



**HAL**  
open science

# Contribution à la génération automatique de plans de masse

Philippe Chaisemartin

► **To cite this version:**

Philippe Chaisemartin. Contribution à la génération automatique de plans de masse. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1986. Français. NNT: . tel-00322531

**HAL Id: tel-00322531**

**<https://theses.hal.science/tel-00322531>**

Submitted on 18 Sep 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THESE

*Présentée à*

**L'Université Scientifique, Technologique et Médicale de Grenoble**

*pour obtenir le grade de*  
**DOCTEUR DE L'UNIVERSITE**  
spécialité "Microélectronique"

par

**Philippe CHAISEMARTIN**



**CONTRIBUTION A LA GENERATION  
AUTOMATIQUE DE PLANS DE MASSE**



Thèse soutenue le 19 novembre 1986 devant la commission d'examen.

G. SAUCIER           Président

J.R. BARRA  
G. DREYFUS           Examineurs  
M. FONTET  
C. MASSON



**UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE**

**Année universitaire 1982-1983**

**Président de l'Université : M. TANCHE**

**MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.**

**(RANG A)**

**SAUF ENSEIGNANTS EN MEDECINE ET PHARMACIE**

**PROFESSEURS DE 1ère CLASSE**

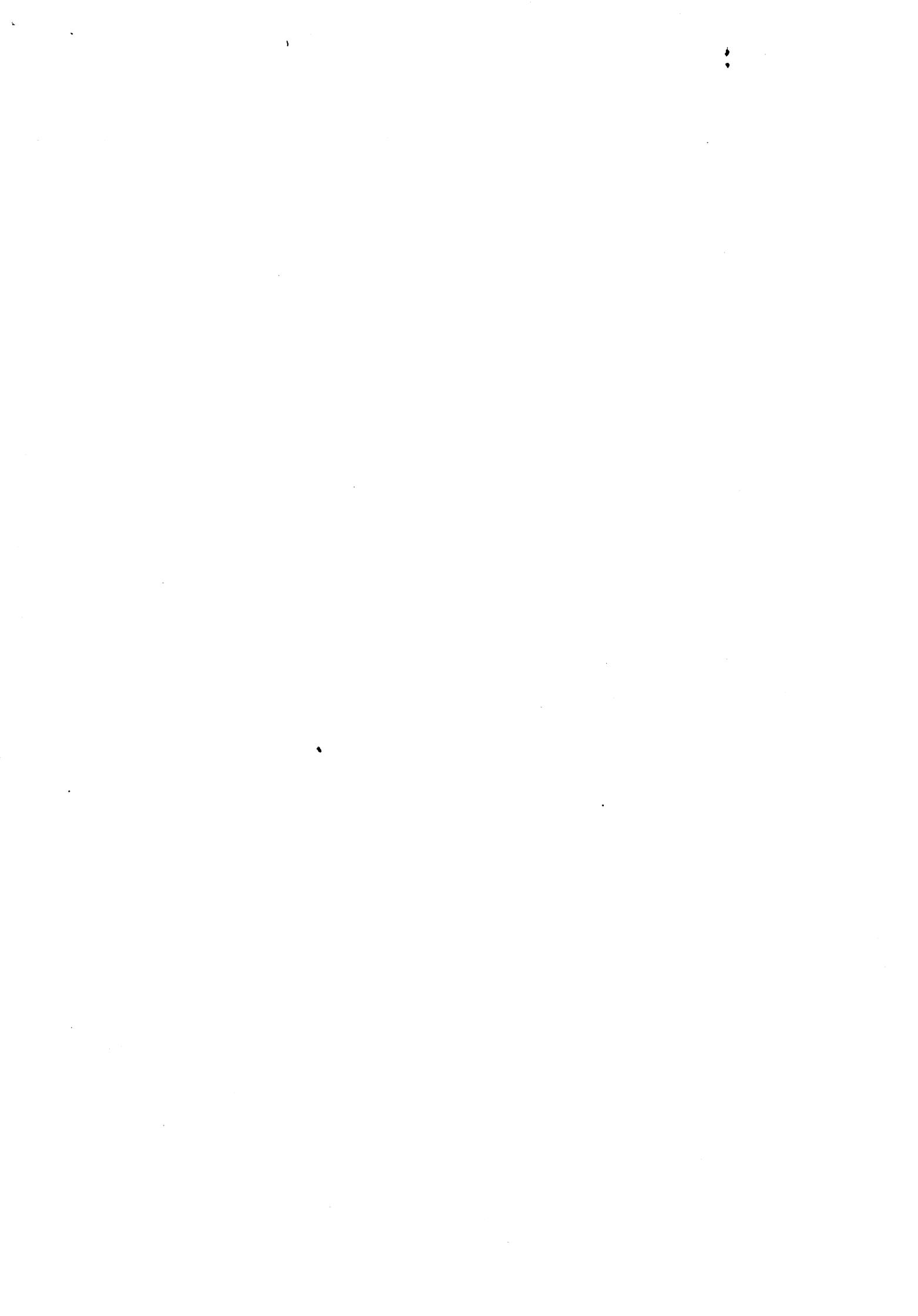
ARNAUD Paul	Chimie organique
ARVIEU Robert	Physique nucléaire I.S.N.
AUBERT Guy	Physique C.N.R.S.
AYANT Yves	Physique approfondie
BARBIER Marie-Jeanne	Electrochimie
BARBIER Jean-Claude	Physique expérimentale C.N.R.S. (labo de magnétisme)
BARJON Robert	Physique nucléaire I.S.N.
BARNOUD Fernand	Biosynthèse de la cellulose-Biologie
BARRA Jean-René	Statistiques - Mathématiques appliquées
BELORISKY Elie	Physique
BENZAKEN Claude (M.)	Mathématiques pures
BERNARD Alain	Mathématiques pures
BERTRANDIAS Françoise	Mathématiques pures
BERTRANDIAS Jean-Paul	Mathématiques pures
BILLET Jean	Géographie
BONNIER Jean-Marie	Chimie générale
BOUCHEZ Robert	Physique nucléaire I.S.N.
BRAVARD Yves	Géographie
CARLIER Georges	Biologie végétale
CAUQUIS Georges	Chimie organique
CHIBON Pierre	Biologie animale
COLIN DE VERDIERE Yves	Mathématiques pures
CRABBE Pierre (détaché)	C.E.R.M.O.
CYROT Michel	Physique du solide
DAUMAS Max	Géographie
DEBELMAS Jacques	Géologie générale
DEGRANGE Charles	Zoologie
DELOBEL Claude (M.)	M.I.A.G. Mathématiques appliquées
DEPORTES Charles	Chimie minérale
DESRE Pierre	Electrochimie
DOLIQUE Jean-Michel	Physique des plasmas
DUCROS Pierre	Cristallographie
FONTAINE Jean-Marc	Mathématiques pures
GAGNAIRE Didier	Chimie physique

GASTINEL Noël	Analyse numérique - Mathématiques appliquées
GERBER Robert	Mathématiques pures
GERMAIN Jean-Pierre	Mécanique
GIRAUD Pierre	Géologie
IDELMAN Simon	Physiologie animale
JANIN Bernard	Géographie
JOLY Jean-René	Mathématiques pures
JULLIEN Pierre	Mathématiques appliquées
KAHANE André (détaché DAFCO)	Physique
KAHANE Josette	Physique
KOSZUL Jean-Louis	Mathématiques pures
KRAKOWIAK Sacha	Mathématiques appliquées
KUPTA Yvon	Mathématiques pures
LACAZE Albert	Thermodynamique
LAJZEROWICZ Jeannine	Physique
LAJZEROWICZ Joseph	Physique
LAURENT Pierre	Mathématiques appliquées
DE LEIRIS Joël	Biologie
LLIBOUTRY Louis	Géophysique
LOISEAUX Jean-Marie	Sciences nucléaires I.S.N.
LOUP Jean	Géographie
MACHE Régis	Physiologie végétale
MAYNARD Roger	Physique du solide
MICHEL Robert	Minéralogie et pétrographie (géologie)
MOZIERES Philippe	Spectrométrie - Physique
OMONT Alain	Astrophysique
OZENDA Paul	Botanique (biologie végétale)
PAYAN Jean-Jacques (détaché)	Mathématiques pures
PEBAY PEYROULA Jean-Claude	Physique
PERRIAUX Jacques	Géologie
PERRIER Guy	Géophysique
PIERRARD Jean-Marie	Mécanique
RASSAT André	Chimie systématique
RENARD Michel	Thermodynamique
RICHARD Lucien	Biologie végétale
RINAUDO Marguerite	Chimie CERMAV
SENGEL Philippe	Biologie animale
SERGERAERT Francis	Mathématiques pures
SOUTIF Michel	Physique
VAILLANT François	Zoologie
VALENTIN Jacques	Physique nucléaire I.S.N.
VAN CUTSEN Bernard	Mathématiques appliquées
VAUQUOIS Bernard	Mathématiques appliquées
VIALON Pierre	Géologie

**PROFESSEURS DE 2ème CLASSE**

ADIBA Michel	Mathématiques pures
ARMAND Gilbert	Géographie

AURIAULT Jean-Louis	Mécanique
BEGUIN Claude (M.)	Chimie organique
BOEHLER Jean-Paul	Mécanique
BOITET Christian	Mathématiques appliquées
BORNAREL Jean	Physique
BRUN Gilbert	Biologie
CASTAING Bernard	Physique
CHARDON Michel	Géographie
COHENADDAD Jean-Pierre	Physique
DENEUVILLE Alain	Physique
DEPASSEL Roger	Mécanique des fluides
DOUCE Roland	Physiologie végétale
DUFRESNOY Alain	Mathématiques pures
GASPARD François	Physique
GAUTRON René	Chimie
GIDON Maurice	Géologie
GIGNOUX Claude (M.)	Sciences nucléaires I.S.N.
GUITTON Jacques	Chimie
HACQUES Gérard	Mathématiques appliquées
HERBIN Jacky	Géographie
HICTER Pierre	Chimie
JOSELEAU Jean-Paul	Biochimie
KERCKOVE Claude (M.)	Géologie
LE BRETON Alain	Mathématiques appliquées
LONGEQUEUE Nicole	Sciences nucléaires I.S.N.
LUCAS Robert	Physiques
LUNA Domingo	Mathématiques pures
MASCLE Georges	Géologie
NEMOZ Alain	Thermodynamique (CNRS - CRTBT)
OUDET Bruno	Mathématiques appliquées
PELMONT Jean	Biochimie
PERRIN Claude (M.)	Sciences nucléaires I.S.N.
PFISTER Jean-Claude (détaché)	Physique du solide
PIBOULE Michel	Géologie
PIERRE Jean-Louis	Chimie organique
RAYNAUD Hervé	Mathématiques appliquées
ROBERT Gilles	Mathématiques pures
ROBERT Jean-Bernard	Chimie physique
ROSSI André	Physiologie végétale
SAKAROVITCH Michel	Mathématiques appliquées
SARROT REYNAUD Jean	Géologie
SAXOD Raymond	Biologie animale
SOUTIF Jeanne	Physique
SCHOOL Pierre-Claude	Mathématiques appliquées
STUTZ Pierre	Mécanique
SUBRA Robert	Chimie
VIDAL Michel	Chimie organique
VIVIAN Robert	Géographie



Toute chose finie n'est jamais entièrement achevée tant qu'elle n'est pas complètement terminée.

Pierre DAC



## Remerciements prompts

Je tiens à exprimer toute ma reconnaissance à **Madame Gabrièle Saucier**, Professeur à l'ENSIMAG, pour m'avoir accueilli dans son laboratoire de recherche, pour avoir assuré l'encadrement scientifique de mon travail et pour m'avoir fait l'honneur de présider le jury de cette thèse.

Je tiens également à remercier :

Monsieur **Christian Masson** d'avoir accepté de faire partie du jury de cette thèse, ainsi que

Monsieur **Jean René Barra**, pour m'avoir apporté les bases théoriques qui me furent très précieuses lors de ce travail et

Messieurs **Max Fontet** et **Gérard Dreyfus** avec qui les réunions furent toujours aussi sympathiques qu'enrichissantes.

Je voudrais également remercier tous mes collègues du *Laboratoire Circuits et Systèmes* pour leurs compétences dont ils ont su me faire profiter et pour l'ambiance agréable qu'ils y ont fait régner.





## TABLE DES MATIERES

<b>Introduction</b>	15
<b>I Le placement automatisé: généralités</b>	19
I.1. Le réseau logique	21
I.2. La structure hôte	23
I.3. Les éléments à placer	27
I.4. Le placement	31
I.5. La fonction coût	37
<b>II. Les méthodes classiques de placement</b>	43
II.1. La démarche	45
II.1.a. Le placement initial	45
II.1.b. L'amélioration itérative	50
II.1.c. La hiérarchie	52
II.2. Les méthodes	60
II.2.a. Le placement initial	60
II.2.b. L'amélioration itérative	64

<b>III. La génération de plans de masse et les méthodes statistiques</b>	<b>69</b>
III.1. L'analyse factorielle d'un tableau de distances	71
III.2. Application à la génération de plans de masse	77
III.2.a. L'inégalité triangulaire	85
III.2.b. Les blocs durs	94
III.2.c. Les plots	102
III.3. La classification	103
III.4. Amélioration du placement	106
III.4.a. Le compactage	107
III.4.b. Le routage global	121
III.4.c. Les perspectives	127
<b>IV Le programme ALFA</b>	<b>131</b>
IV.1. Les données	133
IV.2. Les différentes étapes du programme	137
IV.3. Les limitations et les extensions possibles	142
IV.4. Exemples	144

<b>Conclusion</b>	149
<b>Bibliographie</b>	153



## Introduction

J'aurais aimé commencer cette thèse par autre chose que les désormais classiques "De part la complexité croissante des circuits intégrés..." ou "La densité d'intégration augmentant sans cesse...", mais que dire d'autre lorsque l'on s'est justement penché sur les problèmes posés par l'évolution de ces circuits.

Le but de cette thèse est en effet d'apporter quelques éléments originaux, et je l'espère intéressants, dans le domaine du placement automatisé.

Malgré l'abondance des publications sur ce sujet, la plupart des algorithmes proposés ne sont que des variantes de méthodes classiques, appliquées depuis déjà plusieurs années.

L'approche proposée dans cet ouvrage fait appel à un domaine peu connu des concepteurs de circuits intégrés: celui de l'analyse factorielle de données. Cette méthode, connue depuis longtemps des statisticiens, permet en effet de résoudre rapidement et efficacement des problèmes similaires, du moins en théorie, à ceux posés par le placement automatisé.

Je ne voudrais cependant pas dévoiler maintenant toute la démarche effectuée afin de laisser au lecteur le plaisir (?) de la découvrir au fil des pages.

D'une manière générale, le placement (automatisé ou non) consiste à disposer un ensemble d'éléments de surfaces finies sur un plan en respectant un certain nombre de contraintes. Ces contraintes étant, soit d'ordre géométrique, soit dues aux interactions entre ces éléments.

En fait, le placement automatisé n'est pas restreint à un type de problème bien défini. Il regroupe un ensemble de cas relativement différents et apporte, si possible, une solution spécifique à chacun d'eux.

Les problèmes abordés vont du placement d'éléments réguliers sur une grille (sachant que le nombre de ces éléments peut atteindre plusieurs milliers) au placement de formes géométriques plus ou moins complexes sur une surface restreinte (le nombre des éléments ne dépassant pas quelques dizaines dans ce cas).

Le premier cas est celui du placement sur réseaux structurés ou réseaux prédiffusés, le second est le problème de la génération de plans de masse pour les circuits classiques ("full custom").

Bien que quelques rares tentatives d'applications des méthodes factorielles au problème présenté ici aient été réalisées, aucune étude approfondie n'a été menée. Ceci provient, je pense, du fait qu'il existe un grand pas entre la théorie et son application pratique. C'est essentiellement cette étape qui sera présentée dans les pages qui suivent.

Ainsi, cette approche originale, a permis d'ouvrir de nouvelles portes sur l'implantation automatisée.

Les différents types de problèmes seront exposés dans le chapitre I. Dans cette première partie, le placement sera exposé de façon tout à fait générale puis les différents cas seront mis en évidence en fonction des différences fondamentales qu'ils présentent.

Au deuxième chapitre, l'ensemble des méthodes classiques sera passé en revue. Les différentes solutions répondant aux spécifications présentées dans le chapitre précédant seront ainsi présentées.

Le chapitre trois présente l'analyse factorielle et son application au placement automatisé et plus spécialement à la génération de plans de masse: c'est à dire au placement de surfaces rectangulaires sur un plan (limité ou non), ces surfaces étant susceptibles de subir diverses transformations géométriques.

Le chapitre IV présente enfin le logiciel ALFA développé parallèlement à l'étude théorique présentée au chapitre III. Ce programme a permis de valider ou au contraire de prouver l'inefficacité des différentes options fournies par l'étude théorique.

Ce programme et cette étude ne sont toutes fois pas clos. Ils ont en effet permis d'ouvrir de plus amples perspectives quant à l'utilisation des méthodes statistiques en placement automatisé. Ceci sera présenté en guise de conclusion.

Je voudrais ajouter une remarque à cette introduction: cette thèse s'adresse

d'une part aux concepteurs qui trouveront ici une nouvelle approche du placement automatisé, et d'autre part aux statisticiens qui trouveront une nouvelle application de l'analyse factorielle de données. J'ai donc essayé de rendre les pages qui suivent compréhensibles pour chacun d'entre eux sachant que les langages employés sont souvent différents.



## **CHAPITRE I**

### **LE PLACEMENT AUTOMATISE**



## **I. LE PLACEMENT AUTOMATISE : Généralités**

Le problème qui nous intéresse ici, est celui du placement d'éléments électroniques à l'intérieur d'un circuit intégré. Avant de décrire dans le détail les différentes méthodes permettant d'arriver au but que l'on s'est fixé, il est bon de définir les objets qui seront manipulés.

Le point de départ de tout processus de placement n'est pas seulement une liste de composants mais également une liste d'interconnexions entre ces composants. C'est pourquoi, nous dirons que le placement consiste à "immerger" un graphe logique à l'intérieur d'une structure hôte. En fait, si l'ensemble des composants logiques et de leurs interconnexions peut être facilement modélisé par un graphe (dit graphe logique), plusieurs représentations sont possibles. De même, la surface hôte qui est la surface de silicium sur laquelle seront disposés les composants peut posséder des structures très différentes. Le type des composants est également variable et lié à la structure de la surface d'accueil. La diversité de ces paramètres entraîne une grande disparité dans les problèmes de placement. Chacun des cas qui peuvent se présenter va donc être détaillé.

### **I.1. Le réseau logique ou graphe logique.**

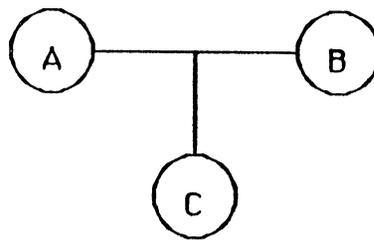
Le réseau logique est décrit par une "net-list" qui associe un ensemble d'équipotentiels à chaque élément. On peut trouver à ce niveau de nombreuses formes de descriptions. Une des plus simples et peut être une des plus courantes consiste en un tableau composants X connexions qui a la forme suivante:

		connexions						
		$c_1$	$c_2$	$c_3$	...	$c_i$	...	$c_n$
éléments	$e_1$							
	$e_2$							
	...							
	$e_j$					$t_{ij}$		
	...							
$e_p$								

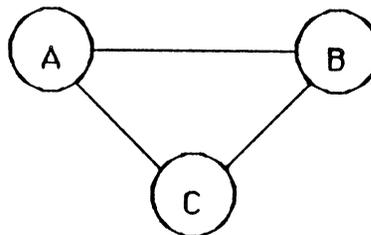
Où  $t_{ij} = 1$  si la connexion  $c_i$  est reliée à l'élément  $e_j$   
= 0 sinon.

La représentation d'un tel tableau peut se faire sous forme de graphe. La solution la plus simple consiste à considérer chaque sommet du graphe comme un élément et chaque arc comme une équipotentielle. Cette représentation, si elle a l'avantage d'être simple, entraîne souvent une perte d'information.

Exemple: A, B et C reliés par une même équipotentielle,



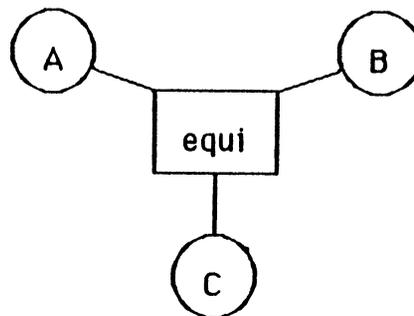
est représenté par:



On constate que cette représentation ne reflète pas la réalité. On est donc souvent amené à créer un autre type de graphe où les sommets peuvent être de deux types différents:

- du type élément logique,
- ou du type équipotentielle.

Les arcs représentent le lien (ou le contact) entre une équipotentielle et un élément logique. Ainsi le cas précédent sera représenté par le graphe suivant.



Cependant, si les équipotentielles sont essentiellement biterminales, c'est à dire ne connectant pas plus de 2 éléments ensemble, un réseau logique du premier type est suffisant.

Il est également intéressant de pouvoir pondérer ces réseaux logiques.

- Soit pour alléger le graphe, en ne faisant plus apparaître toutes les connexions entre deux éléments mais seulement une, affectée d'un poids correspondant au nombre d'équipotentielles qu'elle représente.
- Soit pour ajouter une information concernant la criticité d'une équipotentielle. En effet, pour des raisons électriques ou temporelles, le concepteur peut privilégier certaines connexions. Cette préférence s'accompagne alors d'une pondération plus ou moins importante de l'équipotentielle concernée.

## 1.2. La structure hôte.

Elle correspond à la surface sur laquelle seront placés les différents éléments électroniques [SAU 84]. Au niveau de description le plus bas, un circuit est constitué d'un ensemble de transistors connectés entre eux de façon à réaliser la fonction voulue. A ce niveau, transistors et connexions sont intimement liés et indissociables.

A un niveau de description plus élevé, le circuit sera considéré comme un ensemble de fonctions logiques telles qu'un registre ou un opérateur arithmétique. A ce stade, on pourra différencier plus aisément les modules électroniques des connexions les reliant. Ces modules occuperont des surfaces plus ou moins importantes du circuit, mais relativement bien délimitées. Les fils permettant de relier un module à un autre utiliseront, alors, des passages se faufilant entre ces zones occupées. Ces passages sont appelés "canaux". Un circuit pourra donc être divisé en deux zones de vocations différentes:

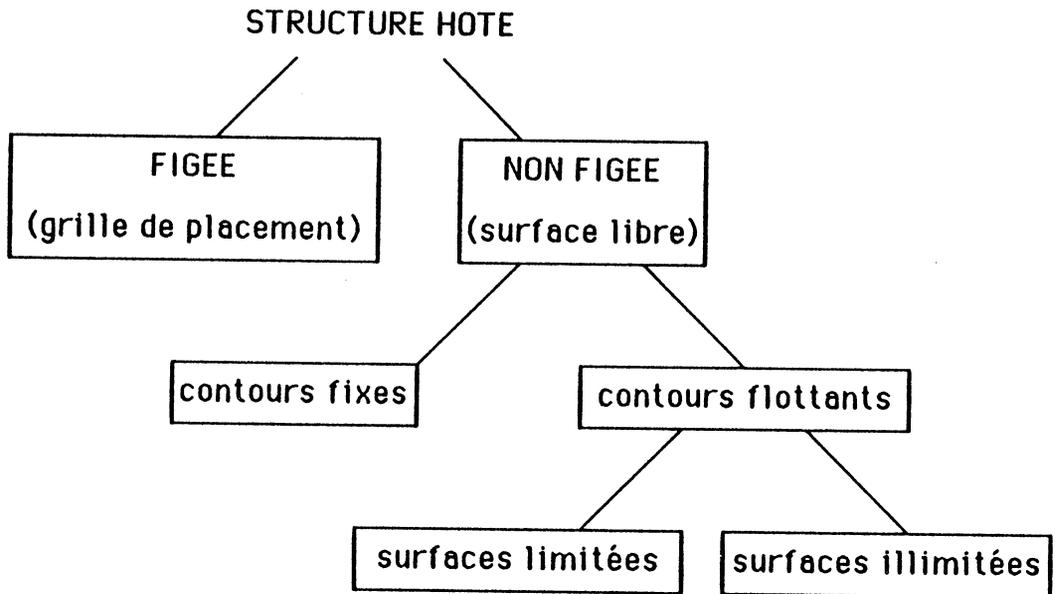
- les zones contenant les fonctions électroniques, dites zones actives.
- les zones ne contenant que des connexions, appelées canaux.

**Remarque** : ce découpage franc d'un circuit en deux zones à vocation fondamentalement différente peut toutefois être nuancé. En effet, il arrive souvent que les éléments possèdent une certaine transparence. Cette transparence définit la faculté d'un composant à pouvoir être traversé par une connexion totalement indépendante de son fonctionnement. Cette transparence est d'autant plus importante que les technologies couramment employées actuellement autorisent au moins deux niveaux de connexions. Il est donc possible de concevoir un composant en laissant libre le niveau supérieur de connexion (totalement ou partiellement) afin de faciliter le tracé ultérieur des connexions entre les différents modules. En pratique cette transparence est relativement faible. Les différents niveaux de connexion étant essentiellement utilisés pour réduire la taille de chaque composant. La transparence est donc limitée à quelques liaisons électriques déjà tracées et dont les contacts sont définis sur la périphérie du module. La notion de canaux reste donc toujours valable et le restera jusqu'à ce que de nouvelles technologies offrent un nombre de niveaux suffisamment important pour pouvoir assurer un routage complet au dessus des zones actives.

Enfin, on pourra ajouter à ces deux zones une zone périphérique réservée aux points de connexions vers l'extérieur du circuit. Ces connexions sont celles permettant

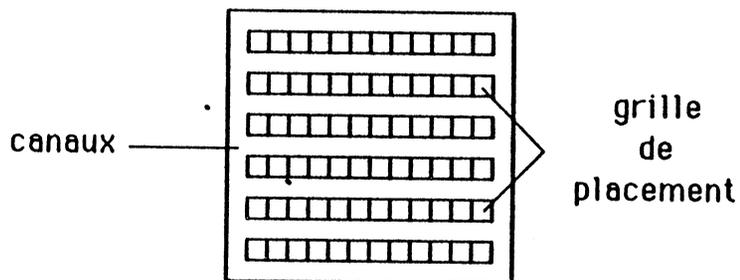
de relier la "puce" aux pattes du boîtier. Ces points sont appelés plots d'entrée/sortie.

En fonction des contraintes propres à chaque structure d'accueil, il est possible de les diviser en plusieurs classes:



### Les structures figées.

Ces structures sont notamment celles des réseaux prédiffusés. En fait, on peut considérer une telle structure comme une grille dont chaque case est susceptible d'accueillir 1 composant ou un ensemble de connexions. En réalité, on différencie deux types de "cases". Certaines zones sont réservées aux composants, d'autres aux connexions. Les premières sont appelées zones actives, les secondes canaux.



Exemple de structure figée: réseau prédiffusé C-MOS

Dans un tel type de structure hôte, les contraintes sont très fortes:

- surface et forme du circuit totalement figées,
- placement régulier des composants (grille de placement).

Il faut également remarquer que dans de telles structures les zones de connexion (ou canaux) possèdent généralement des directions privilégiées. Cette contrainte impose notamment de prendre en compte la transparence des différents composants du circuit (voir paragraphe précédent) afin d'assurer le tracé des connexions entre les zones actives non contiguës.

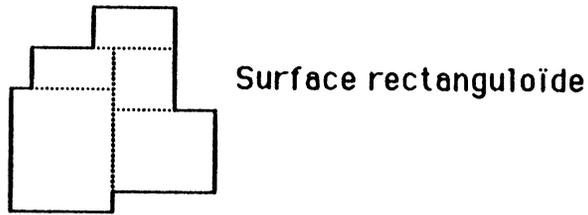
#### **Les structures non contraintes.**

Dans ce cas, la surface d'accueil est libre et la disposition interne des composants n'est pas contrainte. On pourra toutefois différencier quelques cas:

- Les surfaces à contours fixes: dans ce cas, la surface est limitée et sa géométrie est définie. Le placement doit alors impérativement s'effectuer sans déborder de la surface donnée.

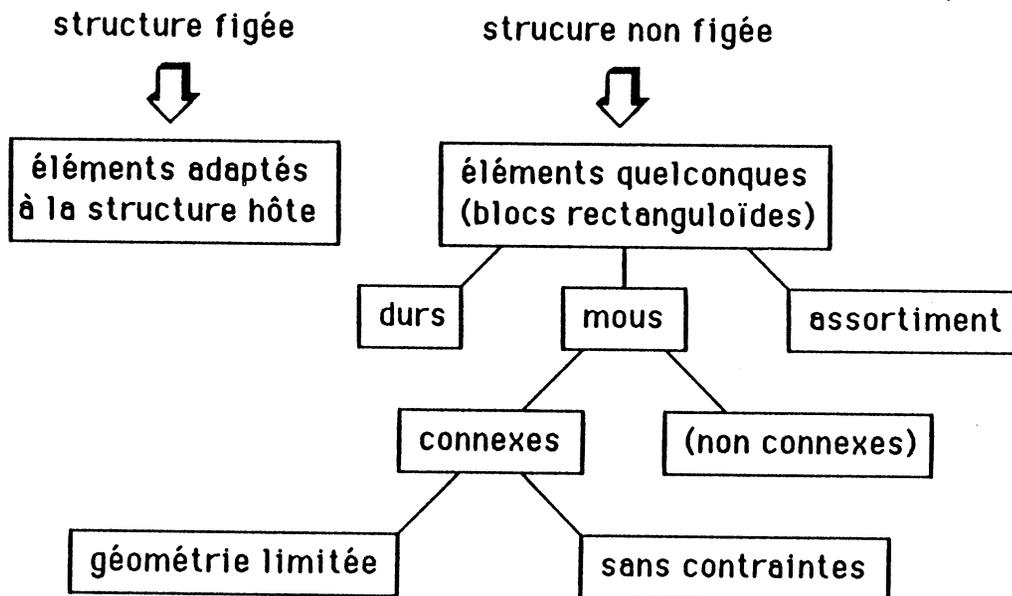
- Les surfaces à contours flottants: dans ce cas, une certaine liberté est laissée au concepteur. Cette liberté est soit totale si la surface finale est quelconque, soit partielle si cette surface est limitée. Dans ce dernier cas, il est possible de jouer sur la hauteur et la largeur du circuit afin de s'adapter au mieux aux contraintes du placement. Il faut noter qu'une limite au rapport de géométrie ( c'est à dire le rapport entre hauteur et largeur) peut être imposée.

**Remarque:** on se limitera généralement aux surfaces rectangulaires, mais il peut être utile de manipuler des surfaces aux contours un peu plus complexes. Cependant, il suffit de considérer des polygones ne comportant que des arêtes verticales ou horizontales et des angles droits. Ce type de surface sera appelé rectanguloïde. En fait, toute surface rectanguloïde peut se décomposer en une juxtaposition de rectangles simples.



### I.3. Les éléments à placer.

Comme la surface hôte, les éléments à placer peuvent posséder des structures très différentes. En fait, ces structures sont liées au type de circuit manipulé.



Dans le cas des structures fortement contraintes, les éléments à placer sont adaptés à la topologie du circuit, c'est à dire qu'ils sont compatibles avec la grille de placement définie par la structure hôte. Un composant occupera une case de la grille, ou éventuellement un nombre entier de cases contiguës.

Dans le cas des structures faiblement contraintes, on se limitera également à des composants rectanguloïdes parmi lesquels on pourra différencier:

- les blocs durs dont la géométrie (forme et surface) est fixée,
- les blocs mous dont la géométrie est non définie. Dans ce cas, la seule indication

fournie est la surface totale occupée par le composant. On se limitera généralement aux blocs dits connexes, c'est à dire que l'on interdit de couper un composant en deux blocs séparés.

Comme dans le cas des structures non figées, on peut être astreint à certaines contraintes concernant le rapport de géométrie du composant. Il est également possible de réduire les déformations apportées à un bloc rectanguloïde en limitant le nombre minimal de rectangles permettant de le constituer.

Le type de blocs mous non connexes correspond à l'éclatement d'un bloc en plusieurs morceaux non contigus. En fait, ce type n'est signalé ici qu'à titre indicatif. Il faut remarquer que le découpage d'un composant en plusieurs composants plus petits n'est possible qu'avec une parfaite connaissance de sa structure interne, notamment concernant la connectique entre les différents modules ainsi constitués. Le bloc mou étant par définition un bloc dont la topologie n'est pas définie, il est impossible d'en fournir une description suffisamment précise pour en autoriser l'éclatement tout en conservant une information cohérente sur le réseau logique.

Le type baptisé assortiment est un cas particulier qui permet le choix entre plusieurs blocs de type dur. Ceci apparaît lorsque l'on dispose de plusieurs composants équivalents au niveau fonctionnel mais différents au niveau topologique [UED 85]. Considérons par exemple un circuit contenant une mémoire de 512 bits. Cette mémoire peut être implantée sous plusieurs formes: soit une matrice de 32 mots de 16 bits, soit une matrice de 16 mots de 32 bits, soit une matrice de 64 mots de 8 bits, etc... Chacune de ces configurations correspond à un modèle existant. Lors du placement, suivant la place disponible et surtout suivant la géométrie de cette place, le concepteur (ou le programme) sera amené à choisir l'un ou l'autre de ces plans mémoires. Il faut remarquer que ce cas ne peut être assimilé à une déformation. L'ensemble des solutions est fini et chaque solution n'occupe pas forcément la même surface.

### **Problèmes liés au nombre des éléments:**

Le placement peut se situer à plusieurs niveaux de la description d'un circuit. On ne pourra cependant pas considérer le niveau le plus bas, c'est à dire le niveau des transistors comme acceptable. Comme il a été dit plus haut, à ce stade de description, les transistors sont trop intimement liés aux connexions. De plus la surface des transistors ne représente pas plus de 30% de la surface totale d'un circuit qui est essentiellement occupée par les connexions. Il n'est donc pas raisonnable de vouloir réaliser un système de placement automatique à ce niveau de description. Il existe également des outils de compaction des dessins de masques, mais il ne s'agit plus à ce niveau, de placement automatisé. On pourra trouver dans [THU 83] et dans la bibliographie qui s'y rapporte, des informations complètes sur ce type d'outil.

Par contre, au niveau des portes logiques ou d'éléments électroniques simples (bascules, multiplexeurs...) il devient possible de mettre au point des programmes de placement automatisé. D'après la taille des circuits actuels, il est facile de constater que le nombre de portes atteint plusieurs dizaines de milliers (15000 pour un 6809, 20000 pour un 68000), d'où une complexité extrême du problème. En fait, le placement au niveau de la porte logique concerne essentiellement le domaine des réseaux prédiffusés dans lesquels le nombre de portes se limite à quelques milliers ce qui atteint déjà une complexité non négligeable.

Le deuxième type de problème concerne les circuits non contraints. Dans ce cas, la description d'un circuit se fait à plusieurs niveaux:

- Le niveau architectural dans lequel on retrouve les modules principaux du circuit comme par exemple:

- registres,
- compteurs,
- opérateurs arithmétiques et/ou logiques,
- mémoires,
- contrôleurs...

- Le niveau logique dans lequel chacun des modules du niveau supérieur sera

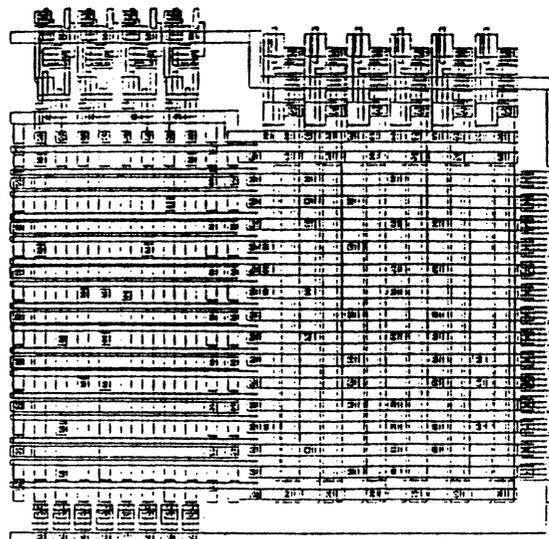
décrit sous forme de portes logiques.

- Le niveau électrique dans lequel chaque porte est décrite sous forme de transistors.

Dans le problème du placement automatisé, nous ne nous intéresserons qu'aux deux premiers cas.

Pour le niveau logique, on peut envisager plusieurs approches: la première consiste à placer une logique anarchique sur une surface sans contraintes. Cette approche est un problème trop complexe pour qu'une réponse correcte puisse être apportée de manière automatique. Dans le cas de la deuxième approche, la surface est non contrainte mais la logique est structurée. C'est le cas des réseaux logiques programmables (PLA), des mémoires, des registres à décalage... Dans ce cas, un dessin minimal pour chaque cellule élémentaire est mis au point "à la main" puis des programmes de génération de PLA ou de ROM construisent la structure complète par répétition de cellules élémentaire et personnalisation du circuit par programmation. C'est également à ce niveau que se situent les assembleurs de silicium qui se contentent de juxtaposer des cellules prédéfinies. On trouvera dans [MON 85] et dans la bibliographie qui s'y rapporte, une description complète des assembleurs de silicium.

Exemple:



PLA en technologie NMOS : on voit la structure répétitive et les transistors qui y ont été ajoutés afin de programmer la fonction souhaitée.

Ce problème ne relève pas du placement automatisé mais de la génération automatique ou de la compaction de PLAs par exemple [PAI 82].

La troisième approche consiste à contraindre fortement la surface hôte. Dans ce cas, on retombe dans un problème du premier type. Soit avec une structure du type réseau prédéfini, soit plus généralement avec une structure du type précaractérisé. Cette structure a l'avantage d'être moins contraignante et assez facile à manipuler [UED 85].

Pour le niveau architectural, le problème est tout à fait différent. En effet, à ce stade, il n'y aura jamais plus de quelques dizaines d'éléments à manipuler mais ces éléments pourront subir de nombreuses déformations géométriques. Ce type de problème constitue ce que l'on appelle la génération de plan de masse. Il est évident qu'un même algorithme ne pourra être utilisé pour deux applications aussi différentes.

#### **1.4. Le placement.**

Le placement consiste donc à immerger le graphe logique dans la structure hôte. Cette immersion doit se faire en respectant toutes les contraintes imposées par chacun des éléments manipulés: la structure d'accueil, les modules à placer et évidemment la connectique.

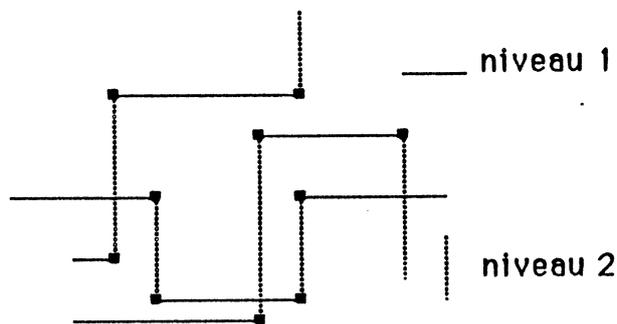
Cette opération consiste à fournir les coordonnées sur un plan de chacun des éléments. Suivant le type de circuit sur lequel on travaille, différentes opérations géométriques peuvent être appliquées:

- La translation : on peut considérer qu'il s'agit de l'opération élémentaire permettant de positionner chaque module. Elle est donc autorisée quelque soit la structure hôte manipulée.

- Les rotations : pour des raisons diverses, on peut être amené à appliquer une rotation aux différents éléments à placer. Il faut noter que dans l'application concernée, seules les rotations correspondant à des multiples de 90° sont autorisées. Une telle opération peut également être interdite. Elle est notamment exclue dans le cas des circuits fortement contraints. Dans le cas des circuits faiblement contraints, la rotation

est généralement autorisée, mais il existe des cas où elle est interdite, et ceci pour la raison suivante:

les connexions empruntant les canaux utilisent au moins deux niveaux de matière différents pour permettre les croisements. Ce peut être un niveau de silicium polycristallin et un niveau de métal, ou deux niveaux de métal, voir deux niveaux de silicium ou trois niveaux de métal. Afin de faciliter le tracé des connexions, chaque niveau a généralement une direction privilégiée (horizontale ou verticale) [DUP 84].



Pour des raisons technologiques, l'espacement entre les lignes de connexion n'est généralement pas le même pour chaque niveau. Il résulte de ceci que le tracé se fait sur une grille dont le pas dépend de la direction.

Donc si des éléments ont été dessinés selon une orientation donnée de sorte que leurs contacts soient positionnés sur le pas de grille correspondant à l'orientation de la face sur laquelle ils se trouvent, il est évident qu'une rotation de  $90^\circ$  ne permettra plus une connexion facile des entrées/sorties de l'élément logique.

- les symétries: on retrouve des contraintes du même type que pour les rotations. De la même façon que pour les rotations, les seules symétries autorisées sont soit des symétries par rapport à l'axe horizontal soit des symétries par rapport à l'axe vertical. On peut également remarquer qu'il est possible de ne prendre en compte qu'un seul type de symétrie. En effet une symétrie par rapport à l'axe vertical est équivalente à une symétrie par rapport à l'axe horizontal, suivie de deux rotations de  $+90^\circ$  (moyennant une translation). Cette propriété permet de réduire le nombre de

transformations géométriques envisageables pour chaque composant et permet généralement de simplifier la représentation.

Ces transformations géométriques ne sont cependant pas les seules à pouvoir être appliquées aux modules à placer. Suivant que le type du module à placer est dur ou mou, on pourra être amené à appliquer ou non une déformation. Cette déformation peut se limiter à faire varier le rapport de géométrie du rectangle représentant le bloc dur ou aller jusqu'à le décomposer en rectangles plus petits. On se limitera généralement à la transformation d'un bloc rectangulaire en un bloc rectanguloïde quelconque, tout en conservant une structure connexe. En effet l'éclatement d'un module en sous-modules non connexes c'est à dire non contiguës n'est envisageable que si une information concernant la connectique entre les sous-modules peut être fournie. Cette éventualité est trop aléatoire pour que cette opération soit autorisée dans le cadre du placement automatisé. Dans le cadre des déformation géométriques, il est possible de prendre en compte les choix entre différents blocs fonctionnellement équivalents.

### **Problèmes liés à la connectique:**

On considère trop souvent que le placement se limite à disposer les éléments électroniques à la surface du circuit. En fait le placement doit également prévoir les places nécessaires au tracé ultérieur des connexions entre les composants c'est à dire au routage. Dans le cas des structures contraintes du type réseaux prédifusés, cette région est prédéfinie. Le rôle du "placeur" est alors d'assurer que le routage sera réalisable c'est à dire que la surface réservée aux connexions sera suffisante pour tracer l'ensemble des équipotentielles du réseau logique. Cette condition est impérative pour valider le placement final des composants.

Dans le cas des structures non contraintes, le problème est quelque peu différent, en effet la surface des canaux étant variable il sera toujours possible de tracer toutes les connexions. Le rôle du placeur est alors de définir le plus précisément possible ces surfaces.

Dans un cas comme dans l'autre, la deuxième contrainte est celle de la longueur

des connexions. On essaiera d'une part de réduire la longueur totale des connexions et d'autre part de réduire la longueur de chacune des connexions. Ceci pour des raisons électriques: plus un fil est court plus le signal se propagera rapidement d'un bout à l'autre et plus les capacités et résistances parasites seront réduites. Mais également pour des raisons topologiques: plus les connexions seront courtes, moins la surface occupée sera importante.

### **Les différents types de problèmes**

Les associations entre éléments à placer et structure hôte vont permettre de différencier deux problèmes différents:

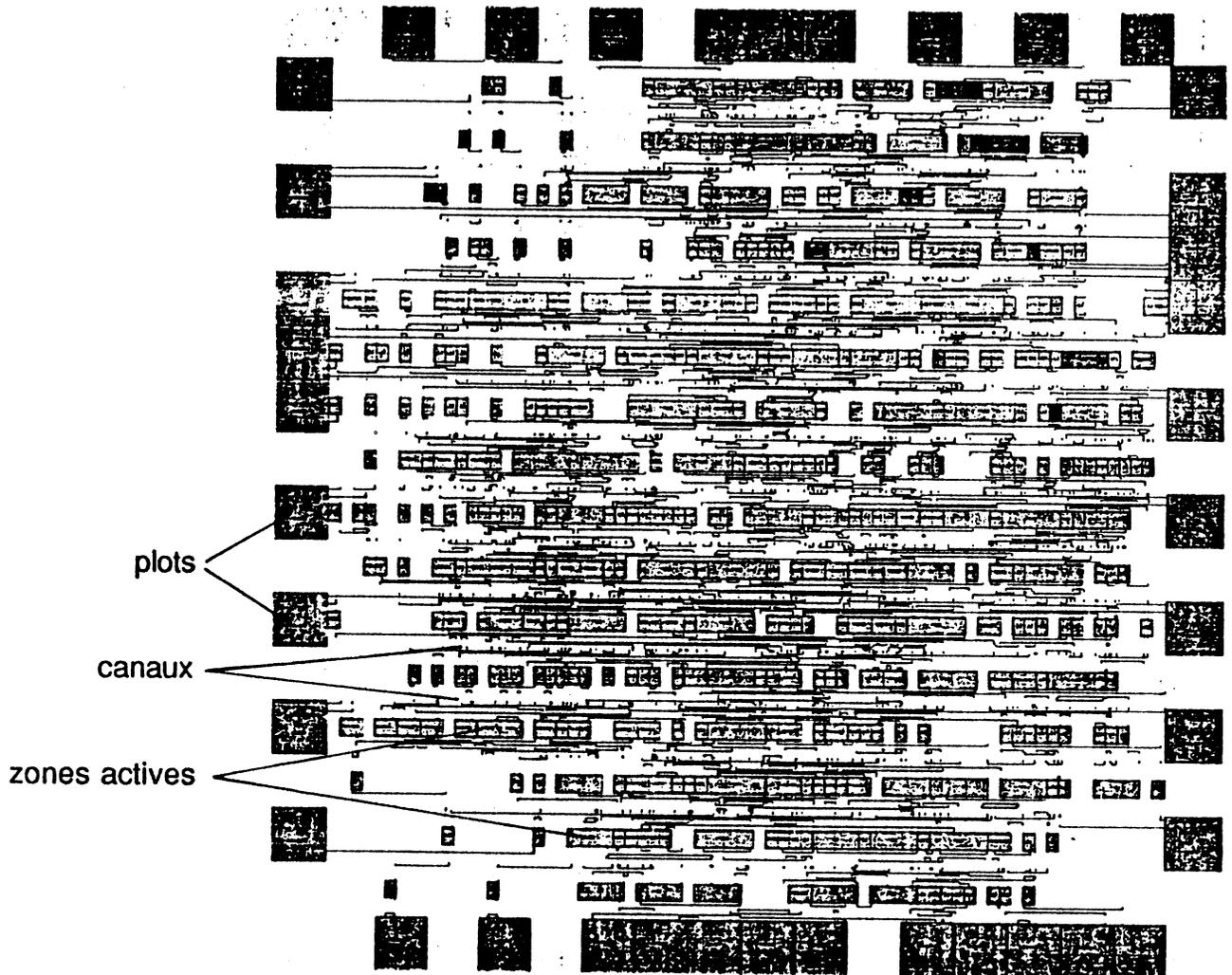
- le placement sur réseaux structurés [BEL84][JAN85],
- la génération de plans de masse [TSI 84][PER 85].

Le placement sur réseaux structurés concerne une grande quantité (plusieurs centaines à plusieurs milliers) de composants à placer sur une "grille" ou "matrice". Ces composants ont tous une taille multiple de la place (ou case) élémentaire de la grille de placement.

Le problème est simplifié du fait que:

- les éléments ont une géométrie figée,
- les rotations et symétries sont généralement exclues.

Exemple:



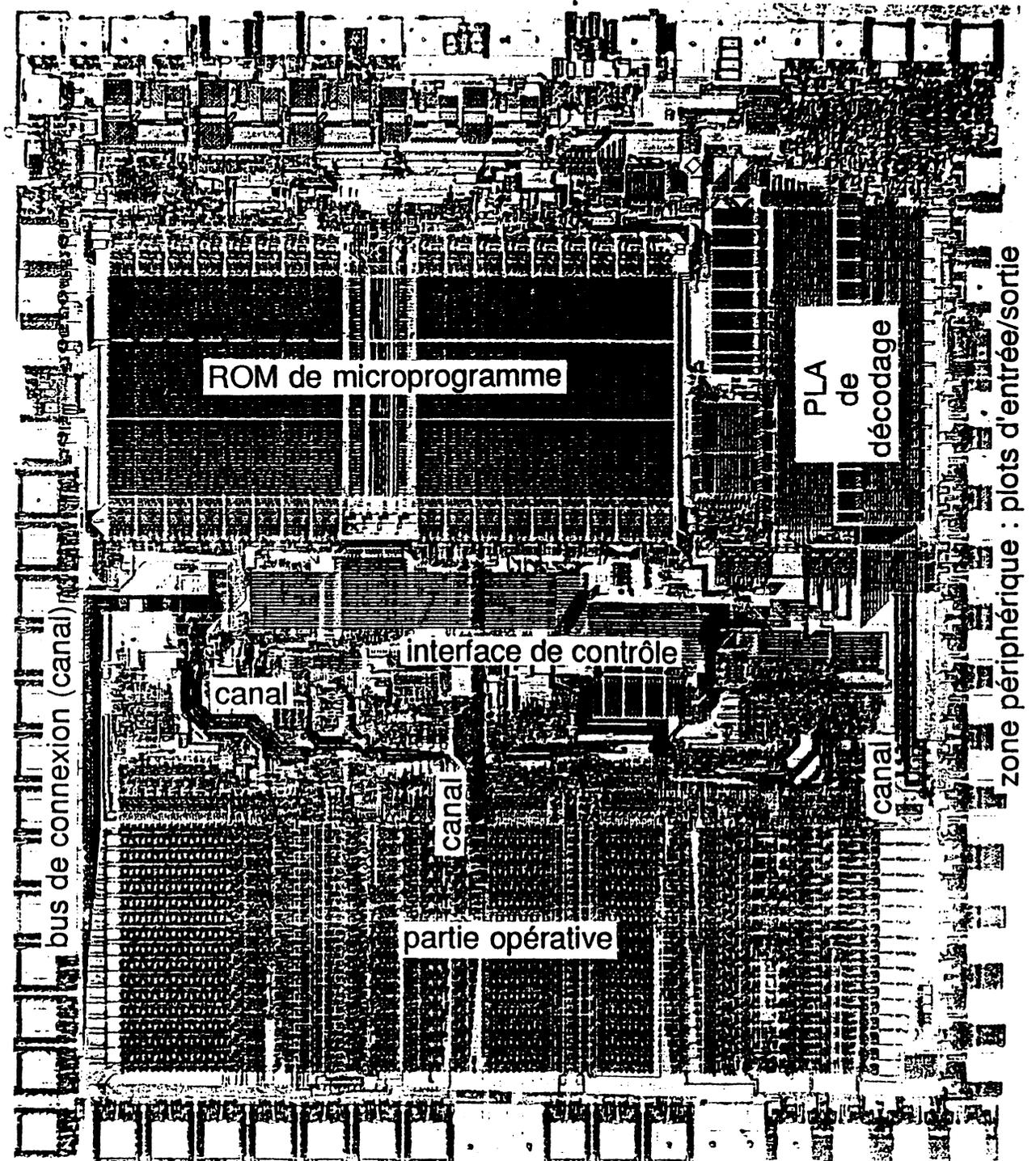
Réseau prédiffusé en technologie CMOS

La génération de plans de masse consiste à disposer une petite quantité de blocs (quelques dizaines au maximum) sur une surface libre. Ces éléments peuvent être déformés et sont susceptibles d'être affectés par des rotations ou des symétries.

Ce problème est généralement abordé en ne prenant en compte que des blocs rectangulaires mais il est souvent utile de pouvoir manipuler des blocs rectanguloïdes c'est à dire constitués de rectangle contigus.

C'est essentiellement ce domaine qui sera développé dans les pages qui suivent mais il faut toutes fois remarquer que si les problèmes sont différents, l'approche n'en reste pas moins semblable dans les deux cas de figures. Les généralités sur les méthodes qui seront présentées plus loin concernent donc aussi bien le placement sur réseaux structurés que la génération de plans de masse.

Exemple: plan de masse du microprocesseur 68000



### **1.5. La fonction coût.**

Le placement consiste donc à réaliser une disposition judicieuse des composants sur la surface hôte. Cette disposition est obtenue en appliquant aux différents éléments du réseau logique un ensemble de transformations géométriques: translations, rotations, symétries et éventuellement déformations. Il est évident que le placement influence très fortement l'étape de routage qui consiste à tracer sur la surface hôte l'ensemble des équipotentielles reliant les composants. La qualité du placement est définie en fonction du respect des contraintes imposées par le concepteur, par la structure utilisée et par les contraintes du routage.

Plusieurs objectifs sont recherchés, parmi ceux-ci, on peut noter:

- 1- minimisation de la surface totale,
- 2- réduction de la longueur totale des connexions,
- 3- réduction de la longueur de chaque connexion,
- 4- faisabilité du placement (non chevauchement des blocs, non débordement du cadre de la structure hôte...),
- 5- faisabilité du routage (conservation de canaux suffisamment large pour assurer le passage de toutes les connexions).

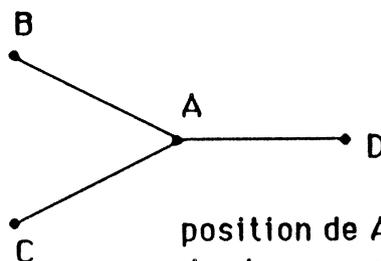
On peut déjà remarquer qu'il existe deux types d'objectifs à réaliser: les objectifs préférentiels comme 1,2 ou 3 et les objectifs impératifs comme 4 ou 5. Ce problème est connu en recherche opérationnelle sous le nom d'optimisation multi-critères sous contraintes. Les contraintes correspondent aux objectifs à respecter impérativement et les critères, aux paramètres à améliorer.

Le premier objectif s'applique essentiellement dans le cas de circuits sans contraintes. Si la géométrie du circuit est définie ou si le circuit est du type réseau prédiffusé, le problème n'est plus d'en réduire la surface mais de réussir à immerger l'intégralité du réseau logique à l'intérieur de la structure fournie.

Les deuxièmes et troisièmes objectifs, même s'ils semblent redondants sont parfois incompatibles. On ne pourra généralement optimiser tous les critères en respectant les contraintes. Une solution consiste donc à remplacer l'ensemble des critères par un compromis de ces derniers.

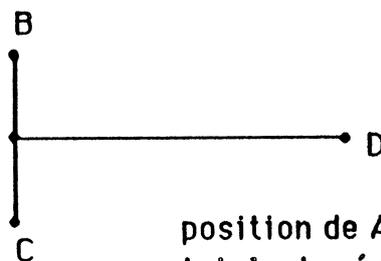
Exemple:

B,C,D fixés:



position de A pour réduire la longueur de chaque equipotentielle.

B,C,D fixés:



position de A pour réduire la longueur totale des équipotentiels.

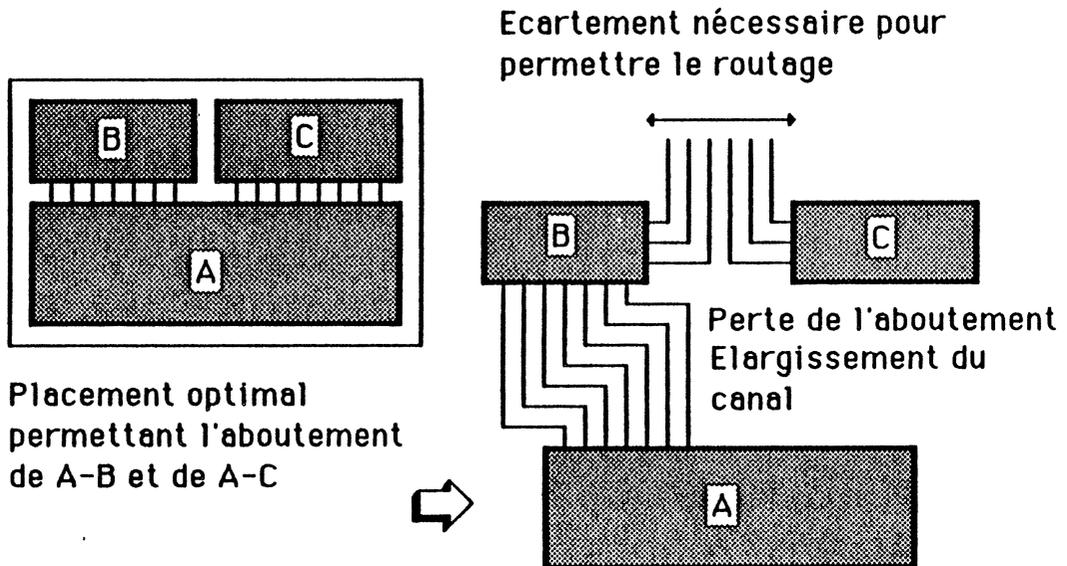
De part la criticité de certaines équipotentiels (délais de propagation devant être réduits, minimisation des capacités parasites...) il est utile de favoriser l'un ou l'autre des critères.

En ce qui concerne la faisabilité du routage, elle dépend fortement du type de circuit concerné.

Dans le cas d'une structure non contrainte, le programme de placement devra définir les largeurs nécessaires à affecter aux canaux pour assurer le routage ultérieur.

Si ces dimensions ne sont pas prises en compte lors du placement, une perturbation importante risque d'être introduite au moment du routage et d'aboutir à une impossibilité de réalisation.

Exemple:



Dans le cas des réseaux prédiffusés, la largeur des canaux est figée, une évaluation du nombre de connexions les empruntant devra également être réalisée lors du placement afin d'éviter toute saturation aboutissant à l'impossibilité de router l'ensemble des connexions. Comme il a été dit dans le paragraphe 1.1.b., cette estimation devra également prendre en compte la transparence des blocs afin de permettre les passages à travers les zones actives.

Dans le but d'évaluer la qualité d'un placement au fur et à mesure de son évolution, une fonction coût est construite. Cette fonction devra être paramétrée par l'ensemble des critères qui définissent la qualité du placement. Elle devra donc prendre en compte les contraintes impératives:

Par exemple, soient S la surface totale de chevauchement des blocs et D la surface de débordement des blocs sur l'extérieur du circuit; on veut à la fin du

placement:

$$S = 0,$$

$$D = 0.$$

Mais également les critères comme par exemple la réduction de la longueur des connexions ou la réduction de la surface totale du circuit.

Par exemple, soient  $L$  la longueur totale des connexions et  $C$  la surface du circuit (ou la surface du rectangle englobant) on veut, à la fin du placement:

$L$  minimale,

$C$  minimale.

La qualité du placement est donc définie par une fonction coût qui dépend des paramètres précédemment cités. L'algorithme de placement doit alors optimiser cette fonction qui est par exemple de la forme:

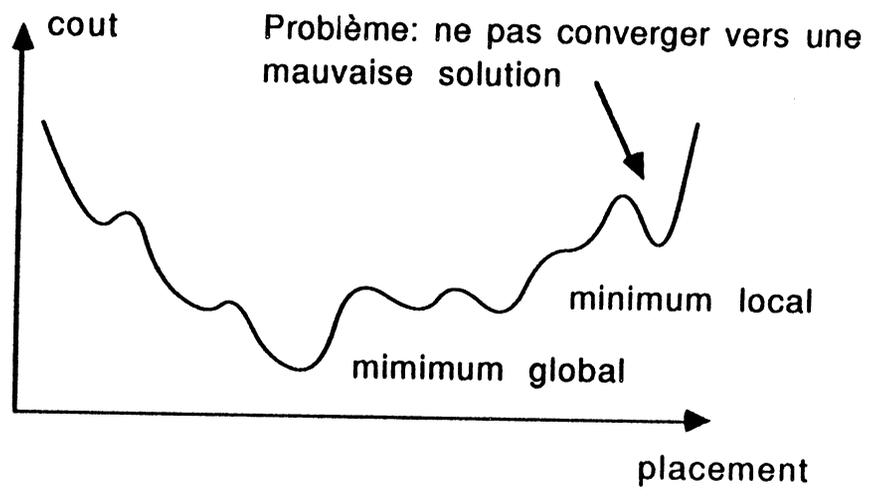
$$F = \alpha S + \beta D + \gamma L + \delta C.$$

Avec  $\alpha$  et  $\beta$  très grands et  $\gamma$  et  $\delta$  adaptés suivant la contrainte primordiale.

En fait, le coût est une fonction des variables qui correspondent aux critères et aux contraintes précédemment définies; mais elle ne pourra être optimisée que sur l'ensemble des variables (coordonnées des blocs) qui satisfont aux contraintes.

Une telle fonction possède évidemment de nombreux minima locaux et un seul minimum global. Le but du placement n'est pas forcément d'atteindre cette meilleure solution mais d'en trouver une suffisamment bonne. Le problème est alors d'éviter les blocages dans des minima locaux ne fournissant pas un résultat satisfaisant.

Exemple:





## **CHAPITRE II**

# **LES METHODES CLASSIQUES DE PLACEMENT**



## **II. LES METHODES CLASSIQUES DE PLACEMENT**

### **II.1 La démarche.**

Comme il a été exposé plus haut, le placement automatisé recouvre un ensemble de problèmes relativement différents. Du placement sur réseaux prédéfinis à la génération de plan de masse. Il est alors impossible de proposer une méthode universelle répondant de façon satisfaisante à tous les cas de figures pouvant se présenter. C'est pourquoi le domaine plus spécialement présenté dans cette thèse est celui de la génération de plan de masse. On pourra trouver dans [BEL 84] une étude plus approfondie sur le placement dans les structures fortement contraintes.

Il faut toutefois remarquer que si les solutions apportées sont très différentes dans le détail, la démarche générale du placement reste toujours à peu près la même. Elle peut se décomposer en deux parties distinctes :

- le placement initial (ou pré-placement),
- l'amélioration finale.

Ces deux parties vont être détaillées dans les paragraphes suivants [SAU 84].

#### **II.1.a. Le placement initial**

Cette étape consiste en fait à dégrossir le problème. Il est en effet extrêmement compliqué, voir impossible, de réaliser en une seule étape un placement respectant toutes les contraintes imposées si l'on ne possède aucune base solide de départ.

En effet le placement des composants à l'intérieur d'un circuit consiste à attribuer à chacun d'entre eux un ensemble de coordonnées permettant de définir leurs positions et éventuellement les transformations géométriques qui leur ont été appliquées (symétrie, rotation, déformation). Le point de départ du placement ne fournissant que des contraintes relatives entre les modules.

Le but du placement initial est de fournir des coordonnées, même

approximatives, pour les éléments à placer en fonction de leurs positions relatives. Ces positions relatives sont calculées à partir de leurs contraintes deux à deux et ne sont pas optimales au niveau global.

Ce problème peut être quelque peu simplifié si on possède quelques points de repères. Ces repères sont souvent fournis par les positions des plots d'Entrée/Sortie. En effet, lors de la conception d'un circuit il est fréquent que des contraintes soient imposées quant à la disposition de ces plots:

- position des fils d'un même bus de façon contiguë et ordonnée,
- respect d'un éventuel standard sur la position des plots d'alimentation,
- etc...

Ces contraintes fournissent donc une base à partir de laquelle il sera possible de construire l'ensemble du circuit. Il faut toutefois faire attention car ces contraintes ne s'accompagnent pas forcément de positions absolues mais souvent de positions relatives.

Cette étape est généralement réalisée de façon constructive, c'est à dire que l'on ne cherchera pas à manipuler simultanément tous les composants à placer et toutes les places disponibles sur la structure hôte mais à les prendre successivement en évaluant à chaque fois la meilleure association place/élément [MAL 82].

Dans une telle approche un ordonnancement est indispensable. C'est en fonction de cet ordre que seront placés les éléments et que seront affectées les places. On voit apparaître deux types d'ordonnancement possibles:

- l'ordonnancement topologique,
- l'ordonnancement logique.

L'ordonnancement topologique consiste à définir un ordre sur les places du circuit [LEB 83]. L'affectation de ces places, c'est à dire le remplissage du circuit avec les différents composants à placer va être réalisé en choisissant pour chaque nouvelle place, dans l'ensemble des composants non encore placés celui qui respecte au

mieux les contraintes, ou tout du moins celui qui permet d'optimiser la fonction coût.

**procédure pré-placement avec ordonnancement topologique;**

**début**

*définir un ordre de progression à l'intérieur de la  
structure hôte;*

*PLACE := première place;*

**tant que tout n'est pas placé**

**début**

*choisir ELEMENT dans l'ensemble des éléments non placés  
tel que le couple (ELEMENT,PLACE)*

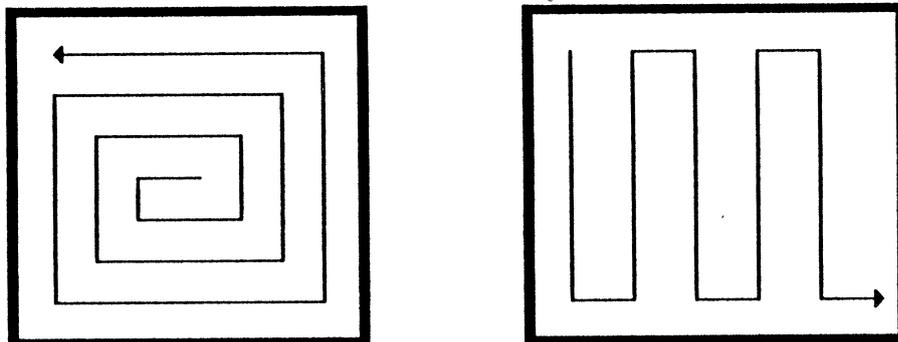
*optimise la fonction coût;*

*PLACE := place suivante ;*

**fin;**

**fin;**

Dans une telle approche l'ordre des places est défini par une progression régulière à l'intérieur de la structure hôte.



Exemple d'ordonnancement des places à l'intérieur d'un circuit.

L'ordonnancement logique consiste à définir un ordre de placement sur les éléments du graphe logique. Dans ce cas le placement consistera à choisir la meilleure place pour chaque nouveau composant. Cet ordre peut être défini à partir de nombreux critères: nombre de connexions aboutissant à chaque composant, éloignement des plots...

*procédure pré-placement avec ordonnancement logique;*

**début**

*définir un ordre de placement pour les éléments;*

*ELEMENT := premier élément;*

*tant que tout n'est pas placé*

**début**

*choisir PLACE dans l'ensemble des places disponibles tel*

*que le couple (ELEMENT,PLACE) optimise la fonction coût;*

*ELEMENT := élément suivant;*

**fin;**

**fin;**

Une méthode mixte peut également être employée. Elle consiste à définir un ordre topologique sur les places et un ordre logique sur les composants. A partir de quoi on placera chaque nouvel élément non encore placé sur chaque nouvelle place [PAI 84][SER 82]. Cette méthode est toutefois extrêmement restrictive. L'ordonnancement ne se fera donc plus sur les éléments et sur les places mais sur des ensembles de ceux-ci. On se définit donc une suite ordonnée d'ensembles de places et une suite ordonnée d'ensembles de composant. Le placement consiste alors à trouver dans les deux ensembles, le couple place/composant optimisant la fonction coût.

*procédure placement par ordonnancement mixte ;*

**début**

*définir des classes sur les places;*

*définir des classes sur les éléments;*

*ordonner les classes places;*

*ordonner les classes éléments;*

*ENSEMBLE PLACES := première classe de places;*

*ENSEMBLE ELEMENTS := première classe d'éléments;*

***tant que tout n'est pas placé***

**début**

*choisir PLACE dans ENSEMBLE PLACES et*

*ELEMENT dans ENSEMBLE ELEMENTS tel que*

*le couple (ELEMENT,PLACE) optimise la fonction coût;*

***si ENSEMBLE PLACES vide***

***alors ENSEMBLE PLACES := classe de places suivante;***

***si ENSEMBLE ELEMENTS vide***

***alors ENSEMBLE ELEMENTS := classe d'él<sup>t</sup> suivante;***

***fin ;***

***fin;***

A la fin du placement initial une position absolue de tous les composants est fournie. Il faut cependant remarquer que ce placement n'a été obtenu que par des optimisations locales de la fonction coût. Il est évident qu'une telle construction ne peut aboutir à un placement optimal c'est pourquoi il est généralement indispensable d'améliorer ce placement par une deuxième étape.

### **II.1.b. L'Amélioration Itérative**

Lors de cette étape on applique quelques perturbations sur le placement précédemment fourni de façon à améliorer globalement le placement. Ces perturbations sont à choisir dans l'ensemble des transformations géométriques

autorisées. Elles peuvent également être du type permutation.

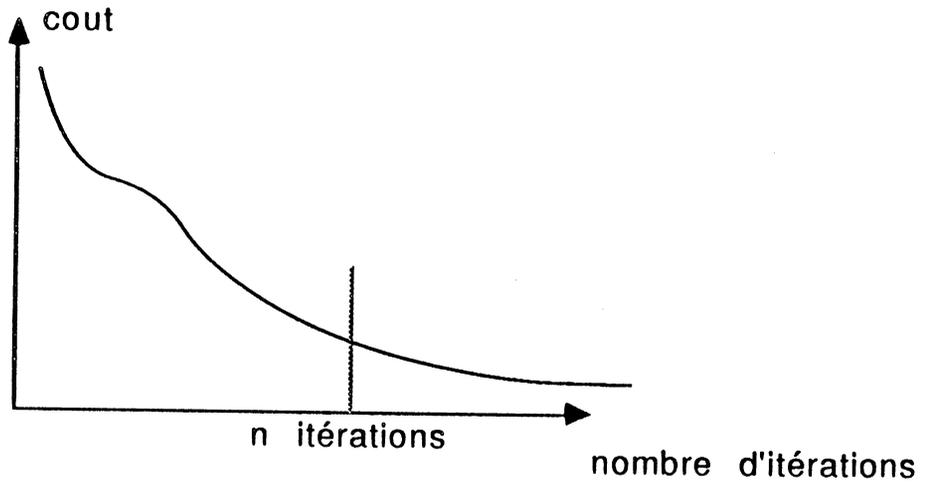
Dans ce cas on choisit deux composants placés à l'intérieur de la structure hôte et on les échange si cette opération améliore effectivement la fonction coût. Il est également possible de réaliser des permutations circulaires avec plus de deux éléments. L'expérience a cependant montré qu'une telle approche risque d'accroître excessivement la complexité du placement en raison du grand nombre de possibilités offertes. On se limitera généralement à des permutations circulaires mettant au plus en jeu trois composants [BEL 84].

Il faut toutefois noter que les permutations ne sont intéressantes qu'entre éléments de tailles équivalentes.

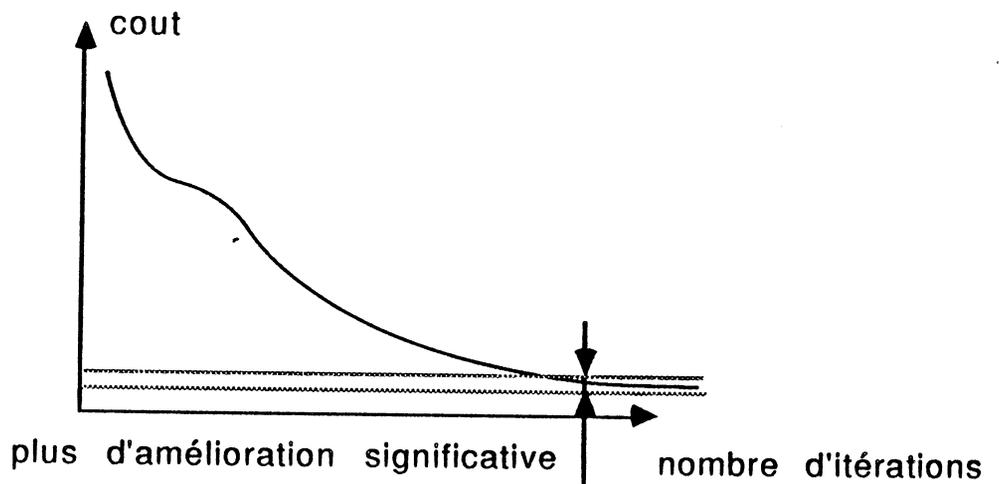
Il est impossible de savoir à un instant donné, si le placement obtenu est optimal et en règle général on ne cherchera pas le meilleur placement mais un placement respectant au mieux les contraintes imposées. Le critère d'arrêt d'une telle méthode ne devra donc pas être choisi de cette façon. Plusieurs options sont par contre possible.

- arrêt au bout d'un certain nombre d'itérations,
- arrêt si stabilisation de la fonction coût (plus d'amélioration possible).

Dans le premier cas, on se donne un certain nombre de pas de calcul pendant les quels on essaie les différentes perturbations possibles afin d'améliorer la fonction coût.



Dans le second cas, on applique ces mêmes perturbations jusqu'à ce que l'on ne puisse plus améliorer de façon significative la fonction coût.



### II.1.c. La hiérarchie

Dans le paragraphe II.1.a. on a déjà remarqué qu'il était plus simple et plus efficace de résoudre un problème de placement initial en réalisant au préalable un ordonnancement sur les éléments et sur les places du circuit. Ce partitionnement revient en fait à simplifier un problème important en plusieurs problèmes plus simples. Il faut noter que cette simplification ne se fait pas au détriment de la qualité puisque ce sont à priori les cas de figures les moins intéressants pour le placement qui ne seront

pas pris en compte. Cette méthode de découpage du problème en sous problèmes est fréquemment utilisée et connue sous le nom de hiérarchisation [UED 83][KOZ 84].

Considérons une hiérarchie à deux niveaux. On a alors les deux étapes suivantes:

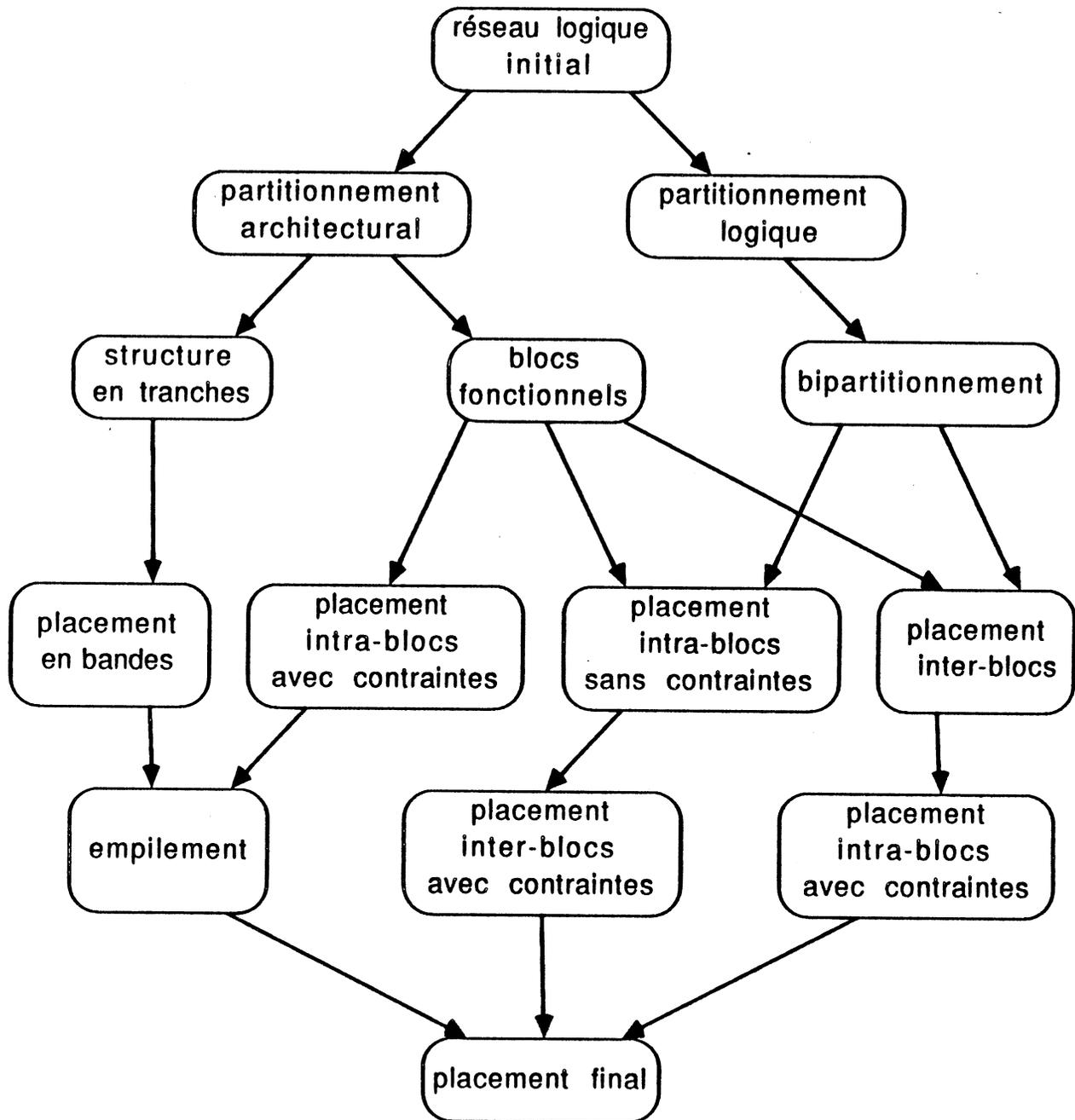
- placement des composants à l'intérieur de chaque classe ( ou bloc),
- placement de l'ensemble des blocs.

On peut considérer deux types de hiérarchie: le type ascendant qui consiste à réaliser la première étape avant la seconde. Cette méthode est ainsi appelée car on commence d'un niveau fin: les composants, pour finir par un niveau plus général: les blocs ou classes de composants.

Le deuxième type est évidemment le type descendant dans lequel la deuxième étape est réalisée avant la première.

On peut également envisager une hiérarchisation du placement à plusieurs niveaux.

En fait, le problème le plus important d'une telle méthode est le découpage du circuit en sous circuits. De ce découpage va dépendre la qualité finale du placement. Suivant les cas, différentes méthodes sont envisageables. Le graphe suivant montre l'ensemble des cas qui peuvent se présenter.



Chacun des chemins de ce graphe représente un choix possible de méthode d'implantation. Chacune des étapes sera détaillée par la suite.

On peut cependant remarquer les deux types importants de partitionnement :

- logique,
- architectural,

et les deux types de hiérarchie :

- ascendante
- descendante.

Le réseau logique: celui ci a déjà été amplement présenté dans le premier chapitre, il est donc inutile d'y revenir. Il faut juste noter que cette méthode de hiérarchisation n'est utile que lorsque ce réseau logique est trop important pour pouvoir être manipulé rapidement en entier.

Le partitionnement architectural: afin de diviser le réseau initial en sous-réseaux, plusieurs méthodes sont possibles. La méthode la plus intuitive est certainement le partitionnement dit architectural. Elle consiste à diviser le circuits en blocs homogènes au niveau fonctionnel.

Structure en tranches: dans ce cas le circuit est divisé en "tranches de bits". Dans chaque classe, on retrouve tous les composants concernant un même bit comme par exemple: buffer, point registre, opérateur arithmétique... Dans un cas tel que celui ci, toutes les classes sont représentées par des réseaux logiques identiques ou presque.

Cette méthode ne peut évidemment s'appliquer qu'aux circuits manipulant des valeurs numériques sur plusieurs bits. Cette méthode apparaît dans presque tous les circuits dessinés "au micron" mais beaucoup plus rarement dans les réseaux prédiffusés. Pourtant, si on regarde la structure des réseaux en technologie C-MOS, donc spécialement adaptés aux circuits numériques, on remarque un découpage en bandes parallèles. Ce découpage topologique naturel s'adapte particulièrement bien à un découpage architectural sous forme de tranches de bits. Une telle méthode a été expérimentée manuellement avec succès [CHA 85] mais elle est rarement (jamais?...) employée de façon automatique.

Le placement en bande: une fois une tranche définie, le placement à réaliser à l'intérieur d'une classe a ceci de particulier qu'il n'est plus à réaliser sur une structure

bidimensionnelle mais uniquement sur un axe. Pour ce cas, de nombreuses solutions originales ont été proposées dans [SER 82].

Les blocs fonctionnels: ce mode de partitionnement consiste à regrouper les éléments logiques du réseau initial par fonction. Ainsi, pour un circuit du type micro-processeur par exemple, on peut définir deux classes bien distinctes: une partie contrôle et une partie opérative à l'intérieur de laquelle on peut à nouveau différencier deux classes: partie mémorisation et partie calculatoire. Ce type d'opération de hiérarchisation peut être appliqué aussi souvent qu'il est nécessaire pour arriver à un problème suffisamment simple.

L'empilement: si les blocs ont été construits de façon homogène ce qui est le cas pour un découpage en tranches ou pour un découpage en blocs fonctionnels si une compatibilité a été préalablement établie au niveau des positions des points de contact de chaque bloc le placement relatif des blocs est grandement simplifié. Il suffit en effet d'empiler les blocs par simple juxtaposition. Il faut cependant remarquer que si cette méthode est particulièrement simple et efficace elle ne peut malheureusement pas être appliquée dans tous les cas.

Le placement intra-blocs: si la méthode de hiérarchisation choisie est du type ascendante, on commencera par réaliser le placement à l'intérieur des blocs. Dans ce cas il est possible d'imposer ou non des contraintes. Ces contraintes concernent la topologie de chaque bloc. On rejoint à ce niveau la notion de bloc mou ou de bloc dur. Si l'on impose ni la géométrie du bloc, ni la position des entrées/sorties propres à ce bloc, on sera en présence d'un bloc mou alors considéré comme circuit sans contraintes pour le niveau inférieur de la hiérarchie (voir chapitre I). Si au contraire, on impose ces différents paramètres, on sera ramené toujours pour le niveau inférieur de la hiérarchie, au placement sur circuit fortement contraint.

On peut remarquer que lors de la conception d'un circuit de nombreux composants sont standards, comme par exemple un registre ou un compteur. Dans ce cas, des bibliothèques d'éléments sont disponibles et cette étape du placement est

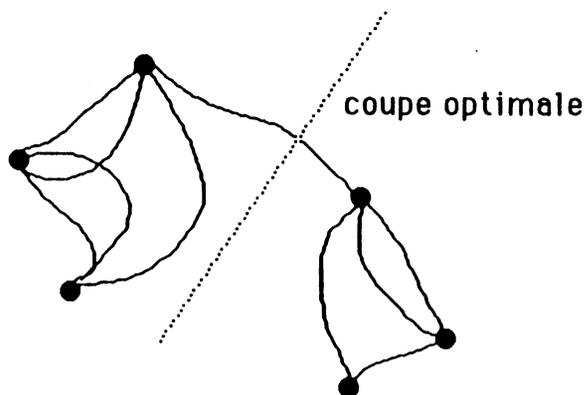
supprimée. On travaille alors sur des circuits du type précaractérisés (standard-cell).

Le placement inter-blocs: il s'agit là du domaine du placement de masse. Suivant les contraintes imposées ou non lors du placement intra-blocs et suivant qu'il s'agit d'une hiérarchie ascendante ou descendante, les blocs manipulés seront du type dur ou mou.

Le partitionnement logique: dans ce type de découpage seuls les critères de connectique sont pris en compte. Le but est d'obtenir des classes d'éléments fortement connectés de façon interne et très faiblement connectés de façon externe. De cette manière, on assure une très faible influence entre les classes. Ainsi les placements à l'intérieur de chaque classe peuvent se faire indépendamment les uns des autres.

La méthode de découpage logique la plus couramment employée est celle dite du bipartitionnement [PIE 85][KRI 83]. Elle consiste à partir du graphe logique et à le découper en sous graphes de sorte que le nombre d'arcs reliant deux sous graphes soit minimisé.

Exemple:



Il faut cependant remarquer que cette coupe doit se faire en conservant un équilibre entre les deux sous graphes obtenus. C'est à dire que les tailles respectives de ces sous-graphes doivent être équivalentes. Par taille il faut entendre nombre de sommets du graphe. On peut toutefois prendre en compte la taille réelle des blocs et

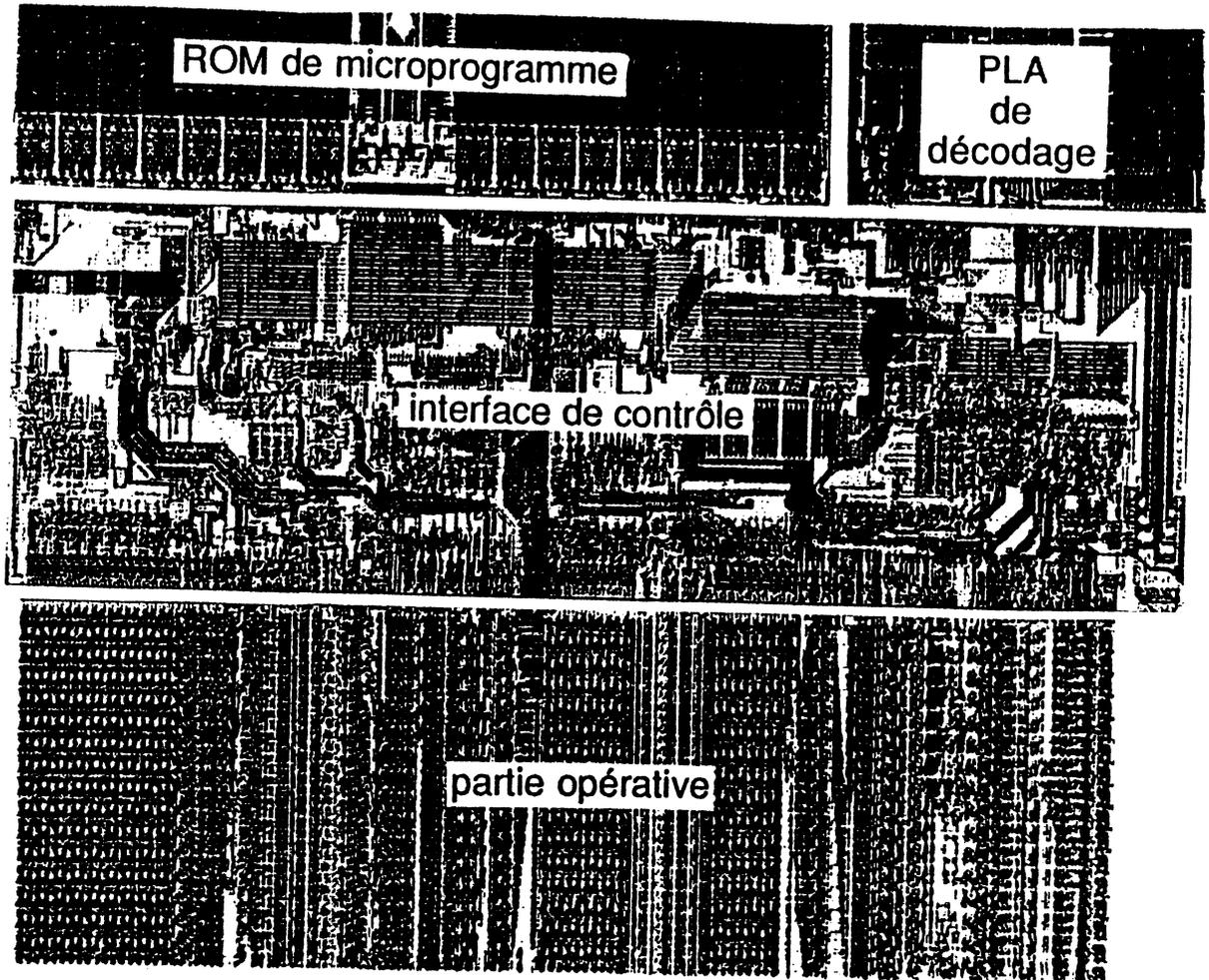
ainsi équilibrer les surfaces occupées par chaque sous-graphe. L'opération est recommencée jusqu'à ce que chaque sous-graphe soit limité à un élément.

Cette méthode a l'avantage d'être universelle et c'est celle qui est couramment employée de façon automatique. Il faut toutefois remarquer que dans un tel cas, on risque de perdre toute structure régulière d'un circuit et de nuire à l'efficacité du placement en ne bénéficiant pas des facteurs de répétition d'un motif.

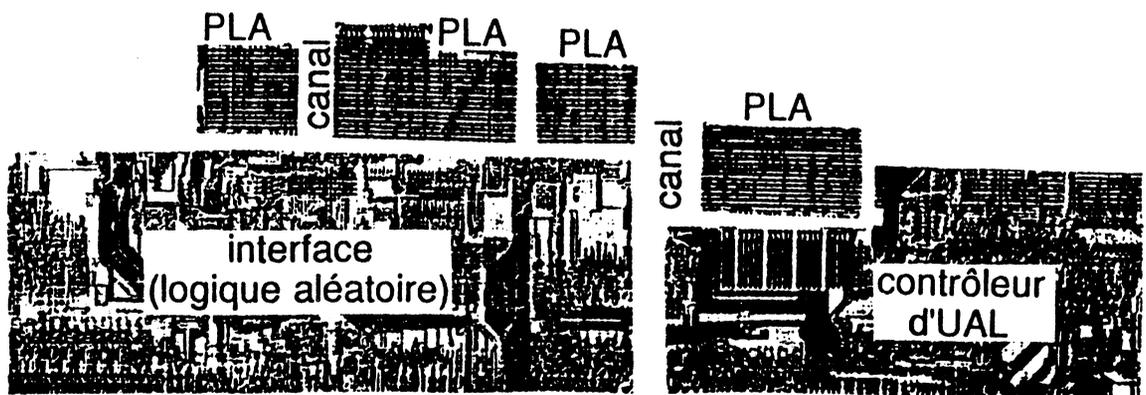
Une autre méthode de découpage logique sera présentée dans le chapitre III.

A partir d'un tel découpage, il est possible d'utiliser les méthodes classiques de placement qu'elles soient ascendantes ou descendantes.

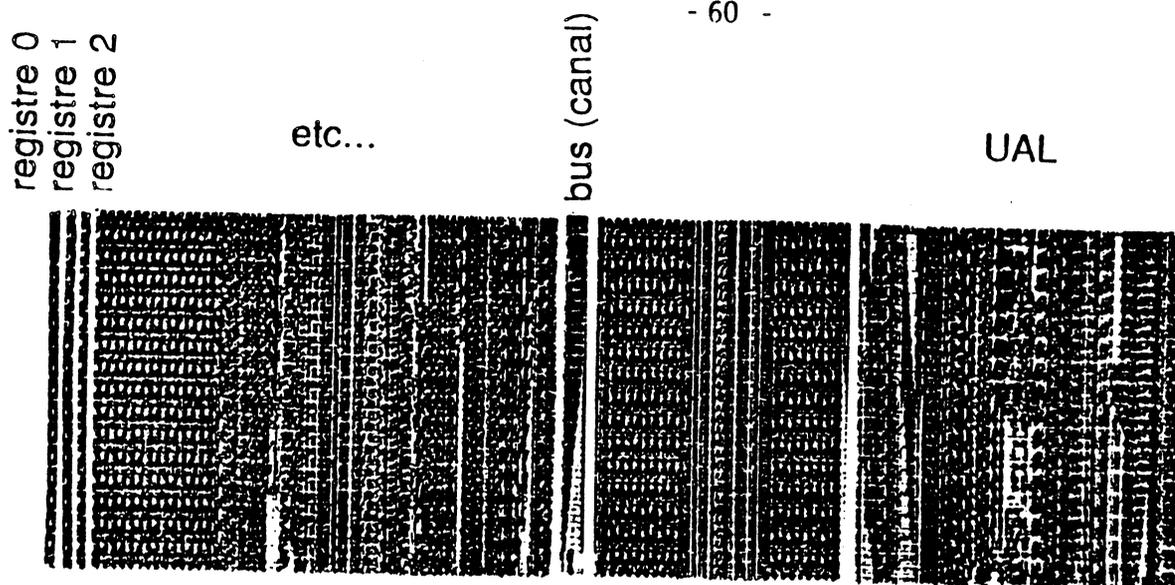
En fait la hiérarchisation n'introduit pas de méthodes de placement différentes. Elle se contente simplement de les utiliser à différents niveaux de la description d'un circuit. Il ne sera donc plus question de découpage hiérarchique dans la suite mais uniquement des différentes méthodes proposées pour résoudre les différents types de problèmes exposés plus haut.



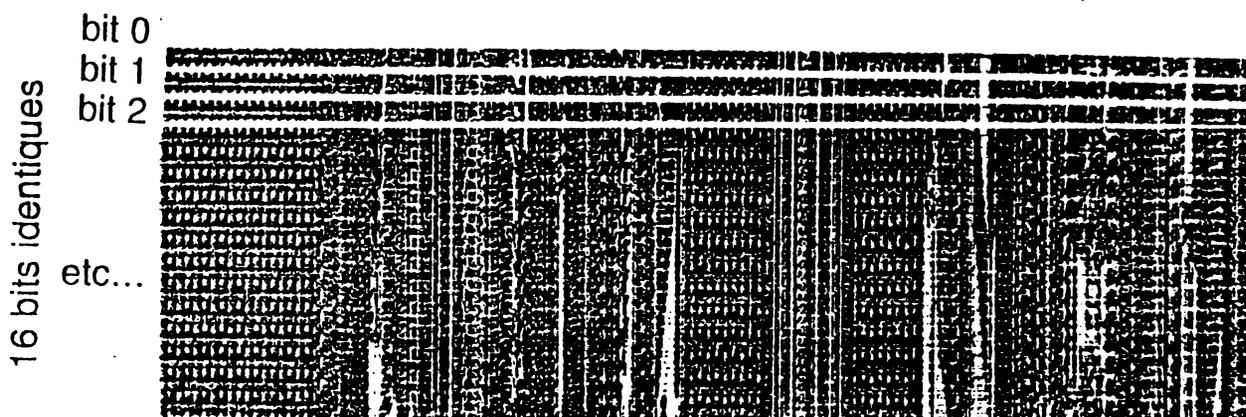
découpage architectural d'une partie du microprocesseur 68000



découpage de l'interface de contrôle



découpage de la partie opérative en unités fonctionnelles



découpage de la partie opérative en tranches de bits

## II.2. Les méthodes

Ce paragraphe se contente de passer rapidement en revue les principales méthodes de placement utilisées. De très nombreuses bibliographies existe sur ce sujet et on trouvera dans [TSI 84],[PER 85] ou [ZEE 85] une étude plus approfondie, précisant notamment les complexités des différents algorithmes employés.

### II.2.a. Les méthodes de placement initial

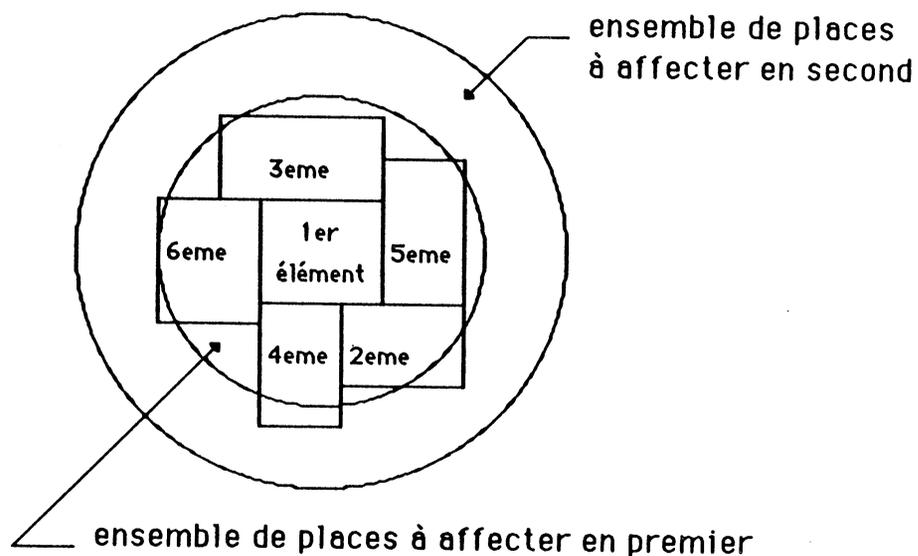
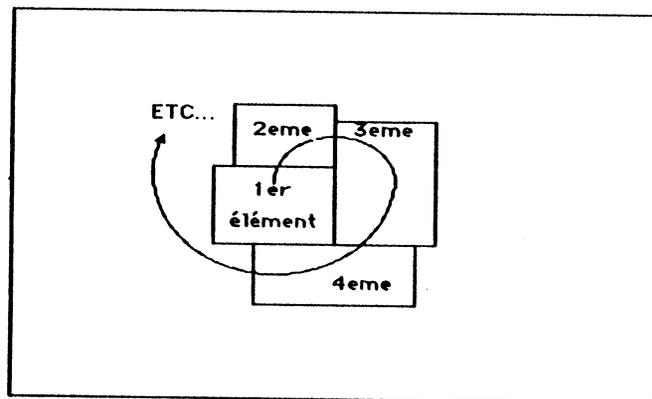
Comme il a été dit plus haut, le placement initial est généralement effectué par une méthode dite constructive.

On pourra différencier grossièrement trois approches:

- le placement constructif par agrégat,
- le placement constructif par barycentres,
- le placement par bi-partitionnement.

Dans le premier cas, l'approche utilise un ordonnancement topologique du circuit ou éventuellement un ordonnancement mixte [JAN 85][KOZ 83].

L'idée consiste à placer un élément au centre du circuit puis à remplir progressivement les places contiguës en progressant, selon une spirale si l'ordre topologique est complètement défini, ou selon des anneaux concentriques si seul un ordre sur des classes de places a été établi.



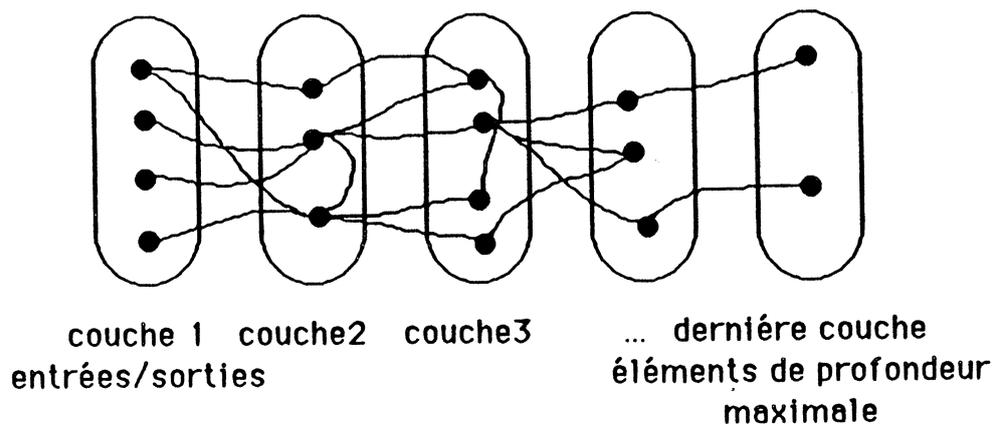
Dans le cas d'un ordonnancement mixte, le problème est de trouver l'ordre sur les éléments. Pour ceci on a souvent recours au "graphe en couche". Cette méthode permet de construire un ensemble de classe d'éléments ordonnées en fonction de leur éloignement des points d'entrée/sortie. Cet éloignement est défini à partir du graphe logique. La construction se fait de la manière suivante:

1°- Dans la première classe, sont rangés tous les éléments du type plot d'entrée/sortie.

2°- Dans la seconde classe, sont rangés tous les éléments possédant au moins une connexion avec les éléments de la couche précédente.

3°- On recommence cette opération jusqu'à avoir classé tous les éléments.

On obtient alors un graphe du type :

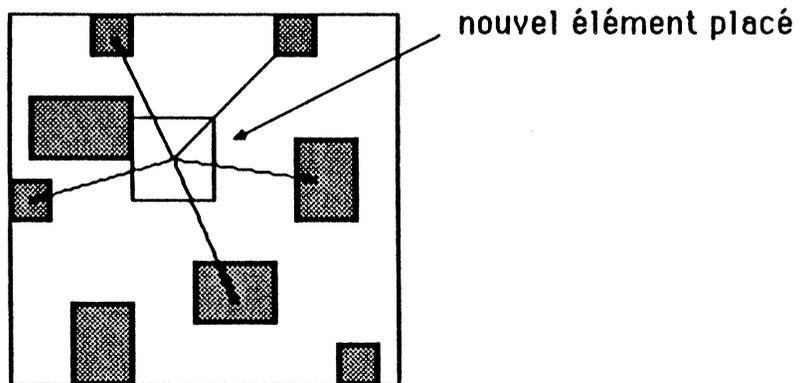


Si la construction par agrégat commence à partir du centre du circuit, ce qui est généralement le cas, on placera les classes d'éléments par ordre décroissant de leur profondeur. On peut toutefois imaginer des agrégats commençant par les bords du circuit. Auquel cas, l'ordre de traitement des classes d'éléments est inversé [HOR 81].

Dans le second cas, l'approche utilisée est du type ordonnancement logique. Le principe consiste à placer successivement les éléments sur la structure hôte de sorte à réduire l'ensemble des forces d'attraction avec les éléments déjà placés. Pour ceci on

place chaque nouvel élément le plus proche possible du barycentre de tous les éléments déjà placés auxquels il est connecté [BOW 83][MOU 84].

Exemple:

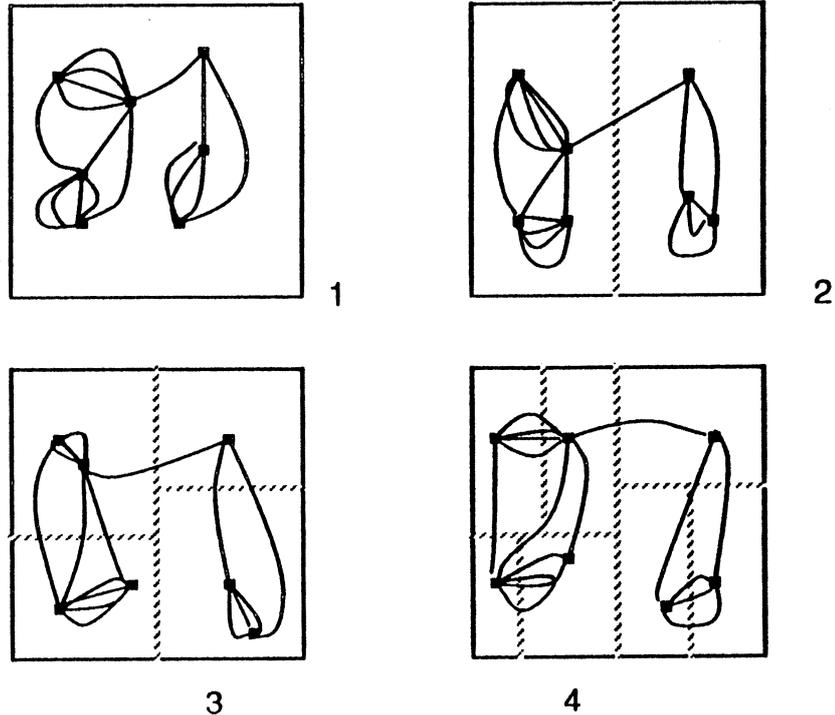


En fait, chaque connexion agit comme un "élastique" et le bloc est placé au point d'équilibre de toutes les forces interagissant.

Cette méthode ne peut évidemment être utilisée que si l'on possède un point de départ. Ce point de départ est généralement donné par la position des points d'entrée/sortie.

Dans le cas du placement par bipartitionnement exposé plus haut, il est possible de faire évoluer un découpage topologique en même temps que le découpage logique. Pour ceci, chaque ligne de coupe logique est associée à une ligne de coupe topologique dans la structure hôte. Ces lignes de coupe peuvent être parallèles ou perpendiculaires, c'est à dire alternativement horizontales et verticales. En pratique, c'est cette seconde méthode de coupes alternées qui permet d'obtenir les meilleurs résultats.

Exemple:



Cet exemple montre l'association entre les lignes de coupe dans le réseau logique et les lignes de coupes topologiques dans la structure hôte:

1: réseau logique initial et structure hôte.

2: premier bipartitionnement du graphe associé à une ligne de coupe verticale de la structure hôte.

3 et 4: nouvelles coupes logiques et topologiques.

Des variantes peuvent être introduites dans cette méthode afin d'adapter au mieux les coupes en fonction de la structure du circuit.

### II.2.b. Les méthodes d'amélioration

Comme il a été dit dans le paragraphe de présentation des méthodes d'amélioration, cette étape est généralement réalisée en appliquant diverses perturbations sur l'ensemble des composants déjà placés. Il est évident que ces

perturbations dépendent fortement du type de structure manipulée: faiblement ou fortement contrainte (voir chapitre I).

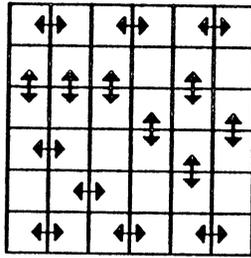
La première idée consiste à essayer méthodiquement chaque perturbation sur chaque composant. Pour chacun de ces cas, on évalue le gain. Ce gain correspond à la différence de la fonction coût entre les deux placements: avant et après perturbation. Si le gain est négatif, c'est à dire si la modification détériore le placement, celle-ci est abandonnée. Si par contre le gain est positif, la modification est effectivement prise en compte.

Peu d'originalités peuvent être apportées dans ce domaine si ce n'est l'utilisation de machines parallèles spécialisées [BRE 83][IOS 83][UED 83].

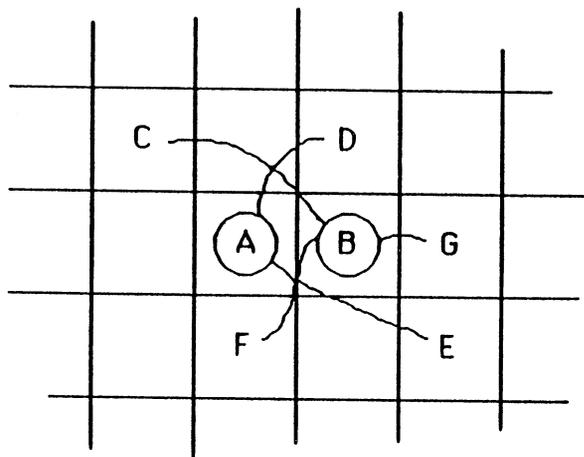
Dans ce type de machine on associe un processeur spécialisé à chaque place du circuit. Un élément est alors affecté à chaque place et donc à chaque processeur. Cette affectation correspond au pré-placement. Elle est soit réalisée après une étape de placement constructif soit simplement aléatoire.

Les processeurs évaluent alors le coût local dû au bloc qui lui est affecté. L'amélioration se fait alors en réalisant des permutations locales entre éléments: les processeurs sont associés par paires;chaque paire évalue le nouveau coût local en cas de permutation des composants qui leur sont affectés . Cette évaluation est réalisée à partir de la différence des forces d'attraction appliquées sur chaque composant en fonction de leur positions respectives.

Exemple:



réseaux de processeurs et différents types d'association par paires



Gain si permutation:

A +2  
B +2 -1

total +3

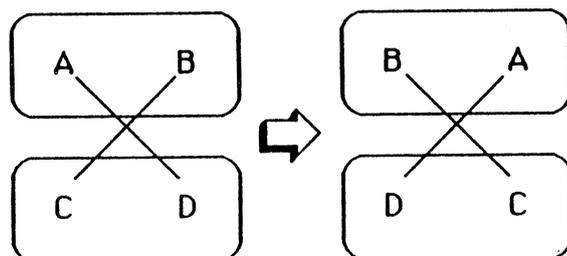
⇒ permutation acceptée

L'opération se répète jusqu'à obtenir une stabilisation de l'ensemble des éléments sur le réseau de processeurs. Pour ceci, à chaque nouveau pas de calcul, les processeurs sont associés de façon différentes. Ces associations doivent permettre à n'importe quel composant de se déplacer à travers tout le réseau.

Quelques variantes ont été apportées à ce type de machines parmi lesquels on pourra noter une méthode de repliement torique du plan de processeurs afin que chacun puisse traiter plusieurs places et donc plusieurs éléments.

Cette méthode est cependant extrêmement limitée:

- nombre de processeurs réduit d'où structure hôte limitée,
- places régulières et composants de tailles uniformes,
- aucune perturbation possible autre que la permutation,
- problème d'oscillations:



Il faut de plus remarquer que si les calculs sont extrêmement rapides grâce au grand nombre de processeurs, chaque permutation doit s'accompagner d'un transfert d'information vers tous les autres processeurs de façon à fournir les nouvelles coordonnées de chaque élément à tous ceux qui lui sont connectés. Comme on ne pourrait décemment envisager une liaison entre chaque couple de processeurs, un bus commun permet de réaliser ces transferts de données. On imagine rapidement la quantité d'information circulant sur un tel bus et on constate qu'une grande partie du temps gagné lors de l'étape de calcul est perdu en transferts de données. Ceci explique pourquoi la majeure partie des machines parallèles réalisées pour une telle application sont restées à l'état de prototypes de laboratoires [CHA 84].

On trouvera dans [BLA 84] un catalogue complet des machines parallèles utilisées pour la conception automatisée de circuits.

Une autre méthode, beaucoup plus intéressante, s'est développée ces dernières années. Il s'agit de la méthode des recuits successifs plus connue sous le nom "d'annealing" [KIR 83][GAY 85]. Cette méthode est dite stochastique. C'est à dire que l'ensemble des perturbations que l'on peut introduire sont appliquées aléatoirement à l'ensemble des blocs. Le but est en fait d'assimiler le système des blocs et de leurs interactions dues à la connectique avec un système de particules interagissant entre elles par leurs forces d'attraction. A ce stade, on fait appel aux théories de la mécanique statistique qui dit qu'un tel système doit tendre vers un état d'équilibre correspondant à une énergie minimale de l'ensemble des particules. L'obtention de cette énergie minimale revient à minimiser l'ensemble des forces d'attraction internes au système. L'analogie avec le problème du placement est assez évidente.

Cette technique est utilisée avec succès dans de nombreux systèmes [SIA 85] [WON 86]. On pourra cependant regretter les temps de calculs très longs nécessaires à la génération d'une bonne solution.

Bien que l'efficacité (plus ou moins bonne) de toutes ces méthodes ne soit plus à prouver, on peut toutefois regretter qu'elles dépendent énormément du point de départ. Autrement dit, suivant l'initialisation du problème, qui est toujours faite de façon plus ou moins aléatoire, les résultats finaux pourront être assez médiocres. Il existe une méthode tout à fait différente qui a été présentée par Otten [OTT 80] il y a déjà quelques années. Cette méthode s'appuie sur des bases mathématiques de traitement de matrices et permet une représentation planaire d'un ensemble d'individus décrits par de simples corrélations internes (la méthode complète sera décrite dans le chapitre suivant). Cette méthode semble tout à fait optimale en théorie et on peut s'étonner de ne pas l'avoir vu se développer plus rapidement. En fait, le pas est assez important entre la théorie et la pratique puisque, contrairement aux autres méthodes qui ont été établies à partir d'un problème donné, on dispose dans ce cas d'une méthode efficace permettant de résoudre des problèmes voisins et qu'il faut tenter d'adapter sans quoi les résultats pratiques sont tout à fait médiocres. Cette étape est tout à fait fondamentale et c'est sur ce point que l'étude décrite dans le chapitre suivant a été menée.

**CHAPITRE III**

**LES METHODES STATISTIQUES**  
**ET**  
**LA GENERATION DE PLANS DE MASSE**



### III. Les méthodes statistiques

L'analyse de données est un ensemble de méthodes statistiques permettant de traiter et d'étudier certains tableaux de valeurs numériques. En particulier les tableaux qui décrivent une liste d'individus en fonction d'un ensemble de caractères. C'est à dire qu'à chaque couple individu, le caractère est associé à une variable numérique. Le but de l'analyse factorielle est de mettre en évidence, à partir de ces données, des informations supplémentaires concernant les corrélations (ou les proximités) entre les individus ou entre les variables. Par exemple, chaque personne d'une population peut être décrite par les variables suivantes: âge, sexe, poids, taille... L'analyse factorielle permettra alors de regrouper les individus par similitude ou d'établir des relations entre les différentes variables (entre le poids et la taille par exemple). De nombreux ouvrages traitent de ce problème, parmi eux, on peut citer [LEB 82] et [FOU 82]. Dans le cas qui nous intéresse, nous examinerons plus précisément les deux méthodes suivantes: l'analyse d'un tableau de distance et la classification.

#### III.1. L'analyse factorielle d'un tableau de distances.

L'analyse d'un tableau de distance que l'on note ATD, est un cas particulier de l'analyse en composantes principales qui a pour but de représenter sous forme graphique un ensemble d'individus ou de variables de sorte que deux individus (ou variables) soient d'autant plus proches qu'ils se ressemblent. L'analyse en composantes principales (notée ACP) concerne un tableau du type individus X variables.

Dans le cas de l'ATD, les données de départ sont fournies sous la forme d'un tableau de distances.

$$T = \begin{pmatrix} & & & & j \\ & & & & \vdots \\ i & \dots & d(i,j) & & \end{pmatrix}$$

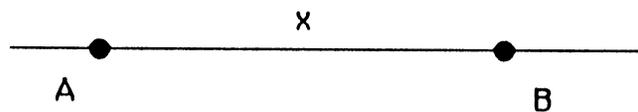
Dans un tel type de tableau, à chaque couple d'individus  $(i,j)$  ( $i,j = 1,2,\dots,n$ ), est associée une distance. Cette notion de "distance" signifie ici que les variables du tableau ont les propriétés suivantes:

- $d(i,j) \geq 0$
- $d(i,j) = d(j,i)$
- $d(i,i) = 0$

Il est donc possible de fournir un tableau de distances réelles; c'est à dire dans lequel chaque distance vérifie l'inégalité triangulaire mais il est également possible de créer un tableau dans lequel on aura défini pour chaque couple, un indice de distance d'autant plus faible que les individus se ressemblent. Le but de l'ATD est de fournir une représentation euclidienne [FOU 82] de l'ensemble des individus dans un espace de dimension réduite de sorte que les distances du tableau soient respectées au mieux. L'espace choisi pour la représentation des individus sera généralement de dimension 2 pour sa facilité de "lecture". On obtiendra alors une "carte" des individus.

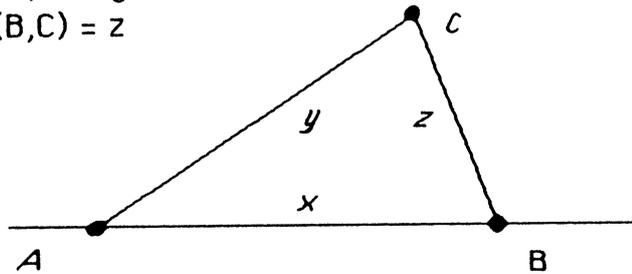
En fait, cette représentation sous forme de tableau de distances correspond à une représentation des  $n$  individus dans un espace de dimension  $n-1$ . En effet, pour pouvoir positionner deux individus ayant une contrainte de distance mutuelle, il suffit d'un espace de dimension 1.

$$d(A,B) = x$$

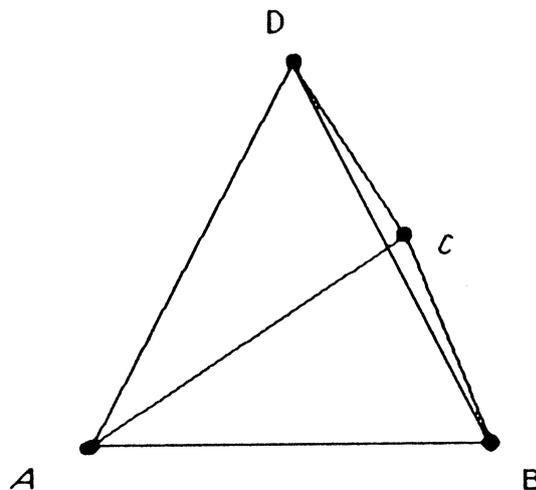


Par contre, si on veut placer un troisième élément ayant une contrainte de distance avec les deux éléments déjà placés, on sera généralement obligé de choisir une représentation dans un espace de dimension 2.

$$d(A,C) = y$$
$$d(B,C) = z$$



De la même façon, si on ajoute un troisième point, on sera obligé d'ajouter une dimension à l'espace de représentation afin de pouvoir vérifier contraintes avec les individus déjà placés. Ainsi, pour représenter fidèlement un nuage de  $n$  points, on a dans le pire cas, besoin d'un espace de dimension  $n-1$ .



### **Théorie de l'ATD**

Cette représentation d'un ensemble d'individus dans un espace de dimension réduite revient à projeter un nuage de points sur le plan qui déforme le moins possible ce nuage. Le problème consiste donc à déterminer le repère orthonormé du plan de projection également appelé plan factoriel.

La théorie utilisée pour la résolution de ce problème ne sera pas exposée ici. Elle est suffisamment connue des statisticiens et n'intéressera guère les micro-électroniciens. On pourra toutefois trouver des informations complètes dans [FOU 82] et [LEB 82].

Il suffit de savoir que le plan factoriel est défini par les vecteurs propres associés

aux deux plus grandes valeurs propres de la matrice des distances. Les coordonnées des points sur le plan sont données par les composantes de ces deux vecteurs.

Cette représentation est d'autant plus réaliste que la deuxième valeur propre est plus grande que la troisième. On peut ainsi définir un critère de qualité de la représentation sur le plan factoriel par la fonction suivante:

$$I = \frac{\lambda_1 + \lambda_2}{\sum_{i=1}^n \lambda_i}$$

Où  $\lambda_i$  est la  $i^{\text{ème}}$  Valeur propre.

Il faut cependant remarquer que les valeurs propres ne sont toutes positives que si la matrice de distances fournie vérifie les propriétés d'une distance euclidienne. Ceci n'étant pas toujours le cas, l'expression précédente n'a plus de sens, on la remplacera alors par:

$$I = \frac{\lambda_1 + \lambda_2}{\sum_{i=1}^n |\lambda_i|}$$

Cette fonction de qualité représente l'inertie du plan factoriel par rapport au nuage de points. Une inertie de 100% , et l'ensemble des valeurs propres positives signifie une représentation parfaite du nuage de points, c'est à dire que toutes les distances du tableau sont respectées dans le plan.

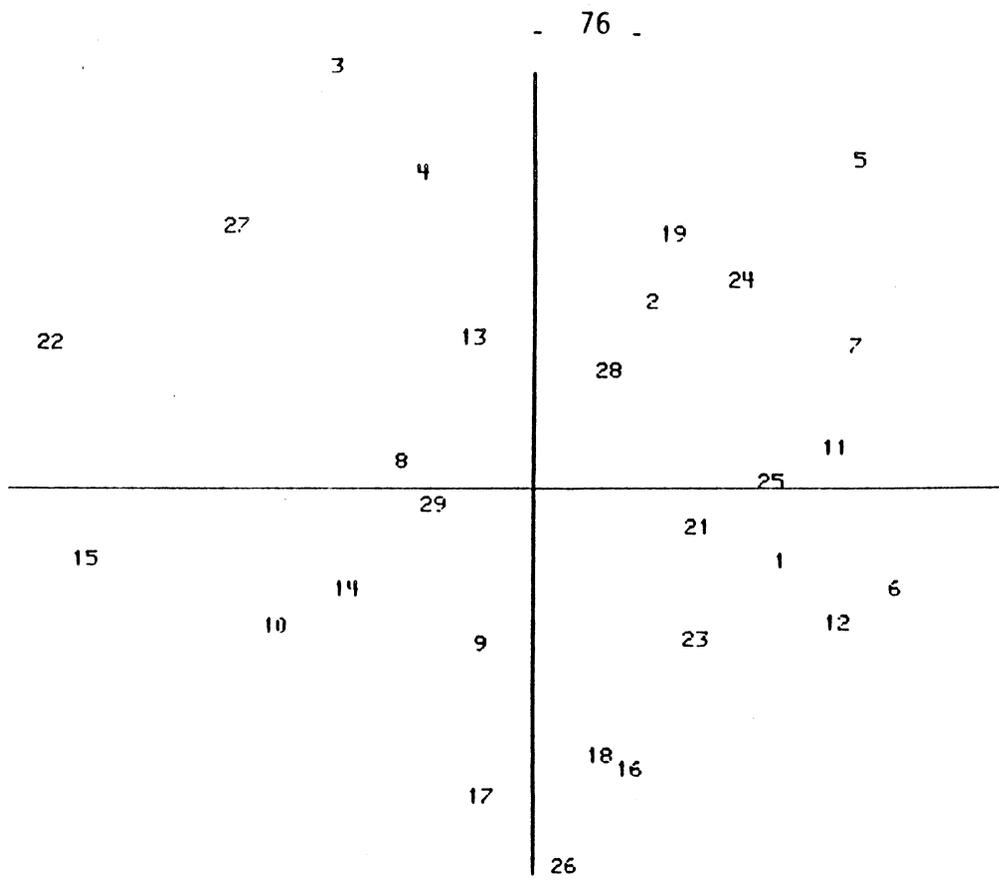
En résumé, l'analyse d'un tableau de distances va permettre d'obtenir une représentation planaire d'un ensemble de points dont seules sont connues les distances deux à deux. Cette opération se réduit à une diagonalisation de matrice pour laquelle de nombreux algorithmes ont été développés.

Afin d'illustrer cette méthode et d'en montrer l'efficacité, il est bon de donner un exemple. Considérons donc un tableau de distances tel qu'on en trouve sur les guides routiers.

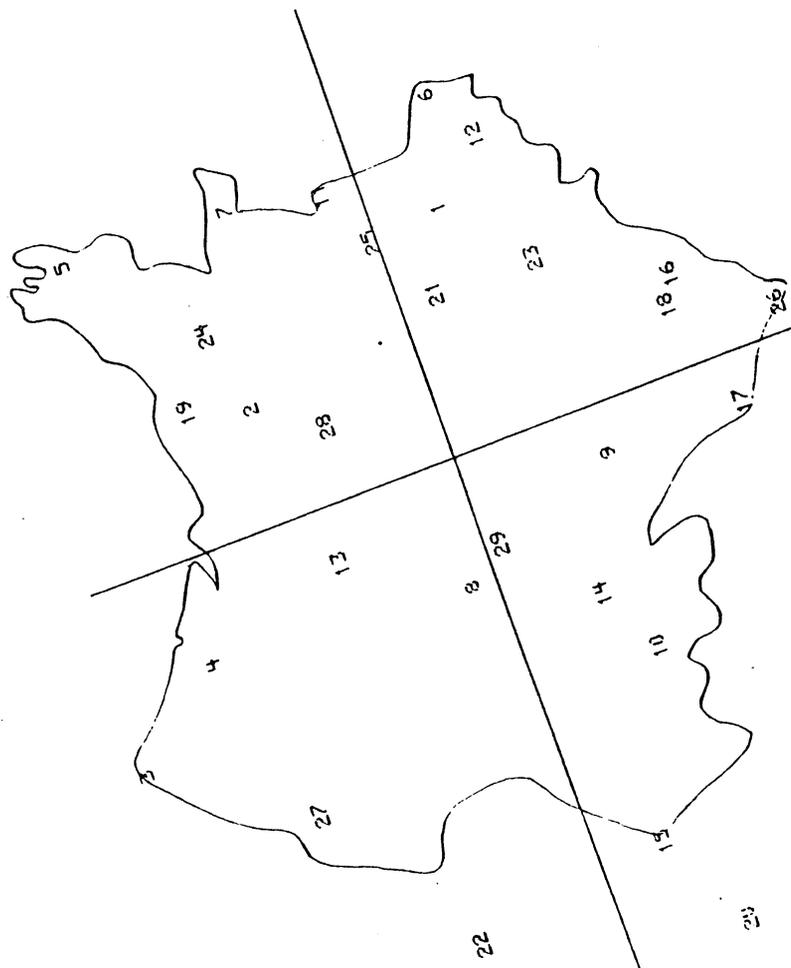
ANGERS	BIARRITZ	BORDEAUX	BREST	CALAIS	CHERBOURG	CLERMONT F.	DIJON	GRENOBLE	LE HAVRE	LILLE	LIMOGES	LYON	MARSEILLE	METZ	MULHOUSE	NANCY	NANTES	NICE	PARIS	PERPIGNAN	REIMS	RENNES	ROUEN	STRASBOURG	TOULOUSE	TOURS	VICHY			
399	862	679	619	159	766	524	417	674	180	113	571	587	910	149	559	358	489	1067	170	1041	167	411	117	490	927	352	485	AMIENS		
518	335	371	494	290	402	498	644	297	503	246	538	842	621	722	602	90	999	303	767	454	176	282	742	552	106	399	ANGEAIS			
183	817	997	808	555	815	820	800	975	407	735	697	1020	997	1001	512	879	744	445	805	618	785	1141	308	510	609	609	BIARRITZ			
634	814	625	369	632	655	617	792	219	549	662	837	814	818	329	844	561	460	702	435	602	958	250	327	412	761	412	BORDEAUX			
	714	402	752	812	996	468	723	602	890	1104	899	1024	886	305	1351	581	1094	734	245	502	1026	884	456	761	633	633	BREST			
		461	672	556	840	275	109	661	753	1076	438	670	469	584	1233	274	1209	278	506	212	609	967	487	533	639	639	CALAIS			
		647	661	874	215	470	552	768	1072	663	811	655	310	1229	340	1068	481	204	249	795	858	351	639	639	639	639	CLERMONT F.			
			280	286	586	613	184	180	463	516	481	472	489	820	382	464	463	507	498	574	389	296	59	59	220	220	220	DIJON		
			204	547	471	396	197	570	226	216	192	588	677	323	653	278	567	454	309	644	392	392	392	392	392	392	392	GRENOBLE		
			776	755	470	106	277	320	412	476	713	337	557	449	562	751	688	505	535	538	538	538	538	538	538	538	538	LE HAVRE		
			204	548	676	980	502	692	511	357	1137	211	1062	320	269	88	651	852	230	547	547	547	547	547	547	547	547	LILLE		
			634	668	991	353	585	384	593	1148	273	1124	193	515	221	524	940	465	574	574	574	574	574	574	574	574	574	574	LIMOGES	
			364	622	647	597	605	297	779	375	516	500	372	458	690	306	201	225	225	225	225	225	225	225	225	225	225	225	LYON	
			323	433	341	389	607	480	470	456	475	645	588	434	542	432	159	159	159	159	159	159	159	159	159	159	159	159	MARSEILLE	
			756	684	712	911	197	793	376	798	949	892	757	412	736	457	456	456	456	456	456	456	456	456	456	456	456	456	METZ	
			239	57	711	913	312	884	182	678	414	161	900	510	456	456	456	456	456	456	456	456	456	456	456	456	456	456	MULHOUSE	
				178	812	749	467	797	332	779	604	106	883	611	421	421	421	421	421	421	421	421	421	421	421	421	421	421	NANCY	
					892	869	308	845	191	659	423	140	856	491	412	412	412	412	412	412	412	412	412	412	412	412	412	412	NANTES	
									1068	393	789	544	1008	372	832	579	196	468	468	468	468	468	468	468	468	468	468	468	NANTES	
																													NICE	
																													PARIS	
																													PERPIGNAN	
																													REIMS	
																													REIMS	
																													RENNES	
																													RENNES	
																													ROUEN	
																													ROUEN	
																													STRASBOURG	
																													TOULOUSE	
																													TOULOUSE	
																													TOURS	
																													VICHY	

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

Les distances fournies correspondent aux distances routières mais il est raisonnable d'estimer qu'elles fournissent une bonne approximation de l'éloignement ou de la proximité des villes. Si on fournit ce tableau en donnée d'un programme d'ATD on obtient le résultat suivant:



Ce qui, moyennant une rotation permet de retrouver la carte suivante :



Constatations:

On remarque tout de suite que l'ATD a permis de fournir assez fidèlement le résultat que l'on espérait. Ceci moyennant une symétrie et une rotation ce qui n'est dû qu'à l'arbitraire des orientations cartographiques. Quelques perturbations ont été introduites pour la raison expliquée plus haut. Par exemple Nice est déporté au sud car les liaisons nord-sud traversent les Alpes introduisent un écart important entre la distance aérienne et la distance routière tandis que les distances est-ouest sont plus proches de la réalité. Après cette courte diversion touristique, il est temps d'aborder l'application de l'analyse factorielle d'un tableau de distances au domaine du placement.

### **III.2. Application de l'ATD au problème de la génération de plan de masse.**

Comme il a été montré plus haut, l'ATD permet une représentation planaire la plus fidèle possible d'un ensemble de points décrit par une matrice de distances. Si maintenant, on regarde le problème du placement, la donnée de départ est un graphe de connexion, c'est à dire un ensemble d'éléments liés deux à deux par des équipotentiels. Le but est d'obtenir une représentation planaire de ces éléments de façon à respecter les contraintes dues aux connexions. C'est à dire que l'on cherche à rapprocher les éléments en proportion du nombre d'équipotentiels qui les relie. La similitude avec l'ATD est évidente. Le problème est finalement le suivant:

- à partir d'une matrice de connexion, trouver une fonction permettant d'obtenir une matrice de distances.

- Soit  $E = \{ e_1, e_2, \dots, e_n \}$  l'ensemble des éléments à placer.
- Soit  $C$  la matrice de connexion telle que  $C_{ij} = l$ , où  $l$  est le nombre de liaisons (équipotentiels) entre  $e_i$  et  $e_j$ . On peut lui associer une matrice de distances par l'intermédiaire d'une fonction  $f$  telle que:

$$f : E \times E \rightarrow \mathbb{R}$$

$$(e_i, e_j) \mapsto d(i, j) = f(l)$$

La distance choisie doit être compatible avec les critères de qualité du placement. Dans un premier temps la fonction a été choisie inversement proportionnelle au nombre de connexions:

Soit :

$$d : E \times E \rightarrow \mathbb{R}$$

$$(e_i, e_j) \mapsto d(e_i, e_j) = \begin{cases} \frac{1}{C_{ij}} & \text{si } i \neq j \\ 0 & \text{sinon} \end{cases}$$

Afin d'éviter d'obtenir une valeur infinie, il est raisonnable de se ramener à l'intervalle [0,1]. La fonction de calcul de distance devient alors:

$$d' : E \times E \rightarrow [0, 1]$$

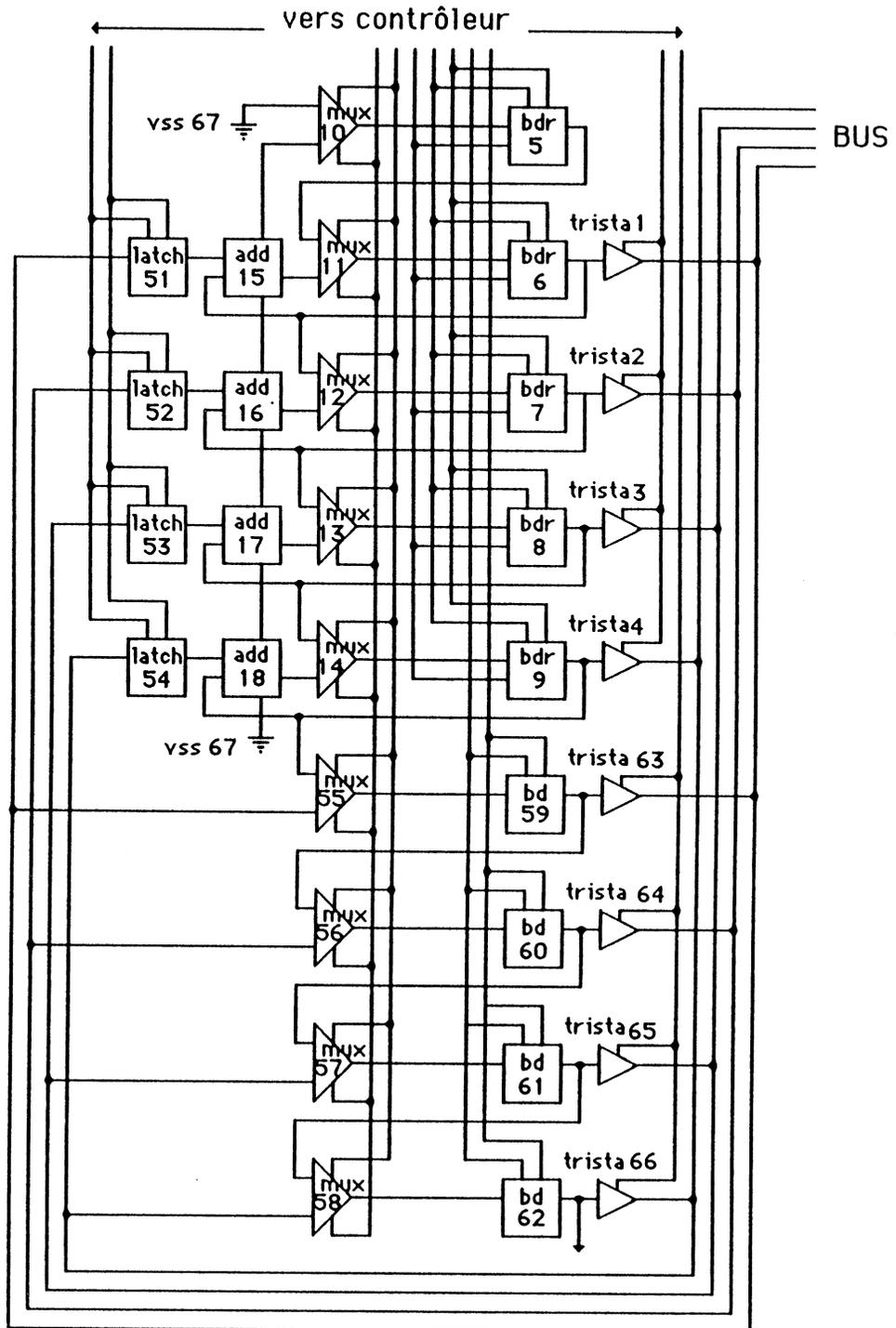
$$(e_i, e_j) \mapsto d'(e_i, e_j) = \begin{cases} \frac{1}{1 + C_{ij}} & \text{si } i \neq j \\ 0 & \text{sinon} \end{cases}$$

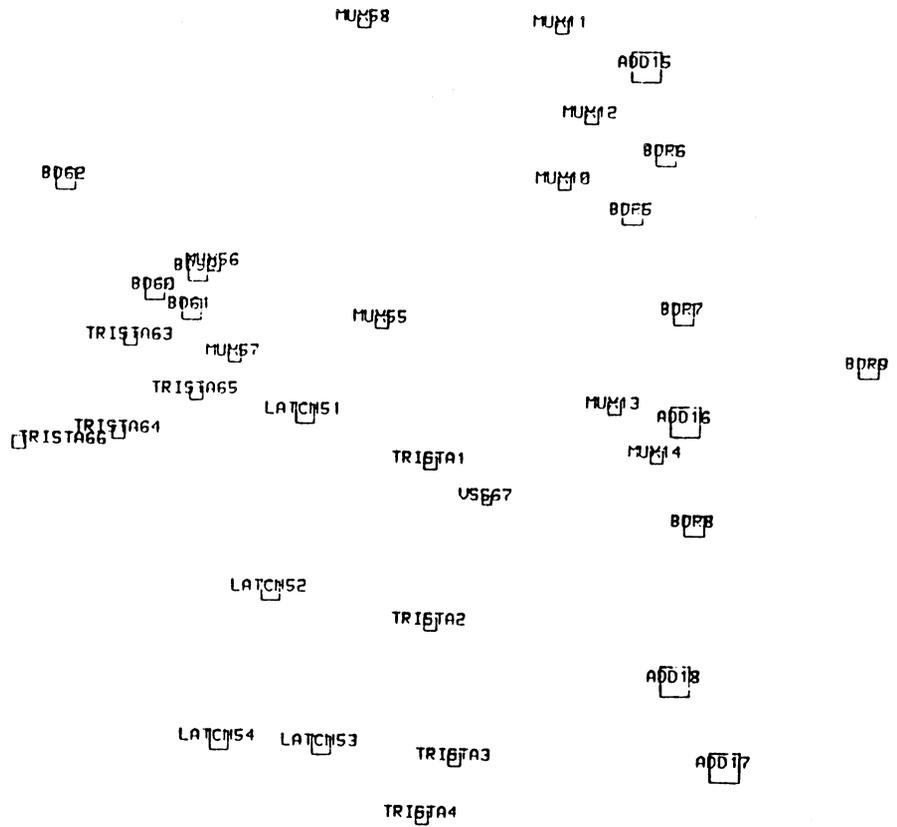
Cette distance correspond effectivement à la contrainte primordiale du placement, c'est à dire rapprochement des éléments proportionnel à leur interconnectivité et éloignement maximal des éléments non liés. Cette première fonction d'évaluation des distances a permis de valider la méthode en l'appliquant à de nombreux exemples.

En voici quelques uns:

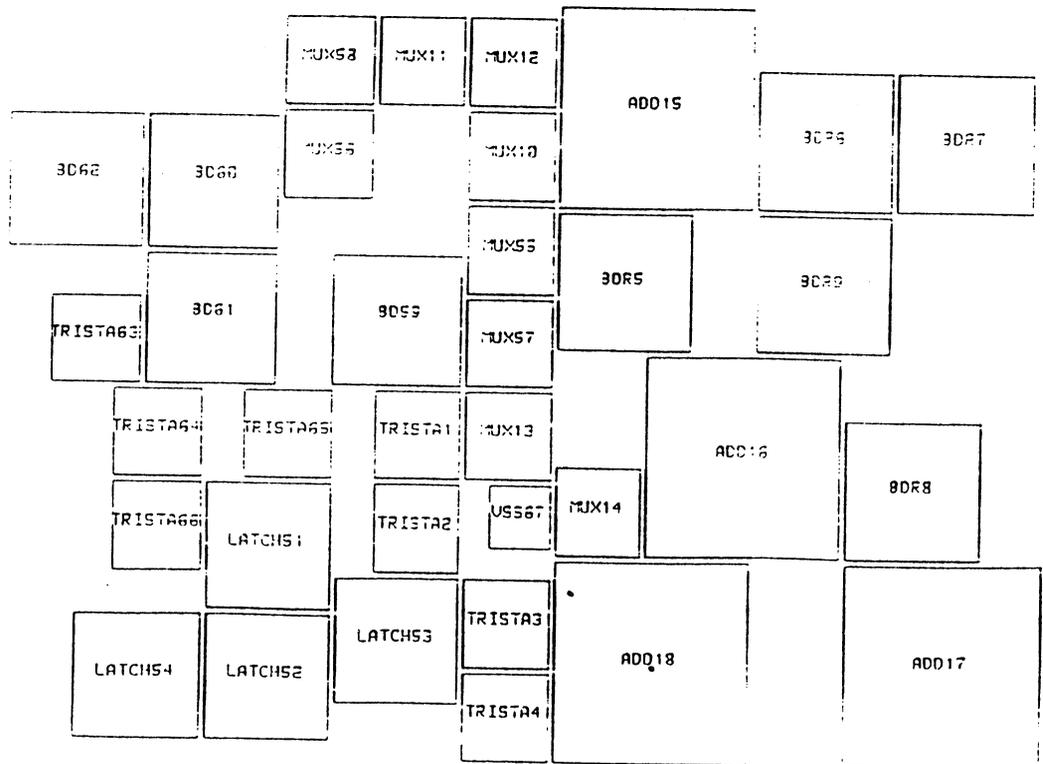
Exemple 1: circuit POMULT (partie opérative d'un multiplieur).

Ce circuit contient 34 blocs de type mou.



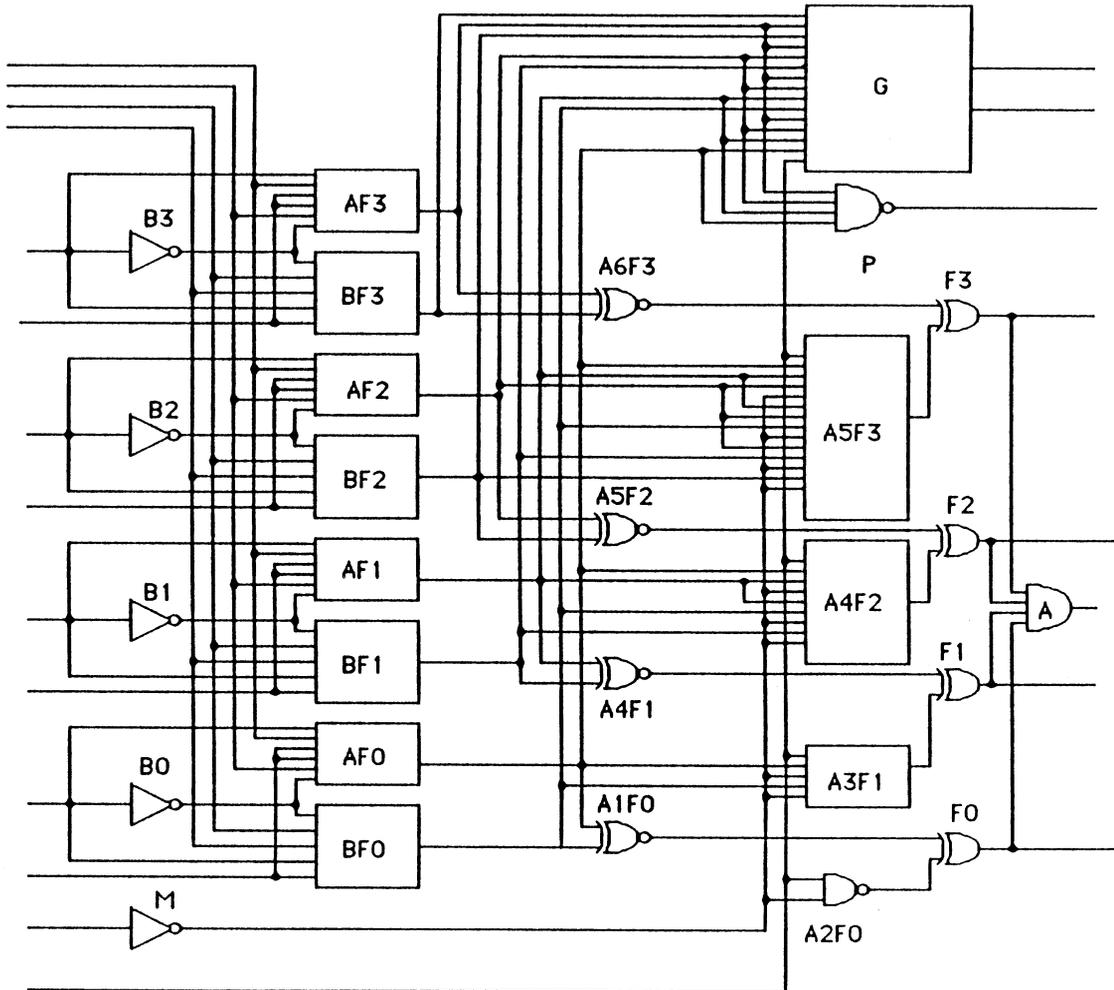


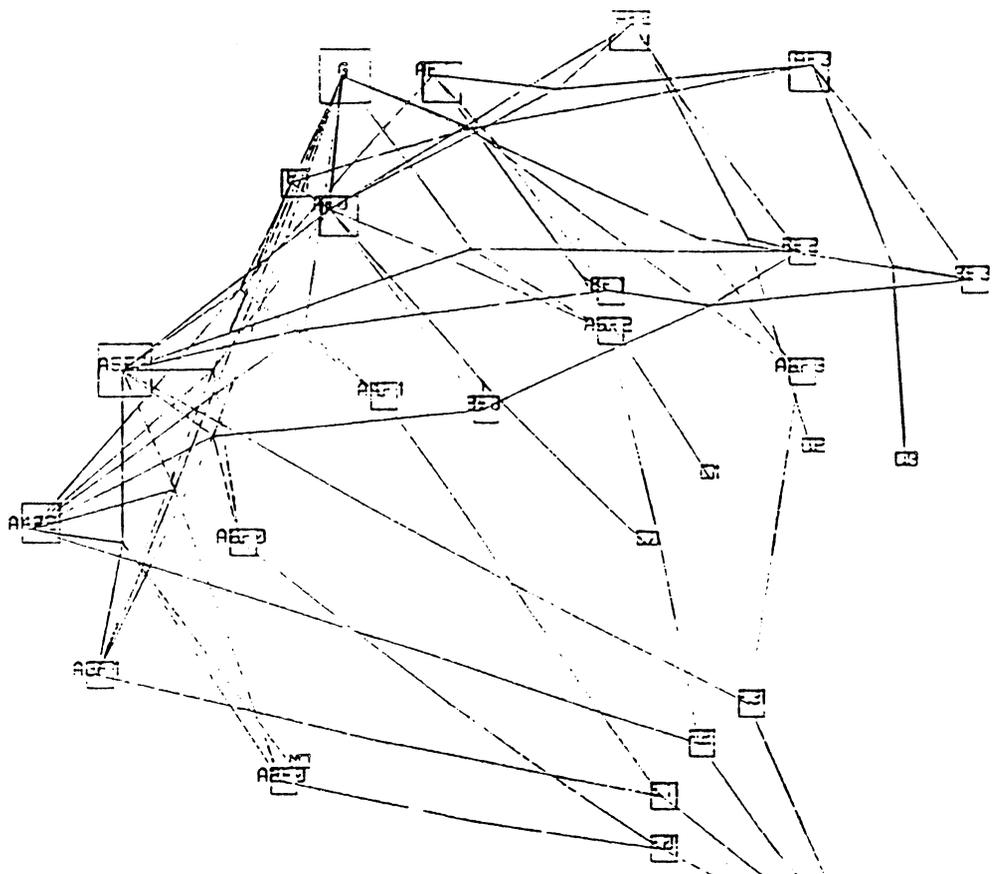
résultat obtenu en sortie d'analyse factorielle



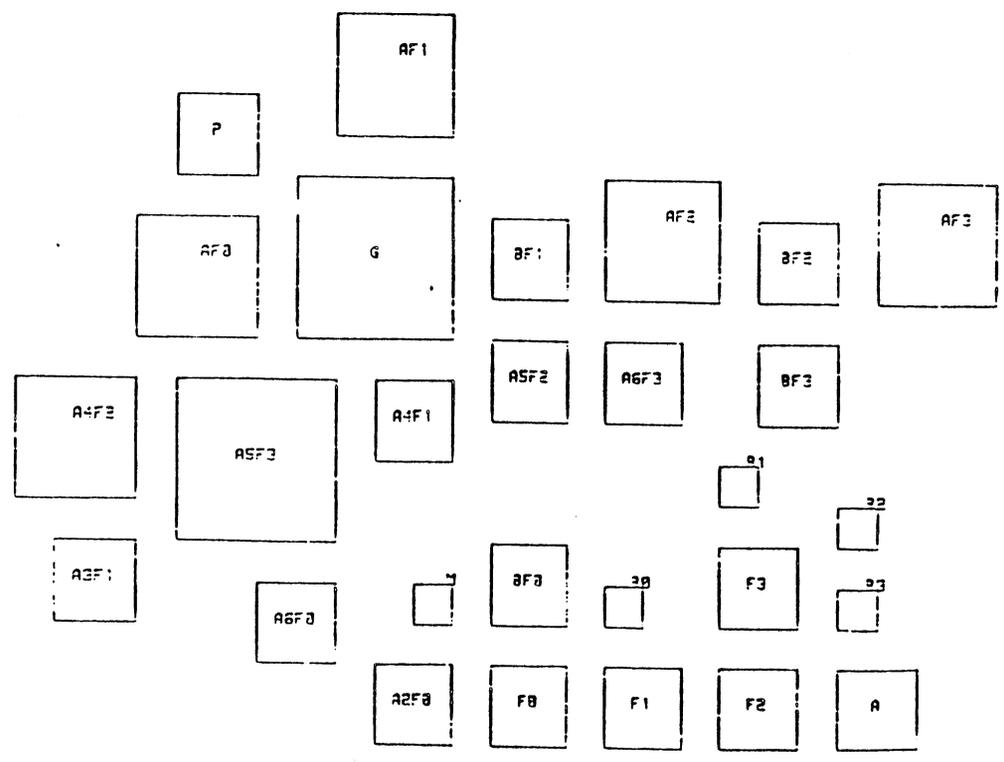
résultat obtenu après compaction

Exemple 2: circuit UAL74181 (unité arithmétique et logique, architecture du circuit 74181 d'après Texas )





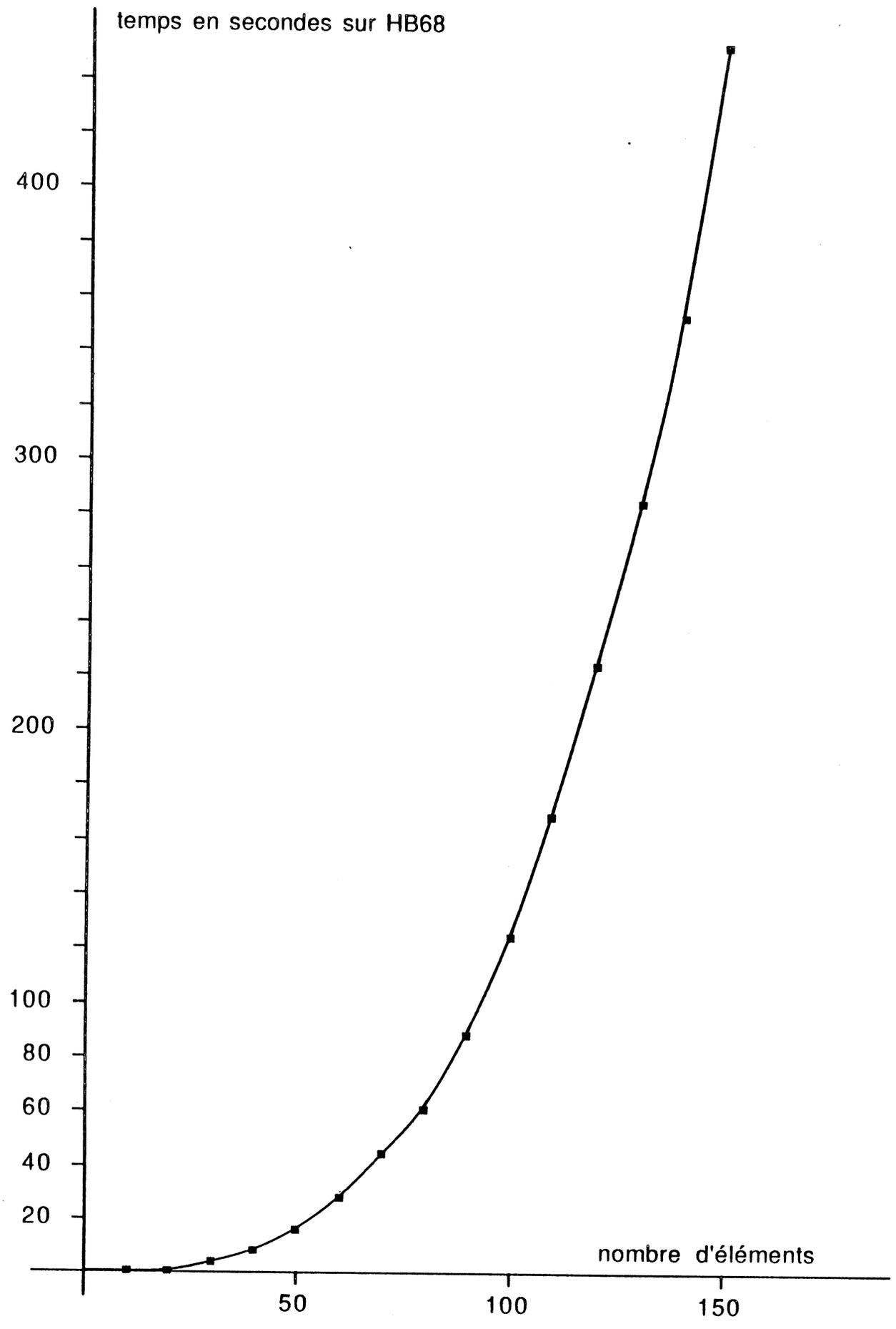
résultat obtenu en sortie d'analyse factorielle avec le chevelu



résultat obtenu après compaction (un espace de routage est laissé entre les blocs)

Pour confirmer la validité d'une telle méthode dans le domaine de la génération de plans de masse il est indispensable d'évaluer la complexité et les temps de calculs de l'algorithme d'analyse des tableaux de distances. Pour ceci, les temps d'exécution ont été mesurés sur HB68 à partir du programme d'analyse factorielle SPAD. Les résultats sont les suivants:

nombre d'éléments	temps d'exécution (en secondes)
10	< 1
20	1
30	3
40	9
50	16
60	26
70	41
80	59
90	89
100	124
110	166
120	224
130	285
140	352
150	450
160	541
170	659
180	766
190	834
200	969



On constate que l'on a approximativement  $t=(n/20)^3$  d'où une complexité de l'algorithme en  $n^3$ . Même si ceci ne semble pas très performant il faut noter que jusqu'à une centaine d'éléments à placer le temps de calcul reste inférieur à deux minutes. Ceci est d'autant plus valable que ce nombre d'éléments est bien supérieur à ce que l'on peut rencontrer dans les problèmes réels de génération de plan de masse.

Afin de mettre en évidence la qualité d'un placement réalisé par cette méthode, il est intéressant de mesurer la déformation introduite par la projection sur le plan factoriel. Cette déformation est représentée par l'inertie du plan factoriel comme il a été expliqué dans le paragraphe précédent.

Dans le cas des deux exemples présentés, les inerties sont les suivantes:

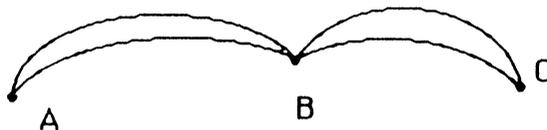
POMULT : 19%

UAL74181 : 37%

Alors que dans le cas de la carte de France, l'inertie est proche de 100%. Ceci provient du fait que les distances fournies dans ce dernier cas sont de vraies distances euclidiennes vérifiant l'inégalité triangulaire alors que dans les exemples de placement à partir de la matrice de connexion seules des relations de proximité sont établies. Une modification de la fonction d'évaluation de distance semble donc utile si on veut assurer une représentation planaire plus fidèle et donc plus proches des contraintes imposées.

### III.2.a. L'inégalité triangulaire.

Ce qui introduit les déformations apportées par l'ATD, sont les incompatibilités du tableau de distance. Considérons l'exemple suivant :



$$d(A,B) = 1/2$$

$$d(B,C) = 1/2$$

$$d(A,C) = \infty$$

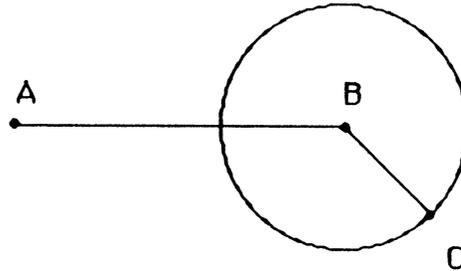
$$d'(A,B) = 1/3$$

$$d'(B,C) = 1/3$$

$$d'(A,C) = 1$$

On voit tout de suite, dans un cas comme dans l'autre, que l'inégalité triangulaire n'est pas respectée. Nous n'avons pas:

$$d(A,B) - d(B,C) \leq d(A,C) \leq d(A,B) + d(B,C).$$



Et ceci introduit un conflit qui ne peut être résolu par l'analyse factorielle que moyennant une déformation de la représentation des trois points. Il est donc indispensable de prendre en compte les contraintes imposées par la représentation planaire. Puisque l'on doit affecter une distance précise à chaque couple de points, nous allons examiner le cas le plus favorable vis à vis des contraintes du placement.

- Contraintes dues au placement :

- A doit être le plus proche possible de B,

- B doit être le plus proche possible de C,

- C peut être aussi éloigné que l'on veut de A.

- Contraintes géométriques dues à la représentation euclidienne :

$$d(A,B) - d(B,C) \leq d(A,C) \leq d(A,B) + d(B,C).$$

Un compromis consiste à prendre :

$$d(A,C) = d(A,B) + d(B,C).$$

Si on considère maintenant, un 4<sup>ème</sup> élément D tel que  $C_{AD} > 0$  et  $C_{CD} > 0$ , l'inégalité triangulaire donne:  $d(A,D) - d(D,C) \leq d(A,C) \leq d(A,D) + d(D,C)$ . Les deux inégalités obtenues ne sont pas forcément vérifiées simultanément, il faut donc à nouveau choisir un compromis se rapprochant au mieux de l'ensemble des solutions possibles pour chacune des inégalités. Ce compromis est donné par:

$$d(A,C) = \min ( d(A,B) + d(B,C) , d(A,D) + d(D,C) ).$$

D'une manière plus générale, il faudra définir pour chaque couple de points dont

la distance ne peut être évaluée, le plus court chemin permettant de les relier. En effet, dans le cas précédent, deux éléments non connectés étaient repoussés au maximum en leur affectant arbitrairement une distance infinie (où supérieure à toutes les autres). Cette solution, bien que donnant des résultats satisfaisants puisqu'elle permet de rapprocher les éléments fortement connectés, aboutit à des conflits et donc à des déformations critiques lors de la projection sur le plan factoriel. Le but est donc de créer dans un premier temps, une matrice dans laquelle sont initialisées toutes les distances connues c'est à dire les distances entre éléments connectés. Les autres distances conservant provisoirement une valeur infinie. Puis dans un second temps, de "lisser" cette matrice de façon à vérifier au mieux les inégalités triangulaires grâce aux compromis décrits plus haut. La fonction de lissage permettant de passer de la matrice de distances initiale  $D$  à la matrice de distances "améliorée"  $D'$  est définie par:

$$l : \mathbb{R}^+ \rightarrow \mathbb{R}^+ \\ D_{ij} \mapsto \min_{k \in E} (D_{ij}, l(D_{ik}) + l(D_{kj}))$$

Ce qui signifie que chaque distance entre deux points est redéfinie comme le plus court chemin passant par un nombre quelconque d'intermédiaires entre ces deux points. Cette fonction de lissage exprimée sous forme récursive se traduit très facilement par l'algorithme suivant :

**fonction** lissage ( A, B, profondeur) ;

**début**

*si* profondeur > 0 **alors**

*pour tout* élément  $C \neq A$  et  $\neq B$  **faire**

**début**

*lissage* := min (distance (A, B),

*lissage* (A, C, profondeur -1) + *lissage* (C, B, 0));

**fin**

**sinon**

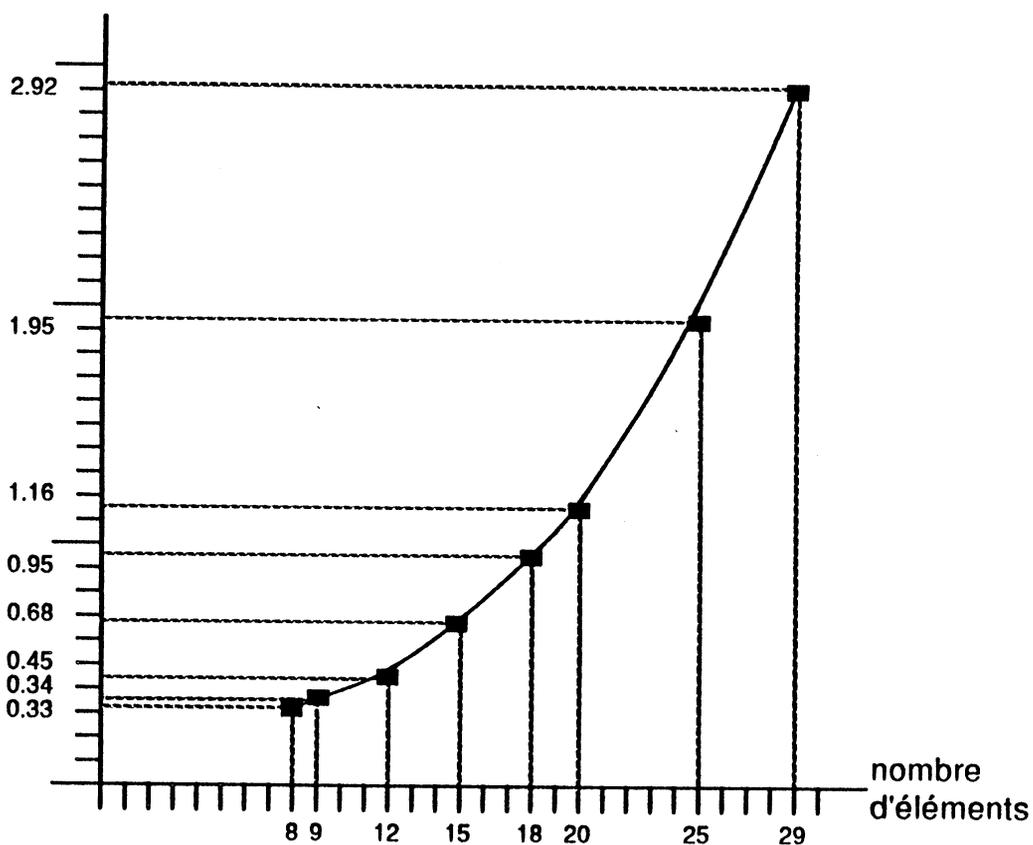
*lissage* := distance (A, B);

**fin;**

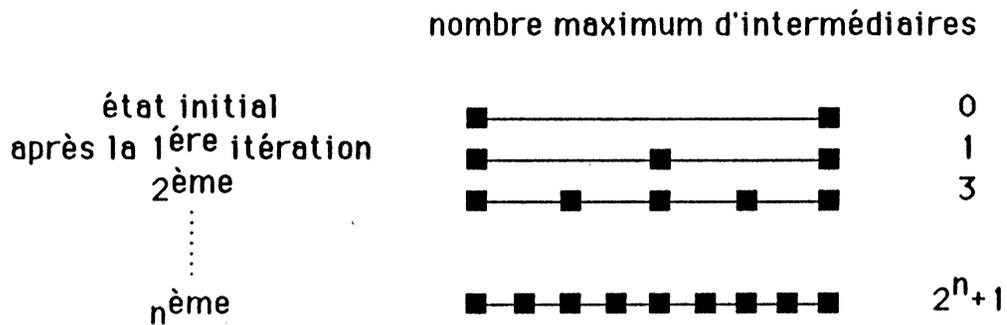
où "profondeur" représente le nombre maximum d'intermédiaires autorisés pour les chemins.

Cependant, un tel lissage est relativement coûteux, étant donné que la complexité est en  $n^2$  pour une profondeur de 1, ce qui pour des chemins de profondeur quelconque devient exponentiel.

temps d'exécution en secondes mesuré sur HB68



Par contre, si on applique l'algorithme plusieurs fois, on traitera, au pas  $n$ , tous les chemins passant par au plus  $2^n - 1$  intermédiaires.



Le nombre d'itérations à appliquer est donc exprimé en  $\log(n)$ . Et si l'on veut être sûr de prendre en compte tous les chemins possibles d'un graphe de  $n$  points, il faut une profondeur d'exploration de  $n-1$  intermédiaires, d'où une complexité globale de l'algorithme en  $n^2 \log n$ . Ce qui reste tout à fait acceptable étant donné la faible complexité de calcul à chaque pas.

De part la méthode utilisée, les distances obtenues sont stables dès le premier lissage (de profondeur maximale) de la matrice de distances. Ce qui signifie que l'algorithme ne risque pas de diverger et de provoquer des perturbations sur les distances calculées d'après la matrice de connexion.

En fait, cette méthode permet de calculer les distances "manquantes" de la matrice et de réajuster les distances "trop longues". Ces distances sont évaluées en fonction de celles qui sont connues. Il est évident que plus la proportion de distances connues sera importante, plus l'estimation sera bonne. Si par exemple, on supprime quelques distances du tableau des villes de France, en les remplaçant par des distances "infinies", l'algorithme permettra de les déduire approximativement des distances connues.

Exemple :

Si on supprime les distances de Paris à Grenoble et de Tours à Bordeaux, on obtient après lissage:

Paris - Grenoble = 557 Km

Tours - Bordeaux = 327 Km

Soit les distances réelles entre ces villes.

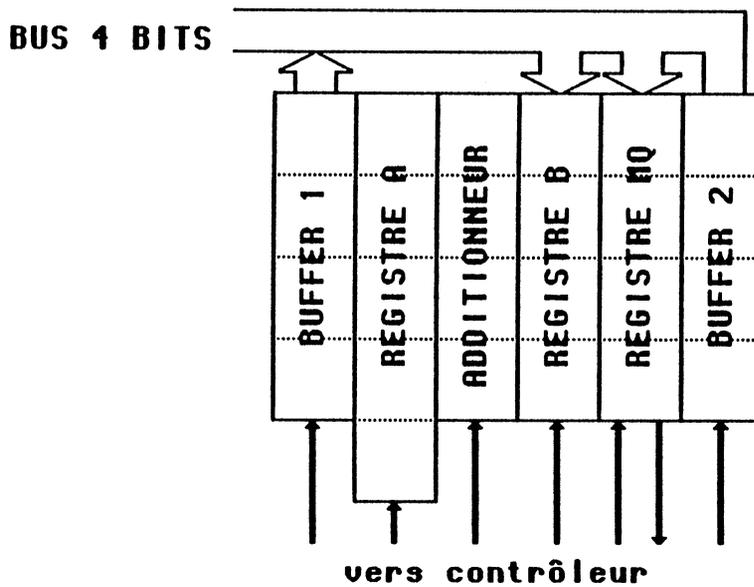
### Retour au problème du placement.

Si maintenant on essaie de placer par analyse factorielle les exemples présentés plus haut et que l'on regarde les déformations introduites par la projection sur le plan, on obtient:

	avant lissage	après lissage
POMULT	19% d'inertie	100% d'inertie
UAL74181	37% "	73% "

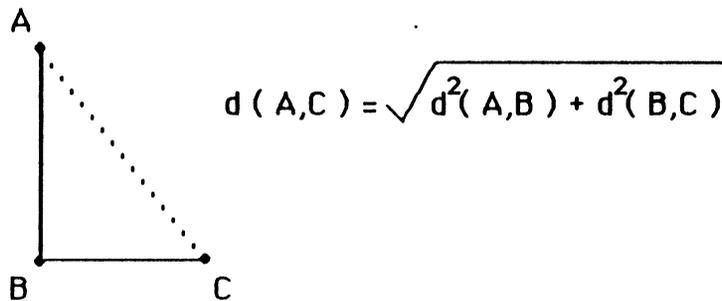
On voit tout de suite la nette amélioration des résultats. Il est également possible d'examiner l'inertie du premier axe factoriel. Dans le cas du circuit "pomult", cette inertie est de 75% ce qui veut dire que le circuit se retrouve essentiellement disposé sur une droite. Cette tendance à aligner les éléments du circuit provient de la méthode de lissage choisie. En effet le compromis choisi pour satisfaire à la fois les contraintes du placement et les inégalités triangulaires est prévu pour disposer les éléments en ligne. On pourrait supposer que cet alignement excessif perturbe le placement. En fait, si l'inertie du plan factoriel est de 100% c'est que l'ensemble des points peut effectivement être placé sur ce plan en respectant la matrice de distances. De plus l'alignement des éléments est une chose naturelle si on regarde que la plupart des connexions manipulées sont du type "bus" c'est à dire que la description du circuit prévoit implicitement un empilement des différents éléments.

Exemple : architecture de pomult.



Ce lissage des matrices de distances permet donc une meilleure représentation des éléments sur le plan factoriel en favorisant les "aboutements", fréquents dans la génération de plan de masse.

Une deuxième étude a été faite en utilisant une autre fonction d'évaluation des distances manquantes ou infinies. Cette fonction respecte également l'inégalité triangulaire en considérant les éléments non plus alignés, mais disposés en triangle:



Cependant les résultats obtenus expérimentalement sont moins satisfaisants que dans le cas précédent. Je ne m'attarderai donc pas d'avantage sur ce cas.

• **Remarque.**

Il a été prouvé qu'à partir d'une matrice de distances quelconques,  $d_{ij}$ , il était toujours possible de créer une matrice  $\delta_{ij}$  vérifiant l'inégalité triangulaire. Cette deuxième matrice s'obtient par:

$$\begin{aligned} \delta_{ij}^2 &= d_{ij}^2 + 2 \lambda_n^2 && \text{si } i \neq j \\ &= 0 && \text{si } i = j \end{aligned}$$

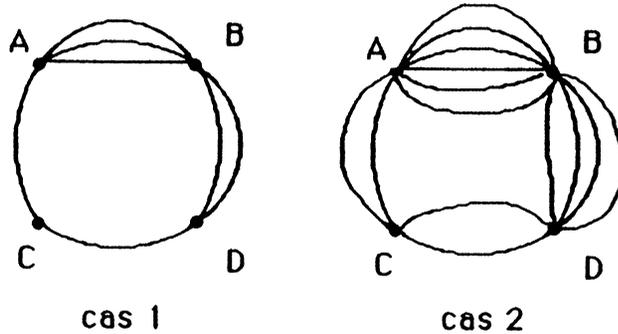
Où  $\lambda_n$  est la valeur propre négative la plus grande en valeur absolue dans l'analyse du tableau de distances  $d$ .

Ceci implique toutes fois de rechercher successivement les valeurs propres de deux matrices  $n \times n$ : une première fois sur la matrice  $d$  pour déterminer la valeur propre  $\lambda_n$ , et une seconde fois sur la matrice modifiée  $\delta$  afin de calculer les coordonnées des points. Cette opération étant relativement coûteuse en temps de calcul, il est préférable de ne pas avoir à la répéter. Cette méthode n'a donc pas été retenue afin de réduire les temps de réponse de l'ensemble du programme.

• **Remarque Importante.**

Il est possible d'utiliser une autre fonction d'évaluation de la distance. Cette fonction n'est plus seulement liée au nombre de connexions existant entre deux éléments mais dépend aussi de la proportion de ces connexions par rapport au nombre total de connexions aboutissant à l'un ou à l'autre des éléments. Cette fonction a pour but de ne pas trop "éclater" ou "tasser" un circuit suivant la densité de connexion qu'il possède.

Exemple:



Les distances calculées par la méthode précédente étaient (avant lissage):

cas 1	cas 2
$d(A,B) = 1/3$	$d(A,B) = 1/6$
$d(B,D) = 1/2$	$d(B,D) = 1/4$
$d(A,C) = 1$	$d(A,C) = 1/2$
$d(C,D) = 1$	$d(C,D) = 1/2$

Les distances obtenue dans les deux cas sont différentes. Une telle différence n'est pas critique si elle apparaît dans deux problèmes différents. Mais elle devient beaucoup plus critique si les deux cas apparaissent dans le même problème. Si l'on pondère ces distances de la façon suivante:

$$d(A,B) = \frac{\text{nombre de connexions aboutissant à A ou à B}}{\text{nombre de connexions reliant A à B}}$$

on obtient des résultats identiques pour les deux cas de figure:

$$\begin{aligned}d(A,B) &= 6/3 = 2 \\d(B,D) &= 6/3 = 2 \\d(A,C) &= 4 \\d(C,D) &= 5\end{aligned}$$

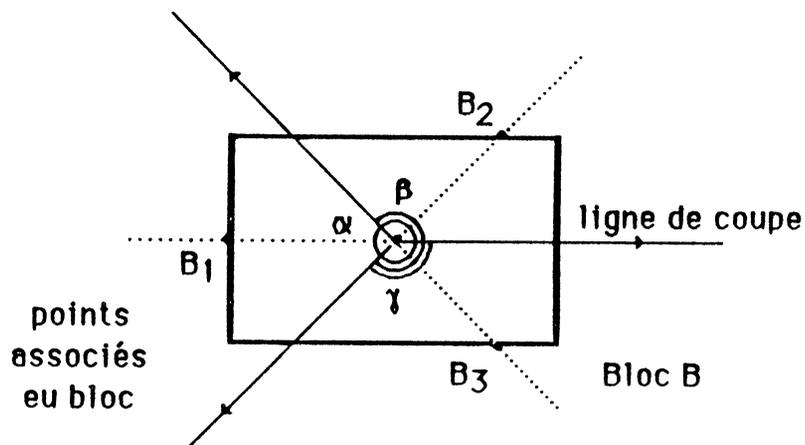
En fait les résultats finaux de placement obtenu à l'aide de cette deuxième fonction d'évaluation de distance sont meilleurs (au sens des contraintes du placement) qu'avec la fonction précédente. Il faut remarquer que cette distance se rapproche de celle fréquemment utilisée en statistiques et connue sous le nom de distance du  $\chi^2$ .

### III.2.b. Prise en compte des blocs durs.

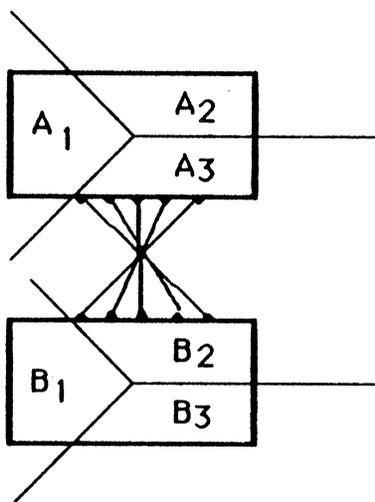
L'analyse d'un tableau de distances, bien qu'étant rapide et efficace, ne peut toutefois donner, à ce stade, un placement définitif des blocs. En effet, dans le cas des blocs durs, des opérations de rotation et de symétrie sont indispensables afin d'optimiser les longueurs des connexions. Or jusqu'à présent, les blocs étaient assimilés à des points, et ces opérations ne pouvaient être prises en compte.

Remarque : pour les blocs de type mou, aucune information concernant la topologie n'étant connue, ces opérations de symétrie et de rotation n'ont aucune raison d'être appliquées.

Le problème consiste donc à traiter ces transformations par un algorithme suffisamment rapide. Dans ce cas, les méthodes du type annealing sont à proscrire. L'idée consiste donc à utiliser l'ATD pour prendre en compte ces opérations. Pour ceci, il faut être capable d'orienter un bloc. Un point n'ayant aucune orientation, le bloc ne devra plus être représenté par un seul point mais par plusieurs. La première idée consiste donc à éclater un bloc en trois points ce qui permet de définir toute rotation ou symétrie de manière unique.

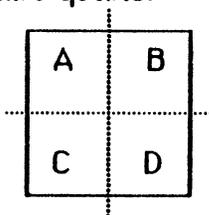


Cette représentation ne permet toutefois pas de prendre en compte toutes les transformations possibles :

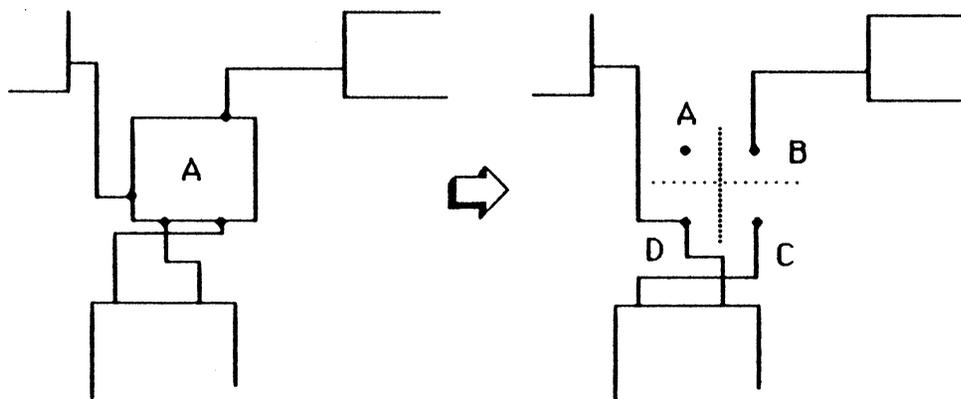


La symétrie horizontale qui s'impose pour l'un des deux blocs passera inaperçue et ainsi une grande partie des aboutements que l'on attend lors du placement, risque de ne pas être réalisée correctement. De plus ce découpage en trois parties dans laquelle le choix des angles  $\alpha$ ,  $\beta$  et  $\gamma$  est arbitraire, s'applique assez mal à la structure rectangulaire des blocs.

Ainsi un bloc est éclaté en quatre quarts.

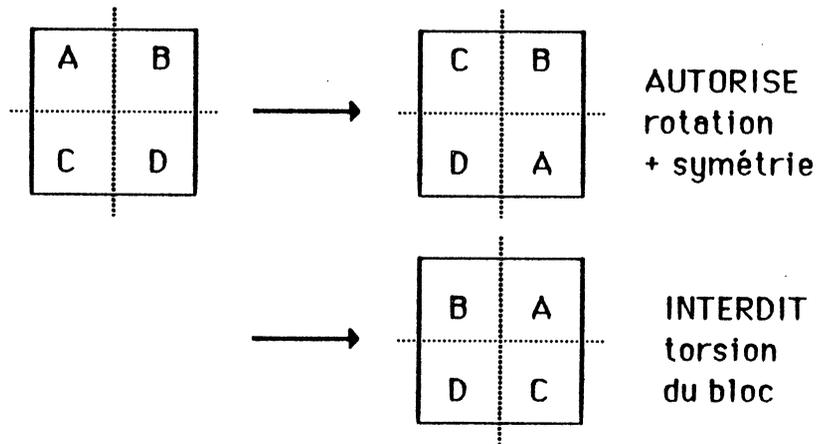


Ces quatre coins sont représentés par quatre points, auxquels sont associées les équipotentiels aboutissant sur le secteur correspondant. Ainsi, chaque coin du bloc sera attiré indépendamment des autres vers d'autres blocs du circuit.

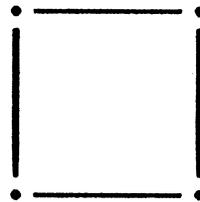


Afin de conserver une cohérence au placement obtenu par l'ATD, c'est à dire pour conserver une certaine cohésion des quatre coins des blocs, on est amené à créer des connexions fictives entre ces quatre points.

De plus certaines déformations sont interdites :

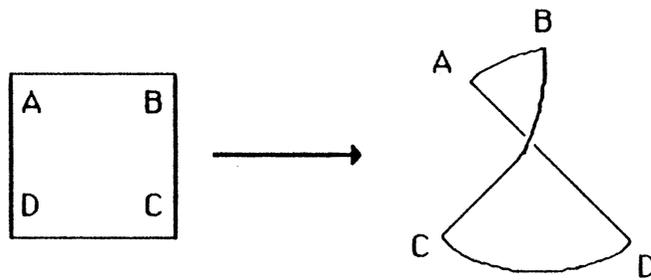


Pour éviter de telles configurations, les liaisons fictives entre les quatre coins sont réalisées de la manière suivante :



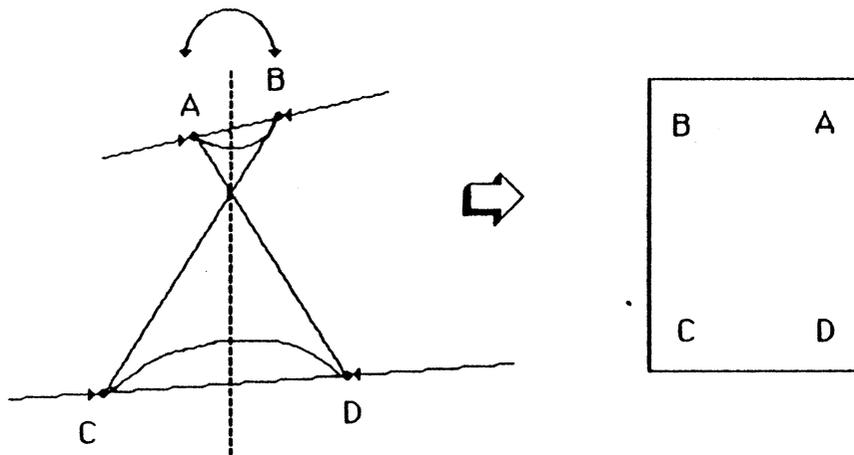
Les liaisons entre coins diagonalement opposés sont, soit considérées comme étant infinies de façon à assurer l'éloignement des coins tout en rapprochant les coins adjacents, soit calculées à partir de la taille du bloc, afin de respecter au mieux la représentation planaire. On crée ainsi un tableau de distances en éclatant les blocs durs en quatre blocs fictifs. Ceci permet, en n'utilisant que l'ATD, de traiter les problèmes de symétrie et de rotation.

Cependant, malgré les précautions prises, des configurations interdites peuvent apparaître.



Dans ce cas, on définit un axe d'inertie qui passe par le barycentre du bloc. C'est le couple de points ( ici (A,B) ou (C,D)) possédant le moment d'inertie maximal qui définit s'il y a ou non symétrie. Ceci peut s'expliquer en disant que ce sont les points les plus excentrés qui ont subi l'attraction la plus forte de l'extérieur, et sont donc plus fortement connectés et donc prioritaires.

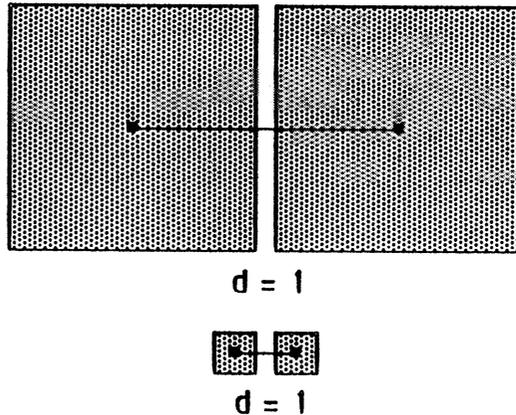
Exemple:



• **Remarque sur le calcul des distances.**

Jusqu'à présent, dans l'évaluation des distances inter-blocs, la taille des éléments n'a pas été prise en compte. Il peut être judicieux de la faire intervenir car elle influe beaucoup sur la position finale.

Exemple:



Afin d'approximer les distances finales entre blocs, on peut donc pondérer chaque distance  $d(e_i, e_j)$  par les tailles respectives de  $e_i$  et  $e_j$ . Pour ceci, il faut considérer que les deux blocs les plus fortement connectés sont situés l'un à côté de l'autre et que les blocs les plus faiblement connectés sont situés au pire à la distance la plus grande, c'est à dire dans deux coins diagonalement opposés du circuit. Ceci oblige donc à trouver une fonction  $f$  telle que:

$$f(\max(d(e_i, e_j))) = \text{diagonale du circuit},$$

$$f(\min(d(e_i, e_j))) = 0.$$

En fait, la distance minimale entre deux points n'est nulle que si la distance entre chaque point et le bord du bloc est nulle, ce qui ne peut arriver que dans le cas de blocs durs puisque chaque point est assimilé à un des coins du bloc. Dans le cas d'un bloc mou, cette distance est augmentée du demi-côté du bloc. Cette fonction de correction des distances en fonction de la taille de chaque bloc devra être linéaire entre les deux extrêmes cités plus haut. En fait, les essais pratiques réalisés en évaluant une distance à l'aide de cette méthode, n'ont pas montré une amélioration sensible des résultats par rapport à une méthode utilisant une distance sans prise en compte de la taille des blocs. Il peut cependant être intéressant, pour la création d'un système modulaire de génération de plan de masse utilisant les méthodes d'analyse factorielle, d'offrir à l'utilisateur cette possibilité.

Finalement, le problème majeur pour la réalisation d'un plan de masse par une méthode factorielle est la création d'une "bonne" matrice de distances: c'est à dire

d'une matrice de distances qui permet de respecter toutes les contraintes du placement planaire en prenant en compte un maximum de paramètres, notamment pour permettre les opérations de symétrie et de rotation. La création de cette matrice de distances peut se résumer par les étapes suivantes:

1° - Eclater les blocs:

soit  $E = \{e_1, e_2, \dots, e_n\}$  l'ensemble des blocs, on considère

$E_m = \{e_1 \dots e_p\}$  le sous ensemble des blocs mous

et  $E_p = \{e_p \dots e_n\}$  le sous ensemble des blocs durs.

On crée l'ensemble  $E' = E_m \cup E'_d$  tel que

$$E'_d = \{e_{p1}, e_{p2}, e_{p3}, e_{p4}, e_{p+11} \dots e_{n4}\}$$

où  $e_{i1} \dots e_{i4}$  correspondent aux quatre coins du bloc  $e_i$ .

2° - Génération de la matrice de connexion:

$$\mathcal{C} : E \times E \rightarrow \mathbb{N}$$

$$(e_i, e_j) \mapsto C_{ij} = \text{nombre de connexions allant de } e_i \text{ à } e_j$$

3° - En déduire une matrice de corrélation entre les blocs:

$$\mathcal{K} : E' \times E' \rightarrow \mathbb{R}^+$$

$$(e_i, e_j) \mapsto \frac{C_{ij}}{\sum_{k=1}^n C_{ik} + \sum_{k=1}^n C_{kj}}$$

4° - Générer la matrice des distances inter-blocs:

$$\begin{aligned} \mathcal{D} : E' \times E' &\rightarrow \mathbb{R}^+ \\ (e_i, e_j) &\mapsto \begin{cases} \frac{1}{\mathcal{K}(e_i, e_j)} & \text{si } i \neq j \\ 0 & \text{sinon} \end{cases} \end{aligned}$$

5° - Calculer les distances intra-blocs:

$$\mathcal{J} : E'_d \rightarrow \mathbb{R}^+$$

$(e_{i1}, e_{i2}) \rightarrow$  largeur du bloc  $e_i$

$(e_{i2}, e_{i3}) \rightarrow$  hauteur du bloc  $e_i$

$(e_{i3}, e_{i4}) \rightarrow$  largeur du bloc  $e_i$

$(e_{i4}, e_{i1}) \rightarrow$  hauteur du bloc  $e_i$

$(e_{i1}, e_{i3}) \rightarrow$  diagonale du bloc  $e_i$

$(e_{i2}, e_{i4}) \rightarrow$  diagonale du bloc  $e_i$

On obtient à ce stade une matrice de distance  $D_{ij}$ .

6° - Lissage de la matrice:

$$\begin{aligned} \mathcal{L} : E' \times E' &\rightarrow \mathbb{R}^+ \\ (e_i, e_j) &\mapsto \min_{k \in E} (D_{ij}, \mathcal{L}(D_{ik}) + \mathcal{L}(D_{kj})) \end{aligned}$$

Il est à noter que toutes les matrices manipulées sont triangulaires. Une fois la matrice finale obtenue elle sera traitée par un programme d'analyse factorielle de tableaux de distances.

Le cheminement décrit correspond à certains choix, notamment concernant l'utilisation

de la fonction de distance du type  $\chi^2$ . Ce choix a été réalisé en fonction de résultats pratiques obtenus à l'aide de chacune des méthodes décrites. Il correspond aux différentes étapes du programme ALFA qui a été développé afin de valider la méthode. Il est cependant intéressant, et c'est là une des prochaines étapes de l'introduction des méthodes factorielles dans la génération de plans de masse, d'offrir à l'utilisateur un ensemble de fonctions diverses. Des méthodes d'analyse suffisamment rapide pour pouvoir assurer leur utilisation interactive ayant été développées [BEL 85], l'utilisateur pourra, au vu du résultat, revenir à tout moment sur une des étapes d'évaluation de la matrice de distances afin d'adapter la méthode à chaque cas particulier. Pour l'instant un tel système interactif de génération de plans de masse n'en est qu'à l'ébauche mais il ne semble pas utopiste de penser bâtir un système expert capable de définir de lui même le cheminement le mieux adapté à chaque problème.

### **III.2.c. Les plots**

Dans tout ce qui a été présenté précédemment, un problème n'a pas encore été résolu: celui du placement des plots d'entrée/sortie. Les blocs de ce type subissent un régime particulier puisqu'ils ne peuvent être placés qu'à la périphérie du circuit. D'autre part, comme il a été dit dans le second chapitre, il est fréquent que la position des plots soit imposée dans le cahier des charges du circuit à réaliser.

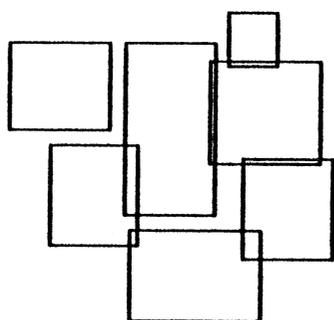
Or jusqu'à présent, l'analyse factorielle et notamment l'analyse d'un tableau de distances ne travaille qu'à partir de positions relatives des éléments et sans contraintes extérieures. Il faut donc être capable de rajouter un certain nombre de contraintes fictives dans le tableau des distances afin de prendre en compte ces nouveaux paramètres.

Considérons par exemple le circuit comme un bloc similaire aux autres: les plots d'entrée/sortie sont représentés comme des points de contact à sa périphérie. Il est alors possible de placer ce bloc comme les autres en le prenant en compte dans le tableau de distances. Ce bloc, au même titre que les autres subira translation, rotations, symétries afin d'être placé au mieux pour respecter les contraintes.

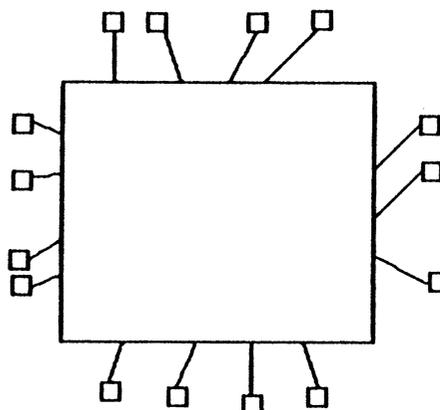
A ce niveau, il est important de constater que l'analyse factorielle elle-même ne permet de placer que des points sur un plan et n'interdit absolument pas le chevauchement des blocs. Si ce chevauchement est gênant dans le cas des blocs internes au circuit, il est au contraire tout à fait recommandé dans le cas présent.

En effet, le but est maintenant d'obtenir le chevauchement maximal entre le bloc représentant le circuit et tous les blocs représentant les composants internes à ce circuit. En pratique, ce chevauchement est assez facile à obtenir étant donnée la taille très importante du bloc "circuit" par rapport aux autres blocs.

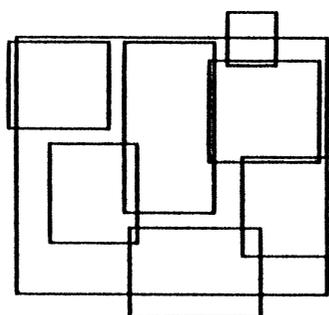
Exemple:



blocs placés avec  
un chevauchement  
minimal



association des plots  
à un bloc représentant  
le circuit



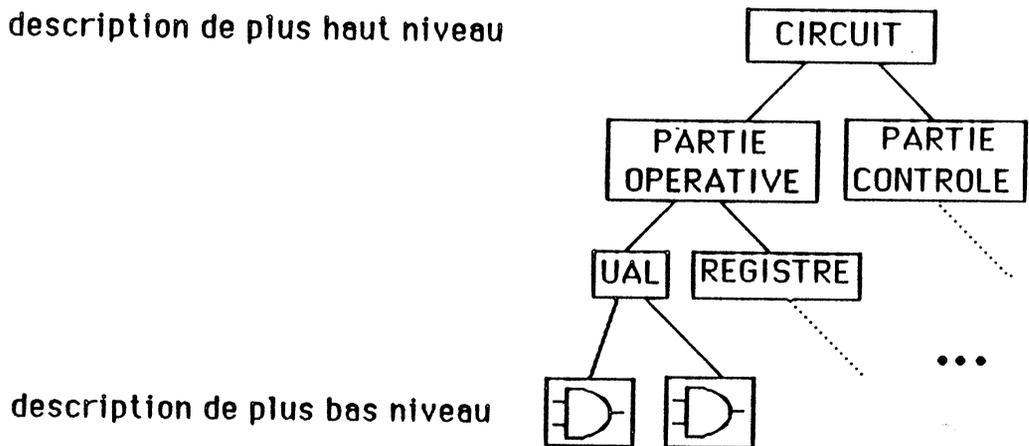
attraction de ce dernier  
bloc par les autres avec  
chevauchement maximal

### III.3. La classification.

Le but de la classification est de créer des groupes (ou classes) d'individus proches, c'est à dire présentant un maximum de similitudes. L'utilisation de cette méthode statistique est entièrement originale dans le domaine du placement automatisé. On en trouvera une description détaillée dans [TRI 86] pour l'approche statistique et dans [JAN 85] pour l'approche conception de circuits. L'intérêt d'une telle méthode dans le domaine du placement automatique est le découpage d'un "gros" problème (c'est à dire d'un circuit comportant un trop grand nombre d'éléments) en plusieurs problèmes plus petits et donc plus simples à résoudre. Ce découpage correspond à une hiérarchisation du circuit. Une hiérarchie évidente existe si l'on

dispose d'une description architecturale complète du circuit.

description de plus haut niveau



Il faut cependant noter que, d'une part cette description ne correspond pas obligatoirement aux critères du placement et que, d'autre part, elle n'existe pas forcément. Il est donc intéressant d'appliquer les méthodes de classification afin de découper le circuit en sous circuits en respectant les contraintes du placement et notamment la connectique.

Avant de présenter cette application de la classification, présentons d'abord la classification:

Une méthode de classification automatique s'appuie essentiellement sur la définition d'une mesure de proximité ou de pseudo-distance entre les éléments. Cette mesure de proximité ne correspond pas forcément à la notion de distance classique. Mais comme dans le cas de l'analyse factorielle, la définition de cette proximité est très importante car c'est elle qui oriente les résultats. En effet une classe est formée d'éléments entre lesquels les pseudo-distances sont faibles. Entre deux éléments appartenant à deux classes différentes la pseudo-distance est grande. Plusieurs définition de cette pseudo-distance basées sur des analogies et des intuitions sont possibles. Seul l'expérimentation saura dire si la fonction utilisée permet de fournir les résultats attendus. Il est donc possible et même recommandé, afin de conserver une certaine cohérence au résultat final d'utiliser les mêmes distances que celle qui ont été décrites dans le paragraphe précédent: c'est dire soit une distance directement

inversement proportionnelle au nombre de connexions entre les éléments, soit une distance du  $\chi^2$ .

Un des algorithmes mathématiques de la méthode de classification est utilisée pour l'application concernée [TRI 86] et se décompose comme suit:

1° - On suppose donnée entre les éléments, une mesure de proximité  $\Delta$  (inverse de la fonction de distance).

2° - On définit une proximité entre deux classes  $C_i$  et  $C_j$  par:

$$D(C_i, C_j) = \min \{ \Delta(x, y) / x \in C_i, y \in C_j \}$$

$$\text{On pose } \alpha = \max_{i,j} \Delta(C_i, C_j)$$

On calcule une estimation de  $\alpha$  en déterminant l'arbre minimal associé au graphe pondéré des éléments [BAR 85].

3° - Pour tout élément  $x$ , on note  $V_x = \{y / \Delta(x,y) \leq \alpha\}$ . On établit une relation  $\mathfrak{R}$ , sur l'ensemble des éléments, réflexive et symétrique:

$$x \mathfrak{R} y \Leftrightarrow \min \left( \frac{|V_x \cap V_y|}{|V_x|}, \frac{|V_x \cap V_y|}{|V_y|} \right) \geq \beta$$

La relation  $\mathfrak{R}$  dépend de  $\beta : \beta \in [0,1]$ .

4° - La fermeture transitive de  $\mathfrak{R}$ ,  $R$ , forme une relation d'équivalence sur l'ensemble des éléments dont les classes constituent la classification recherchée.

On aura donc:

$$x R y \Leftrightarrow \exists z_1, z_2 \dots z_k \text{ tels que } x \mathfrak{R} z_1, z_1 \mathfrak{R} z_2 \dots z_k \mathfrak{R} y.$$

R est une relation d'équivalence. Il faut remarquer que le critère d'agrégation choisi n'est pas unique, on aurait pu choisir, par exemple:

$$\frac{|v_x \cap v_y|}{\sqrt{|v_x| \cdot |v_y|}}$$

Ceci est une sorte de corrélation. Le premier critère sera choisi pour l'application au problème du placement.

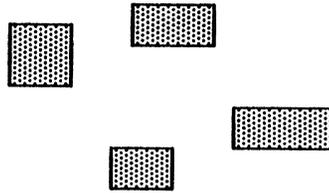
En fait, la classification dépend fortement des paramètres choisis par l'utilisateur et notamment:

- de la mesure de proximité  $\Delta$ ,
- des deux paramètres  $\alpha$  et  $\beta$ ,
- du critère R d'agrégation choisi comme relation entre les modules,
- d'une relation d'ensemble sur les modules.

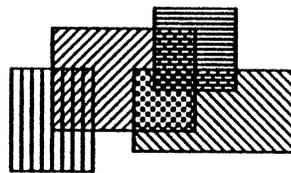
#### **III.4. Amélioration d'un placement.**

Le placement obtenu par l'analyse factorielle, comme la plupart des résultats obtenus à l'aide de méthodes constructives, s'il satisfait la plupart des contraintes demandées n'est toutefois pas forcément optimal, ni même tolérable.

Exemple:



Ce placement n'est pas optimal car une simple compaction permettrait de réduire toutes les longueurs de connexions.



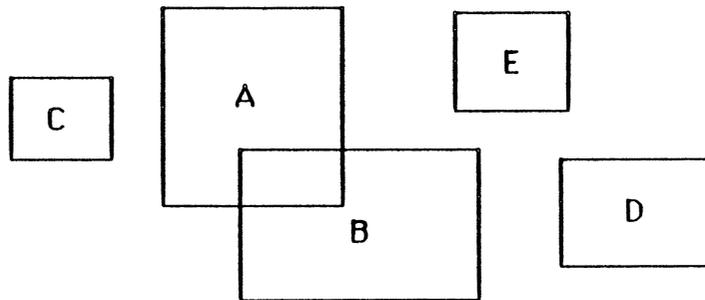
Ce placement n'est pas tolérable car les blocs se chevauchent. Il est alors indispensable d'appliquer une homothétie depuis le centre du circuit.

### III.4.a Le compactage

En fait, les résultats fournis par l'analyse factorielle comme elle a été présentée au paragraphe précédent sont suffisamment bons pour se passer totalement de ces méthodes complexes. Les seuls cas de figure à éliminer sont les chevauchements et les écarts excessifs entre les blocs. Seul un compactage est nécessaire. Pour ceci une méthode de bipartitionnement topologique a été mise au point. L'idée est de découper le circuit en zones, de façon à ce que chaque bloc appartienne à une zone et que le positionnement relatif des blocs fourni par le placement initial soit conservé.

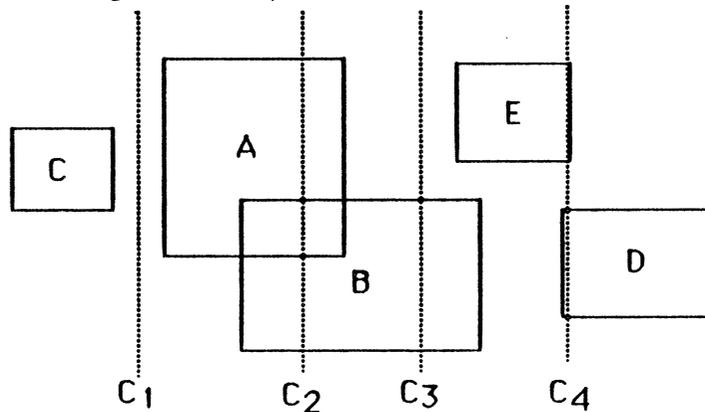
Le procédé est le suivant:

1°- L'ensemble des blocs est d'abord trié sur un axe selon les coordonnées croissantes de leurs centres.



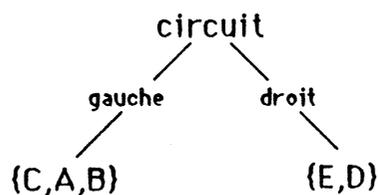
L'ordre est C, A, B, E, D.

2°- Une ligne de coupe est définie entre deux blocs consécutifs.

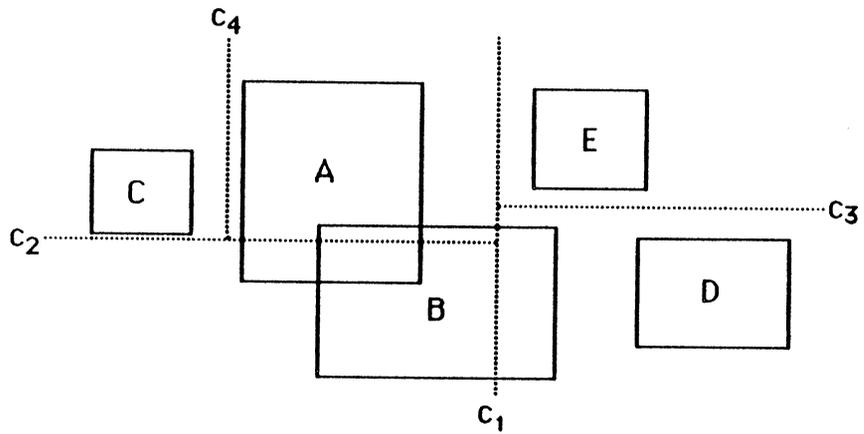


3°- Une ligne de coupe est choisie, et le circuit est partitionné en deux demi circuits appelés cellules.

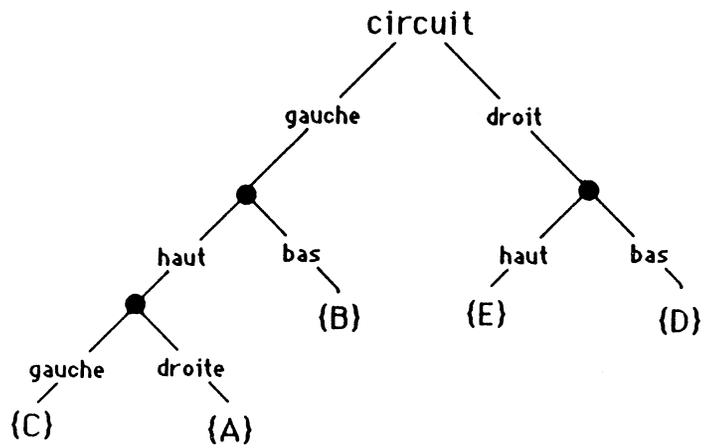
Exemple: coupe suivant C<sub>3</sub>.



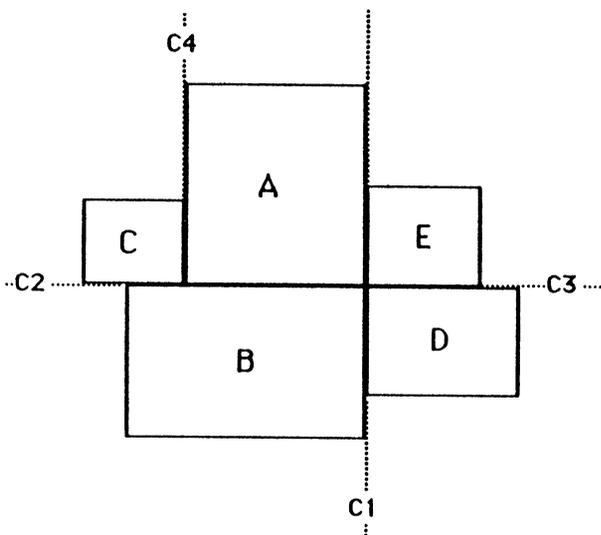
4°- La même opération est effectuée sur chaque cellule avec un axe perpendiculaire au précédent. Ceci est répété jusqu'à obtenir des cellules indivisibles, c'est à dire ne comportant qu'un seul bloc. Ces cellules seront appelées cellules terminales.



Ce qui donne l'arbre suivant:



D'où une disposition finale des éléments:



### Choix de la ligne de coupe.

Après la définition de toutes les bisections, une doit être choisie. On peut trouver de nombreux critères de choix parmi lesquels je retiendrai les deux suivants:

- Espacement maximum entre les blocs [OTT 82].

Définitions:

Soit  $B = \{ b_1, b_2 \dots b_n \}$  l'ensemble des blocs.

$C = \{ c_1 \dots c_{n-1} \}$  l'ensemble des lignes de coupes.

Nous appellerons  $S(b_j)$ , la surface du bloc  $b_j$ ,

$X(b_j)$ , la coordonnée du centre du bloc  $b_j$  sur l'axe des  $x$ ,

$X(c_j)$ , la coordonnée de la ligne de coupe  $c_j$  sur l'axe  $x$ .

Il est possible d'ordonner les ensembles  $B$  et  $C$  tels que:

$$\forall i, j = 1 \dots n, \quad i \leq j \Rightarrow X(b_i) \leq X(b_j)$$

$$\forall i, j = 1 \dots n-1, \quad i \leq j \Rightarrow X(c_i) \leq X(c_j)$$

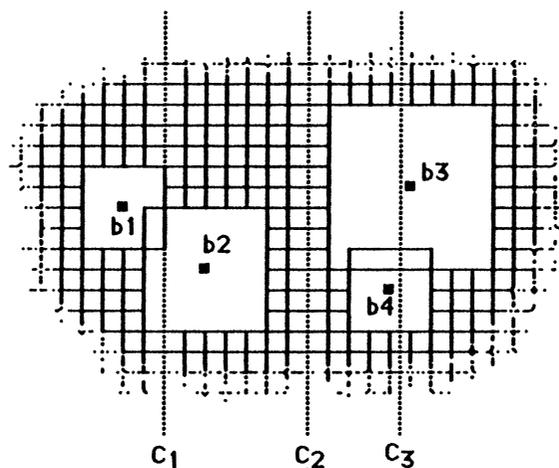
on aura alors,

$$\forall i = 1 \dots n-1, \quad X(b_i) \leq X(c_i) \leq X(b_{i+1}).$$

On peut définir, pour chaque ligne de coupe, un poids  $P$ .

$$P_i = P(c_i) = \min_{j=i+1}^n \left( X(b_j) - \frac{\sqrt{S(b_j)}}{2} \right) - \max_{j=1}^i \left( X(b_j) + \frac{\sqrt{S(b_j)}}{2} \right)$$

Exemple:



On a:  $P_1 = P(c_1) = -1$

$$P_2 = P(c_2) = 3$$

$$P_3 = P(c_3) = -4$$

Ce qui correspond à l'intervalle minimal situé entre deux blocs situés respectivement à gauche et à droite de la ligne de coupe. La ligne de coupe choisie est celle ayant le poids le plus fort.

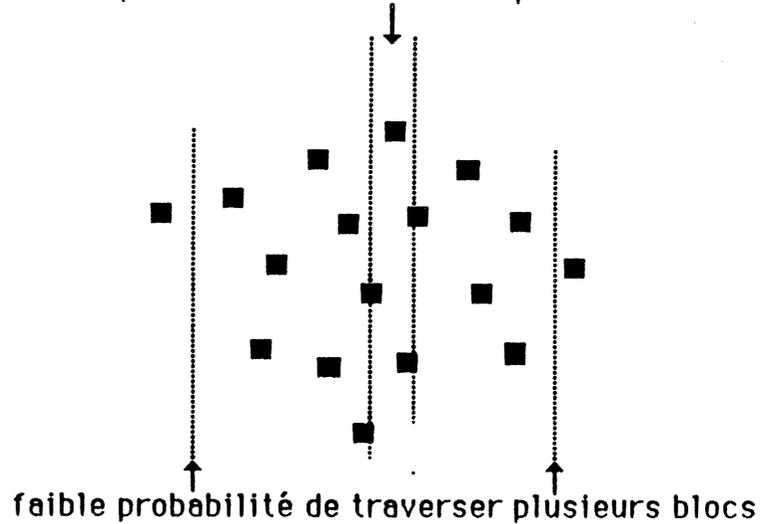
Remarque: il est évident qu'un tel calcul est justifié si les blocs sont carrés ce qui est le cas par défaut lorsqu'il sont mous. On pourra, dans le cas de blocs durs, remplacer l'évaluation de taille du bloc (racine de sa surface) par ses dimensions réelles c'est à dire hauteur ou largeur.

Cette méthode permet de découper le circuit en modules relativement éloignés les uns des autres. Ce en quoi, on se rapproche des méthodes de classification exposées plus haut. Cependant les résultats pratiques sont assez médiocres car un tel découpage crée des cellules de tailles non homogènes et difficilement "empilables".

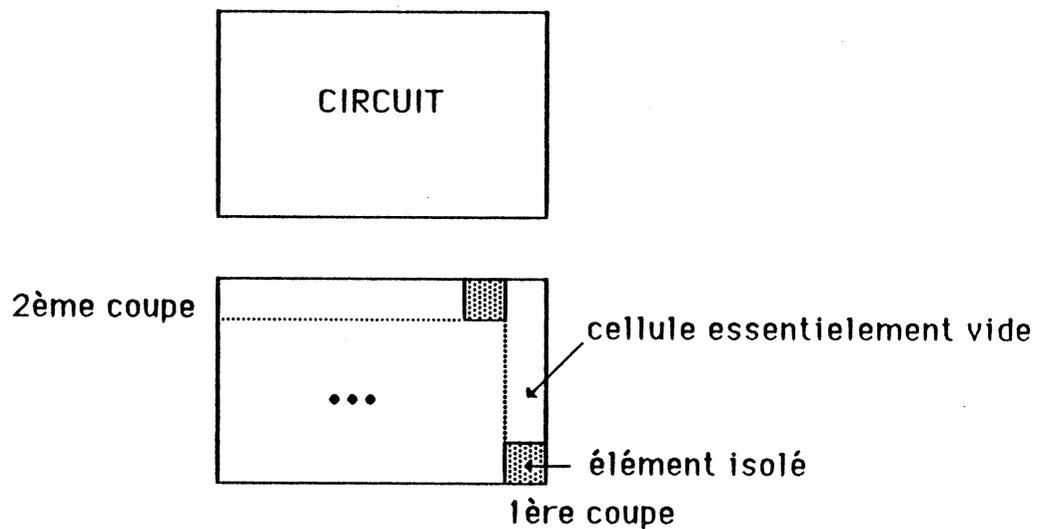
En effet, même si le nuage de points obtenu par l'analyse factorielle a une densité uniforme, la probabilité pour qu'une ligne de coupe située au centre du circuit traverse des blocs dont les centres sont situés de part et d'autre de cette ligne, est beaucoup

plus importante que si la ligne est située sur un des bords du circuit.

forte probabilité de traverser plusieurs blocs



Avec cette méthode on aboutit donc à un isolement des éléments périphériques ce qui rend le compactage assez critique.



En fait, il est préférable de couper le circuit en cellules homogènes au point de vue de leurs tailles. La méthode suivante a donc été développée.

- **Equilibrage des surfaces.**

Si on reprend les définitions et notations précédentes on peut définir un nouveau

poids  $P'$  pour chaque coupe:

$$P'_i = P'(c_i) = \left| \sum_{j=1}^i S(b_j) - \sum_{j=i+1}^n S(b_j) \right|$$

On choisira alors la ligne de coupe ayant le plus faible poids c'est à dire la ligne de coupe telle que la somme des surfaces des blocs situés à gauche de cette ligne soit la plus proche possible de la somme des surfaces des blocs situés à sa droite. Par cette méthode on aboutit à une bonne homogénéisation des surfaces des cellules ce qui permet un bon compactage.

Les algorithmes utilisés procèdent par appels récursifs croisés:

*procedure coupe\_verticale (cellule);*

*debut*

*si cellule contient plus d'un bloc alors*

*début*

*trier les blocs selon l'axe Y;*

*calculer les poids des lignes de coupe;*

*choisir une ligne de coupe;*

*diviser cellule en cellule\_gauche et cellule\_droite;*

*coupe\_horizontale (cellule\_gauche);*

*coupe\_horizontale (cellule\_droite);*

*fin;*

*fin;*

*procedure coupe\_horizontale (cellule);*

*debut*

*si cellule contient plus d'un bloc alors*

*début*

*trier les blocs selon l'axe X;*

*calculer les poids des lignes de coupe;*

*choisir une ligne de coupe;*

*diviser cellule en cellule\_haut et cellule\_bas;*

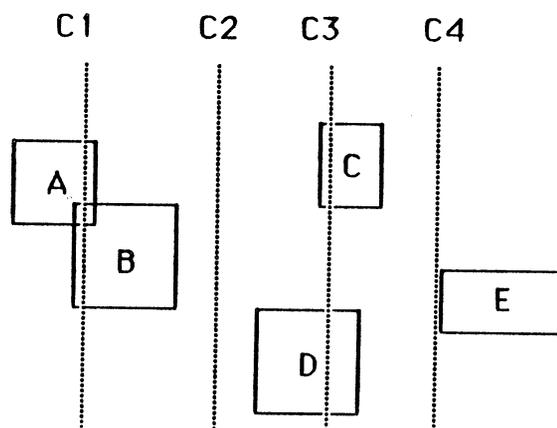
*coupe\_verticale (cellule\_haut);*

*coupe\_verticale (cellule\_bas);*

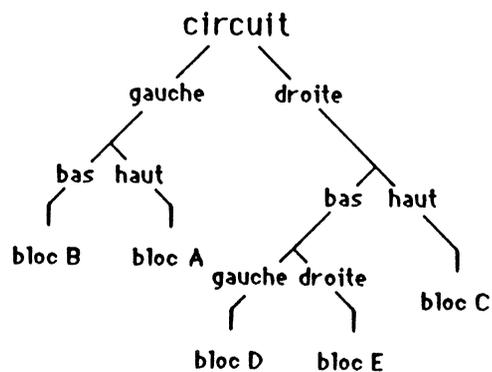
*fin;*

*fin;*

EXemple: a partir du pré-placementv suivant,



on obtient une structure arborescente du type:

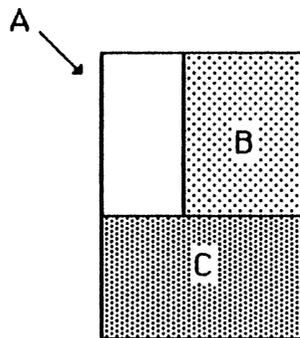


De cette façon, on ne modifie pas, ou peu, les positions relatives des blocs:

- Un bloc situé au dessus d'un autre reste au dessus.
- Un bloc situé à gauche d'un autre reste à gauche.

On peut maintenant connaître facilement la taille de chacune des cellules du circuit en utilisant des procédures récursives qui évaluent comme étant la largeur (ou hauteur) d'une cellule, soit la somme des deux largeurs (ou hauteurs) des cellules de niveau inférieur, soit le maximum de ces deux largeurs (ou hauteurs), ceci en fonction du type de coupe (horizontale ou verticale).

Exemple:



largeur de A = max ( largeur de B, largeur de C )

hauteur de A = hauteur de B + hauteur de C

La fin de la récursivité est donnée par l'arrivée sur une cellule terminale dont largeur et hauteur sont connues puisqu'il s'agit de celles du blocs correspondant. Il est évident que la connaissance des largeurs des cellules à n'importe quel niveau de l'arbre permet de déterminer leurs coordonnées exactes et donc d'obtenir les coordonnées des blocs.

**procedure largeur (cellule) ;**

**debut**

**choix ligne\_de\_coupe**

**horizontale :**

**largeur := max (largeur (cellule.superieure),  
largeur (cellule.inferieure));**

**verticale :**

**largeur := largeur (cellule.superieure)+  
largeur (cellule.inferieure);**

**fin;**

**procedure hauteur (cellule) ;**

**debut**

**choix ligne\_de\_coupe**

**verticale :**

**hauteur := max (hauteur (cellule.superieure),  
hauteur (cellule.inferieure));**

**horizontale :**

**hauteur := hauteur (cellule.superieure)+  
hauteur (cellule.inferieure);**

**fin;**

Cette méthode a l'avantage d'être extrêmement rapide et de ne pas trop perturber le placement obtenu en sortie d'analyse factorielle. De plus, on peut définir des canaux entre les blocs pour assurer le tracé ultérieur des connexions. Pour ceci, les dimensions d'une cellule terminale ne sont plus les mêmes que celles du bloc correspondant, mais celles du bloc augmentées d'un certain epsilon calculé en fonction du nombre de connexions estimées autour du bloc.

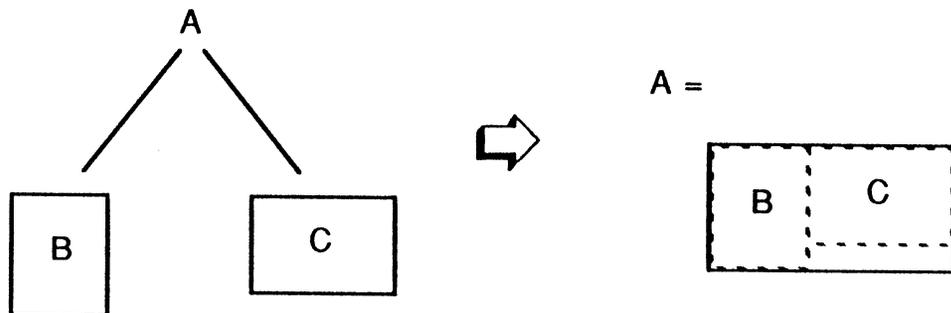
Lors de cette étape de compaction il est indispensable de prendre en compte

l'existence des blocs mous. En effet ces blocs n'ont pas reçu d'orientation lors du placement initial et n'ayant pas de dimensions fixes ils ont été par défaut assimilés à des surfaces carrées dont le côté est égal à la racine de la surface précisée initialement.

C'est donc à ce stade que les déformations et donc les dimensions finales de chacun de ces blocs mous pourront être définies avec précision. Pour ceci un ensemble de règles est défini. Ces règles utilisent la description arborescente obtenue par la méthode décrite précédemment.

Règle 1: une cellule terminale correspondant à un bloc dur est définie comme cellule **dure**. De même une cellule terminale correspondant à un bloc mou est définie comme une cellule **molle**.

Règle 2: si une cellule est constituée de deux cellules dures ses dimensions sont calculées selon la méthode exposée plus haut. Cette cellule est alors considérée elle même comme cellule de type dur.



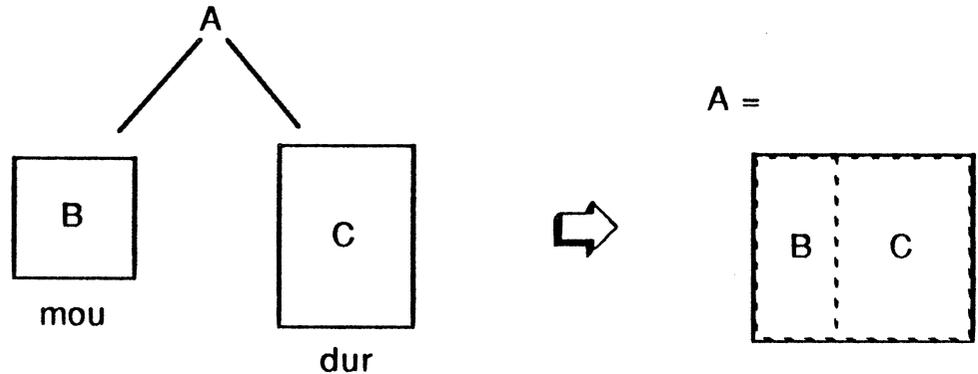
Règle 3: si une cellule est constituée d'une cellule de type dur et d'une cellule de type mou les dimensions de cette dernière sont calculées de la façon suivante:

soit  $S$  la surface du bloc mou,

soit  $A$  et  $B$  les longueurs de ses deux cotés,

soit  $L$  la longueur du coté du bloc dur contigu au bloc mou,

$$A = L,$$
$$B = S/A.$$



La cellule de type mou devient alors dure et on se ramène à la règle numéro 2.

Règle 4: si une cellule est constituée de deux cellules molles, elle est définie du type mou et sa surface est égale à la somme des surfaces des deux cellules qui la constituent.

Règle 5: lorsqu'une cellule molle passe comme cellule de type dur les dimensions des cellules molles qui la constituent sont calculées de la façon suivante:

soit  $A$  et  $B$  les dimensions de la cellule englobante,

soient  $S_1$  et  $S_2$  les surfaces des deux sous cellules,

soient  $A_1, B_1, A_2, B_2$  les dimensions respectives des deux sous cellules,

$$A_1 = A_2 = A,$$

$$B_1 = S_1/A_1 = B - B_2.$$

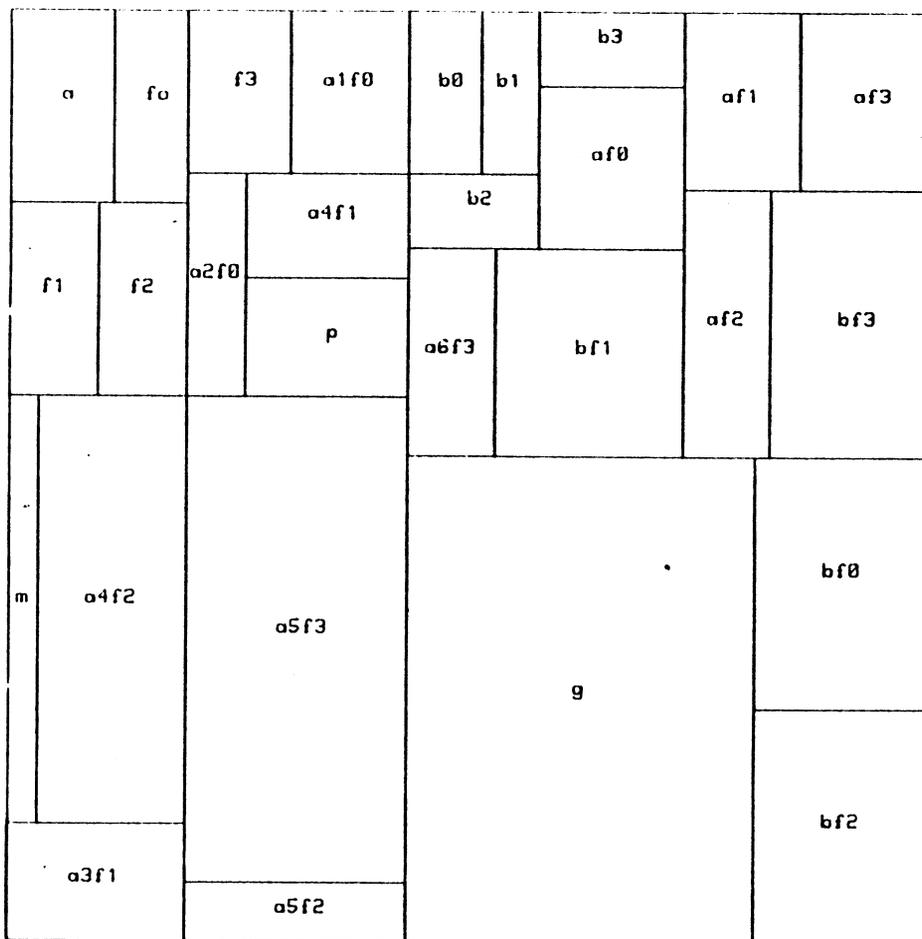
Si ces sous cellules sont terminales, les dimensions calculées sont affectées aux blocs correspondants.

Règle 6: si l'ensemble du circuit correspond à une cellule molle (si tous les blocs composant le circuits sont de type mou) ses dimensions sont celle d'un carré dont la surface est égale à la somme des surfaces de tous les blocs.

Ces six règles sont appliquées à la structure arborescente décrivant le circuit. Un premier parcours ascendant de l'arbre permet de définir toutes les cellules dures et leurs tailles respectives. Un parcours descendant total ou partiel permet de définir les tailles des cellules molles et finalement des blocs de types mous.

Cette méthode permet d'obtenir la meilleure compaction possible des blocs mous puisqu'aucun vide n'est laissé entre eux.

Exemple de plan de masse obtenu sur un circuit ne comportant que des blocs mous:



Un problème peut cependant apparaître à cause des rapports de géométrie maximaux autorisés pour chacun des blocs mous. En effet les dimensions calculées ne font pas intervenir ce paramètre. On pourra palier cet inconvénient en modifiant quelque peu les règles 3 et 5 de façon à vérifier le rapport de géométrie maximal autorisé.

Soient  $A$  et  $B$  les dimensions du bloc mou calculées comme ci-dessus,  
soit  $S$  sa surface,  
soit  $R$  le rapport de géométrie maximal autorisé,

Si  $1/R \leq A/B \leq R$  alors pas de problème

Sinon si  $A/B < 1/R$  les contraintes à respecter sont:

$$A/B = 1/R$$

$$A.B = S$$

$$\text{d'où } B^2 = R.S \text{ et } A^2 = S/R$$

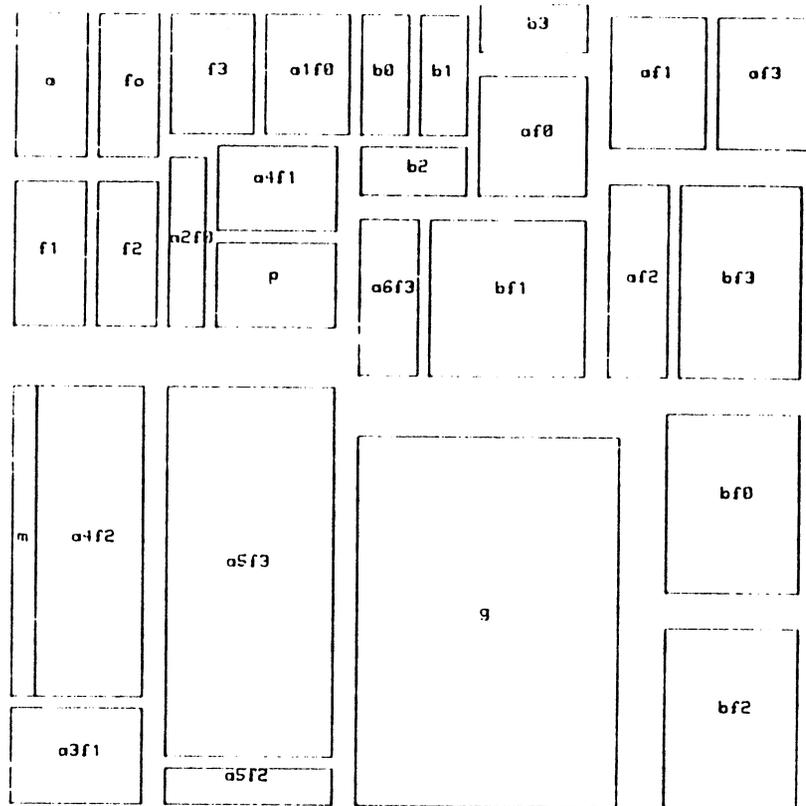
sinon  $A/B > R$ , les contraintes sont alors:

$$A/B = R$$

$$A.B = S$$

$$\text{d'où } B^2 = S/R \text{ et } A^2 = S.R$$

De cette façon, toutes les contraintes géométriques seront respectées mais un problème supplémentaire se pose alors. En effet dans le cas précédent, les surfaces des cellules molles étaient calculées comme étant la somme des cellules molles inférieures sachant qu'aucun vide ne pouvait être laissé entre elles. Cette "propriété" n'étant plus respectée on ne peut plus faire d'estimation précise. La solution consiste alors à majorer les surfaces des blocs mous (pourcentage ou constante) de façon à permettre la prise en compte de ces contraintes. Cette majoration pourra également être utile pour assurer le routage entre les blocs. Une étape finale de compaction est alors appliquée dans le but de supprimer les espaces inutilement laissés libres entre les blocs. Le résultat obtenu est le suivant:



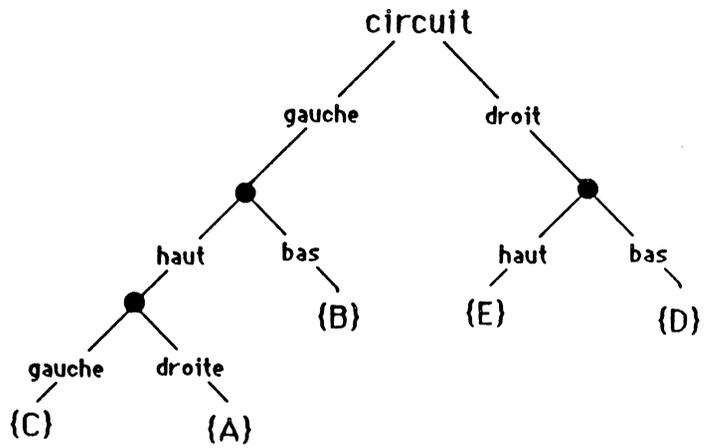
### III.4.b. Le routage global

Sans routage, un placement ne peut être validé et ne sert à rien. Cependant, sans être obligé de réaliser un routage fin de l'ensemble des connexions, il est possible et même recommandé de faire une estimation de ce tracé.

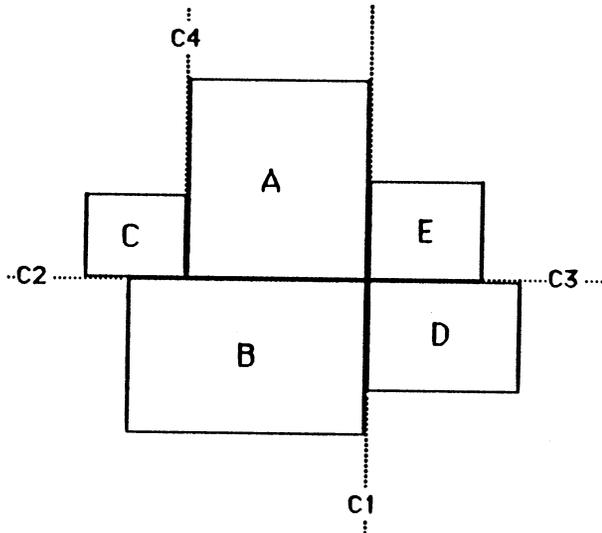
C'est le rôle du "global router" qui prévoit un itinéraire pour chacune des connexions. Cet itinéraire est défini par la liste des canaux empruntés. Une fois l'itinéraire établi pour chaque équipotentielle, on peut déterminer ou au moins évaluer approximativement, la taille de chaque canal. En effet, cette taille (c'est à dire la largeur du canal) dépend du nombre de connexions qui l'empruntent et de la largeur occupée par chacune.

De la largeur des canaux va dépendre le placement final. C'est à ce stade que seront affinées les positions de chaque bloc.

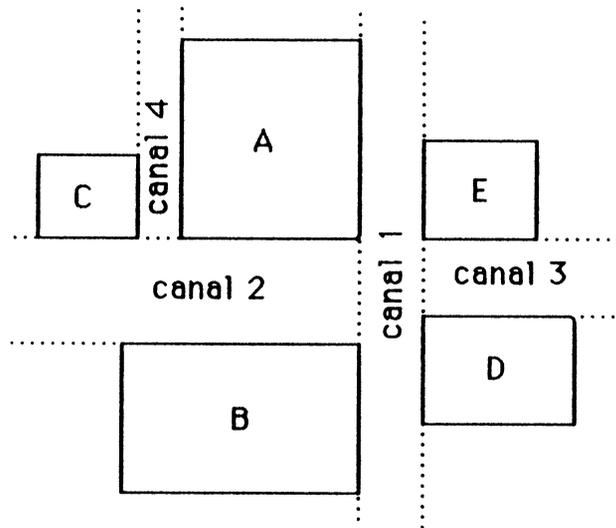
Si on regarde la structure définie dans le paragraphe précédent et utilisée pour le compactage des blocs:



Et si on regarde la structure topologique correspondante :



On remarque que chaque ligne de coupe ( $C_1 \dots C_n$ ) et donc chaque noeud de l'arbre binaire définit un canal.



Donc, si l'on connaît la largeur de chaque canal, il est très simple de calculer les nouvelles positions de chaque composant en introduisant ce nouveau paramètre dans les fonctions exposées plus haut.

**procedure** largeur (cellule) ;

**debut**

**choix** ligne\_de\_coupe

horizontale :

largeur := max (largeur (cellule.superieure),  
largeur (cellule.inferieure));

verticale :

largeur := largeur (cellule.superieure)+  
largeur (cellule.inferieure) +  
occupation canal (ligne de coupe);

**fin;**

**procedure** hauteur (cellule) ;

**debut**

**choix** ligne\_de\_coupe

verticale :

hauteur := max (hauteur (cellule.superieure),  
hauteur (cellule.inferieure));

horizontale :

hauteur := hauteur (cellule.superieure)+  
hauteur (cellule.inferieure)+  
occupation canal (ligne de coupe);

**fin;**

Comme il a été dit plus haut à propos des blocs de type mou, une majoration de la surface occupée est indispensable afin de respecter les contraintes géométriques. Cette majoration peut être suffisamment importante pour permettre l'insertion ultérieure des canaux.

On voit que cette opération est particulièrement simple et rapide. La seule difficulté réside dans l'évaluation de l'encombrement de chaque canal. Pour ceci, de nombreuses solutions sont offertes [DUP 84]. On pourra par exemple utiliser un algorithme du type Lee [LEE 61].

Cet algorithme offre la possibilité de réaliser une répartition homogène des connexions sur l'ensemble des canaux en prenant en compte, non seulement un critère de longueur, mais également un critère de densité. Il est également possible de fixer la largeur de certains canaux en autorisant seulement un nombre fixé de connexions, de cette façon il est possible de forcer la position relative de deux blocs voisins afin d'éviter la perte d'un aboutement par exemple. Perte qui provoquerait des perturbations trop importantes au niveau du placement (voir chapitre I).

Le principe d'un tel algorithme est le suivant [BEL 84]:

- on se positionne à l'une des extrémité d'une équipotentielle,
- on propage un front d'onde dans les canaux autour de ce point,

- on associe un coût à chaque itinéraire ainsi établi,
- une fois la deuxième extrémité de l'équipotentielle atteinte, on choisit le chemin de plus faible coût.

Le coût d'un chemin est incrémenté à chaque étape d'évolution de front d'onde. Il dépend:

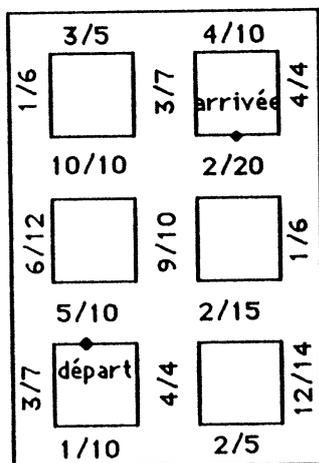
- de la distance ainsi parcourue,
- de l'encombrement des canaux utilisés.

Ce deuxième paramètre est introduit afin d'homogénéiser les connexions sur la surface du circuit. Ainsi, la valeur ajoutée au coût à chaque étape de calcul est du type :

$$\frac{\text{occupation\_max}}{\text{occupation\_max} - \text{occupation}}$$

Où  $\text{occupation\_max}$  est le taux d'occupation maximum autorisé pour le canal courant et  $\text{occupation}$  est le nombre de connexions empruntant déjà ce canal. Il faut noter que cette dernière valeur doit être mise à jour après que chaque itinéraire ait été défini.

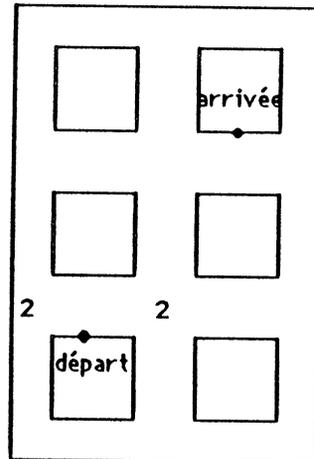
Exemple:



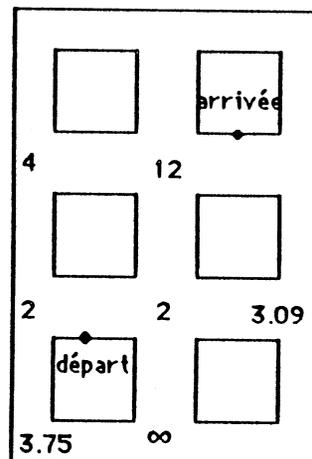
Les chiffres dans les canaux indiquent le taux d'occupation maximale et le taux d'occupation courant sous la forme:

max / courant

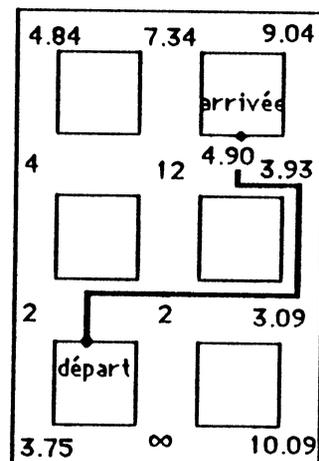
premier pas  
de calcul



second pas  
de calcul



dernier pas  
de calcul et  
chemin choisi



### III.5 Perspectives

Malgré toutes les améliorations qui ont pu être apportées à l'analyse factorielle d'un tableau de distance pour pouvoir être utilisée efficacement dans le domaine de la génération de plans de masse, certains défauts peuvent subsister. Ces défauts concernent d'une part la prise en compte des canaux de routage puisque l'on ne peut se baser que sur une estimation très approximative de leurs encombrement, et d'autre part le fait que le concepteur ne peut guère imposer les positions ou les dimensions définitives des blocs mous. Pour ceci il est possible de travailler par affinements successifs.

En effet, le point de départ de l'algorithme de placement est une matrice de distances. Ces distances, bien que raffinées au maximum, restent approximatives et ne reflètent pas forcément la volonté du concepteur puisqu'elles sont toutes évaluées de façon automatique. Par contre, après le premier passage à travers l'algorithme, les bases de travail sont beaucoup plus sérieuses puisque qu'une représentation géométrique concrète est fournie à l'utilisateur. Une nouvelle matrice de distances (euclidiennes cette fois) peut alors être établie. Il est évident que d'utiliser cette matrice comme point de départ d'une procédure d'analyse factorielle ne changera rien au résultat (voir exemple de la carte de France) mais on peut parfaitement envisager d'appliquer quelques modifications à cette matrice de façon à prendre en compte des contraintes supplémentaires lors d'une deuxième passe.

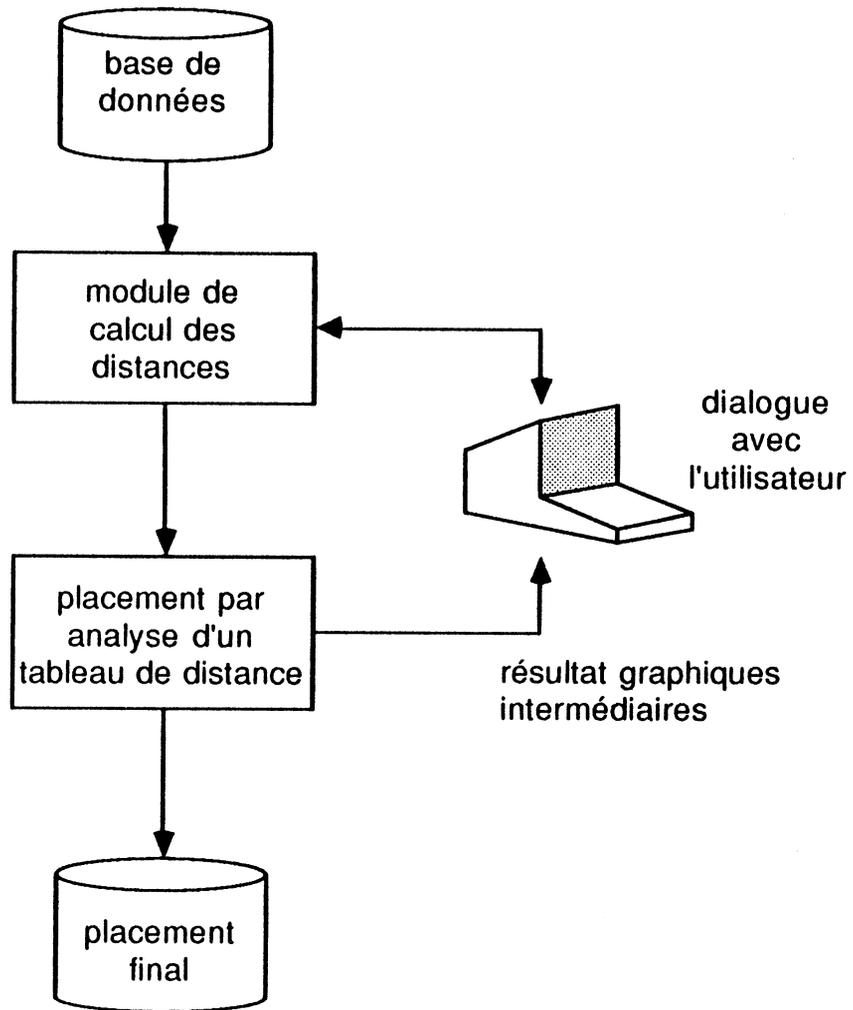
Prenons l'exemple de deux blocs voisins distants d'une largeur de canal  $L$ . S'il s'avère après passage d'un global router que cette largeur est insuffisante, il est tout à fait possible de modifier la matrice des distances obtenues après le premier passage en posant  $L' = L + k$  ou  $L' = k.L$  ou  $k$  représente une constante d'accroissement de la largeur du canal. Cette opération peut évidemment être appliquée sur l'ensemble des distances réelles mesurées sur le premier plan fourni.

A ce stade, deux types de modifications peuvent être envisagées. Des modifications automatiques comme celle exposées plus haut; ces modifications sont le

résultats d'évaluation numériques précises quant à la structure finale du circuit. Et des modifications manuelles; ces modifications sont apportées par le concepteur soit par souci "d'esthétique" (aboutement exact de deux blocs) soit pour des raisons plus techniques (problèmes liés à la dissipation thermique, aux lignes d'alimentation, aux parasites haute-fréquence, etc...).

Ces transformations étant effectuées, un second plan, plus "précis" que le premier peut être fourni. Tant qu'il ne donne pas totalement satisfaction, la même étape peut être répétée. On se rend compte que pour qu'une telle méthode soit utilisable il faut impérativement que les temps de calculs utilisés pour la génération de chaque nouveau plan de masse soit suffisamment brefs. Moyennant cette contrainte qui est bien respectée grâce à l'utilisation des méthodes statistiques, il est possible d'obtenir un résultat respectant l'ensemble des contraintes possibles.

Un tel système n'a malheureusement pu encore être développé. Un projet utilisant ce qui a déjà été fait et s'appuyant sur les idées émises ici et notamment celle proposant les itérations successives est toutefois lancé.





## **CHAPITRE IV**

### **LE PROGRAMME ALFA**



## **IV. Le programme ALFA**

Afin de valider la méthode d'analyse factorielle appliquée au plan de masse et présentée dans le chapitre précédant, un programme a été développé. Ce programme reprend la plupart des opérations décrites. De nombreuses extensions permettraient toutefois de le rendre plus encore plus performant.

Le but de ce programme est de pouvoir fournir le plan de masse d'un circuit de façon interactive. C'est à dire avec des temps de calcul les plus réduits. Ceci est rendu possible grâce aux méthodes factorielles très rapides.

### **IV.1. Les données**

Afin de répondre au mieux à toutes les contraintes citées au premier chapitre, il est nécessaire de définir un langage de description spécifique à la génération de plans de masse.

Ce langage doit permettre de décrire les composants à placer: on doit donc différencier deux groupes fondamentaux: les blocs "mous" et les blocs "durs".

Un bloc mou est défini par la surface qu'il occupe. Ces blocs sont considérée comme des rectangles. Des surfaces plus complexes sont exclues afin de limiter le problème et ainsi pouvoir assurer des temps de calcul raisonnables pour un système interactif.

Un paramètre supplémentaire peut toutefois être pris en compte. Il s'agit du rapport de géométrie maximal autorisé, c'est à dire le rapport maximal entre largeur et hauteur du bloc. Ceci afin de ne pas introduire de déformation trop importante sur les blocs. Déformation qui risquerait de modifier de manière excessive l'évaluation de surface préalablement établie en augmentant les longueurs de connexions internes au bloc.

Le deuxième type de bloc que l'on est amené à considérer est le type dur. Dans ce cas, il s'agit de blocs dont la topologie est parfaitement déterminée. Ainsi, ces blocs sont définis par les deux paramètres suivants: hauteur et largeur.

La description des blocs doit être complétée par la description des points d'entrée/sortie. Comme pour le type des blocs, deux types de contacts sont envisageables: les contacts flottants et les contacts fixes.

Les contacts flottants sont des contacts dont la position n'est pas définie sur la périphérie du bloc. Les blocs mous n'auront que ce type de contact puisque leur topologie n'est pas définie. Même si ce choix semble être quelque peu restrictif il reste tout de même justifié.

Les contacts fixes, eux, ont une position parfaitement connue. Elle est donnée par les deux paramètres suivants: face et coordonnée sur cette face. Les blocs durs n'auront généralement que des contacts fixes. On peut cependant penser qu'il est raisonnable d'autoriser les contacts flottants sur des blocs durs. Soit que la position du contact n'est pas établie, soit que le contact peut être pris en de multiples points de la périphérie du bloc.

Un cas particulier est fait des plots d'entrée/sortie du bloc. En effet ces plots ne peuvent être considérés comme des blocs standards puisqu'il sont obligatoirement situés à la périphérie du circuit. D'autre part leur taille est généralement faible devant les autres blocs à placer. Le choix de leur description est donc le suivant: le circuit est considéré comme un bloc et les plots d'entrée/sortie comme des contacts sur ce bloc.

Ceci à l'avantage de permettre de traiter des circuits possédant les mêmes caractéristiques que les blocs: c'est à dire durs ou mous, avec des contacts fixes ou flottants.

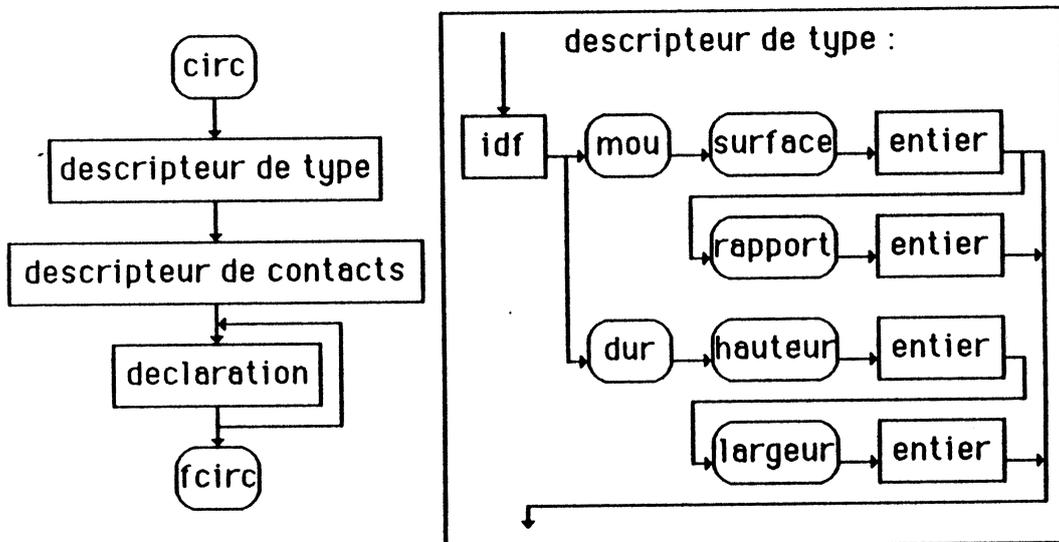
Le deuxième avantage est de donner exactement la même structure à l'ensemble du circuit qu'à un bloc et de pouvoir ainsi assurer une hiérarchisation facile puisqu'un bloc pourra à son tour être vu comme un circuit pour des blocs d'un niveau inférieur.

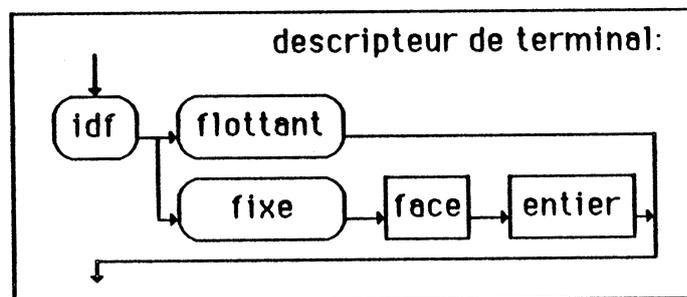
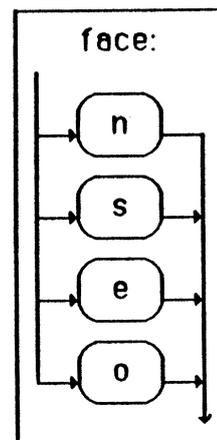
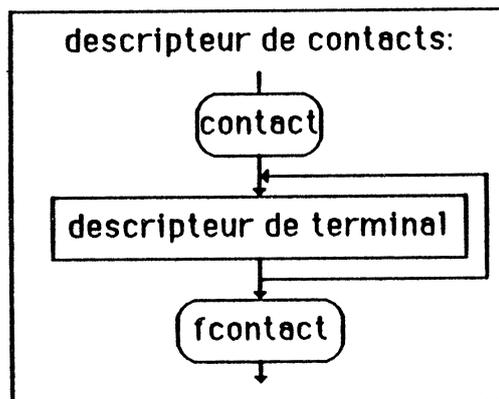
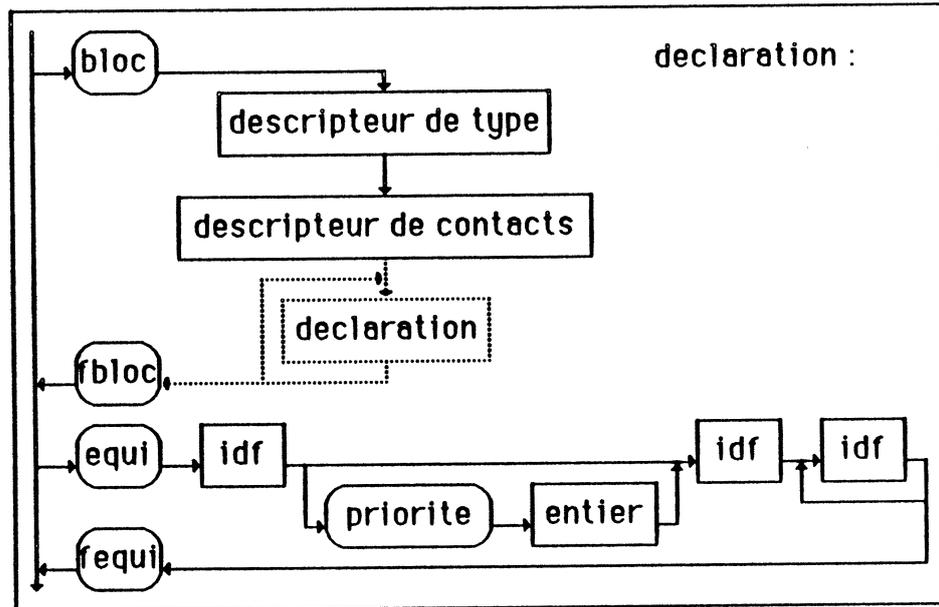
Le langage de description doit également permettre de décrire les connexions entre ces blocs. Ces blocs sont reliés entre eux par des équipotentiels. Ces

équipotentielle sont décrites à partir de la liste des points de contacts situés à la périphérie des blocs et auxquels elles sont reliées.

Un paramètre supplémentaire peut être ajouté. Il s'agit de la priorité d'une équipotentielle. Ce paramètre permet soit de prendre en compte la criticité d'une équipotentielle, soit de représenter un ensemble de connexions (bus par exemple) par une seule équipotentielle.

Une grammaire réduite répondant à ces contraintes a donc été définie:





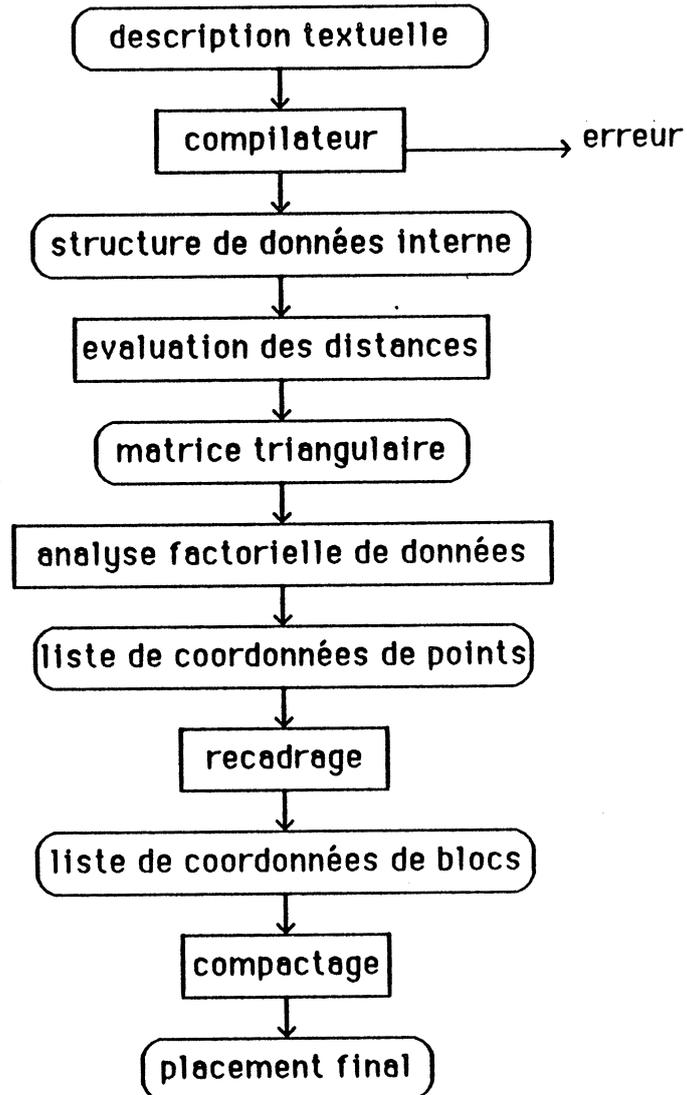
On remarque sur cette grammaire que le rapport de géométrie pour un bloc est optionnel. Lorsqu'il n'est pas précisé, il vaut par défaut 2.

Le paramètre de priorité affecté à chaque équipotentielle est également optionnel, il est par défaut de 1.

La partie en pointillé qui permet une hiérarchisation au niveau des blocs est prévue mais non encore réalisée.

#### **IV.2. Les différentes étapes du programme**

Le programme ALFA regroupe les modules suivants:



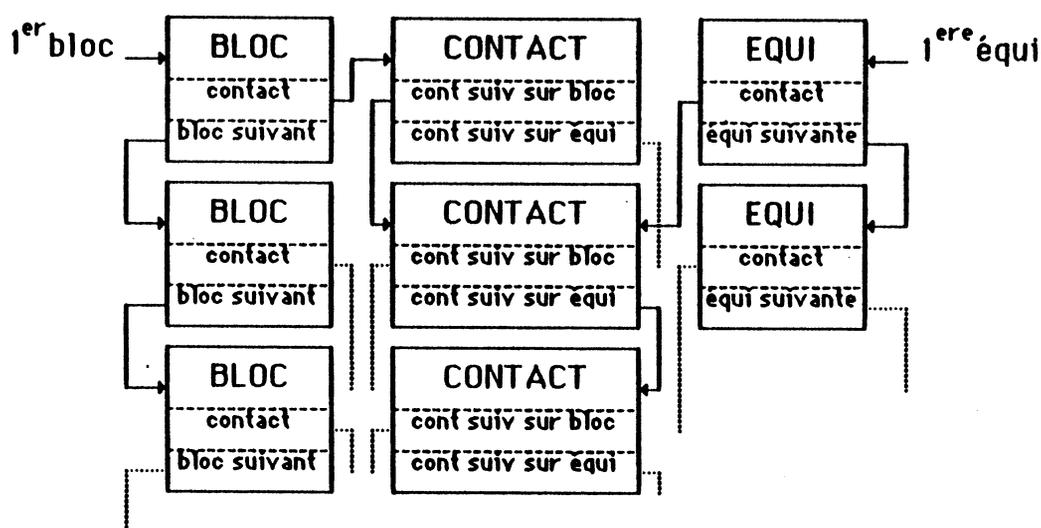
Le module compilateur permet de fournir une structure de donnée interne à partir d'une description textuelle du circuit. Il permet en outre de vérifier toutes les éventuelles erreurs de description:

- erreur de syntaxe,
- double déclaration d'un bloc, d'une équipotentielle ou d'un contact.

De plus le module compilateur vérifie que chaque contact déclaré par l'intermédiaire d'un bloc l'est également par l'intermédiaire d'une équipotentielle.

La structure interne créée à partir de la description textuelle utilise différentes listes :

- Une liste correspondant aux blocs.
- Une liste correspondant aux équipotentielle.
- Un ensemble de listes doublement chaînées correspondant aux contacts appartenant à un bloc (premier chaînage) et à une équipotentielle (deuxième chaînage).



Ainsi tout bloc pointe sur une liste de contacts qui lui appartiennent, et toute équipotentielle pointe sur une liste de contacts auxquels elle est raccordée. On peut de cette façon tout connaître sur le circuit en parcourant l'une ou l'autre des listes. Par la suite, un chaînage supplémentaire sera ajouté depuis les blocs pour accéder aux niveaux inférieurs de la hiérarchie.

Le module d'évaluation de distances permet de calculer les corrélations entre tous les blocs.

Pour ceci, une première étape d'éclatement des blocs est réalisée: les blocs durs sont représentés par 4 points alors que les blocs mous sont toujours représentés par un seul point comme il a été expliqué dans le chapitre III. A chaque point représentant un des coins d'un bloc dur, sont associées toutes les connexions arrivant sur le quart correspondant.

Puis les corrélations sont déterminées à partir du nombre de connexions. La fonction de distance utilise les différents critères exposés au chapitre III. Afin d'utiliser la fonction désirée chacun des critères précités (nombre de connexions, taille des blocs...) est affecté d'un coefficient que l'utilisateur peut modifier à son gré.

Ce module génère un tableau bloc/bloc triangulaire. La fonction de lissage également décrite dans le chapitre précédent est appliquée à cette matrice.

Le module d'analyse factorielle traite alors ce tableau de distances pour générer l'ensemble des coordonnées des points. On trouvera de plus amples renseignements sur la méthode utilisée dans [BEL 85] .

Une fois les coordonnées de tous les points connues, il est évident que dans le cas des blocs durs celles-ci ne correspondent pas à une position réelle du bloc. Le module de recadrage détermine la position du bloc parallèlement aux axes tel que celle-ci se rapproche le plus possible de la position fournie. Le programme détermine ainsi les opérations de symétrie et de rotation à appliquer au bloc. Les torsions éventuelles sont éliminées par la méthode présentée au chapitre III.

A ce stade le programme fournit la liste des coordonnées des blocs, mais des chevauchements ou des espacements excessifs peuvent toutefois apparaître. Le module de compaction permet de supprimer ces problèmes. Il utilise la méthode des lignes de coupe homogénéisant les surfaces.

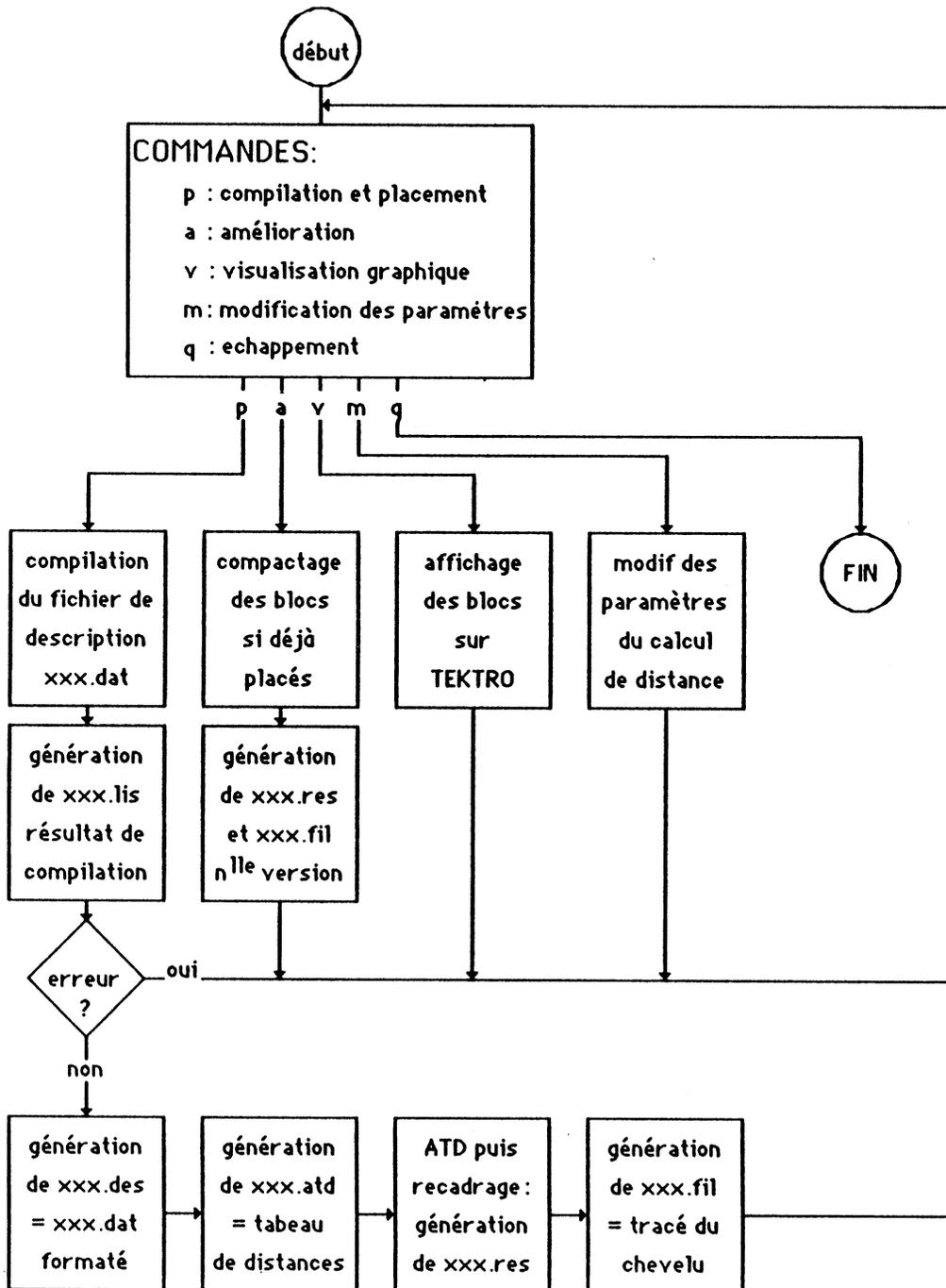
Le module de compaction permet également de définir les dimensions exactes de tous les blocs mous. En effet, jusqu'à cette étape, ces blocs étaient considérés comme carrés. L'application, lors de la compaction, de l'ensemble des règles décrites au chapitre III, permet de définir hauteur et largeur de chaque bloc mou en respectant la surface initiale et les limitations éventuelles du rapport de géométrie.

On obtient finalement la liste des coordonnées réelles des blocs.

Afin de rendre plus conviviale l'utilisation d'un tel programme, un module

graphique a été ajouté . Ce dernier permet de visualiser le placement obtenu soit en sortie de recadrage des blocs, soit en sortie de compaction.

L'organigramme général du programme est le suivant :



Le programme est écrit en Pascal standard et est implanté sur VAX-780 avec système VMS. Il ne nécessite aucune procédure de commande, l'ensemble des attachements de fichiers avec vérification étant intégrés. Il utilise pour les sorties graphiques un terminal du type TEKTRONIX 4112 et un traceur 4663. Les procédures graphiques sont incluses au programme qui est ainsi totalement transportable puisque les procédures d'analyse factorielle ne font pas appel non plus à une bibliothèque extérieure.

### **IV.3. Les limitations d'un tel programme**

Il est évident qu'un programme de placement utilisant des méthodes statistiques ne peut à coup sûr donner des résultats parfaits dans tous les cas. Une telle méthode doit en fait servir de guide au concepteur en lui donnant dans un temps très bref une solution acceptable à son problème. Cette solution permet de voir un peu plus clair dans un problème complexe et peut servir de base solide à une solution définitive. Il est donc prévu de coupler ce programme à un éditeur graphique permettant d'ajuster le placement de façon manuelle. D'autre part, les temps de calcul étant très courts, une solution non satisfaisante peut être remise en question en modifiant les paramètres de la fonction d'évaluation de distance et en recommençant le placement.

De même, comme il a été dit à la fin du chapitre précédent, il est prévu de créer un système complet utilisant les méthodes employées dans ALFA.

Ce système fournira de nombreuses options à l'utilisateur:

- Découpage éventuel du problème:
  - de façon manuelle,
  - de façon automatique par classification. Dans ce cas, les paramètres utilisés

dans l'algorithme seront accessibles et ajustables par l'utilisateur.

- Possibilité de choisir la fonction de calcul des distances entre blocs. Les différentes options présentées au chapitre III seront proposées.

- Visualisation des résultats intermédiaires avec évaluation des différents paramètres: encombrement et taille des canaux, surface englobante...

A ce stade, il sera possible de modifier, soit de façon manuelle, soit de façon automatique, comme il a été dit à la fin du chapitre III, les distances euclidiennes obtenues. Ceci permettra d'affiner un placement en rebouclant tant que le résultat ne sera pas totalement satisfaisant, sur la procédure d'analyse du tableau de distances.

- Un global router suivit d'un channel router devraient clore le système en fournissant un résultat définitif complet.

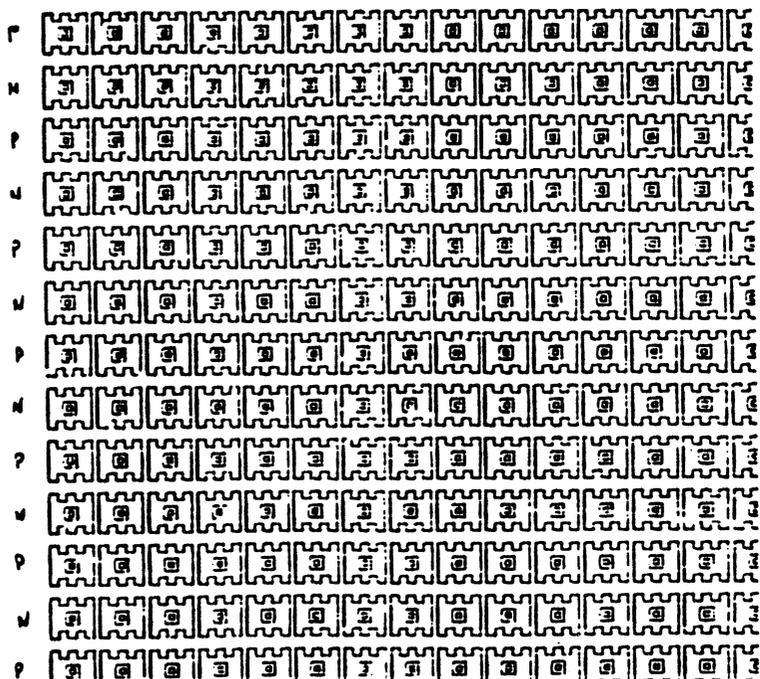
Remarque: applications aux PILs

Une nouvelle génération de réseaux prédiffusés est en train d'apparaître, il s'agit des réseaux PIL, c'est à dire les Prédifusés à Interconnexions Libres connus également sous le nom de mers de transistors.

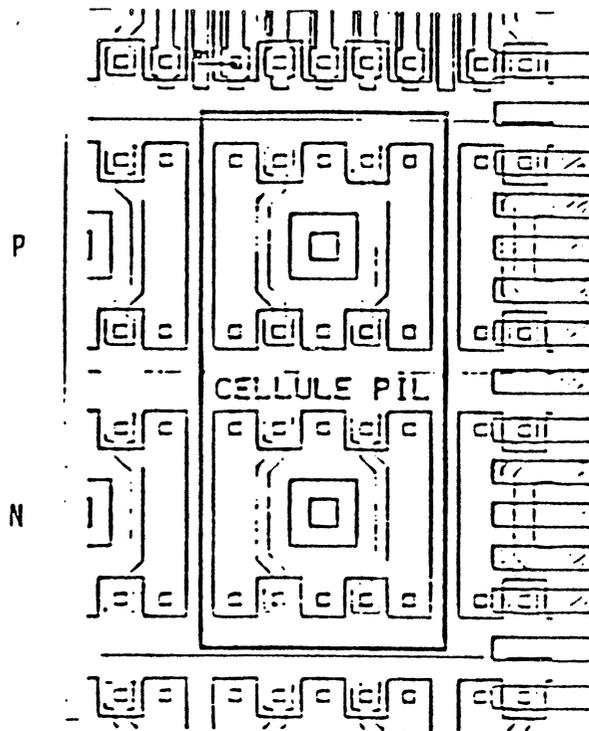
Dans ce type de réseau il n'y a plus de canaux prévus. Des transistors sont implantés sur toute la surface du circuit. Les zones réservées aux connexions sont obtenues en sacrifiant certaines cellules actives (les transistors sont alors inutilisés)

[AUD 85].

Structure générale  
d'un réseau P.I.L



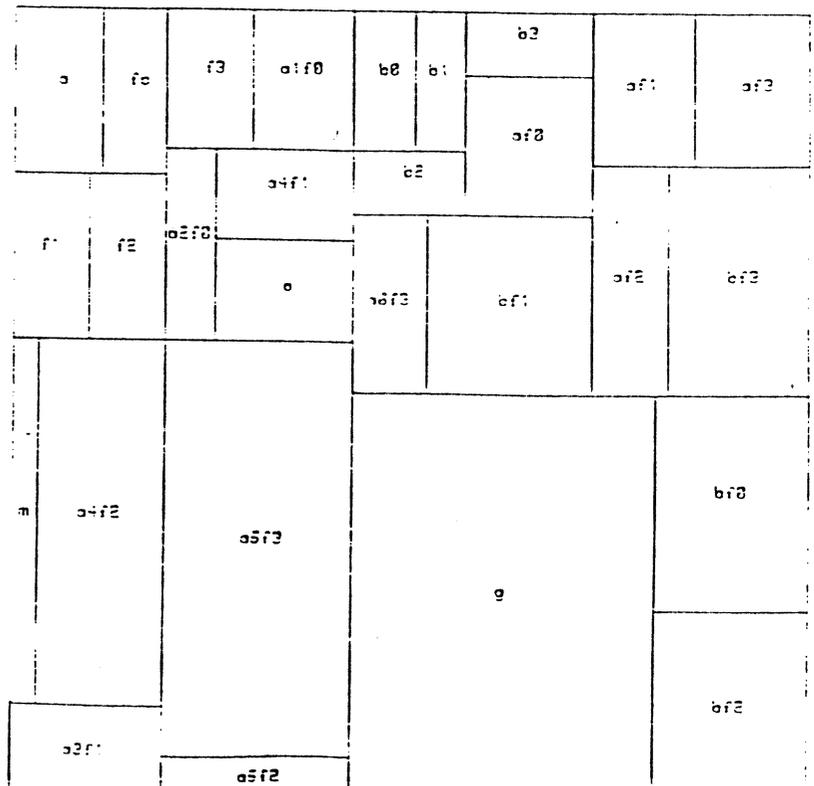
Structure d'une  
cellule de réseau  
P.I.L



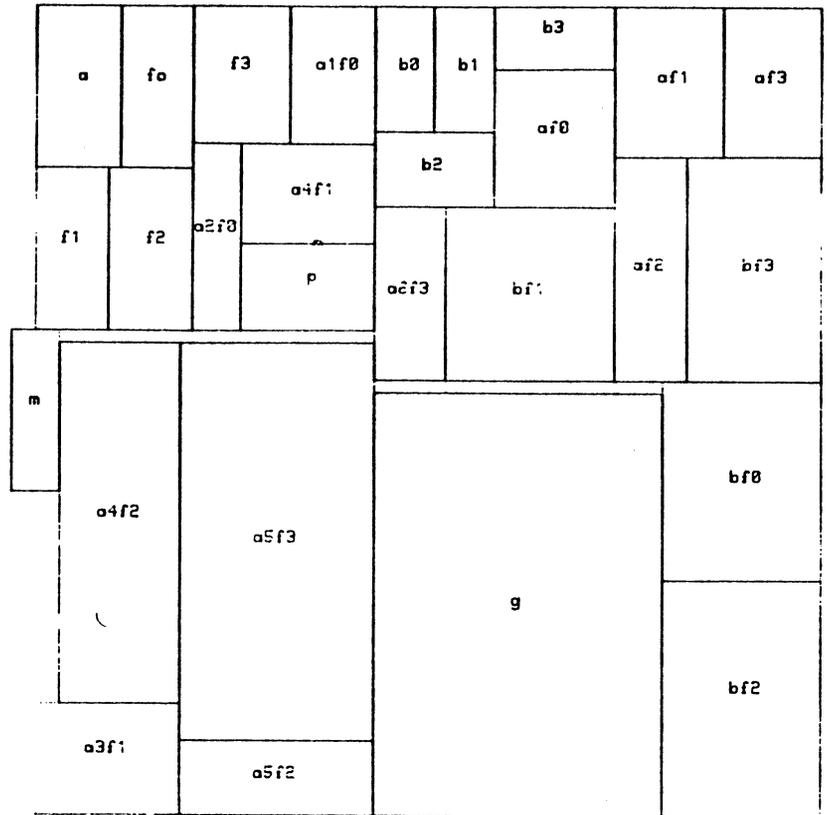
Dans un tel type de circuits, les éléments de bibliothèque sont généralement plus complexes et surtout plus importantes que dans les réseaux prédifusés classiques. Le placement fait alors partie du domaine de la génération de plans de masse. La nouvelle contrainte à respecter provient de la présence d'un pas de grille égal aux dimensions de la cellule élémentaire. Un programme utilisant l'analyse factorielle de données comme ALFA peut donc sans de grandes modifications apporter une solution intéressante aux problèmes apportés par ces nouveaux réseaux.

IV.4. Exemples

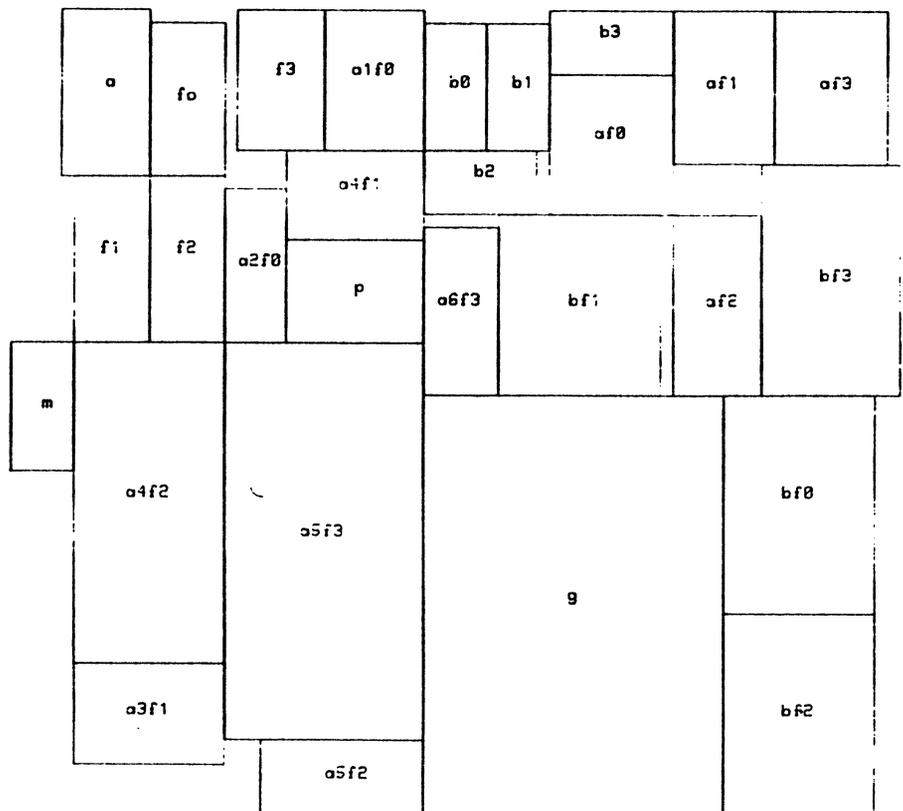
Circuit UAL 74181  
composé de  
blocs mous avec un  
rapport de géométrie  
quelconque



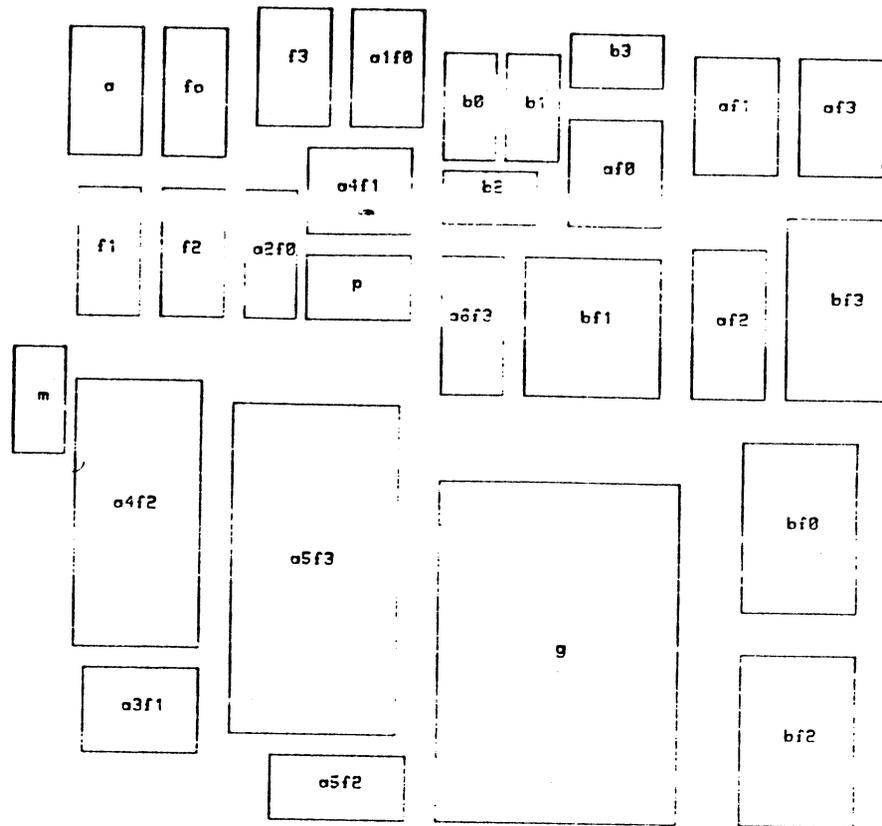
Même circuit, mais  
le rapport de géométrie  
est limité à 4



Rapport de géométrie  
limité à 2

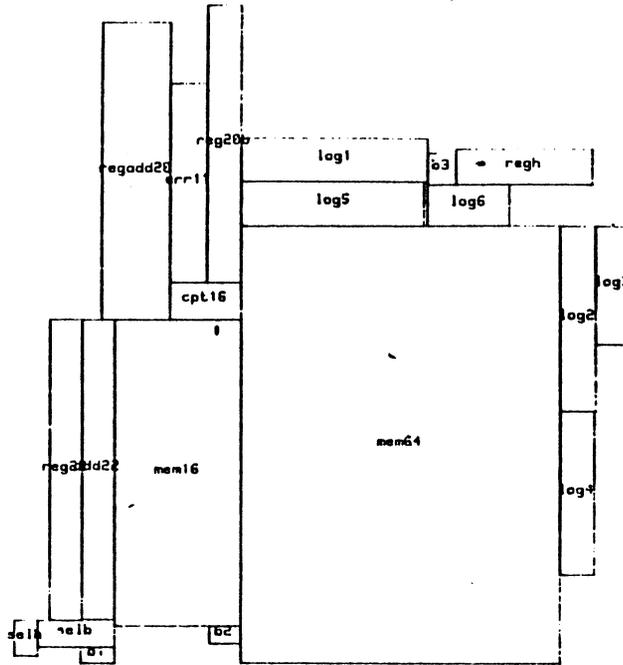


Ajout d'un  
espace  
entre les blocs  
pour permettre  
le routage ultérieur

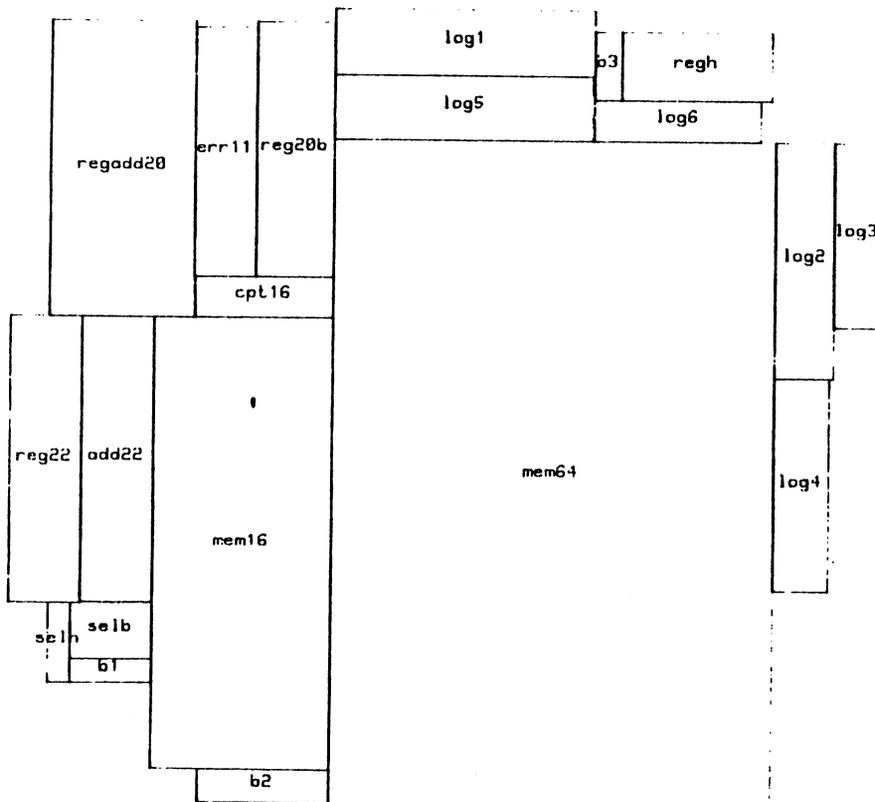


Le temps CPU, mesuré sur un micro-VAX, pour cet exemple est de l'ordre de 30 secondes. Ce temps comprend: la compilation du fichier texte, le placement par analyse factorielle et la compaction.

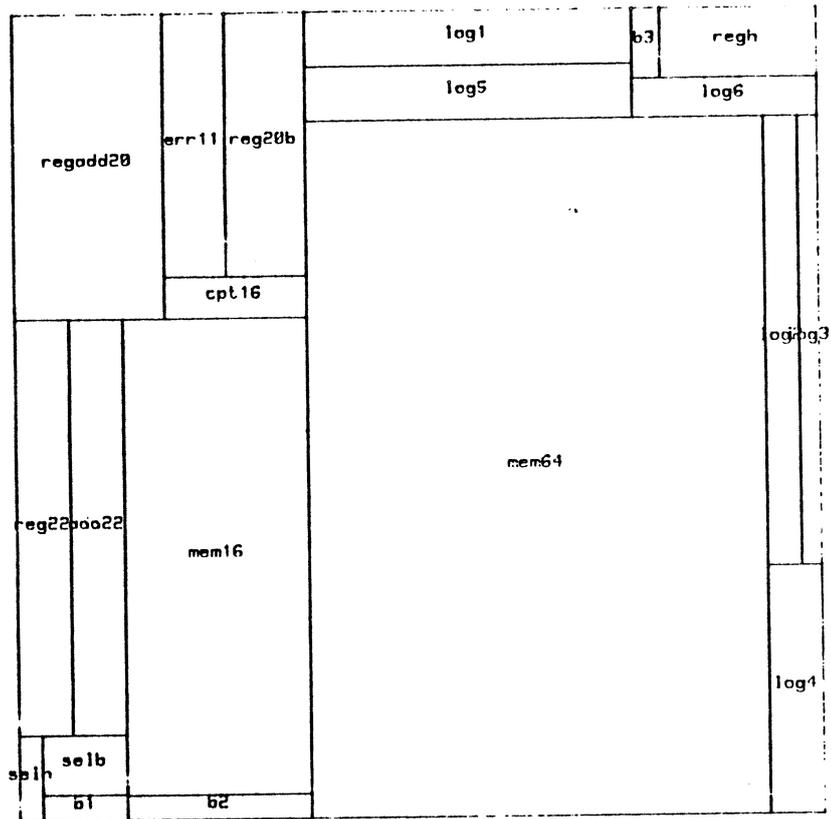
Exemple de circuits constitué de blocs durs:



Même circuit, mais les blocs sont considérés comme mous avec un rapport de géométrie limité à 4.



Dans le cas ou le rapport de géométrie est quelconque, on obtient:



Le temps CPU pour cet exemple traité avec des blocs durs est de l'ordre de 3 minutes  
La différence s'explique par le fait que le nombre d'éléments devant être manipulés  
par le programme d'analyse factoriel est multiplié par 4.

## **CONCLUSION**



## CONCLUSION

Bien que l'idée d'introduire l'analyse de données dans le domaine de la génération automatique de plans de masse ne soit pas totalement nouvelle, aucune application concrète entièrement basée sur de telles méthodes n'avait encore été développée.

Les différentes options présentées dans cette thèse et permettant de faire une évaluation précise des distances réelles entre les éléments à placer en fonction de distances virtuelles, ont permis d'obtenir un système mettant en évidence l'efficacité des méthodes mathématiques mises en oeuvre.

Il ne faut cependant pas penser que l'étude menée ici soit close. En effet, toutes les possibilités fournies par l'analyse de donnée et notamment par l'analyse d'un tableau de distances, sont bien loin d'avoir été toutes explorées.

Le but est maintenant de créer un système complet de génération de plans de masse regroupant :

- une bibliothèque de fonctions de calcul de distances à choisir par l'utilisateur,
- une bibliothèque de fonctions statistiques: analyse en composantes principales, analyse de tableaux de distances, classification...
- un système d'évaluation des qualités d'un placement permettant l'ajustement des distances précédemment obtenues afin de respecter au mieux les volontés du concepteur,
- un éditeur graphique permettant une intervention manuelle lors du placement,
- un routeur global permettant de définir l'occupation exacte des canaux,
- un routeur de canaux permettant d'obtenir le tracé fin des connexions entre les constituants du circuit.

Le logiciel ALFA, développé dans le cadre de cette thèse constitue donc le noyau d'un système complet de CAO de circuits intégrés. Son point fort vient non seulement de la qualité des résultats, mais surtout de la rapidité de calcul. Cette efficacité permet une utilisation interactive de l'outil de placement et donc un meilleur contrôle du résultat par le concepteur.

## BIBLIOGRAPHIE



**Références bibliographiques :**

AUD 85 : J.C. Audrix, P. Charransol, J. Michard, P.Delerue

Les P.I.L, une nouvelle génération d'architecture de réseaux

Colloque national conception de circuit à la demande, 1985

pp 209-224

BAR 85 : J.R. Barra, M. Becker, E.F. Kouka, M. Tricot

Elaboration de plan de masse en vue d'un placement automatisé de blocs

Colloque national conception de circuit à la demande, 1985

pp 324-339

BEL 84 : A. Bellon

Optimisation des méthodes d'implantation automatisée sur les réseaux structurés

Thèse de 3° cycle, Grenoble, déc. 1984

BEL 85 : D. Belaid

Progiciel INDA pour l'analyse des données

Thèse de 3° cycle, Université de Grenoble, octobre 85.

BOW 83: J.D. Bowman & al.

Placement of irregular circuit elements on non-uniform gate arrays

DAC 83, pp 637-643

BRE 83 : M.A. Brener, Dah-Juh Chyan

A placement algorithm for array processor

DAC 83, pp182-188

CHA 84 : P. Chaisemartin, G. Saucier

Parallelism in the placement problem in structured logic  
EDA 84, pp 179-183

CHA 85 : P. Chaisemartin, C. Jay

Conception d'un circuit de signature  
Colloque national conception de circuits à la demande, 1985,  
pp 131-148

CHA 86 : P. Chaisemartin

FADDA: an interactive floorplan design system  
Silicon Design Conference, 1986

&

FADDA2: a new interactive floorplan design system  
ICTC, Irlande, septembre 86.

DUP 84 : G. Dupenloup

Tracé automatique de canaux d'interconnexion  
Thèse de 3° cycle, INP Grenoble, juin 84.

FOU 82 : T. Foucart

Analyse factorielle, programmation  
Edition Masson, 1982

GAY 85 : J.G. Gay, R. Richter

Component placement in VLSI circuit using a constant pressure  
Monte Carlo method  
Integration, the VLSI journal, vol. 3, 1985, pp 271-282

HOR 81 : C. Horng, M. Lie

An automatic/interactive layout planning system for arbitrarily  
sized rectangular building blocks

DAC 1981, pp 293-300

IOS 83 : A. Iosupovici

A module interchange placement machine

DAC 1983, pp 171-174

JAN 85 : A. Janati

Implantation automatisée des circuits intégrés sur  
réseaux prédifusés CMOS

Thèse de 3<sup>e</sup> cycle, INP Grenoble, juillet 1985

KIR 83 : Kirkpatrick

Optimization by simulated annealing

Sciences, may 1983, Vol 220, n° 4598

pp 671-680

KOZ 83 : T. Kozawa & all

Automatic placement algorithms for high packing density VLSI

DAC 1983, pp 175-181

KOZ 84 : T. Kozawa, C. Mivra, H. Terai

Combin and top-down block placement algorithms for hierarchical  
logic VLSI layout

DAC 1984, pp 667-669

KRI 83 : B. Krishnamurthy

An improved min-cut algorithm for partitioning VLSI networks  
rapport de General Electric Research & Development Center  
aout 1983

LEB 82 : L. Lebart, A. Morineau, J.P. Fénelon

Traitement des données statistiques  
Edition Dunod, 1982

LEB 83 : A. Leblond

CAF: a computer assisted floorplan tool  
DAC 1983, pp 747-753

LEE 61: C.Y. Lee

An algorithm for path connections and its application  
IEEE transactions on electronic computers, vol EC10, n°3, pp 346-365  
septembre 61.

MAL 82 : R. Malladi, G. Serrero

Placement et interconnexion pour l'implantation automatique  
des circuits  
L'onde électronique, Vol 62, n° 1, 1982

MON 85 : L. Monier, J. Vuillemin

Using silicon assemblers  
VLSI 85, pp 305-318

MOU 84 : M. Moullion

Automatic customization for single layer gate arrays  
VLSI Design, novembre 1984, pp 86-94

OTT 80 : R.H.J.M. Otten

Complexity and diversity in I.C. layout design

ICCC 1980, pp 764-767

OTT 82 : R.H.J.M. Otten

Automatic floorplan design

DAC 1982, pp 261-267

PAI 84 : J.F. Paillotin

Implantation automatique de logiques en bandes

Thèse de 3° cycle, INP Grenoble, dec 1984

PER 85 : P. Perdaems

FLOORLAM: un générateur automatique de planchers de circuits  
intégrés

Thèse de 3° cycle, Université de Montpellier, 1985

PIE 85 : J.O Piednoir, M. Hild

Efficient placement algorithms for VLSI

VLSI Design, avril 1985, pp 46-50

SAU 84 : G.Saucier, A. Bellon, J.F Paillotin, J. Tsitsimis,

A. Janati, P. Chaisemartin

Classification and comparison of different placement strategies

ICCD 1984, pp 745-750

SER 82 : G. Serrero

Implantation symbolique automatisée de circuits intégrés

Thèse de docteur-ingenieur, INP Grenoble, mars 1982

SIA 85 : P. Siarry, L. Bergouzi, G. Dreyfus

Optimisation du placement de blocs par la méthode  
thermodynamique: application à la conception du plan de masse  
d'un circuit  
Colloque national conception de circuits à la demande, 1985  
pp 352-356

THU 83 : G. Thuau

Conception logique et topologique en technologie MOS  
Thèse de 3° cycle, INP Grenoble, decembre 83.

TRI 86 : M. Tricot

A propos de quelques applications de la classification  
automatique  
Thèse de 3° cycle, Université de Grenoble, juin 86.

TSI 84 : J. Tsitsimis

Implantation automatisée de circuits précaractérisés et  
prédifusés  
Thèse de 3° cycle, INP Grenoble, oct 1984

UED 83 : K. Ueda, T. Kamatsubara, T. Hosaka

A parallel processing approach for logic module placement  
IEEE Transaction on CAD, Vol CAD2, n° 1  
pp 39-47, jan 1983

UED 85 : K. Ueda, H. Kitazawa, I. Harada

CHAMP, chipflooplan for hierarchical VLSI layout design

IEEE Transaction on CAD, Vol CAD4, n° 1

pp 12-22, jan 1985

WIP 82 : G.S. Wipfler, M. Wiesel, D.A. Mlynski

A combined force and cut algorithm for hierarchical VLSI layout

DAC 1982, pp 671-677

WON 86: D.F.Wong, C.L.Liu

A new algorithm for floorplan design

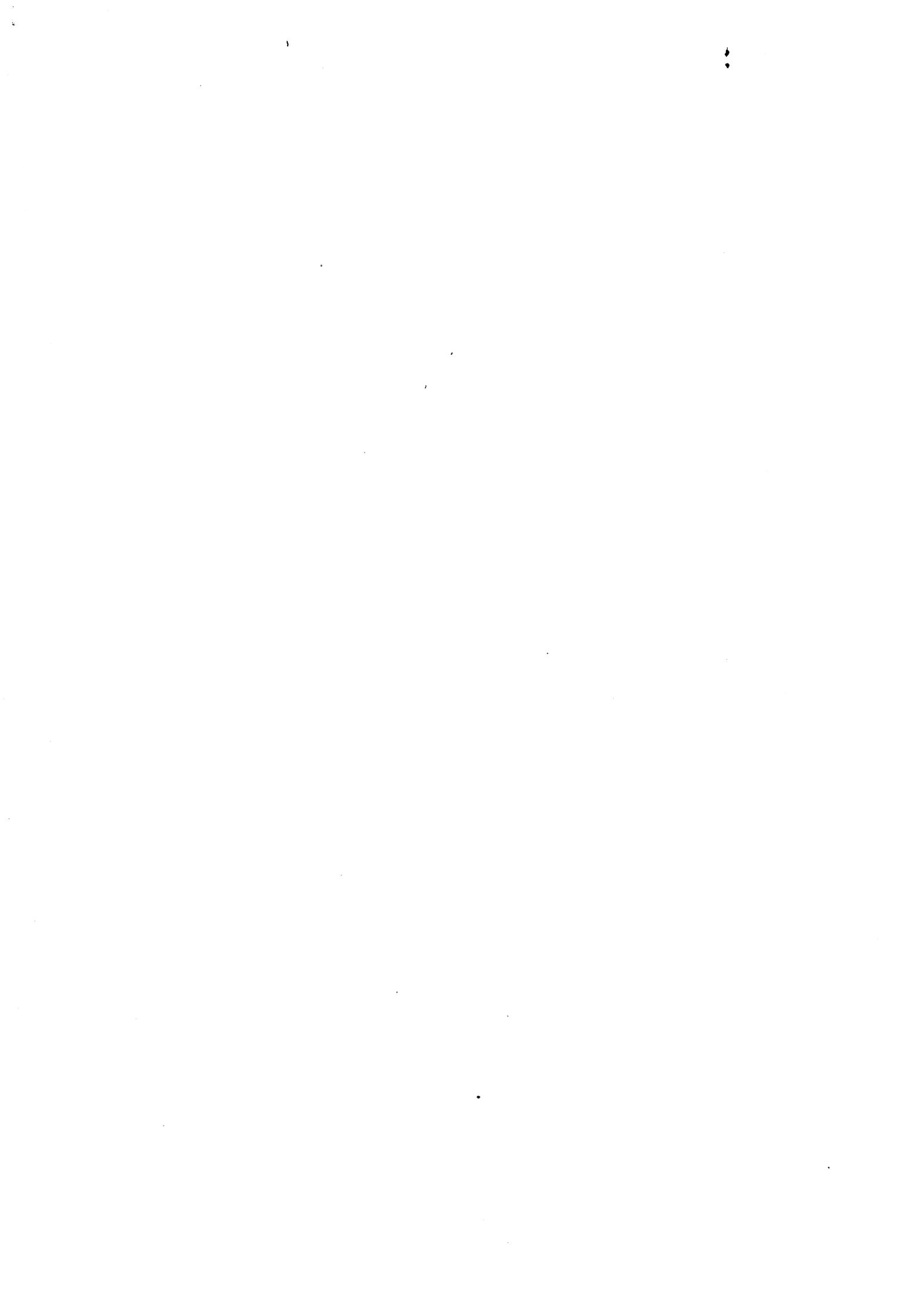
DAC 1986.

ZEE 85 : P.F. Zeegers

Approches hiérarchiques pour le placement routage des circuits

VLSI

Thèse de 3° cycle, Université de Paris VI, 1985



**Bibliographie :**

P. Antognatti, A. De Gloria, F. Ellena, L. Repetto

An expert chip planning tool

EDA 1984, pp 174-178

M. Becker, E.F. Kouka, M. Tricot

Application of data analysis methods and of simulated annealing  
for the automatic layout of circuits

RR LCS/IMAG n°571, janvier 1896

M.A. Brener, M.S. Chondrasekhar

Optimum placement of two rectangular blocks

DAC 82, pp 879-893

W.R. Heller, K. Maling, S.H. Mueller

On finding most optimal rectangular package plans

DAC 1982, pp 663-670

K.H. Khokhani, A. Digiacomo, D.G. Weinel

Automatic placement in non-uniform rectangular components

VLSI Design, may 1985, pp 50-70

S.M. Leinward, Y.T Lai

An algorithm for building rectangular floor-plans

DAC 1984, pp 663-666

L.A. Markov, J.R. Fox, J.M. Blank

Technical optimization for two dimensional placement

DAC 1984, pp 652-650

**B.T. Preas**

**Placement and routing algorithms for hierarchical integrated  
circuit layout**

**Thèse Stanford University, aout 1979**

**K.J Supowitz**

**Placement algorithms for custom VLSI**

**DAC 1983, pp 164-170**

AUTORISATION DE SOUTENANCE

DOCTORAT 3ème CYCLE, DOCTORAT-INGENIEUR, DOCTORAT USTMG

Vu les dispositions de l'Arrêté du 16 avril 1974,

Vu les dispositions de l'Arrêté du 5 juillet 1984,

Vu les rapports de M. J. R. BARRA.....

M. G. DREYFU.S.....

M. CHAISEDARIN PHILIPPE..... est autorisé

à présenter une thèse en vue de l'obtention du titre de.....

Docteur USTMG.....

Grenoble, le 10 OCT. 1988.....

Le Président de l'Université Scientifique  
Technologique et Médicale



M. TANCHE

