



HAL
open science

Implantation symbolique automatisée de circuits intégrés

Gilles Serrero

► **To cite this version:**

Gilles Serrero. Implantation symbolique automatisée de circuits intégrés. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1982. Français. NNT: . tel-00300461

HAL Id: tel-00300461

<https://theses.hal.science/tel-00300461>

Submitted on 18 Jul 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'Institut National Polytechnique de Grenoble

pour obtenir le grade de

DOCTEUR - INGENIEUR

Génie Informatique

par

SERRERO Gilles



IMPLANTATION SYMBOLIQUE AUTOMATISEE

DE CIRCUITS INTEGRES.



Mme G. SAUCIER

Président

G. MAZARÉ

G. NOGUEZ

J.C. RAULT

D. SIEWIOREK

A. VERDILLON

Examineurs

Président : Daniel BLOCH

Vice-Présidents : René CARRÉ
Hervé CHERADAME
Marcel IVANES

PROFESSEURS DES UNIVERSITES

ANCEAU François	E.N.S.I.H.A.G
BARRAUD Alain	E.N.S.I.E.G
BESSON Jean	E.N.S.E.E.G
BLIMAN Samuel	E.N.S.E.R.G
BLOCH Daniel	E.N.S.I.E.G
BOIS Philippe	E.N.S.H.G
BONNETAIN Lucien	E.N.S.E.E.G
BONNIER Etienne	E.N.S.E.E.G
BOUVARD Maurice	E.N.S.H.G
BRISSONNEAU Pierre	E.N.S.I.E.G
BUYLE-BODIN Maurice	E.N.S.E.R.G
CAVAIGNAC Jean-François	E.N.S.I.E.G
CHARTIER Germain	E.N.S.I.E.G
CHENEVIER Pierre	E.N.S.E.R.G
CHERADAME Hervé	M.C.P.P
CHERUY Arlette	E.N.S.I.E.G
CHIAVERINA Jean	M.C.P.P
COHEN Joseph	E.N.S.E.R.G
COUMES André	E.N.S.E.R.G
DURAND Francis	E.N.S.E.E.G
DURAND Jean-Louis	E.N.S.I.E.G
FELICI Noël	E.N.S.I.E.G
FOULARD Claude	E.N.S.I.E.G
GENTIL Pierre	E.N.S.E.R.G
GUERIN Bernard	E.N.S.E.R.G
GUYOT Pierre	E.N.S.E.E.G
IVANES Marcel	E.N.S.I.E.G
JAUSSAUD Pierre	E.N.S.I.E.G
JOUBERT Jean-Claude	E.N.S.I.E.G
JOURDAIN Geneviève	E.N.S.I.E.G
LACOUME Jean-Louis	E.N.S.I.E.G
LATOMBE Jean-Claude	E.N.S.I.H.A.G
LEROY Philippe	E.N.S.H.G
LESIEUR Marcel	E.N.S.H.G
LESPINARD Georges	E.N.S.H.G
LONGUEUE Jean-Pierre	E.N.S.I.E.G
MAZARE Guy	E.N.S.I.H.A.G
MOREAU René	E.N.S.H.G
MORET Roger	E.N.S.I.E.G
MOSSIERE Jacques	E.N.S.I.H.A.G
PARIAUD Jean-Charles	E.N.S.E.E.G
PAUTHENET René	E.N.S.I.E.G
PERRET René	E.N.S.I.E.G
PERRET Robert	E.N.S.I.E.G

CHERCHEURS DU MINISTERE DE L'INDUSTRIE

(Directeurs et Maîtres de recherche - E.N.S Mines Saint Etienne)

LESBATS Pierre	Directeur de recherche
BISCONDI Michel	Maître de recherche
KOBYLANSKI André	Maître de recherche
LE COZE Jean	Maître de recherche
THEVENOT François	Maître de recherche
TRAN MINH Canh	Maître de recherche
LALAUZE René	Maître de recherche
LANCELOT Francis	Maître de recherche

PERSONNALITES HABILITEES A DIRIGER DES TRAVAUX DE RECHERCHE

(Décision du Conseil Scientifique)

E.N.S.E.E.G

BERNARD Claude
BONNET Roland
CAILLET Marcel
CHATILLON Catherine
COULON Michel
EUSTATHOPOULOS Nicolas
HAMMOU Abdelkader
JOUDE Jean-Charles
MALMEJAC Yves (CENG)
RAVAINE Denis
SAINFORT (CENG)
SARRAZIN Pierre
TOUZAIN Philippe
URBAIN Georges (Laboratoire des Ultraréfractaires, ODEILLO)

E.N.S.M Saint Etienne

GUILHOT Bernard
THOMAS Gérard
DRIVER Julian

E.N.S.E.R.G

BOREL Joseph
CHEHIKIAN Alain

E.N.S.I.E.G

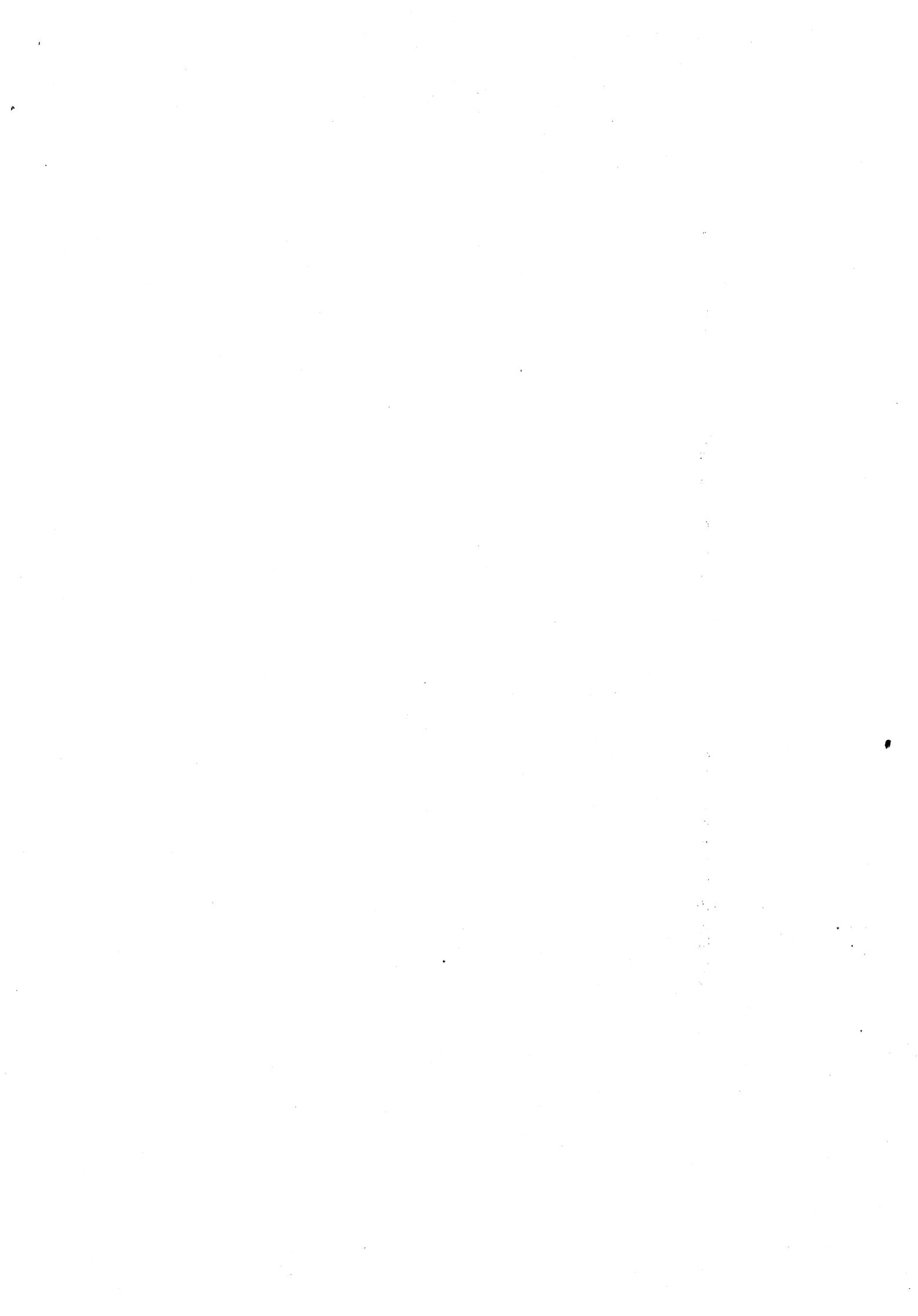
BORNARD Guy
DESCHIZEAUX Pierre
GLANGEAUD François
LEJEUNE Gérard
PERARD Jacques

E.N.S.H.G

DELHAYE Jean-Marc



à mes parents



Je tiens à exprimer toute ma reconnaissance à Madame G. SAUCTER, Professeur à l'ENSIMAG, d'avoir bien voulu m'accueillir dans son équipe de recherche, d'avoir su encadrer ce travail et de m'avoir fait l'honneur d'accepter de présider le jury de cette thèse.

Je tiens à remercier :

- Monsieur G. MAZARÉ, Professeur à l'ENSIMAG, d'avoir accepté d'être rapporteur de cette thèse et d'avoir bien voulu me faire profiter de ses remarques constructives,

- Monsieur A. VERDILLON, Maître Assistant à l'USMG, de s'être intéressé à ce travail, et d'avoir contribué à des améliorations,

- Monsieur G. NOGUEZ, Professeur à l'Institut de Programmation de Paris,

Monsieur J.C. RAULT, chargé de mission à l'Agence de l'Informatique, et

Monsieur D. STEWIOREK, Professeur à Carnegie-Mellon University,

d'avoir accepté de faire partie du jury de cette thèse.

Je tiens également à remercier :

- tous mes collègues et amis de l'équipe "Conception et Sécurité des Systèmes" qui, par leur soutien, ont contribué à ce travail. Qu'il me soit permis de remercier tout particulièrement Monsieur R. MALLADT dont j'ai pu apprécier toutes les qualités au cours de notre collaboration.

- toutes les personnes qui ont assuré la réalisation technique de cet ouvrage : Madame G. DUFFOURD pour la frappe ainsi que Monsieur D. IGLESIAS et l'équipe de reprographie de l'IMAG pour le tirage.

TABLE DES MATIÈRES

INTRODUCTION

CHAPITRE I MÉTHODES ET ALGORITHMES POUR L'IMPLANTATION AUTOMATIQUE
DES CIRCUITS INTÉGRÉS

I - IMPLANTATION D'UN CIRCUIT INTÉGRÉ

I - 1. Définitions

I - 2. Les étapes de l'implantation

II - PLACEMENT

II - 1. Complexité du problème

II - 2. Coût d'un placement et relation d'ordre

II - 3. Composantes du vecteur coût

II - 3.1. Longueur des connexions

II - 3.2. Densité de connexions

II - 3.3. Surface et forme du circuit

II - 3.4. Nombre de croisements de connexions

II - 3.5. Antagonismes entre critères

II - 4. Méthodes de placement

II - 4.1. Recherche d'une solution optimale

II - 4.2. Recherche d'une solution approchée

II - 5. Conclusion

III - INTERCONNEXION

III - 1. Introduction

III - 2. Préparation du tracé

III - 2.1. Liste des connexions

III - 2.2. Affectation aux couches

III - 2.3. Ordre du tracé

III - 3. Méthodes de tracé des connexions

III - 3.1. Algorithme de Lee

III - 3.2. Algorithme de recherche par ligne

III - 3.3. Algorithme d'expansion par ligne

III - 3.4. Méthode par tracé de canaux

III - 3.5. Méthode par zone saturée

III - 4. Conclusion

IV - CONCLUSION

CHAPITRE II LA REPRÉSENTATION SYMBOLIQUE : PRÉSENTATION ET CLASSIFICATION

LE MD-MOS : ÉTUDE DÉTAILLÉE

I - LA REPRÉSENTATION SYMBOLIQUE

I - 1. Introduction

I - 2. Présentation des symbolismes

I - 2.1. Symbolisme par site, sur grille

I - 2.2. Symbolisme par site et connexion, sur grille

I - 2.3. Symbolisme par site et connexion, sur grille virtuelle

I - 2.4. Symbolisme par site et connexion, sans grille

I - 2.5. Symbolisme par porte logique, sur grille

I - 2.6. Symbolisme par fonctions logiques complexes

I - 3. Conclusion

II - LA PORTE MD-MOS ET LES RÈGLES D'IMPLANTATION SYMBOLIQUE

II - 1. L'inverseur en technologie MOS

II - 2. Principe de la porte MD-MOS

II - 3. Réalisation technologique

II - 4. Modèle symbolique utilisé pour l'implantation

II - 5. Dimensionnement du transistor de charge

II - 6. Cellule de portes MD-MOS.

CHAPITRE III A - OASIS. OUTIL AUTOMATISÉ DE SYSTÈMES EN IMPLANTATION SYMBOLIQUE

B - ÉTUDE DE CAS : MÉTHODE DE PLACEMENT ET D'INTERCONNECTION AUTOMATISÉS DE CIRCUITS MD-MOS DÉCRITS SYMBOLIQUEMENT

I - MÉTHODE GÉNÉRALE

- I - 1. Objectifs et méthode générale
- I - 2. Respect des contraintes
- I - 3. Optimisation
- I - 4. Organisation générale

II - ÉTUDE DES CELLULES À UNE LIGNE D'ALIMENTATION

- II - 1. Etude préliminaire
 - II - 1.1. La fonction de Pic; définition
 - II - 1.2. Intérêt du pic
 - II - 1.3. Algorithme de répartition des segments
- II - 2. Placement
 - II - 2.1. Placement initial constructif : coût, algorithme, complexité
 - II - 2.2. Amélioration itérative : coût, algorithme, complexité
- II - 3. Tracé des connexions
 - II - 3.1. Partitionnement des connexions en deux classes
 - II - 3.2. Méthode de tracé
 - II - 3.3. Connexions traitées dans la zone 1
 - II - 3.4. Algorithme TC de tracé des connexions
- II - 4. Exemples - remarques

III - ÉTUDE DES CELLULES À DEUX LIGNES D'ALIMENTATION

- III - 1. Placement
 - III - 1.1. Placement initial
 - a) Affectation de certaines portes à C_H ou C_B
 - b) Partitionnement vertical
 - c) Partitionnement horizontal
 - d) Positionnement des portes
 - e) Méthode générale - complexité
- III - 2. Tracé des connexions
 - III - 2.1. Connexions internes
 - III - 2.2. Connexions transverses entre C_H et C_B
- III - 3. Exemples - remarques

IV - CONCLUSION

CONCLUSION

BIBLIOGRAPHIE

ANNEXE - DESCRIPTION DES CIRCUITS



INTRODUCTION

La densité d'intégration des circuits VLSI (Very Large Scale Integration) rend de plus en plus complexe la conception de tels circuits. Une démarche de conception bien structurée et sûre, c'est-à-dire conduisant sans erreur au circuit à réaliser est donc nécessaire.

Dans cette optique, une démarche de conception descendante est en général adoptée : la synthèse du circuit mène du cahier des charges à la fabrication du circuit. Le processus de conception est décomposé en étapes successives pour lesquelles des modèles du circuit à concevoir sont définis. Leur analyse permet de valider ou modifier un choix (figure 0).

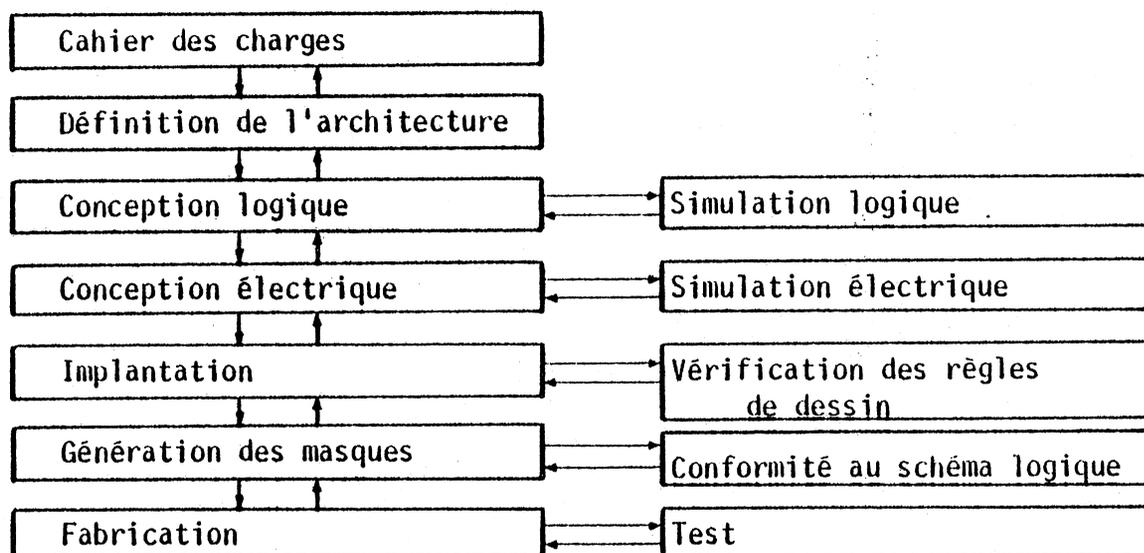


Figure 0. Etapes de la conception des circuits.

De nombreux outils existent déjà, qui assistent le concepteur lors des différentes phases de la conception [Br 81].

La complexité actuelle des circuits rend de plus en plus problématique une de ces étapes : l'implantation du circuit. Il s'agit de placer et de connecter entre eux des milliers d'éléments (transistors par exemple) en réalisant un compromis entre trois objectifs :

- . l'optimisation de surface,
- . la fiabilité, c'est-à-dire la non introduction d'erreur de conception,
- . le temps passé à cette implantation.

Ces deux derniers points rendent obligatoire dans les années à venir l'utilisation d'outils CAO (Conception Assistée par Ordinateur) puissants réalisant automatiquement ce passage. Malheureusement, la difficulté du problème est telle que de nombreux essais dans le passé se sont soldés par des échecs relatifs, à savoir :

- . une optimisation en surface insuffisante,
- . des approches algorithmiques traitant "bien" 95 à 98% des éléments mais rendant impossible une solution finale,
- . des temps de calcul prohibitifs.

Devant une situation aussi difficile, deux tendances nous paraissent prometteuses de succès :

- (i) l'amélioration des algorithmes déjà utilisés,
- (ii) la recherche de solutions plus "structurées".

Nous entendons par solutions plus structurées, l'implantation sur structures hôtes à fortes contraintes (structures répétitives, structures à canaux d'interconnexions figés), le cas "extrême" étant les réseaux prédiffusés. De telles restrictions limitant le choix des solutions d'implantation : restriction sur les emplacements possibles et sur les voies de communication. Elles vont décroître très fortement la complexité du problème de l'implantation automatisée.

Nous avons choisi dans notre approche une solution qui peut être considérée comme intermédiaire entre les solutions très figées et peu optimisées et les solutions entièrement libres (implantation au micron sur surface libre) : l'utilisation d'une implantation symbolique. L'implantation symbolique choisie (celle du MD-MOS du CNET) nous a paru originale et susceptible de conduire non seulement à des résultats probants dans le domaine de l'implantation automatisée, mais également dans la démarche de conception sûre. En effet, cette représentation symbolique présente de nombreux avantages :

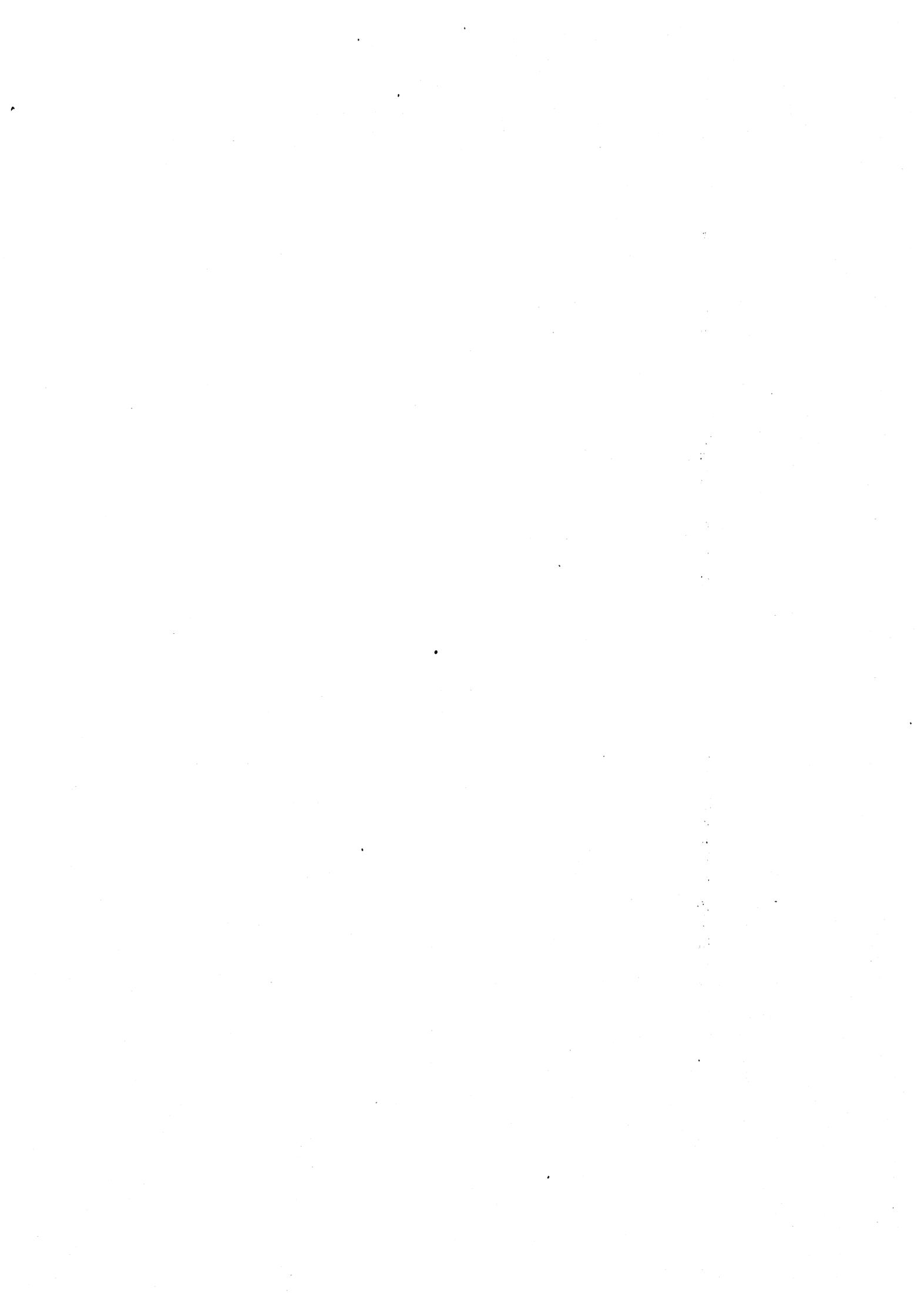
- . elle permet une conception logique qui se traduit immédiatement en portes symboliques,
- . elle permet une conception électrique automatique (dimensionnement des transistors),
- . enfin, face au problème de l'implantation, on peut constater qu'elle permet un gain en temps d'un facteur de quelques dizaines (quelques dizaines de transistors dessinés au lieu de un au micron). Elle permet en outre d'aborder le problème de l'implantation automatisée avec un taux de réussite bien supérieur à une implantation libre au micron. La perte de surface sera elle aussi sans commune mesure avec les structures hôtes à contraintes plus sévères.

Quant à l'amélioration de l'algorithme, elle repose essentiellement sur un abandon partiel de méthodes de théorie des graphes cherchant des solutions quasi-optimales pour s'orienter vers des méthodes très heuristiques proches des méthodes d'intelligence artificielle. Il s'agit de procéder par "pas" en plaçant progressivement les éléments, en optimisant par rapport à des vecteurs coûts en effectuant des améliorations itératives, et en évaluant soigneusement la complexité des algorithmes utilisés. Ce type d'approche est très proche de la démarche d'un concepteur et se prête très bien à l'interaction concepteur/CAO. C'est une telle approche qui sera utilisée tout au cours de ce travail.

Nous trouverons dans le premier chapitre une étude des méthodes et algorithmes connus d'implantation automatique. Cette étude concerne essentiellement les problèmes du placement et de l'interconnexion.

Le second chapitre présente la notion de représentation symbolique des circuits intégrés. Les différents symbolismes existants sont classés et le MD-MOS, sur lequel porte le troisième chapitre, est décrit en détail.

Enfin, dans le troisième chapitre; nous proposons des méthodes et algorithmes de placement et d'interconnexion automatisés de cellules MD-MOS décrites sous forme symbolique. Ce travail concerne essentiellement les cellules à une et deux lignes d'alimentation pour lesquelles des méthodes heuristiques et originales sont définies.



CHAPITRE I

MÉTHODES ET ALGORITHMES POUR L'IMPLANTATION
AUTOMATIQUE DES CIRCUITS INTÉGRÉS



I - IMPLANTATION D'UN CIRCUIT INTÉGRÉ

I -1. DEFINITIONS

I -1.1 Un circuit intégré non implanté est défini par un réseau de modules :

- Les modules sont définis par leur forme extérieure ; il s'agit en général de rectangles, éventuellement de la juxtaposition de rectangles. Ces modules ont des points d'entrée et de sortie ayant ou non une position figée dans le module.

- Les connexions relient d'une part ces modules entre eux, d'autre part certains modules à des points particuliers : point d'entrée-sortie situé à la périphérie du circuit, point d'alimentation, point de masse. Ces connexions relient deux ou plusieurs points dans le circuit et représentent les équipotentielles électriques.

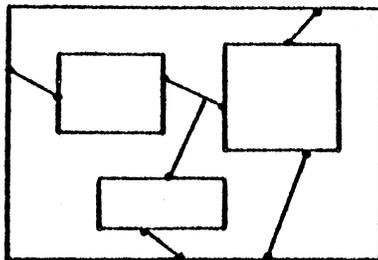


Figure 1 . Réseau de modules

I -1.2 Grphe de connectivité d'un circuit [Kun 72]

Dans certaines études préalables, il peut être intéressant de travailler sur un graphe simplifié traduisant les relations de connectivité de ce réseau. On peut

- (i) construire un graphe dans lequel les modules sont représentés par les noeuds. Ces noeuds sont reliés par une arête lorsque les modules sont connectés. (figure 2-b)

(ii) construire un graphe biparti dans lequel deux types de noeuds représentent les modules et les connexions ; une arête relie un module et les connexions auxquelles il est relié. (figure 2-c)

Le graphe biparti permet de conserver la structure d'équipotentielle entre les noeud -modules.

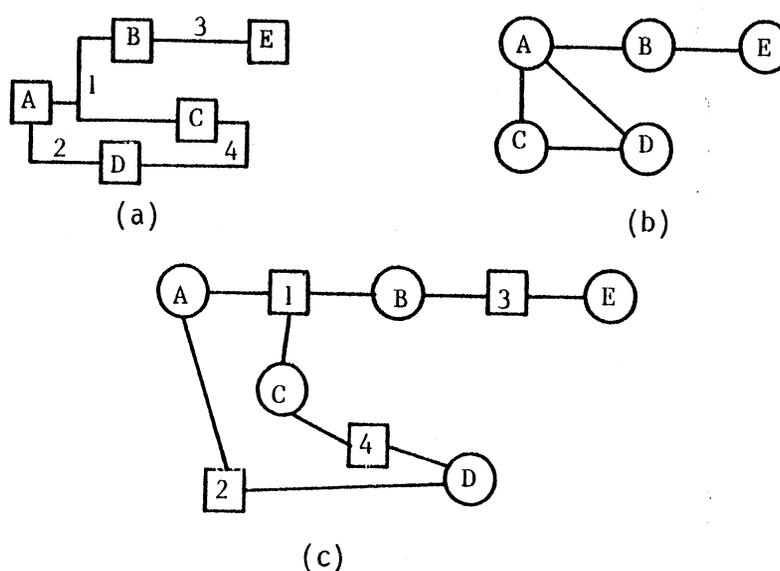


Figure 2 . Deux graphes correspondant au réseau (a).

On peut alors appliquer des méthodes classiques de la théorie des graphes (coupure, plus court chemin ...) pour étudier les propriétés de connectivité du réseau.

1 -1.3 L'implantation d'un circuit intégré consiste à immerger le réseau précédent dans une structure hôte qui sera le "cadre" du circuit définitif. Ces structures hotes peuvent être classées suivant la nature des contraintes associées.

structure à contrainte faible

(i) *contrainte de direction*

Deux directions (verticale et horizontale) sont définies dans la structure hôte (qui est alors rectangle).

Les bords des modules ainsi que les connexions sont parallèles à ces deux directions.

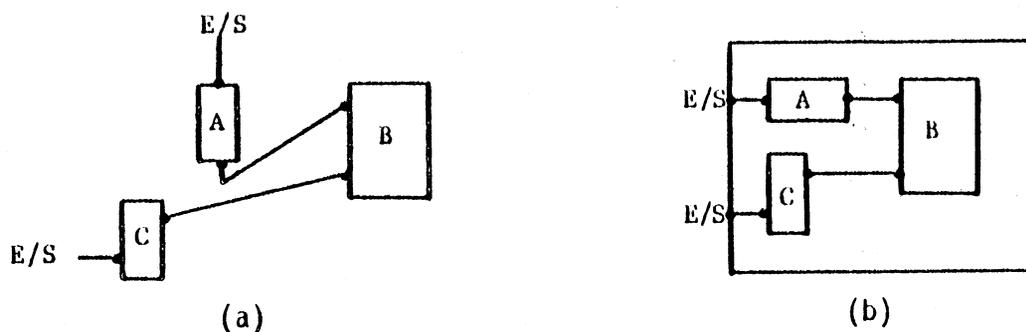


Figure 3 . Contrainte de direction.

(ii) *contrainte de direction et de surface*

La structure hôte est un rectangle de surface fixée ; la surface globale n'est alors par un critère à optimiser car elle est figée.

(iii) *contrainte d'emplacement des points d'entrée/sortie et/ou des points d'alimentation et de masse*

En plus des contraintes précédentes, les contraintes les plus fréquentes sont : emplacement des points d'entrée/sortie, emplacement des points d'alimentation et de masse :

Structure à contrainte forte

(i) *contrainte sur l'emplacement des connexions*

Des zones réservées aux interconnexions sont définies entre les modules : canaux d'interconnexion. Les connexions sont tracées dans ces canaux sur une ou deux couches. Ces canaux peuvent être définis dans des zones précises du circuit et avoir des directions privilégiées. (figure 6a, b, c)

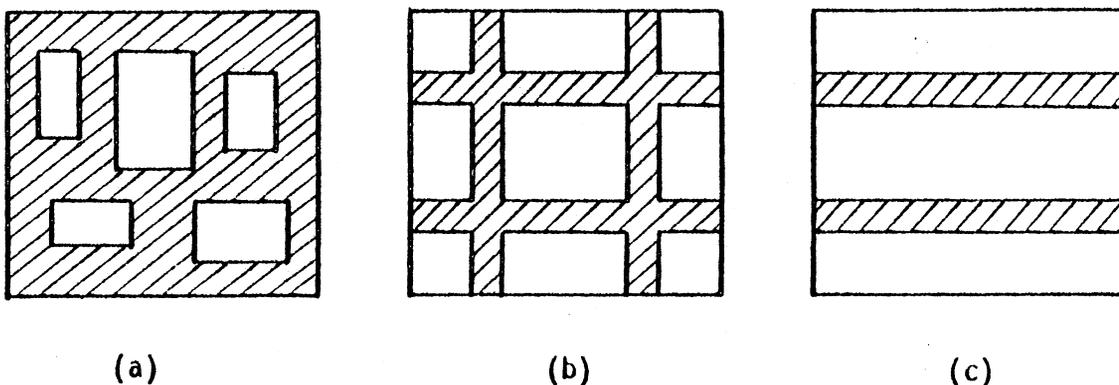


Figure 6 . Canaux d'interconnexion (zones hachurées)

(a) non fixés à priori

(b) fixés dans deux directions

(c) fixés dans une seule direction.

(ii) *contrainte sur l'emplacement des modules*

Des emplacements sont définis dans la structure hôte pour recevoir les modules. Ce type de contrainte se rencontre généralement pour les réseaux prédiffusés de cellules ou de portes.

Par exemple, sur la figure 7, chaque zone non hachurée peut recevoir 4 modules en A, B, C, D ; les formes des modules sont alors compatibles avec les surfaces réservées. Remarquons que cette contrainte sur l'emplacement des modules implique une contrainte sur l'emplacement des connexions.

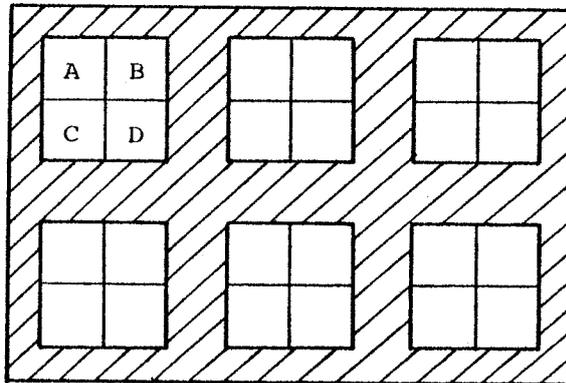


Figure 7 . Contrainte d'emplacement des modules

I -2. LES ETAPES DE L'IMPLANTATION

La phase d'implantation se décompose en

- (i) partitionnement des circuits en sous-circuits,
- (ii) placement des modules dans chaque sous-circuit et placement des sous-circuits,
- (iii) tracé des connexions entre les modules et les sous-circuits.

(i) L'objectif principal du partitionnement est de réduire la complexité des problèmes de placement et d'interconnexion. On cherche à minimiser le nombre d'interconnexion ou les temps de propagation des signaux entre les blocs d'une partition. On effectue le plus couramment un découpage fonctionnel qui présente l'avantage d'être simple et de permettre une lecture facile du circuit implanté.

(ii) L'étape de placement consiste à affecter les modules

- soit à des emplacements non définis a priori et situés sur des parties quelconques de la puce (circuits à la demande),

- soit à des emplacements prédéfinis sur la puce (réseau prédéfinis).

(iii) L'étape de tracé consiste à définir et à construire les emplacements exacts de toutes les connexions entre les modules.

Bien que très dépendantes, les étapes de placement des éléments et de tracé des connexions, à cause de leur complexité, ne peuvent pas être abordées simultanément ; elles sont donc résolues séquentiellement : placement puis tracé, sans pour autant interdire une remise en cause du placement durant l'étape de tracé.

II - PLACEMENT

II -1. COMPLEXITE DU PROBLEME

Le nombre de placements possibles dépend du nombre n de modules à placer et des contraintes de la structure hôte. il est dans la plupart des cas fort important.

On ne cherchera donc pas à générer tous les placements possibles, mais on recherchera un placement optimisant un ou plusieurs critères (cf II -2. *notion de coût*).

La recherche d'un placement, même simplifiée (i.e. placement minimisant un seul critère), reste très complexe. Ainsi la recherche d'un placement d'un ensemble d'éléments (modules) dont la longueur totale des connexions (distance Euclidienne) est inférieure à une valeur fixée L est un problème NP-complet. La longueur des connexions, pour un placement donné, est facilement calculée en un temps polynomial par rapport au nombre de connexions, ce qui permet la comparaison à une valeur fixée L. Ce problème peut donc plus ou moins rapidement être résolu par essais successifs, mais aucune démarche algorithmique efficace ne permet actuellement d'en déterminer les solutions .

Le problème initial de placement peut être fractionné, mais toute transformation conduit toujours à au moins un problème de type NP. La division du problème d'implantation conduit elle aussi à deux problèmes NP-complets : placement et interconnexion [Do'80] .

II -2. COUT D'UN PLACEMENT ET RELATION D'ORDRE

Pour permettre d'affirmer qu'un "placement est meilleur" qu'un autre, il faut pouvoir évaluer les "coûts" des placements et les comparer. Pour cela, une relation d'ordre est définie. Cette définition sera étendue à un "placement partiel" pour permettre l'emploi de méthodes progressives de placement.

De nombreux critères calculés de manière prédictive (le tracé des connexions n'étant pas connu durant l'étape de placement) permettent de caractériser et d'évaluer un placement, parmi lesquels :

- la longueur des connexions,
- la longueur de la connexion la plus longue,
- taux de connexions respectant (ou non) des directions privilégiées,
- nombre de croisements de connexions,
- surface,
- facteurs de dissipation,
- régularité du placement,
- capacité parasite.

La diversité des critères rend difficile l'élaboration d'un vecteur coût pour lequel tous les critères seraient minimisés. Ce vecteur serait de plus lourd à manipuler dans le problème de placement. Nous nous limitons donc, en général, à ne minimiser qu'un nombre réduit de critères simples, jugés importants et conduisant à des solutions acceptables pour les circuits réels.

Définition : coût élémentaire c_j

Soit \mathcal{P} l'ensemble des placements possibles des modules du réseau \mathcal{R} . A chaque placement P de \mathcal{P} , on associe une valeur c_j de \mathbb{R} appelée coût élémentaire qui mesure la grandeur que le critère i associé à c_j optimise.

c_j permet de caractériser :

(i) soit un critère simple, c'est à dire un critère pour lequel on n'optimise qu'une grandeur. Par exemple minimiser la surface, ou la longueur des connexions sont des critères simples.

(ii) soit un critère composé, c'est à dire un critère pour lequel on optimise une fonction de plusieurs critères simples. Les critères composés permettent d'exprimer un compromis entre critères simples. Si cette fonction est linéaire, le critère composé s'écrit $c_j = \sum_i \alpha_i c'_i$ où c'_i est un critère simple et α_i appartenant à \mathbb{R} le poids associé à c'_i .

Dans la suite de l'étude, les c_i , composantes du vecteur coût à définir, seront minimisées pour obtenir une "meilleure" solution. Cette condition n'est pas restrictive, car il est toujours possible de se ramener à ce cas en considérant l'opposé du critère que l'on veut maximiser.

Par exemple, on cherche à maximiser le nombre N_c de connexions entre deux sous-ensembles de modules. Le coût élémentaire considéré sera $-N_c$, que l'on minimisera.

Définition : vecteur coût

Le vecteur $C = \begin{pmatrix} c_1 \\ \vdots \\ c_k \end{pmatrix}$ défini sur \mathbb{R}^k tel que c_i caractérise le critère i et

$(\forall (i,j) \in [1,k]^2, i < j)$ alors (le critère i est plus important que le critère j)

est appelé vecteur coût.

Il permet de juger de la qualité d'un placement. Cette priorité entre critères est établie par le concepteur en fonction de l'application.

Propriété : Relation d'ordre total entre vecteurs coûts.

Soient C et C' deux vecteurs coûts de \mathbb{R}^k

$$C < C' \iff \left\{ \begin{array}{l} C \neq C' \\ \text{et} \\ \exists j \in [1, k] \text{ tq } \begin{cases} c_i = c'_i \text{ pour tout } i < j \\ c_j <_{\mathbb{R}} c'_j \end{cases} \end{array} \right.$$

Démonstration

Soient C et C' deux vecteurs coûts différents.

Donc $\exists i \in [1, k]$ tel que $c_i \neq c'_i$

Soit i_0 le plus petit i remplissant cette condition.

La relation d'ordre entre C et C' se ramène en fait à la relation d'ordre entre c_{i_0} et c'_{i_0} . L'ordre total $<_{\mathbb{R}}$ nous permet donc de conclure que $<$ est une relation d'ordre total pour les vecteurs coûts.

Définition

Soient P et P' deux placements pour lesquels les coûts sont respectivement C et C' .

$$P \text{ est } \underline{\text{meilleur}} \text{ de } P' \stackrel{\text{déf.}}{\iff} C < C'.$$

Définition : Coût partiel

On appellera coût partiel par rapport à un sous-réseau \mathcal{R}' de \mathcal{R} , le vecteur coût $C_p = \left\| \begin{array}{c} c_{p1} \\ \vdots \\ c_{p1} \end{array} \right\|$ où le coût c_{pi} est évalué sur \mathcal{R}' .

Un sous-réseau est défini par un sous-ensemble de modules et les connexions les reliant.

Dans un placement progressif ou constructif, il s'agira du sous-ensemble défini par les modules déjà placés.

Définition : Coût réduit

On appellera coût réduit d'un vecteur coût $C = \begin{pmatrix} c_1 \\ \vdots \\ c_k \end{pmatrix}$, le

vecteur coût $C_r = \begin{pmatrix} c_{r1} \\ \vdots \\ c_{rk} \end{pmatrix}$ où

$(1 < k)$ et $(\forall j \in [1,1]) (c_{rj} \in \{c_i\}_{i \in [1,k]})$.

Autrement dit, le coût réduit C_r est composé d'un sous-ensemble des coûts élémentaires de C .

Remarque : Le coût réduit évalué pour un sous-réseau \mathcal{R}' de \mathcal{R} sera un coût partiel réduit.

II -3. COMPOSANTES DU VECTEUR COUT

Nous nous intéresserons dans ce chapitre à l'évaluation prédictive des coûts élémentaires associés aux critères simples suivants : longueur des connexions, densité de connexions, surface et forme du circuit, nombre de croisements de connexions.

L'évaluation de ces coûts doit être (i) simple à mettre en oeuvre, et (ii) peu coûteuse en temps de calcul.

L'optimisation des critères doit conduire à une bonne solution, c'est à dire un placement permettant le tracé de toutes les connexions.

II -3.1 Longueur des connexions

Deux distances (au sens mathématique) sont généralement utilisées pour estimer la longueur des connexions : la distance de Manhattan et la distance Euclidienne.

Soient $M_1 (x_1, y_1)$ et $M_2 (x_2, y_2)$ deux points du plan, la distance de Manhattan d_m est définie par

$$d_m (M_1, M_2) = |x_1 - x_2| + |y_1 - y_2|$$

et la distance Euclidienne d_e par

$$d_e (M_1, M_2) = \left[(x_1 - x_2)^2 + (y_1 - y_2)^2 \right]^{1/2}$$

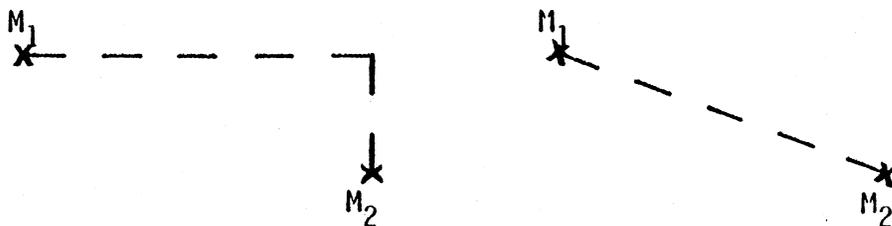


Figure 8 . (a) distance de Manhattan
(b) distance Euclidienne

Dans la pratique, pour des éléments reliés le plus souvent par des connexions horizontales et verticales, la distance de Manhattan est la plus réaliste. De plus, il est possible d'établir une pondération sur les éléments à connecter, ce qui augmente la vraisemblance de la distance pour le problème de placement, et prépare mieux la phase de tracé.

II -3.2 Densité de connexions

La densité de connexions pour une ligne de coupe l_c du circuit est égale au nombre de connexions traversant cette ligne.

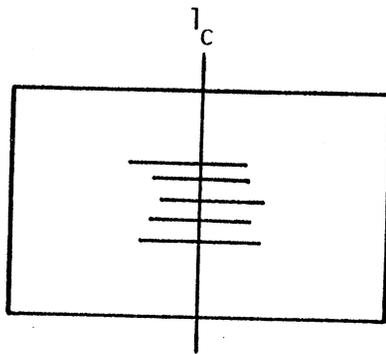


Figure 9 . La ligne de coupe

Cette notion a été introduite pour répondre à deux objectifs :

- tracé des connexions : le tracé des connexions est d'autant plus facile dans les zones où la densité de connexions est peu élevée.
- répartition uniforme des connexions : la répartition uniforme est rendue possible, ce qui permet d'agir directement sur la largeur des canaux d'interconnexion et la surface totale du circuit.

Les algorithmes basés sur la notion de densité de connexions [Br 77] tendent donc en général à faire décroître (i) la densité de connexions, (ii) la longueur totale des connexions, et (iii) la surface totale du circuit.

II -3.3 Surface et forme du circuit

Considérons les deux grandeurs liées à la surface du circuit : ST , la surface totale du rectangle contenant le circuit et SU, la surface utile du circuit, somme des surfaces des modules et des canaux d'interconnexion. Le critère de surface peut être pris en compte en minimisant ST et (ST - SU).

Soient L et l les Longueur et largeur du rectangle contenant le circuit et RT le rapport souhaité entre Longueur et largeur du circuit final. Minimiser (L/l - RT) permet d'agir sur la forme finale du circuit (RT = 1 pour obtenir une forme proche d'un carré).

II -3.4 Nombre de croisements de connexions

Le nombre de croisements est une quantité difficile à estimer correctement avant le tracé effectif des connexions. En fait, un majorant du nombre de croisements est calculé simplement et minimisé. De nombreux croisements pourront être évités lors du tracé par allongement de certaines connexions (les croisements sont ainsi contour-
nés).

Nous proposons un critère simple pour indiquer si deux segments ont ou non une intersection.

Soient $A_1 A'_1$ et $A_2 A'_2$ deux segments dont les extrémités $A_1(x_1, y_1)$, $A'_1(x'_1, y'_1)$, $A_2(x_2, y_2)$, $A'_2(x'_2, y'_2)$ sont distinctes (Figure 10).

Soient L_1 et L_2 les perpendiculaires à $A_1 A'_1$ et $A_2 A'_2$ et passant respectivement par A_1 et A_2 .

Soient P_1 et P'_1 les projections orthogonales de A_1 et A'_1 sur L_2 .

Soient P_2 et P'_2 les projections orthogonales de A_2 et A'_2 sur L_1 .

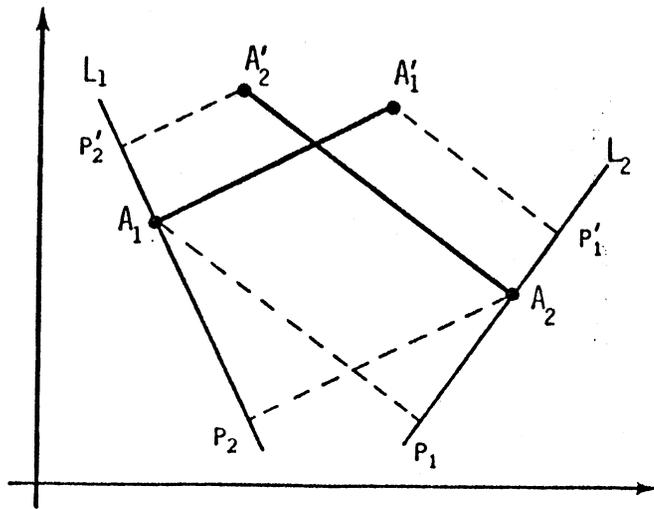


Figure 10 . Conditions d'intersection entre
 $A_1 A'_1$ et $A_2 A'_2$

Les deux segments ont un point commun si les deux conditions suivantes sont satisfaites :

Les segments $P_1 P'_1$ et $P_2 P'_2$ contiennent A_2 et A_1 respectivement.

Ces conditions s'expriment facilement à l'aide des produits scalaires des vecteurs :

$$\left\{ \begin{array}{l} \cdot \overrightarrow{A_1 A'_1} \cdot \overrightarrow{A_1 A_2} \text{ et } \overrightarrow{A_1 A'_1} \cdot \overrightarrow{A_1 A'_2} \text{ sont de signes opposés} \\ \cdot \overrightarrow{A_2 A'_2} \cdot \overrightarrow{A_2 A_1} \text{ et } \overrightarrow{A_2 A'_2} \cdot \overrightarrow{A_2 A'_1} \text{ sont de signes opposés} \end{array} \right.$$

où

$$\begin{aligned} & \overrightarrow{A_1 A_2} (x_2 - x_1, y_2 - y_1), \overrightarrow{A_1' A_2'} (x_2' - x_1', y_2' - y_1'), \\ & \overrightarrow{A_2 A_1} (x_1 - x_2, y_1 - y_2), \overrightarrow{A_2' A_1'} (x_1' - x_2', y_1' - y_2'), \\ & \overrightarrow{A_1 A_1'} (-(y_1' - y_1), x_1' - x_2) \text{ orthogonal de } \overrightarrow{A_1 A_1'} \\ & \overrightarrow{A_2 A_2'} (-(y_2' - y_2), x_2' - x_1) \text{ orthogonal de } \overrightarrow{A_2 A_2'} \end{aligned}$$

On déduit alors le système d'inéquations suivant qui exprime le cas général d'une intersection entre $A_1 A_1'$ et $A_2 A_2'$.

$$\left\{ \begin{array}{l} \left[-(y_1' - y_1)(x_2 - x_1) + (x_1' - x_1)(y_2 - y_1) \right] \\ \quad \left[-(y_1' - y_1)(x_2' - x_1') + (x_2' - x_1')(y_2' - y_1') \right] < 0 \\ \left[-(y_2' - y_2)(x_1 - x_2) + (x_2' - x_2)(y_1 - y_2) \right] \\ \quad \left[-(y_2' - y_2)(x_1' - x_2') + (x_2' - x_2')(y_1' - y_2') \right] < 0 \end{array} \right.$$

Lorsque les coordonnées des points sont entières (implantation sur grille), la méthode précédente conduit à ne manipuler que des grandeurs entières, ce qui simplifie calculs et comparaisons à 0. Les problèmes dûs à l'utilisation de nombres flottants en machine sont alors évités.

II - 3.5 Antagonismes entre critères :

Tous ces différents critères ne peuvent pas être pris en compte simultanément. Pour chaque application, on définira un vecteur coût facile à calculer qui concrétisera un compromis entre ces critères.

Le critère le plus fréquemment utilisé, parce que le plus simple et néanmoins efficace, est sans nul doute, la minimisation de la longueur des connexions, et ce parfois au détriment de la surface du circuit ou du nombre de croisements comme le montrent les figures 11 et 12.

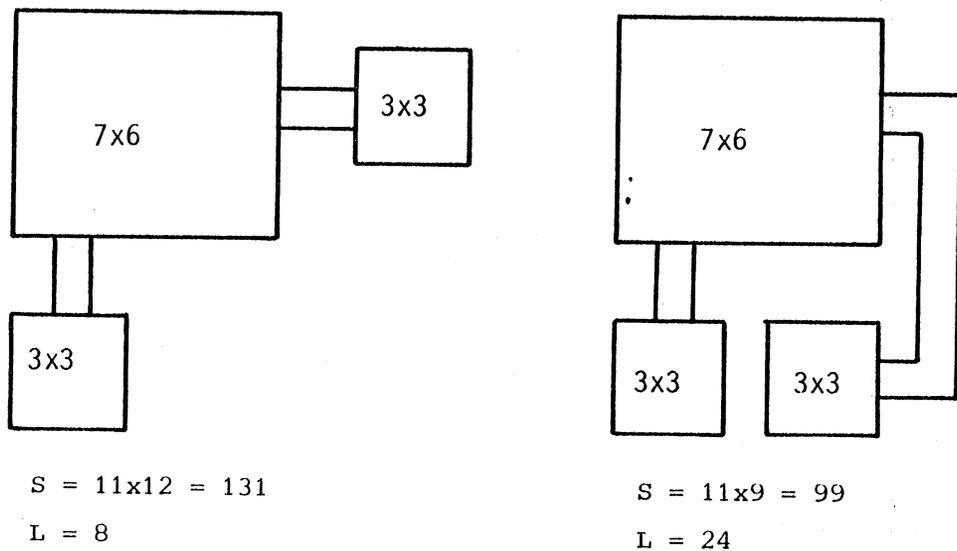


Figure 11 . surface contre longueur de connexions

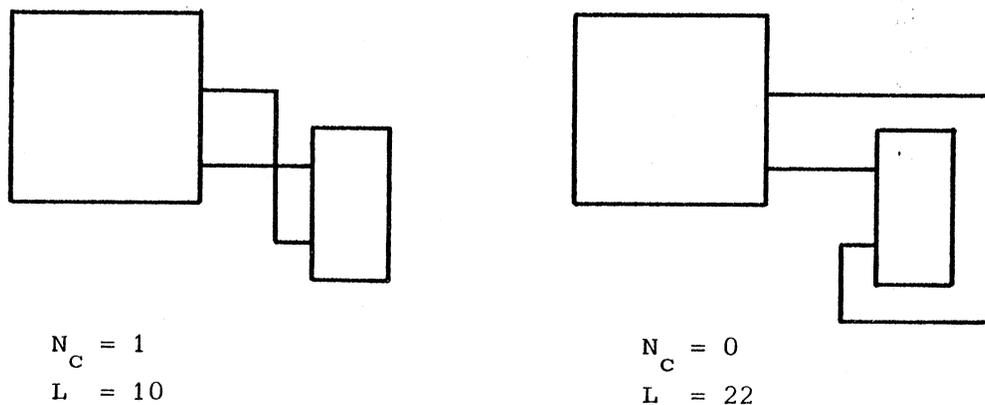


Figure 12 . nombre de croisements contre longueur de connexions

II -4. METHODES DE PLACEMENT

Les méthodes de placement sont classées en deux catégories :

- les méthodes calculant une solution optimale,
- les méthodes calculant une solution le plus souvent non optimale (approchée) et néanmoins acceptable.

Les méthodes de la première catégorie conduisent à une solution optimale au prix d'un temps de calcul prohibitif qui les rend impraticables sauf pour des circuits de très faible complexité. En revanche, les méthodes de la deuxième catégorie, applicables à la résolution du problème de placement, présentent un grand intérêt pratique.

II -4.1 Recherche d'une solution optimale

II -4.1.1 Méthode énumérative

Un arbre sur n niveaux est construit, la racine étant le niveau 0, pour énumérer les $n!$ permutations des n objets.

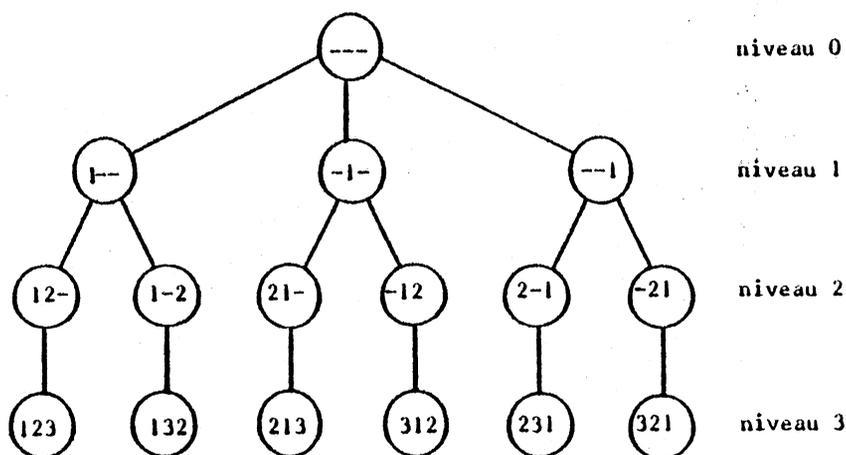


Figure 13 . Exemple de construction de l'arbre complet pour $n=3$

Au $k^{\text{ième}}$ niveau, k objets ont été placés et il reste $(n-k)$ positions possibles pour l'élément suivant, c'est à dire que $(n-k)$ branches partent de chaque noeud du niveau k vers le niveau $k+1$. La solution est générée au $n^{\text{ième}}$ niveau où les $n!$ permutations sont décrites.

II -4.1.2 Méthode semi-énumérative : "branch and bound"

La méthode du "branch and bound" a été définie par Little [Li 63] pour résoudre le problème du voyageur de commerce. On calcule tout d'abord un seuil, c'est à dire le coût d'une solution possible obtenue par une quelconque méthode (constructive, aléatoire, ...). La méthode consiste alors à développer l'arbre défini précédemment en évaluant pour chaque noeud le coût partiel du sous-réseau placé. L'arbre n'est plus développé à partir d'un noeud, si le coût partiel correspondant dépasse le seuil précédemment établi. Cette méthode conduit dans la pratique, sans construire l'arbre complet, à une solution optimale quand le nombre d'objets est limité ($n < 15$). Son efficacité dépend à la fois de la précision avec laquelle coût partiel et seuil sont évalués, ce qui permet d'élaguer plus ou moins l'arbre complet. Si par exemple aucun seuil n'a été calculé, la méthode du "branch and bound" devient alors une méthode entièrement énumérative. Un compromis devra être recherché entre la précision des seuils et des coûts partiels (accroissement du temps de calcul) et le nombre de noeuds à éliminer dans le parcours de l'arbre. Divers algorithmes basés sur cette méthode, ainsi que leurs applications sont clairement exposés dans [Han 72] et [Re 77] .

II -4.2 Recherche d'une solution approchée

II -4.2.1 Méthode stochastique [Han 72]

Plusieurs placements, dont on évalue les coûts, sont générés suivant la méthode de Monte-Carlo. La meilleure des solutions est retenue lorsque le calcul s'arrête :

- soit après un temps fixé à l'avance, c'est à dire après un nombre fixé de solutions générées,
- soit quand la meilleure solution est satisfaisante, c'est à dire que son coût est inférieur ou égal à une valeur fixée à l'avance.

II -4.2.2 Méthodes constructives [Han 72]

La méthode de placement constructif consiste à sélectionner un élément suivant une règle (en général, l'élément le plus connecté à un ou plusieurs éléments de l'ensemble déjà placé), puis à la positionner en minimisant le coût partiel du placement (le coût est en général réduit à la longueur des interconnexions). Les règles de sélection et de positionnement permettent de définir différentes méthodes.

II -4.2.2.1 Construction par paire (pair-linking method)

L'élément le plus connecté à l'un des éléments déjà positionnés est choisi, plus placé le plus près possible de cet élément de manière à minimiser la longueur des connexions. Le placement est initialisé en choisissant comme noyau la paire d'éléments ayant le plus grand nombre de connexions communes. Cette méthode, assez peu utilisée, est présentée dans [kur 65] .

II -4.2.2.2 Méthode par agrégation (Cluster-development Method)

L'élément le plus connecté aux éléments déjà positionnés est choisi, puis placé de manière à ce que les interconnexions soient le plus court possible, par exemple en utilisant la notion de barycentre. Chacun des éléments positionnés est affecté d'un poids, fonction du nombre de connexions avec l'élément sélectionné et leur barycentre est calculé. L'élément sélectionné est alors placé à ce barycentre, ou lorsque cette position est occupée par un autre élément, le plus près possible de ce barycentre.

La figure 14 décrit l'algorithme général correspondant à cette méthode. Voir dans [Kh 77] , [Pr 78] et [Wi 74] des applications de cette méthode à la construction d'un placement initial.

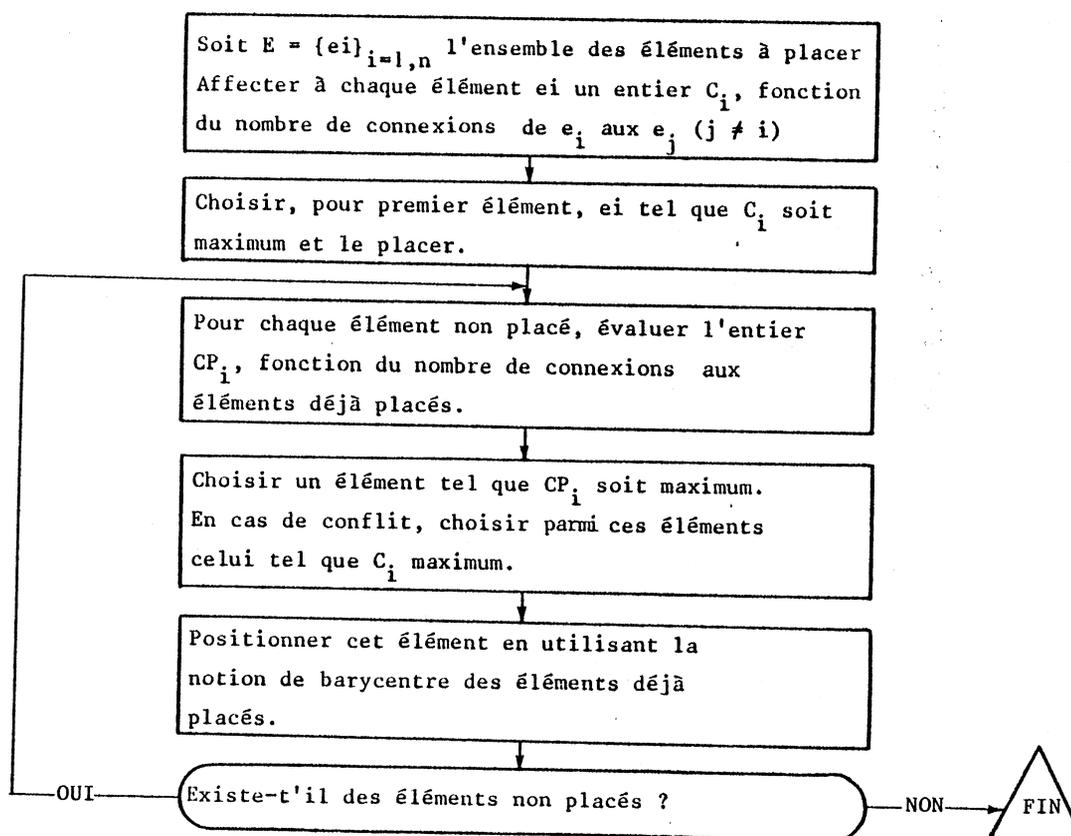


Figure 14 . Algorithme de la méthode par agrégation

II -4.2.2.3 Construction par niveaux

Dans la méthode de construction par niveaux, illustrée dans [Se 81-3] et [Sc 76] , un premier élément choisi en fonction de sa taille et du nombre de ses connexions constitue le noyau de l'étape suivante de construction. Les éléments sont ensuite sélectionnés en fonction de leur "connectivité" au noyau et placés autour pour former un premier niveau.

L'ensemble des éléments alors placés définit le noyau de l'étape suivante et la construction est ainsi effectuée par niveaux successifs . Cette méthode est bien adaptée au placement de blocs rectangulaires de dimensions variables.

II -4.2.3 Méthodes itératives d'amélioration

Un placement initial, obtenu préalablement, est transformé en modifiant l'emplacement d'un élément ou en échangeant certains éléments, de manière à réduire le coût du placement final. La transformation est effectuée jusqu'à ce que le placement ne soit amélioré.

II -4.2.3.1 Echange par paire

A chaque itération, le placement est modifié par l'échange de deux éléments. Deux algorithmes d'itérations sont possibles :

- effectuer tous les échanges que permet la méthode et conserver la meilleure solution. On réitère, à partir de cette configuration, jusqu'à ce qu'il n'y ait plus d'amélioration.

- effectuer les échanges que permet la méthode jusqu'à la première amélioration. On réitère à partir de cette configuration améliorée jusqu'à ce qu'aucun échange ne soit plus accepté.

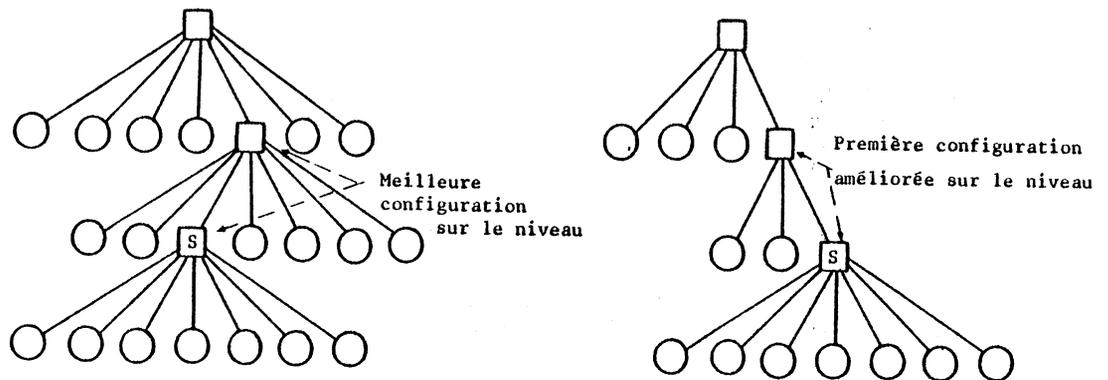


Figure 15 . Arbres développés par les deux méthodes d'itération

Ces algorithmes ne conduisent en général pas à la même solution et ne peuvent pas être comparés de manière absolue.

Le choix des éléments à échanger peut être réalisé de diverses manières :

- échange systématique des $n(n-1)/2$ paires obtenues lors d'un placement de n éléments [Se 80] ;
- choix pseudo-aléatoire des paires d'éléments à échanger [Lu 80] ;
- échange d'éléments vérifiant une certaine propriété :
 - . éléments voisins [St 61] ,
 - . éléments fortement connectés [Gl 59] ,

. éléments tels que chacun des deux éléments se trouve dans le voisinage du barycentre calculé en supprimant l'autre élément (force directed pairwise relaxation) [St 61] .
 Une variante de cette méthode, consistant à effectuer simultanément deux échanges, est proposée dans [Sm 80] .

II -4.2.3.2 Méthode par relaxation

Considérons un ensemble de n éléments placés : $n-1$ éléments étant fixés, un élément est déplacé vers le barycentre des $n-1$ éléments auxquels il est connecté (le poids associé à un élément est fonction du nombre de connexions avec l'élément déplacé). La méthode consiste à modifier le placement en considérant à chaque itération un élément différent, et s'arrête quand les modifications effectuées ne réduisent plus le coût du placement [Fisk 67] , [Go 79] .

II -4.2.3.3 Changement d'orientation d'un élément

Le placement est transformé en appliquant à chacun des éléments des opérations de rotation ou de symétrie [La 79] , [Se81-3] et [Pr 78]. Cette méthode est intéressante en particulier pour le placement de cellules ou de blocs rectangulaires.

II -4.2.4 Méthode constructive puis itérative

L'utilisation d'une méthode constructive suivie d'une méthode itérative permet en général l'obtenir un bon placement. La méthode constructive délivre un placement initial qui ne peut clairement pas constituer une méthode satisfaisante.

En effet, un élément est placé en fonction des seuls éléments déjà positionnés et non pas, par conséquent, en fonction de tous les éléments. Ce placement initial est ensuite amélioré au moyen d'un algorithme itératif. La plupart des méthodes de placement exposées dans la littérature sont basées sur ce principe.

Notons qu'en général les résultats obtenus après amélioration itérative d'un placement initial construit sont meilleurs qu'avec un placement initial aléatoire [Go 78] . Il peut aussi être intéressant de construire plusieurs solutions initiales et de les soumettre à une procédure itérative pour en conserver la meilleure [Go 79] .

II -5. CONCLUSION

La méthode la plus couramment utilisée dans la pratique consiste à construire rapidement un placement initial (construction par paire ou méthode par agrégation) et de l'améliorer par une méthode itérative (échange par paire).

La complexité des algorithmes employés varie de n^2 (n désignant le nombre d'éléments à placer) pour les échanges par paire et certaines méthodes constructives à n^4 pour la méthode "Branch and Bound".

Cependant, ces ordres de complexité ne reflètent que partiellement les temps de calculs requis par ces méthodes.

Par exemple, le placement constructif, bien que du même ordre de complexité que l'échange par paire, traite plus rapidement les problèmes quand le nombre d'éléments à placer est grand. De même, l'évaluation d'une méthode ne doit pas négliger les temps de calculs des critères que l'on veut optimiser. En tout état de cause, cette méthode ne produit en général pas une solution optimale mais une solution sous-optimale souvent acceptable.

III - INTERCONNEXION

III -1. INTRODUCTION

Une fois le placement des éléments réalisé, le traitement des connexions et du tracé peut être fractionné en quatre étapes successives :

- déterminer la liste des connexions,
- affecter chaque connexion ou partie de connexion à une couche,
- déterminer l'ordre du tracé,
- réaliser le tracé effectif sur chaque couche.

Nous aborderons très brièvement les trois premiers points qui préparent le tracé effectif et nous nous intéresserons de manière plus détaillée aux méthodes les plus courantes de tracé de connexions : algorithme de Lee et principales variantes, algorithme de recherche par ligne, algorithme d'expansion par ligne, méthode par tracé de canaux, méthode par zone saturée. Leur but est de déterminer le chemin suivi par une connexion entre deux points.

III -2. PREPARATION DU TRACE

III -2.1 Liste des connexions

Les méthodes de tracé proprement dit ne permettent que le traitement des connexions entre deux points. Il faut donc, à partir de la liste des modules et de leur interconnexion, c'est à dire des équipotentielles, déterminer l'ensemble des connexions (entre deux points) à tracer de longueurs les plus courtes. Le problème ne se pose en fait que pour les équipotentielles reliant plusieurs modules simultanément.

Des méthodes ont été développées à partir de problèmes équivalents de la théorie des graphes : algorithme de recherche de l'arbre de poids minimum dans un graphe et problème du voyageur de commerce [Kr 56] , problème de Steiner [Han 66] . Le problème de Steiner consiste à déterminer l'arbre de poids minimum reliant n points en ajoutant k ($k = 0, 1, 2 \dots$) points fictifs, de manière à ce que la longueur totale des connexions et le nombre de points fictifs soient minimum.

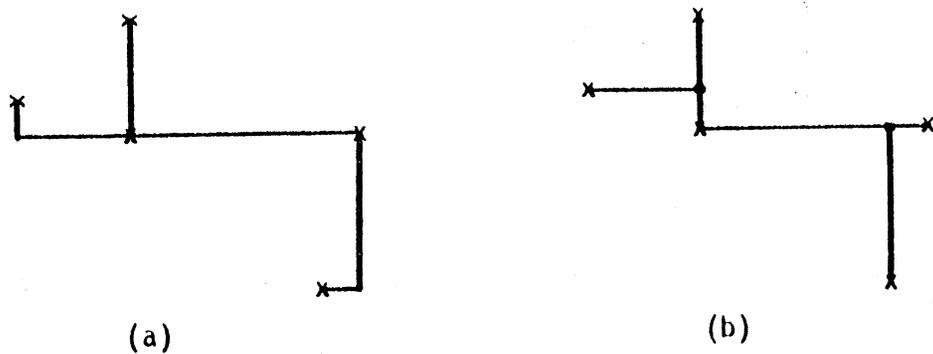


Figure 16 . (a) arbre de poids minimum

(b) arbre de Steiner (2 points fictifs)

III -2.2 Affectation aux "couches"

Une fois déterminée la liste des connexions à tracer, on les affecte aux différentes couches ou niveaux d'interconnexion disponibles. Le nombre de couches varie communément de une à quatre : une ou deux couches Aluminium, une ou deux couches Si poly(le plus souvent une Alu + une Si poly).

Pour les circuits à deux niveaux d'interconnexion, on trace par exemple le plus grand nombre de connexions sur une couche, puis on termine le tracé sur l'autre : cette méthode conduit à un nombre réduit de passages d'une couche à l'autre. Une autre politique peut être employée pour les circuits multi-couches : définir des classes de connexions suivant leur "tendance directionnelle" et affecter chaque classe à une couche. Abel décrit dans [Ab 72] plusieurs méthodes et conclue qu'il n'en existe pas une qui, dans tous les cas, soit supérieures aux autres.

III -2.3 Ordre de tracé

Chaque couche est ensuite examinée pour déterminer l'ordre dans lequel chaque connexion doit être tracée. En effet, comme le montre la figure 17, le tracé effectif dépend fortement de cet ordre.

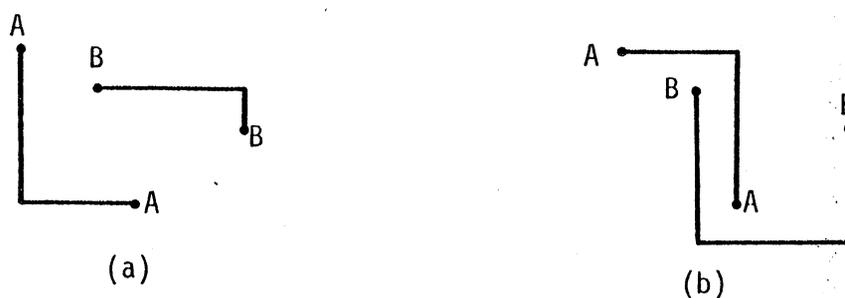


Figure 17 . (a) (BB) puis (AA)

(b) (AA) puis (BB)

Des algorithmes basés sur la longueur des connexions (choix de la plus courte, choix de la plus longue, ...) et simples à exécuter ont été développés. Une méthode proposée par Akers [Ak 72] consiste à construire le rectangle minimum autour d'une connexion et de dénombrer les extrémités (des autres connexions) qu'il contient.

Ces valeurs affectées à chacune des connexions et considérées dans l'ordre croissant déterminent alors l'ordre de tracé (figure 18).

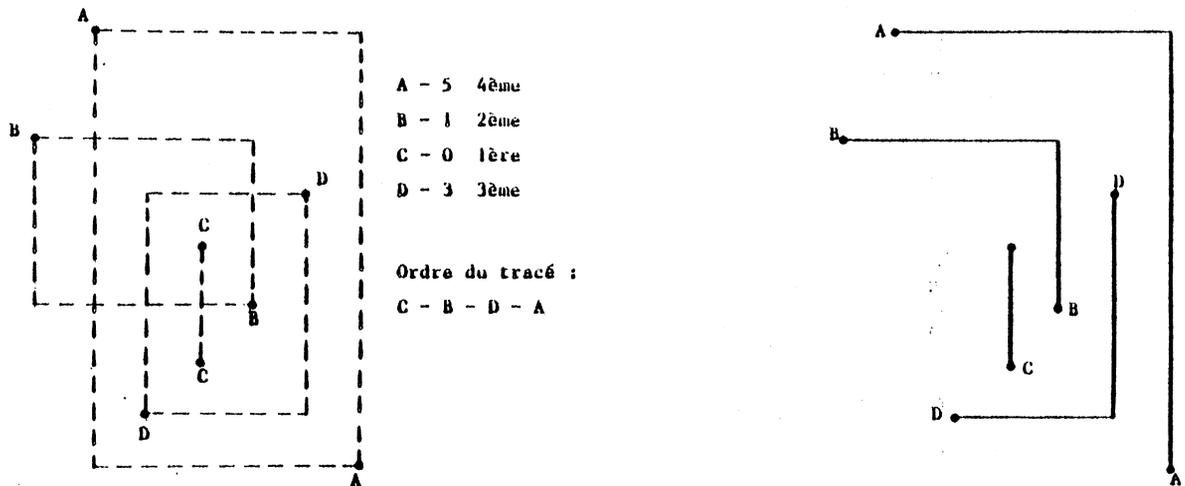


Figure 18 . Exemple de détermination de l'ordre de tracé

Les modifications nécessaires pour tenir compte des interactions entre connexions complique beaucoup ces algorithmes et accroît les temps de calcul même pour les cas simples [Ab 72] , [Kul 79] .

III -3. METHODES DE TRACE DES CONNEXIONS

Les méthodes de tracé que nous nous proposons d'étudier peuvent être classées suivant la structure hôte pour laquelle elles sont utilisées.

- Les algorithmes de type Lee sont les mieux adaptés et faciles à mettre en oeuvre pour une structure hôte à contrainte faible pour laquelle une grille est définie.

- Les algorithmes de recherche par ligne et d'expansion par ligne peuvent indifféremment être utilisés pour une structure hôte à contrainte faible ou forte.

- On préférera cependant les algorithmes de tracé de canaux et zone saturée sur les structures à contrainte forte pour lesquelles des canaux d'interconnexion sont définis.

III -3.1 Algorithme de Lee [Lee 61] , [Rub]

L'algorithme de Lee est sans doute le plus connu pour rechercher un chemin entre deux points A et B disposés sur un plan. Cet algorithme est en fait utilisé le plus souvent sur une grille rectangulaire, chaque case pouvant être vide, ou contenir un obstacle fixe ou une connexion.

Le processus est initialisé avec l'un des points A ou B à relier (les deux points ne sont pas équivalents quant au calcul nécessaire pour déterminer le chemin entre A et B). Une zone appelée onde est développée autour du point choisi (par exemple A) : un 1 est inscrit dans chaque case vide adjacente à A. Puis un 2 est inscrit dans chaque case vide adjacente aux cases contenant un 1. Et ainsi de suite jusqu'à l'une des deux situations suivantes :

- le processus se bloque, c'est à dire qu'au $k^{\text{ième}}$ pas, on ne peut plus trouver de case vide adjacente à celles contenant $k-1$. Il n'existe dans ce cas aucun chemin entre A et B ;
- le point B est atteint : les plus courts chemins entre A et B ont été déterminés. La connexion est alors considérée comme un obstacle pour les suivantes.

La figure illustre sur un exemple cette méthode et on y a fait apparaître une zone "privilégiée" des plus courts chemins possibles (figure 19 a). Un plus court chemin est montré figure 19 b.

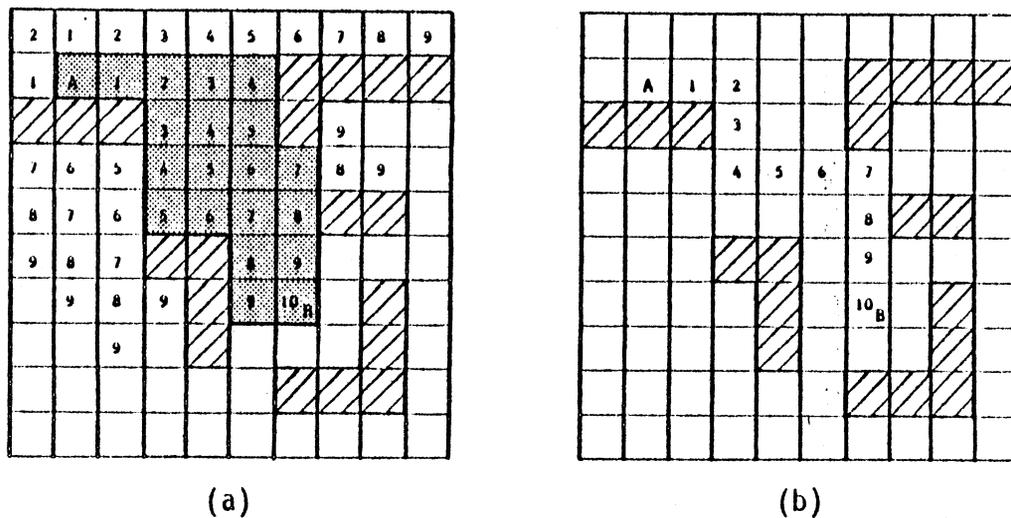


Figure 19 . Illustration de l'algorithme de Lee

Quand un chemin existe, la connexion est tracée en remontant de B vers A, en choisissant une des cases adjacentes du pas précédent. Si en théorie, le choix d'une case plutôt qu'une autre importe peu, en pratique quelques règles sont respectées. Par exemple le nombre de coudes de la connexion est minimisé en effectuant le moins possible de changement de direction. Quand un changement de direction est nécessaire pour contourner un obstacle, il sera effectué suivant une priorité pré-établie. Ceci permet de privilégier un nombre limité de directions, ce qui, on l'a constaté empiriquement, a pour effet d'augmenter le nombre de connexions tracées.

III -3.1.1 Réduction du temps de calcul

La phase d'expansion de l'algorithme de Lee précédemment exposé prend environ 90% du temps utilisé pour le tracé. Plusieurs modifications simples de l'algorithme original permettent une réduction de ce temps.

III -3.1.1.1 Choix des origines de l'expansion

- Un choix judicieux du point autour duquel l'onde est développée peut permettre, comme l'illustre la figure 20 un gain appréciable en temps. En règle générale, le point le plus éloigné du centre du circuit est choisi comme départ de l'expansion.

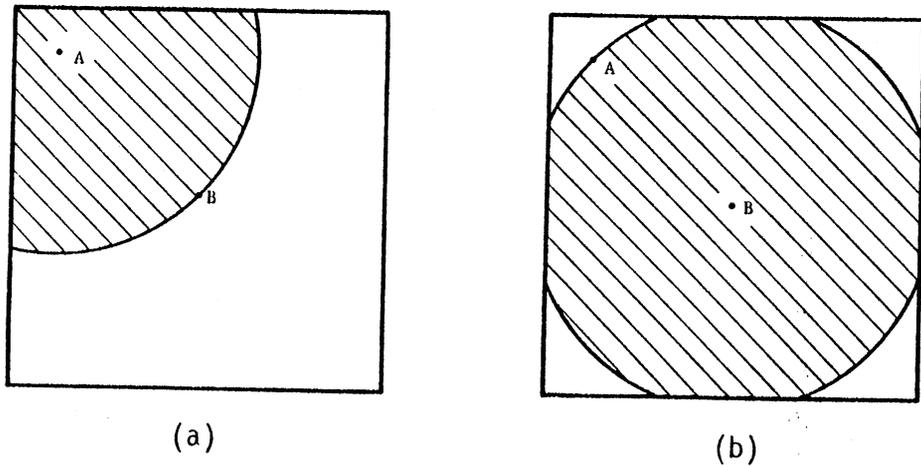


Figure 20 . (a) point de départ de A
(b) point de départ de B

- Les deux points A et B sont considérés simultanément comme point de départ, et l'onde est développée autour de chacun d'eux (figure 21).

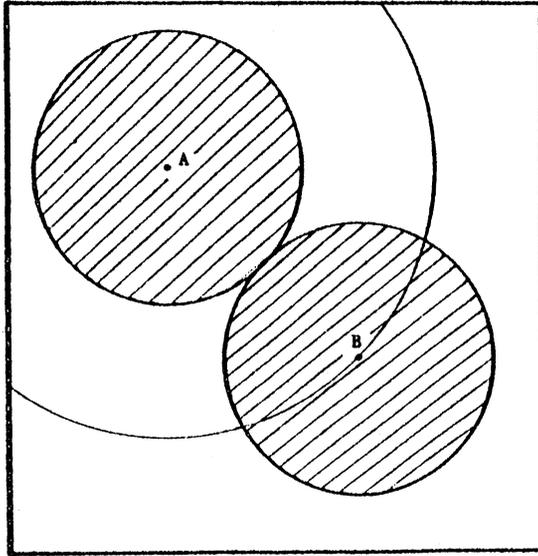


Figure 21 . L'onde est développée simultanément à partir de A et de B

III -3. 1.1.2 Limitation de la zone de recherche

- Un moyen pratique efficace consiste à limiter la zone de recherche du chemin à une fenêtre rectangulaire contenant A et B. Si aucun chemin n'est obtenu à l'intérieur de cette fenêtre, elle est agrandie pour poursuivre la recherche (figure 22).

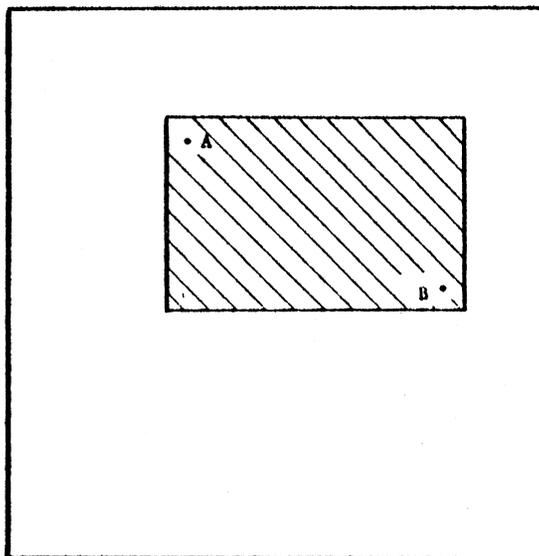


Figure 22 . Limitation de l'expansion à un rectangle

- La méthode proposée dans [Ta 80] considère une zone de recherche du chemin en forme de L (figure 23) dont la largeur varie à chaque itération.

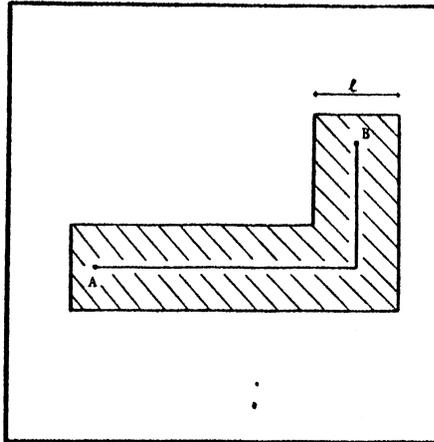


Figure 23 . Limitation de l'expansion à une zone en forme de L

L'algorithme (figure 24) consiste à rechercher le plus grand nombre de connexions (par la méthode de Lee) dans les zones de largeur fixée et de réitérer le procédé en augmentant la valeur de 1. Cela a pour effet d'établir un pourcentage élevé (> 90%) de connexions : en effet, chaque étape détermine des chemins dont le tracé est d'autant plus simple que 1 est petit, ce qui facilite le tracé des connexions suivantes. De plus, un gain en temps d'environ 25% est réalisé.

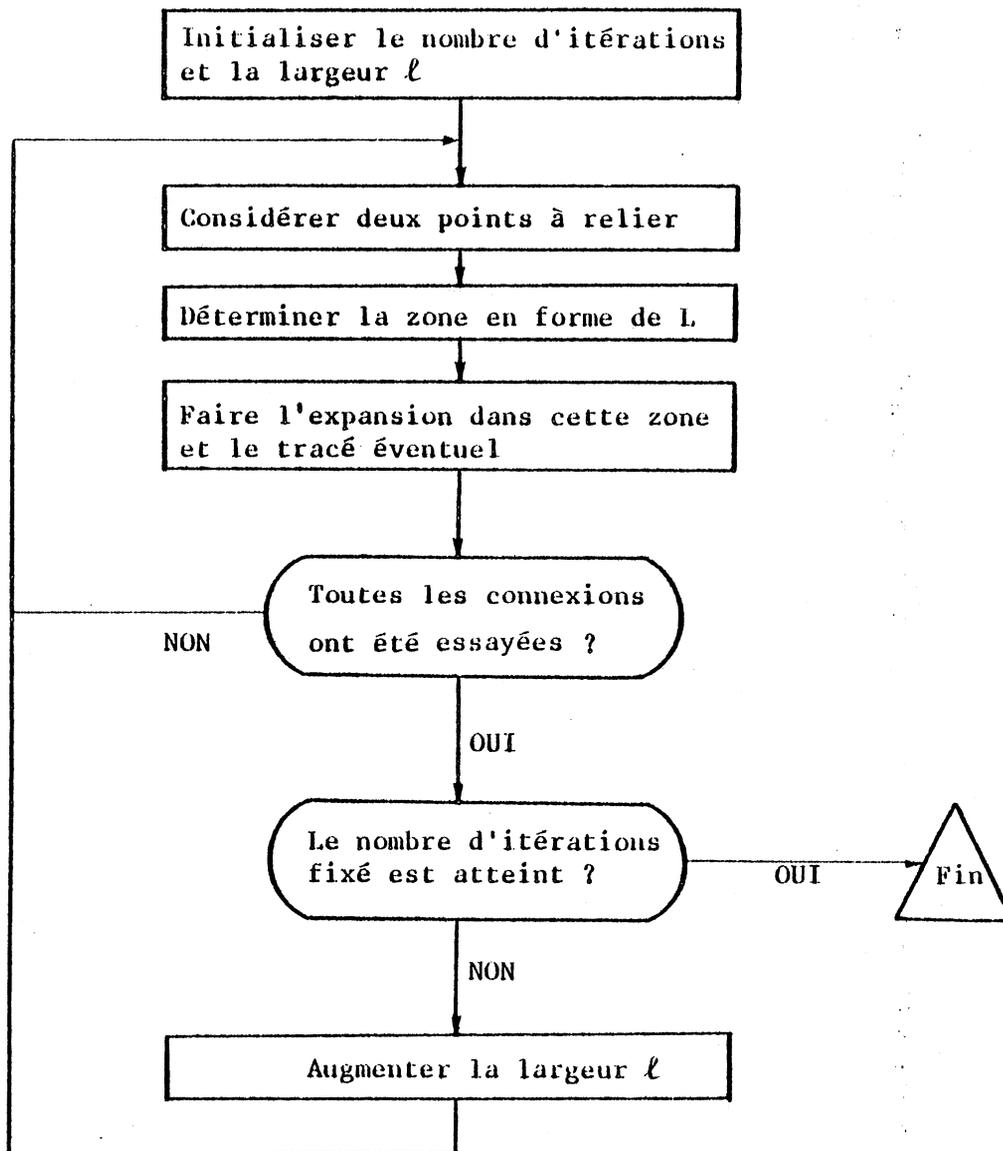
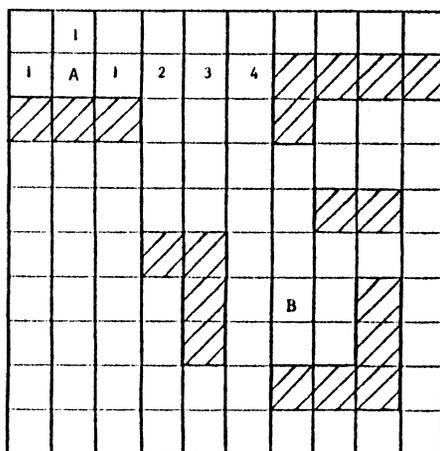
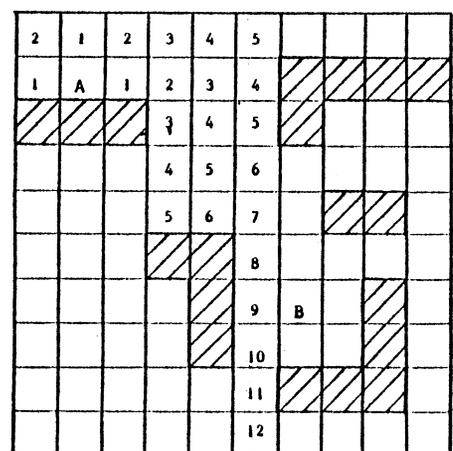


Figure 24 . Variante de l'algorithme de Lee

- Au lieu de développer une onde autour de la case de départ, l'expansion est effectuée horizontalement et verticalement le plus loin possible (figure 25 a) à partir du point de départ. L'expansion continue ensuite, si besoin est, perpendiculairement à l'axe le plus long obtenu précédemment, à partir de chacune des cases de cet axe (figure 25 b). Cette modification proposée par Mikami [Mi 68] permet de réduire de moitié le temps d'expansion de l'onde par rapport à l'algorithme original.



expansion verticale et
horizontale



expansion à partir des cases
de l'axe horizontal

Figure 25 . Variante de l'algorithme de Lee

III -3.1.2 Réduction de l'encombrement de la mémoire

Un gros inconvénient de la méthode de Lee est la place mémoire importante utilisée. En effet, chaque case de la grille est représentée à l'aide de plusieurs bits indiquant son état (vide, obstacle, atteinte) et son rang dans l'expansion lorsqu'elle est atteinte.

Supposons qu'on étiquette successivement les cases de l'expansion suivant la séquence 0-0-1-1-0-0-1-1-... (figure 26). On peut à l'aide de cette séquence remonter de B vers A une fois le point B atteint lors de l'expansion : en effet, il est suffisant de pouvoir distinguer un "prédécesseur" d'un "successeur" pour réaliier le tracé de la connexion, un fois déterminé un chemin entre A et B. Cela permet un étiquetage de chaque case plus simple (le rang dans l'expansion est supprimé) pour lequel un "prédécesseur" a un étiquette différente d'un "successeur" [Ak 67] .

Deux bits seulement sont alors nécessaires pour coder chaque case :

00 case disponible

01 obstacle

10 case atteinte étiquetée 0

11 case atteinte étiquetée 1

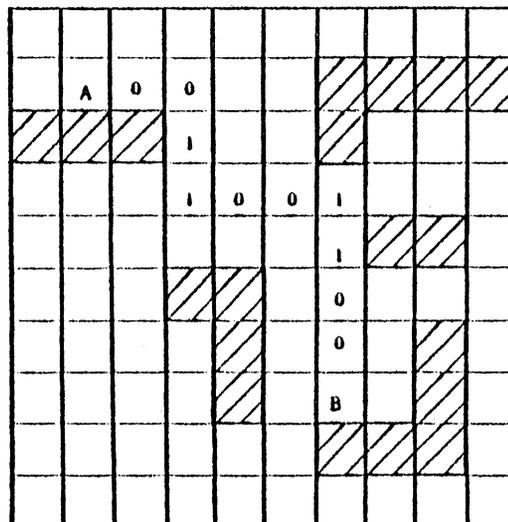


Figure 26 . Utilisation d'un étiquetage minimum dans l'algorithme de Lee

Dans les méthodes précédentes, une connexion une fois tracée devient un obstacle fixe pour les autres et peut par conséquent allonger intempestivement une autre connexion, voire empêcher son tracé. Pour éviter ce type de conflit, Rubin dans [Ru 74] détermine pour chaque connexion le chemin le plus court (chemin alors dit "privilegié") et supprime les croisements en déformant sans trop s'écarter du chemin privilégié ou en retraçant certaines des connexions en ayant alors une "vision" globale du circuit. Cette méthode, au prix d'une programmation difficile, semble prometteuse, car elle permet une vision globale du tracé.

L'intérêt de la méthode de Lee réside dans le fait que, quand une solution existe, la plus courte est déterminée. Cependant, cette méthode est d'autant plus coûteuse que le nombre de cases occupées est faible.

Aussi, il sera préférable de déterminer le maximum de connexions par des méthodes moins onéreuses et d'utiliser l'algorithme de Lee pour achever le tracé.

III -3.2 Algorithme de recherche par ligne (line-search algorithm) [Hi 69]

La recherche d'un chemin n'est pas effectuée point par point (utilisation d'une grille) comme dans la méthode précédente, mais au moyen d'un balayage par lignes, ce qui ne nécessite pas le stockage du plan. On ne génère pas une onde autour d'un point mais des "lignes de dégagement" pour contourner un obstacle.

Considérons le point A et les segments l_1 vertical, et l_2 horizontal (figure 27) dont les extrémités sont les obstacles rencontrés (bord du circuit, autre composant, autre connexion). Considérons ensuite les droites perpendiculaires à l_1 et l_2 en des points déterminés suivant un pas fixé à l'avance (a, b, c, d, e, f, g, h, i, j, k). Si le point B à relier n'appartient pas à une de ces droites, on détermine celles (appelées lignes de dégagement) qui permettent de franchir les obstacles rencontrés par l_1 et l_2 (a, e, f, g, k).

L'une de ces droites est alors choisie (par exemple a), ce qui définit sur l_1 un point de dégagement (noté A'). L'ensemble des perpendiculaires à a est ensuite construit. Si un chemin n'est pas déterminé, on réitère le procédé à partir d'un nouveau point de dégagement sur a. Dans la pratique, les lignes de dégagement sont générées à partir des deux points A et B à relier, ce qui définit un réseau de segments verticaux et horizontaux. Lorsque deux segments générés à partir de A et B se croisent, un chemin est alors établi et la connexion est définie.

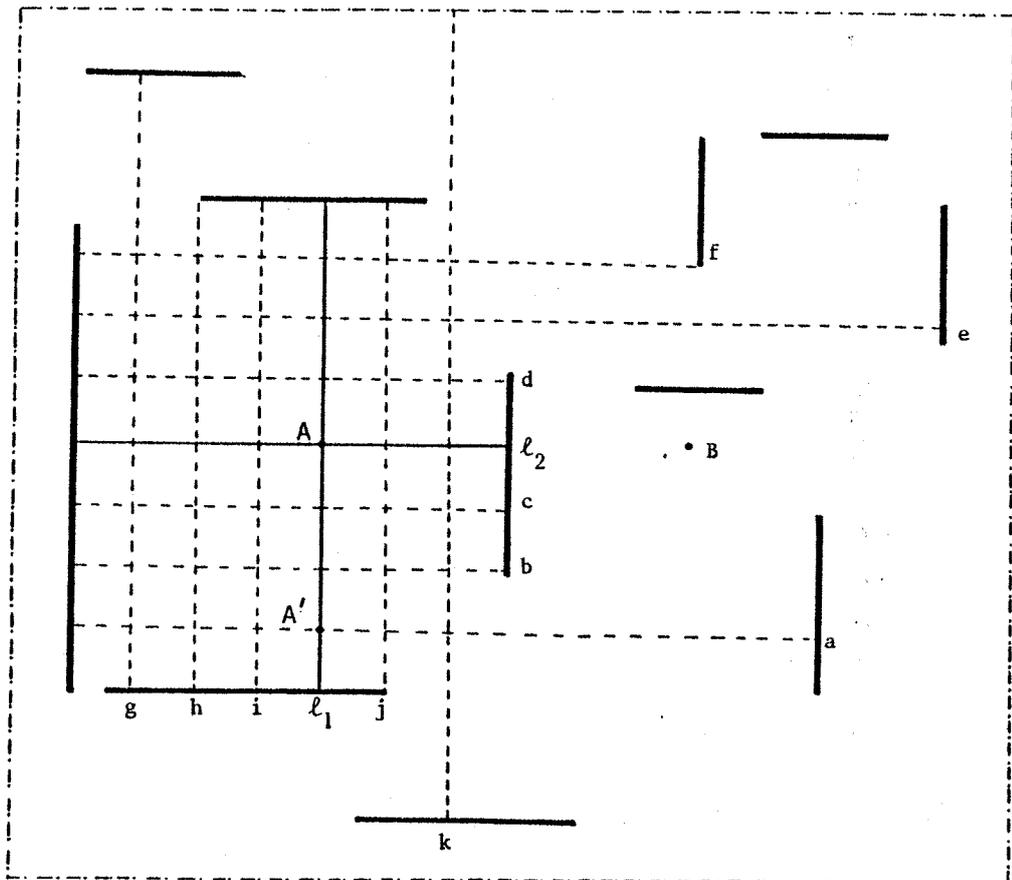


Figure 27 . Méthode de recherche par ligne

L'algorithme consiste donc à rechercher des points de dégagement, à construire des lignes de dégagement, à déterminer si elles se croisent, puis à répéter le cas échéant ce procédé jusqu'à ce qu'un chemin soit trouvé.

Cet algorithme, plus rapide et moins encombrant que les algorithmes basés sur la méthode de Lee, ne trouve cependant pas toujours une solution même s'il en existe une. De plus, le chemin déterminé peut ne pas être le plus court.

III -3.3 Algorithme d'expansion par ligne (line-expansion algorithm) [He 80]

Comme dans la méthode précédente, des droites verticales ou horizontales sont construites (lignes d'expansion), mais au lieu de ne considérer qu'une seule ligne de dégagement, une zone d'expansion est définie : c'est l'ensemble des points atteints par toutes les lignes d'expansion générées.

Soit le segment l_1 (figure 28) contenant le point A et limité par les obstacles b_{01} et b_{02} . L'ensemble des droites perpendiculaires à l_1 (suivant un pas fixé) limitées par les obstacles b_{11}, \dots, b_{15} définit une zone d'expansion ZEI. Les bords de ZEI qui ne sont pas des obstacles sont appelés lignes actives. Ces lignes que les connexions peuvent traverser permettent de générer de nouvelles zones d'expansion, et par conséquent de nouvelles lignes actives. Par exemple, a_{15} permet de construire la zone ZEa_{15} au moyen de lignes verticales et définit a_{21}, a_{22}, a_{23} .

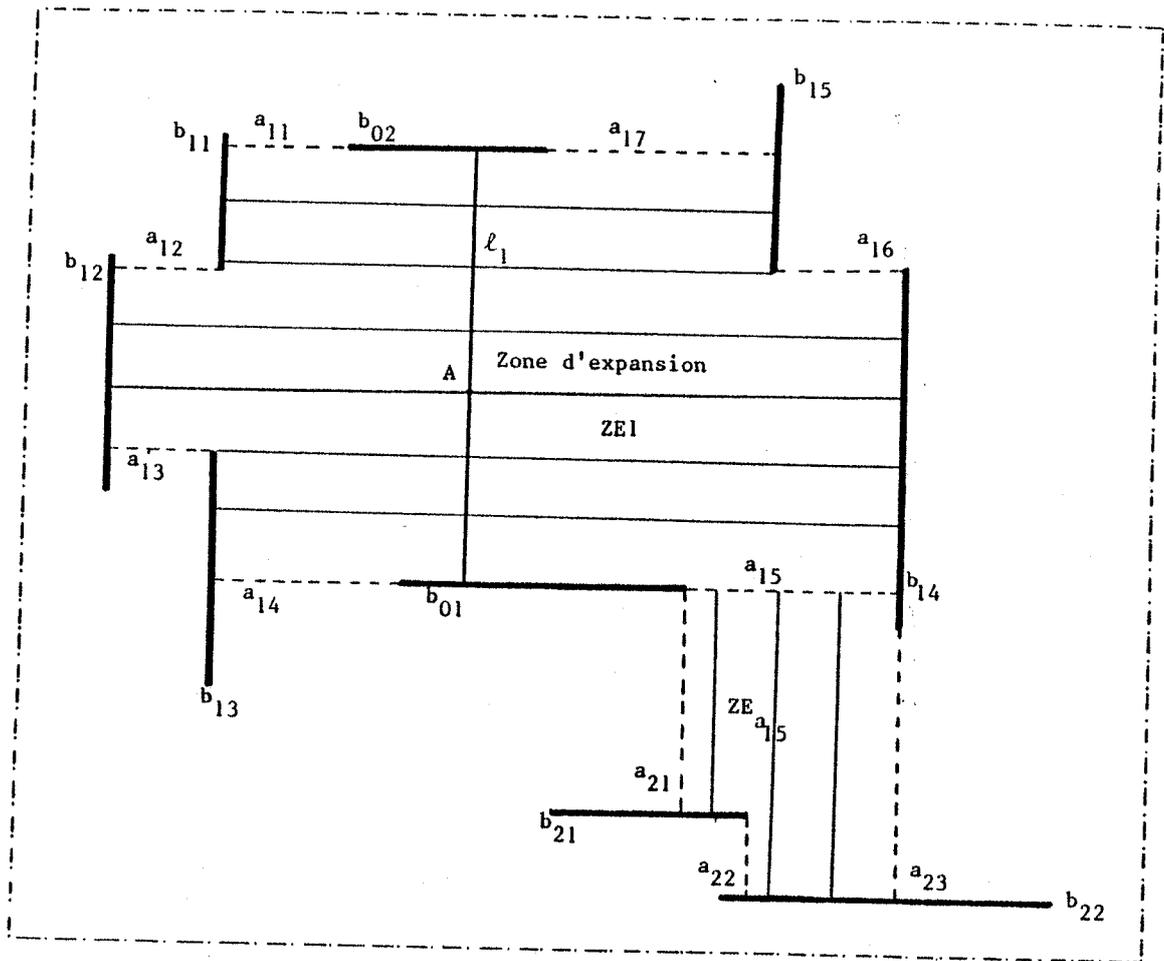


Figure 28 . Methode d'expansion par ligne

L'expansion à partir des lignes actives n'est effectuée que dans un seul sens : il n'y a pas de recouvrement de zones d'expansion. De plus, les lignes actives une fois définies sont considérées comme des obstacles pour les expansions suivantes. Remarquons que la direction de l'expansion d'une ligne active est orthogonale aux lignes d'expansion qui l'ont générée.

Dans la pratique, deux zones d'expansion sont construites à partir d'un segment passant par A et d'un segment passant par B. Ces deux zones sont élargies par la procédure d'expansion décrite ci dessus jusqu'à l'une des deux situations suivantes lorsqu'il existe un chemin :

- une des lignes actives d'une zone est confondue avec une ligne active (devenue obstacle franchissable) de l'autre zone,
- un des points à relier se situe dans la zone d'expansion générée à partir de l'autre point.

On peut alors construire un chemin entre A et B.

L'organigramme de l'algorithme d'expansion entre les points A et B est décrit figure 29. S_A et S_B sont les ensembles des lignes actives définies à partir de A et de B.

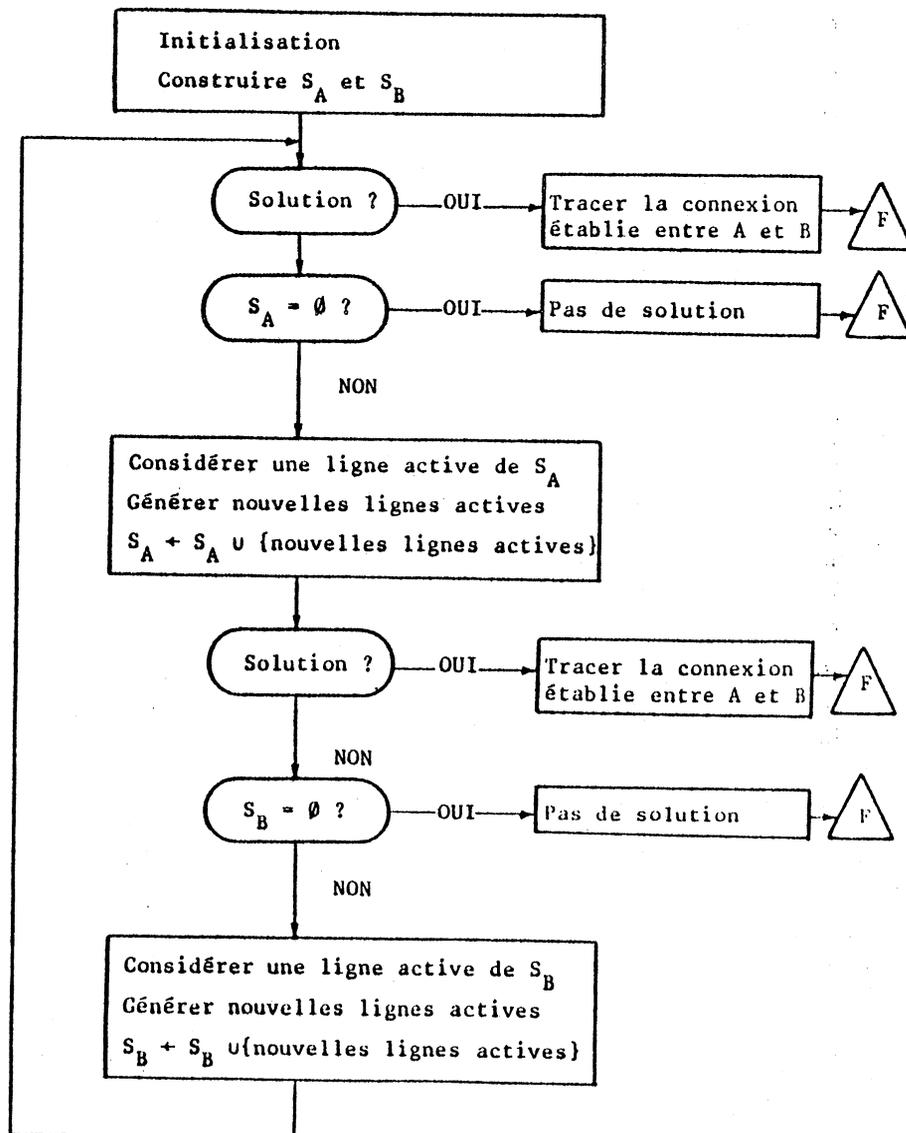


Figure 29 . Algorithme d'expansion par ligne

Plusieurs stratégies permettent le choix d'une ligne active pour l'élargissement de la zone d'expansion :

- toutes les possibilités d'un même niveau d'expansion sont explorées avant de passer au niveau suivant,
- une seule ligne active est considérée à chaque niveau d'expansion avec retour au niveau précédent en cas de blocage (retour-arrière),
- les lignes actives sont choisies suivant une méthode heuristique, par exemple en établissant une priorité en fonction de la proximité des lignes actives de chaque zone au point à atteindre.

Cette méthode présente, par rapport à la recherche par ligne, l'avantage de toujours trouver une solution s'il en existe une. De plus, elle permet, grâce au faible encombrement de la mémoire, de traiter des circuits de grande complexité et à forte densité de connexions.

III -3.4 Méthode par tracé de canaux (channel routing)

Cette méthode fut introduite par Hashimota et Stevens [Has 71] pour le tracé de connexions sur cartes bi-couches. Elle peut être utilisée avec succès sur les réseaux prédéfinis permettant deux niveaux d'interconnexion et pour lesquels des canaux d'interconnexion sont définis verticalement et horizontalement.

La capacité d'un canal désigne le nombre de pistes parallèles qu'il peut contenir. Cette capacité est donc définie d'une part par la longueur physique du canal et d'autre part par la largeur des pistes et leur espacement.

Chaque piste permet le tracé d'un ou plusieurs connexions ne se recouvrant pas.

Une première étape affecte les connexions (sous forme de segments horizontaux et verticaux) à chaque canal, en tenant compte de leur capacité. La notion de densité de connexions permet une estimation réaliste du nombre de pistes nécessaires au tracé.

Une deuxième étape répartit ensuite les segments dans les canaux de façon à occuper le minimum de pistes et à assurer la continuité d'une même connexion (trous de passage d'une couche à une autre).

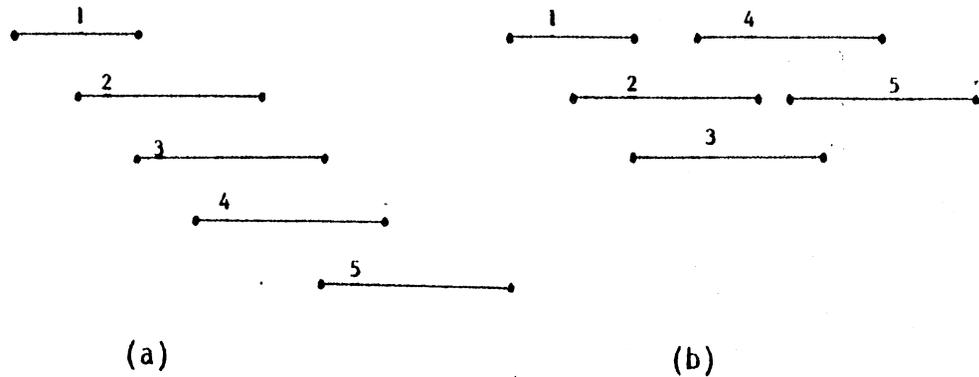


Figure 30 . Les 5 segments (a) occupent 3 pistes (b)

Le tracé effectif est réalisé le plus tard possible pour pouvoir utiliser au mieux la surface disponible et réduire le nombre de passage d'une couche à l'autre.

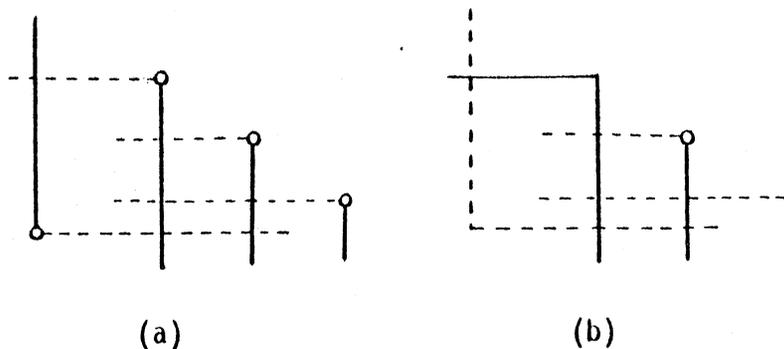


Figure 31 . Réduction du nombre de trous de passage

(a) 4 trous

(b) 1 trou

Fisher [Fish 78] propose une classification des différents types de conflits entre connexions (capacité, passages d'une couche à l'autre) résultant de la première étape et accroît l'efficacité de la méthode originale en remplaçant la deuxième étape par autant de procédures que de classes de conflits définies.

La notion de canaux d'interconnexion est exploitable pour résoudre le tracé lors d'une approche hiérarchisée de l'implantation de circuits intégrés [Pr 79] . Le circuit complexe est partitionné en blocs ou cellules rectangulaires de complexité moindre. L'interconnexion de ces blocs se fait à travers des canaux définis par le placement, ces canaux pouvant être modifiés pour faciliter le tracé.

Cette méthode ne produit pas les solutions les plus courtes mais permet une bonne répartition des connexions et donc une meilleure utilisation de l'espace disponible.

III -3.5 Méthode par zone saturée (saturated zone method)

Une zone saturée, définie par Wu et Schmidt [Wu 76] , est une portion du plan ne contenant pas les extrémités d'une même connexion non tracée. Si toutes les connexions entre les points de deux zones saturées adjacentes sont tracées, l'union de ces deux zones forme une nouvelle zone saturée (figure 32).

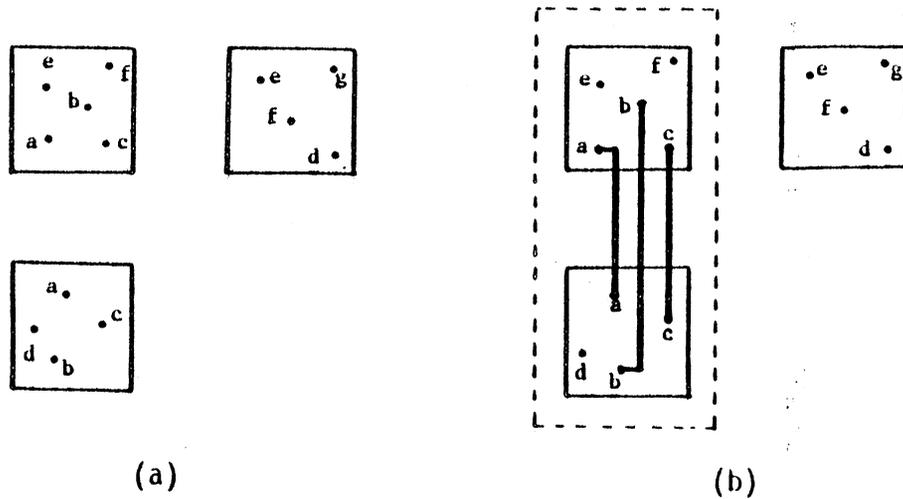


Figure 32 . Combinaison de deux zones saturées

(a) 3 zones

(b) 2 zones

En combinant ainsi les zones saturées adjacentes, on obtient finalement, lorsque toutes les connexions sont tracées, une zone saturée couvrant le plan tout entier (i.e. le circuit). La méthode consiste donc à considérer simultanément les zones saturées adjacentes, ce qui revient globalement à sélectionner les connexions des plus courtes aux plus longues et permet de traiter simultanément des connexions susceptibles d'utiliser les mêmes ressources physiques (chemin, passage d'une couche à l'autre). Cinq procédures spécialisées sont utilisées pour le tracé.

La *procédure 1* traite les connexions entre deux points alignés verticalement ou horizontalement (figure 33-a).

La *procédure II* construit des chemins (figure 33-b) entre deux points non alignés verticalement ou horizontalement.

La *procédure III* détermine les connexions comportant trois coudes (figure 33-c) par la méthode d'expansion par ligne. Durant cette étape, le canal entre les deux zones saturées adjacentes verticalement (horizontalement) n'est pas utilisé pour le tracé de segments horizontaux (verticaux). Ceci permet de préserver, si possible, ce canal pour le tracé ultérieur de connexions horizontales (verticales).

La *procédure IV* traite les connexions comportant 2, 3 ou 4 coudes et dont un segment occupe le canal précédemment défini (figure 33-d).

La *procédure V* recherche les connexions à quatre coudes par la méthode d'expansion par ligne avec possibilité d'utiliser le canal défini précédemment. Les procédures I, II et IV traitent simultanément les connexions des zones considérées, ce qui permet d'éviter certains blocage. Ces cinq procédures sont appliquées tour à tour pour chaque paires de zones saturées adjacentes.

Il est important de définir une stratégie pour sélectionner les zones saturées à traiter. Dans la solution adoptée par Lorenzetti et Smith II dans [Lor80] le traitement est initialisé en considérant chaque extrémité de connexion comme une zone saturée. Les zones adjacentes d'une même ligne (ou colonne) sont alors combinées dans un ordre prédéfini (exemple figure 34).

Cette méthode, par le mode de sélection des connexions qu'elle impose, permet un traitement parallèle local, ce qui réduit le nombre de cas de blocage par rapport aux méthodes entièrement séquentielles.

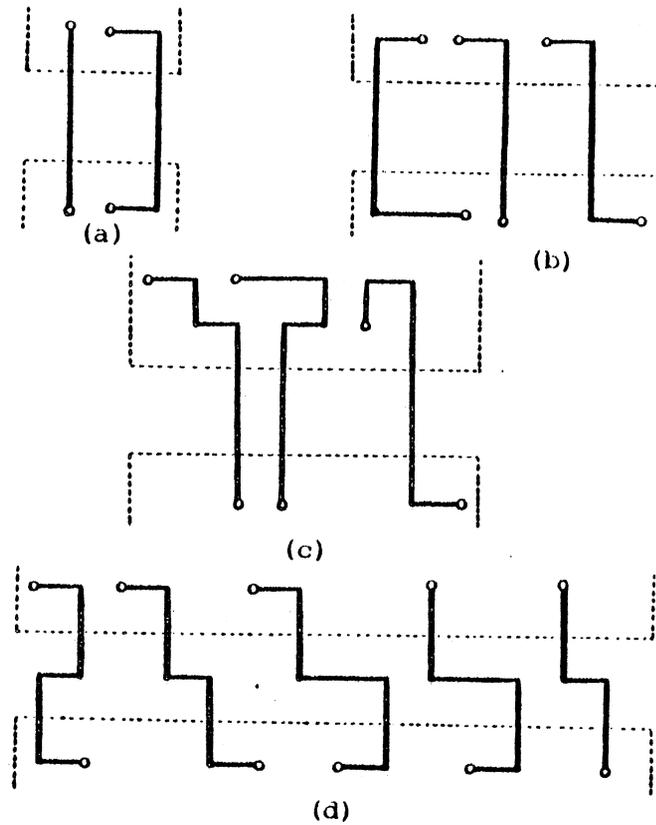


Figure 33 . Classement des connexions entre deux zones adjacentes verticalement

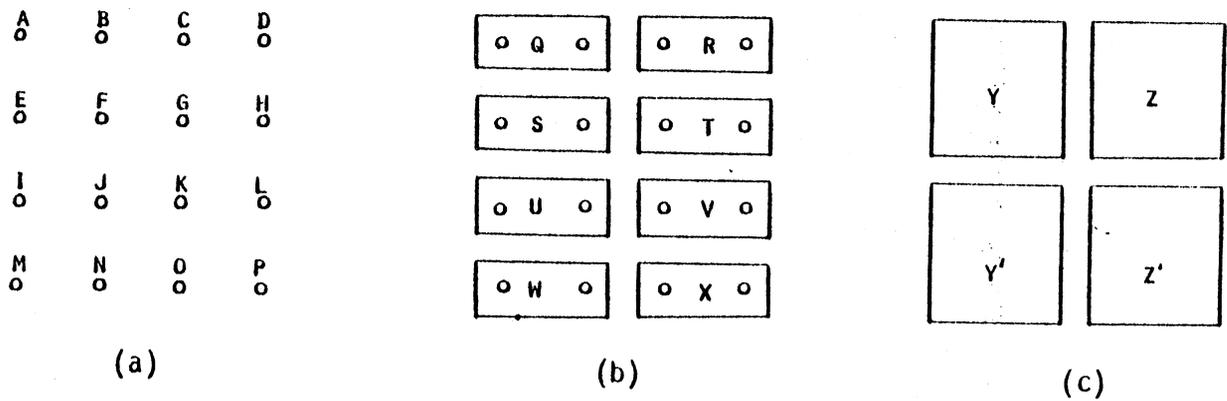


Figure 34 . (a) Répartition initiale des zones

(b) Combinaison des zones adjacentes d'une même ligne

(c) Combinaison des zones adjacentes d'une même colonne

III -4 CONCLUSION

L'étude précédente nous montre qu'un bon outil de tracé automatique doit remplir quatre conditions :

- connexion de longueur minimale,
- rapidité d'exécution
- encombrement mémoire peut important,
- et surtout tracé de toutes les connexions.

Aucune des méthodes exposées ne satisfait simultanément à ces quatre points : l'algorithme de Lee est encombrant et peu rapide, les algorithmes de balayage par lignes ou d'utilisation de canaux ne déterminent pas toujours le tracé de longueur minimale. De plus, elles ne garantissent pas un taux de réussite de 100%. Dans la pratique, l'efficacité du tracé est augmentée en combinant ces méthodes : Higtower et Boyd [Hi 80] résolvent le problème de tracé de connexions entre cellules par l'assignation de canaux et l'algorithme de Lee définit les connexions de passage à travers les cellules. Dans [Ni 80], l'algorithme de recherche par ligne traite le plus grand nombre de connexions et l'algorithme de Lee est utilisé pour tracer les restantes.

Dans les méthodes précédentes, une connexion, une fois tracée, devient un obstacle pour les suivantes et peut entraîner des cas de blocage.

Le circuit est alors terminé manuellement du prix de grandes difficultés et d'un accroissement de temps de conception. Pour éviter cet inconvénient, deux améliorations sont proposées :

- Une analyse prédictive des effets du tracé d'une connexion sur la suite du processus met en évidence les blocages et permet alors de définir une stratégie pour les éliminer.

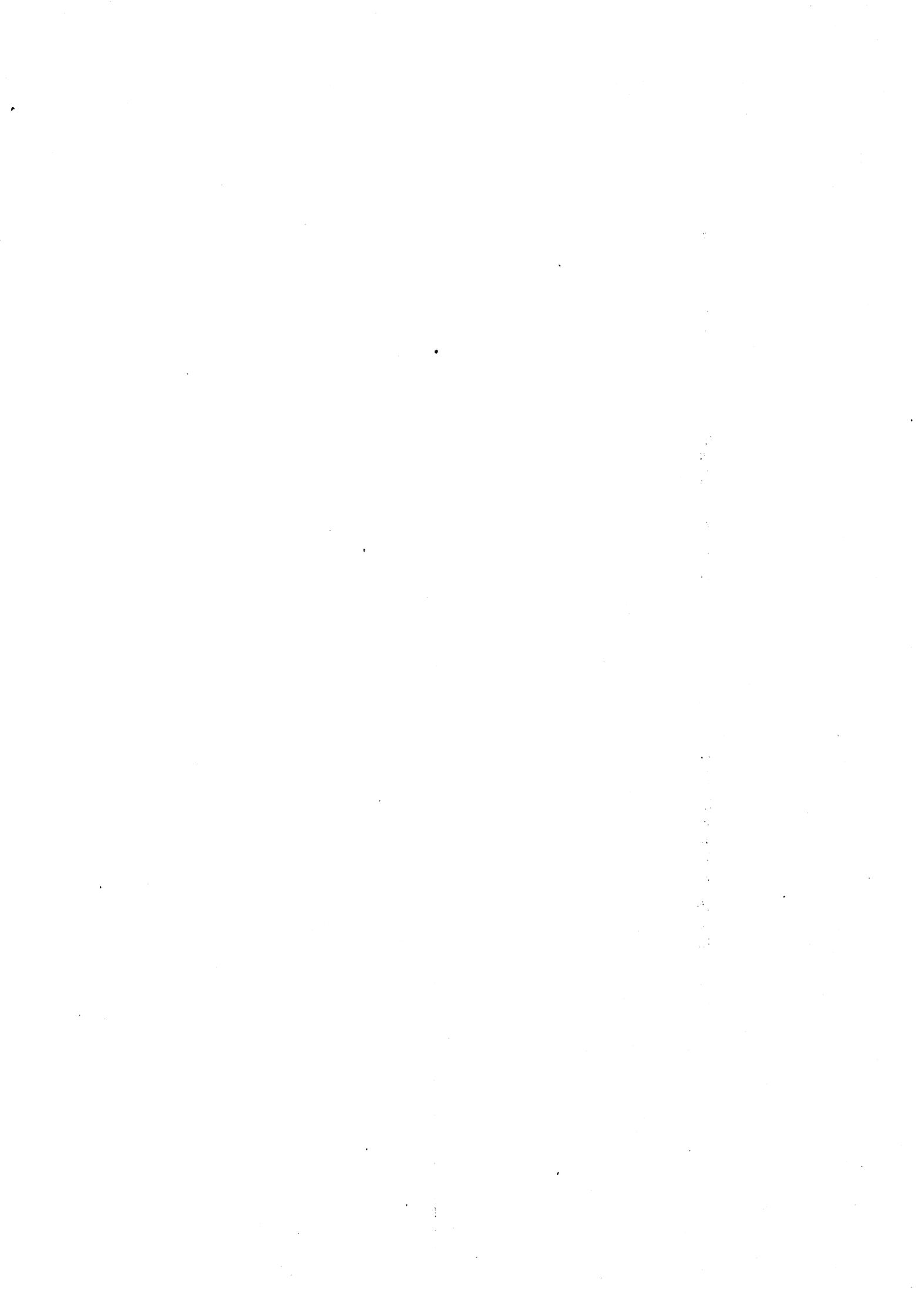
- Le tracé n'est pas effectué séquentiellement mais "globalement". Les connexions définies sont susceptibles de modifications en fonction du tracé des autres connexions.

L'application des méthodes précédentes en tenant compte de ces deux principes permettrait alors de définir un outil de tracé automatique et efficace.

IV - CONCLUSION

Les diverses méthodes d'implantation que nous avons présentées et discutées ont pour la plupart initialement été développées pour les circuits imprimés. Elles peuvent cependant efficacement être utilisées pour résoudre certains problèmes posés par le placement et l'interconnexion automatiques dans les circuits intégrés, comme nous le montrons dans le chapitre III concernant l'étude de cas (MD-MOS).

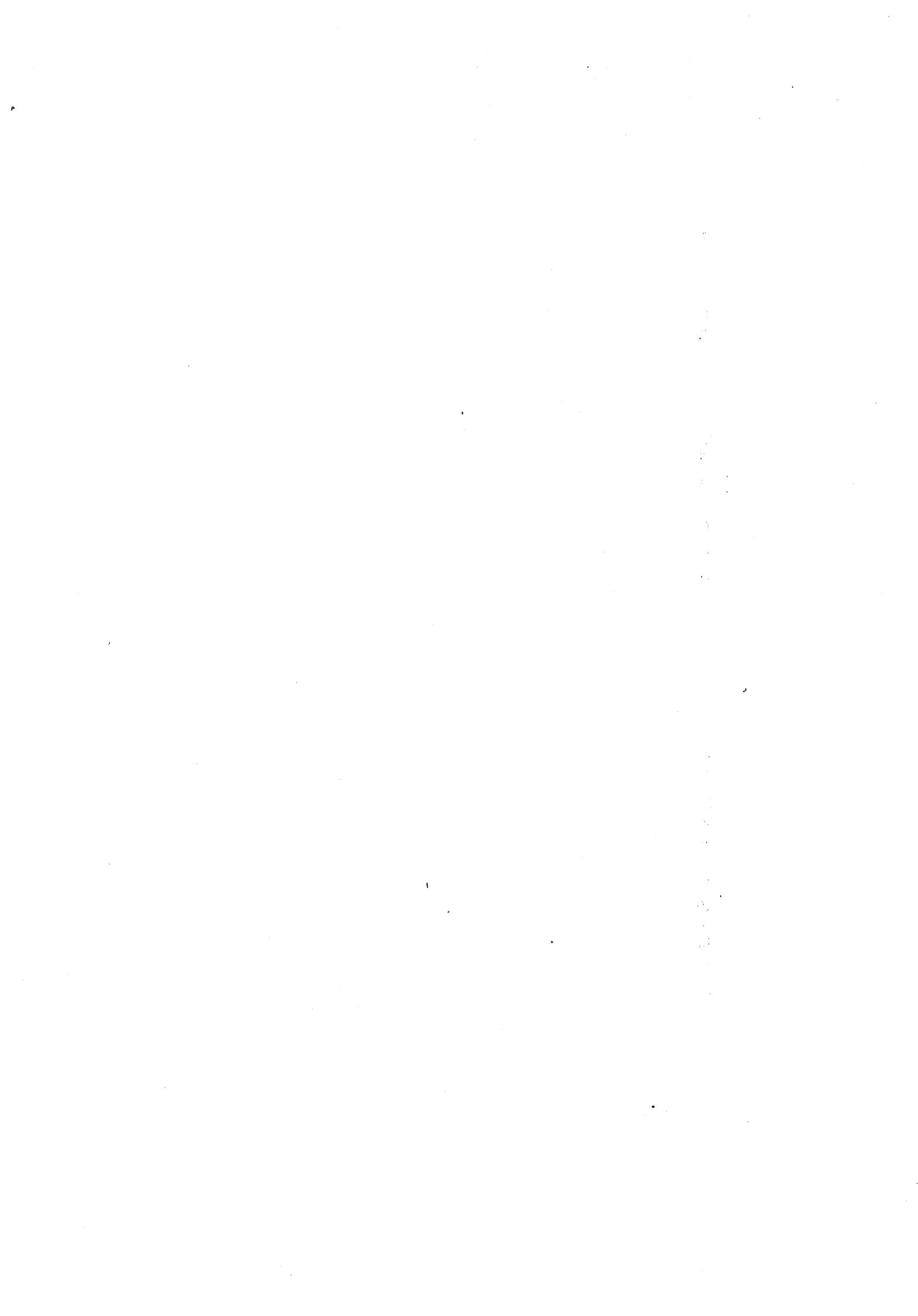
La notion de vecteur coût que nous avons introduite, utilisée avec des méthodes constructives et itératives ont permis de répondre efficacement au problème du placement des portes MD-MOS. Pour le tracé des connexions, des méthodes plus originales ont été définies pour tenir compte des contraintes particulières des circuits MD-MOS.





CHAPITRE II

LA REPRÉSENTATION SYMBOLIQUE :
PRÉSENTATION ET CLASSIFICATION
LE MD-MOS : ÉTUDE DÉTAILLÉE



I - LA REPRÉSENTATION SYMBOLIQUE : PRÉSENTATION ET CLASSIFICATION

I - 1. INTRODUCTION

Dans une approche classique de la conception des circuits intégrés, les différents niveaux de masques technologiques pour la fabrication doivent être dessinés de manière très précise (dessin au micron). Suivant les filières technologiques utilisées, un ensemble de règles de dessin sont définies et doivent être respectées; l'obtention des masques est alors une tâche longue et fastidieuse, souvent génératrice d'erreurs.

L'implantation symbolique est une méthode de conception des masques utilisant des symboles contenant les détails technologiques. Ces symboles sont ensuite transformés de manière automatique pour finalement obtenir les masques. Dans cette approche, la tâche fastidieuse du dessin au micron est supprimée et le respect de toutes les règles géométriques est assuré par programme, conformément à la structure topologique des éléments du circuit dessiné sous forme symbolique. On obtient donc un schéma des masques conforme aux règles par construction, ce qui permet de s'affranchir des programmes compliqués et coûteux de vérification du dessin [Tr 81], [Be 79].

Les règles que le concepteur doit assimiler sont peu nombreuses et d'un apprentissage facile. Elles permettent de définir un langage simple de description pour une utilisation automatique ou interactive des différents symboles. Le temps de conception des masques est alors considérablement réduit par rapport à une conception au micron.

De plus, la complexité du problème du dessin des masques et le volume des données à manipuler sont beaucoup plus faibles dans le cas d'une implantation symbolique, ce qui facilite la réalisation d'outils automatiques d'implantation.

Tous ces avantages ne doivent cependant pas éclipser un inconvénient qui freine une plus large utilisation de la représentation symbolique pour l'implantation : la densité d'intégration des circuits obtenus est plus faible. En effet, les règles simplifiées utilisées doivent toujours tenir compte du pire cas et le concepteur ne peut pas tirer parti de toutes les finesses permises dans la technologie : les dimensions des transistors doivent être précisées explicitement par la valeur de W/L , que l'on choisit parmi un ensemble de valeurs préalablement fixées. Souvent, les circuits implantés symboliquement sont soumis à des programmes plus ou moins sophistiqués de compactage pour en accroître la densité d'intégration.

L'utilisation de l'implantation symbolique est d'autant plus appropriée qu'une grande rapidité de conception est requise et que la surface de silicium ou les performances ne sont pas des paramètres critiques (par exemple lors de certaines petites séries).

Plusieurs types de symbolismes, correspondant à des niveaux différents de représentation, ont été développés par les concepteurs, et peuvent être classés comme suit :

- . par site, sur grille;
- . par site et connexion, sur grille;
- . par site et connexion, sur grille virtuelle;
- . par site et connexion, sans grille (sticks);
- par porte logique, sur grille;
- par fonctions logiques complexes.

Remarquons que les grilles qui supportent ces symbolismes sont à quadrillage carré.

1 - 2. PRESENTATION DES SYMBOLISMES

1 - 2.1. Symbolisme par site, sur grille

Les différents symboles (contact, transistor, connexion, croisement...) utilisés pour décrire le circuit sont disposés sur une grille dont le pas est déterminé à partir des règles technologiques. Chaque symbole est associé à un ou plusieurs masques et (i) les symboles adjacents représentent des topologies connectées sur les niveaux communs de masque, (ii) les symboles non adjacents ne sont pas connectés, et les distances minimales devant séparer les topologies qu'ils représentent sont respectées.

Le nombre de symboles varie suivant la technologie, le degré de liberté souhaité pour l'implantation, et le type de traitement pour générer les masques. Concernant ce dernier point, deux cas peuvent être envisagés :

(i) le symbole permet directement le passage aux masques, car il contient toutes les informations nécessaires,

(ii) le symbole est associé aux symboles qui lui sont adjacents pour pouvoir générer les masques.

Le premier cas permet un traitement simplifié mais accroît sensiblement le nombre de symboles, ce qui a pour effet de compliquer la description de l'implantation.

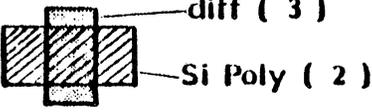
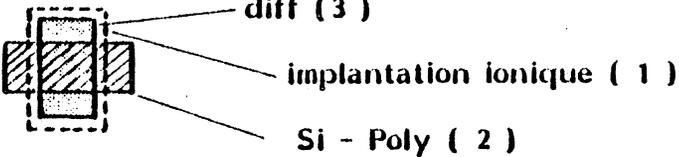
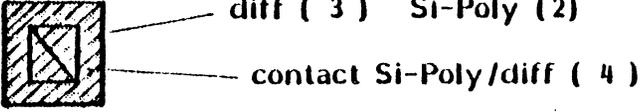
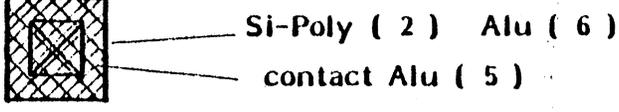
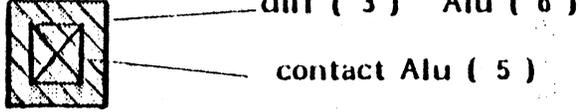
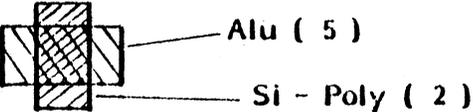
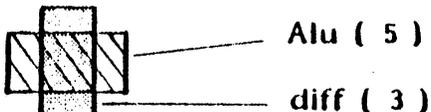
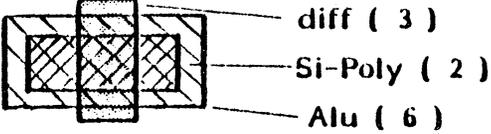
Des symbolismes par site sont décrits dans [Gi 76] pour une technologie P-MOS grille aluminium et dans [Lop 80] pour une technologie C-MOS grille Si-polycristallin.

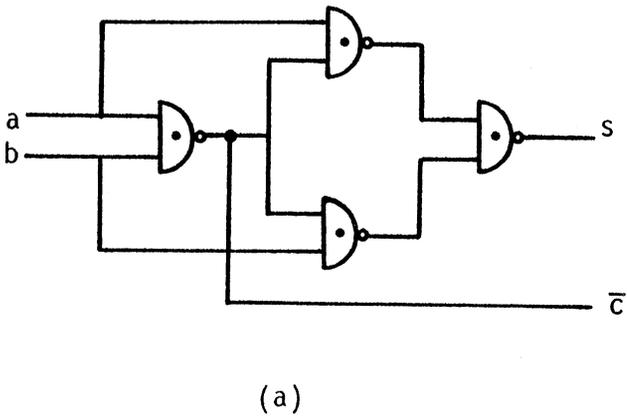
Nous nous proposons d'en définir un pour des circuits en technologie N-MOS grille Silicium Polycristallin (Si Poly) dont les règles de dessin au micron sont basées sur celles utilisées par Mead et Conway [ME 80]. Le pas de la grille d'implantation est alors fixé à 4λ .

Le processus technologique nécessite six masques différents :

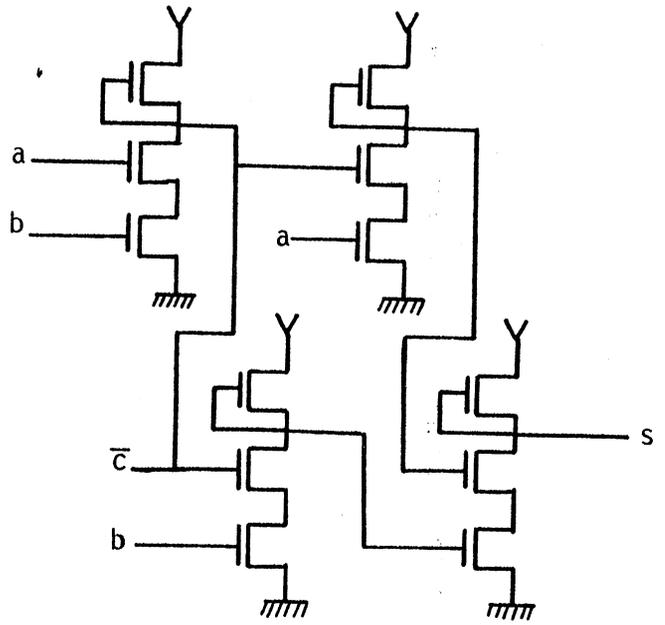
- 1) Implantation ionique
- 2) Silicium polycristallin
- 3) Diffusion
- 4) Contact diffusion - Si Poly (précontact)
- 5) Contact aluminium
- 6) Aluminium

Les symboles décrits figure 1 sont utilisés pour l'implantation d'une cellule demi-additionneur, dont les différentes étapes sont illustrées figure 2.

SYMBOLE	DEFINITION	GEOMETRIE à une rotation près (niveau de masque)
A	Aluminium	 (5)
S	Si polycristal- lin	 (2)
D	Diffusion	 (3)
T	Transistor	 diff (3) Si Poly (2)
I	Transistor déplété	 diff (3) implantation ionique (1) Si - Poly (2)
P	Pré-contact (Si-Poly /diff)	 diff (3) Si-Poly (2) contact Si-Poly /diff (4)
C	Contact Si-Poly /Alu	 Si-Poly (2) Alu (6) contact Alu (5)
O	Contact (Diff / Alu)	 diff (3) Alu (6) contact Alu (5)
#	Croisement (Si-Poly /Alu)	 Alu (5) Si - Poly (2)
+	Croisement (Alu/diff)	 Alu (5) diff (3)
∞	Croisement (grille Si-Poly / Alu)	 diff (3) Si-Poly (2) Alu (6)



(a)



(b)

		\bar{c} (retenu)										s (sortie)						
		S										S						
A	A	A	A	O	O	A	A	#	A	A	A	A	A	O	O	A	#	A
	S	S	S	T	T			S	S	S	S	S	T	T			S	
	S			D	D		S	S	S				D	D			S	
	S		S	S	T	T	S		S		S	S	T	T			S	
	S		S		D	D	D	P	S	S		S		D	D	D	P	
	S		S		S	I	S				S		S	I	S			
	S		S		S	I	S				S		S	I	S			
A	#	A	#	A	A	O	A	A	A	A	A	#	A	A	O	A	A	A
	S		S	S	S	I	S				S	S	S	I	S			
	S			P	S	I	S	S	S		P	S	I	S				
	S			D	D	D			S			D	D	D				
	S	S	S	S	T	T		S	S	S	S	S	T	T				
	S			D	D								D	D				
	S			T	T	S	S	S	S	S	S	S	T	T				
A	#	A	A	A	O	O	A	#	A	A	A	A	A	O	O	A	A	A
	S							S										S

b (entree)

a (entree)

(c)

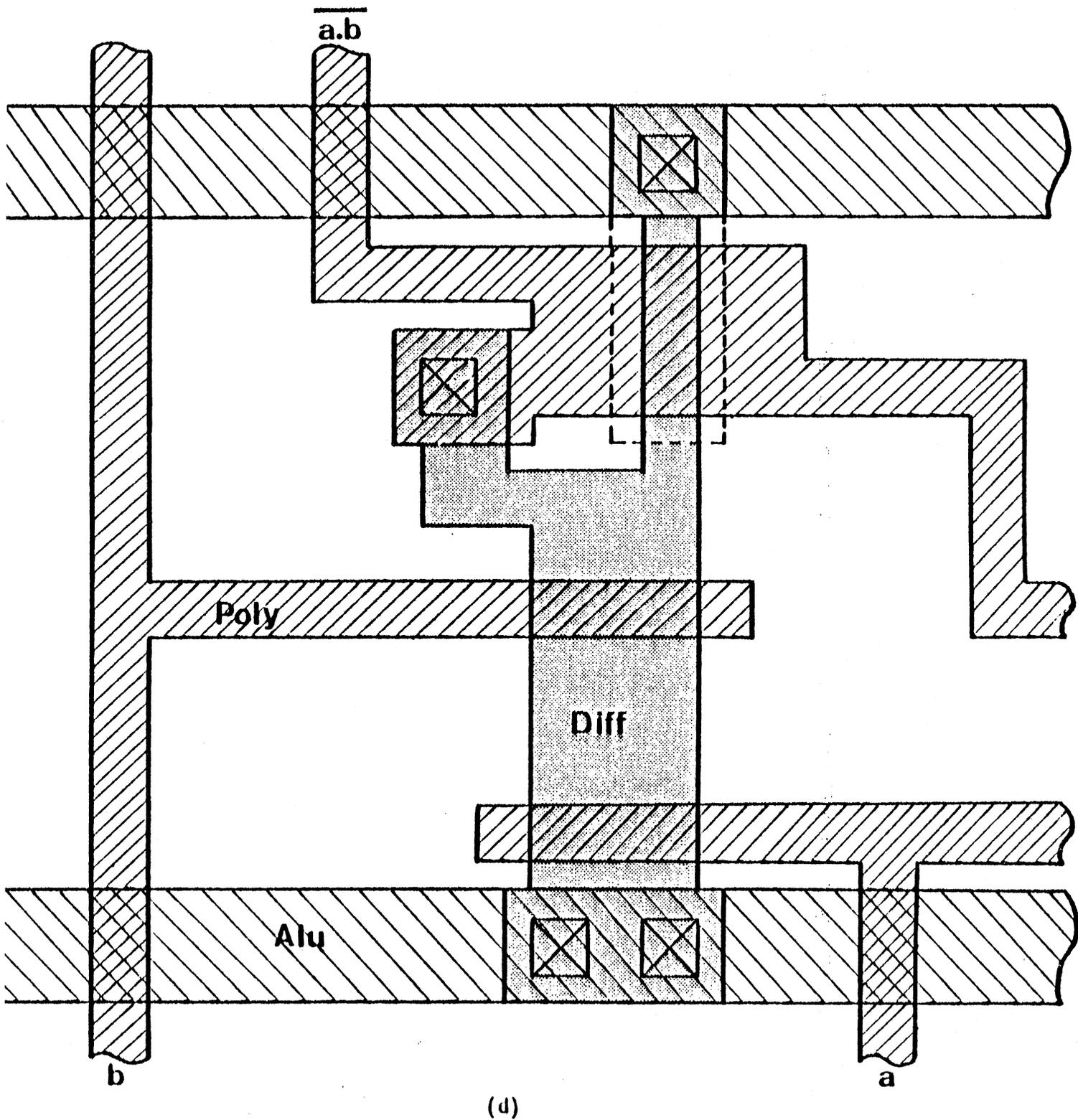


Figure 2. Demi-additionneur.

- (a) schema logique
- (b) schema transistor
- (c) schema symbolique
- (d) schema au micron (partie)

1 - 2.2. Symbolisme par site et connexion, sur grille

Le circuit est comme précédemment décrit sur une grille à l'aide de symboles correspondant à des sites (contact, transistor) et à des connexions (Aluminium, Si Poly, Diffusion) les reliant. Le croisement ou la superposition de deux connexions est représenté sans recourir à des symboles particuliers. Ce type de symbolisme ne nécessite donc pas de définir un nombre trop grand de symboles et est bien adapté à des technologies pour lesquelles le symbolisme par site exclusivement se révèle lourd à manipuler. Une application est actuellement développée pour une technologie C-MOS double Si Poly au Centre Norbert Segard à Grenoble (Figure 3). Les symboles ont été définis en couleur (Poly I en rouge, Poly II en vert, Alu en bleu,..) et la figure reflète mal les caractéristiques du circuit décrit symboliquement. Elle donne tout de même un aperçu de ce qu'est le symbolisme par site et connexion.

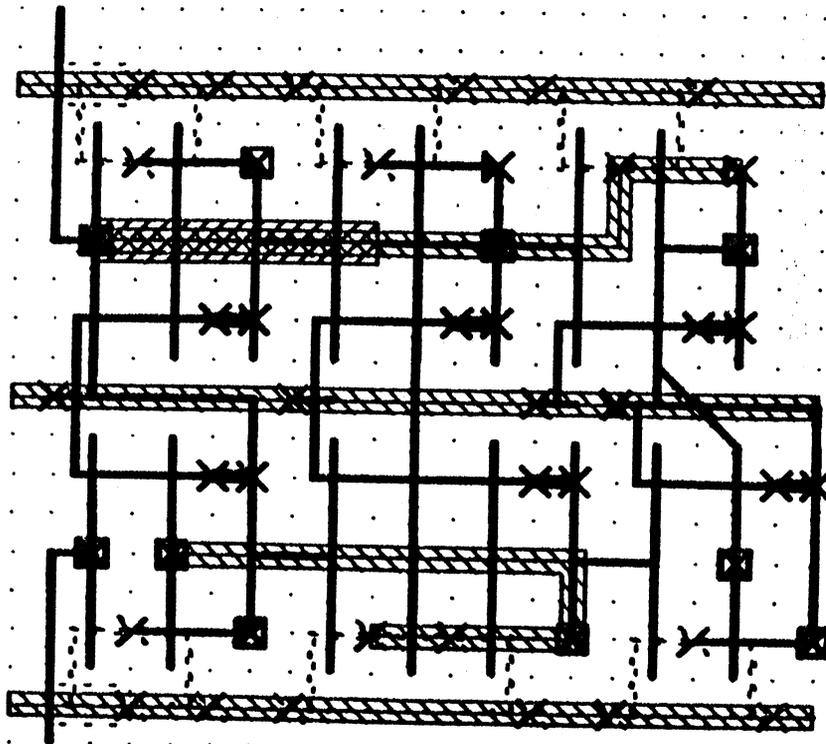


Figure 3. Exemple de symbolisme pour le C-MOS

1 - 2.3. Symbolisme par site et connexion, sur grille virtuelle

Comme précédemment, le concepteur effectue l'implantation symbolique sur une grille, mais son pas n'est pas fixé à priori, d'où l'appellation grille virtuelle. Un programme de compactage transforme alors le circuit représenté symboliquement en calculant au mieux les espacements entre les éléments voisins de nature différente, ce qui permet finalement d'obtenir des circuits dont la densité d'intégration est relativement élevée.

Le système MULGA [We 81] des Laboratoires de la Bell est basé sur ce principe et permet l'implantation de circuits en technologie N-MOS et C-MOS.

1 - 2.4. Symbolisme par site et connexion, sans grille

Le concepteur dessine son circuit à l'aide de segments représentant les connexions (Aluminium, Si Polycristallin, Diffusion) et de sites (contacts, implantation ionique). Les symboles sont placés relativement les uns par rapport aux autres et non de manière absolue. Les transistors sont définis implicitement aux croisements des lignes de diffusion et de Si Polycristallin. Le circuit se présente alors symboliquement sous la forme d'un ensemble de segments ou "sticks", dont une description détaillée est donnée par Mead et Conway [Me 80] pour la technologie N-MOS grille Si Poly. Les différents niveaux de masques sont ensuite obtenus par application de programmes de compactage qui tiennent compte des règles technologiques de dessin, tel que le système CABBAGE [Hs 80] développé par l'université de Berkeley.

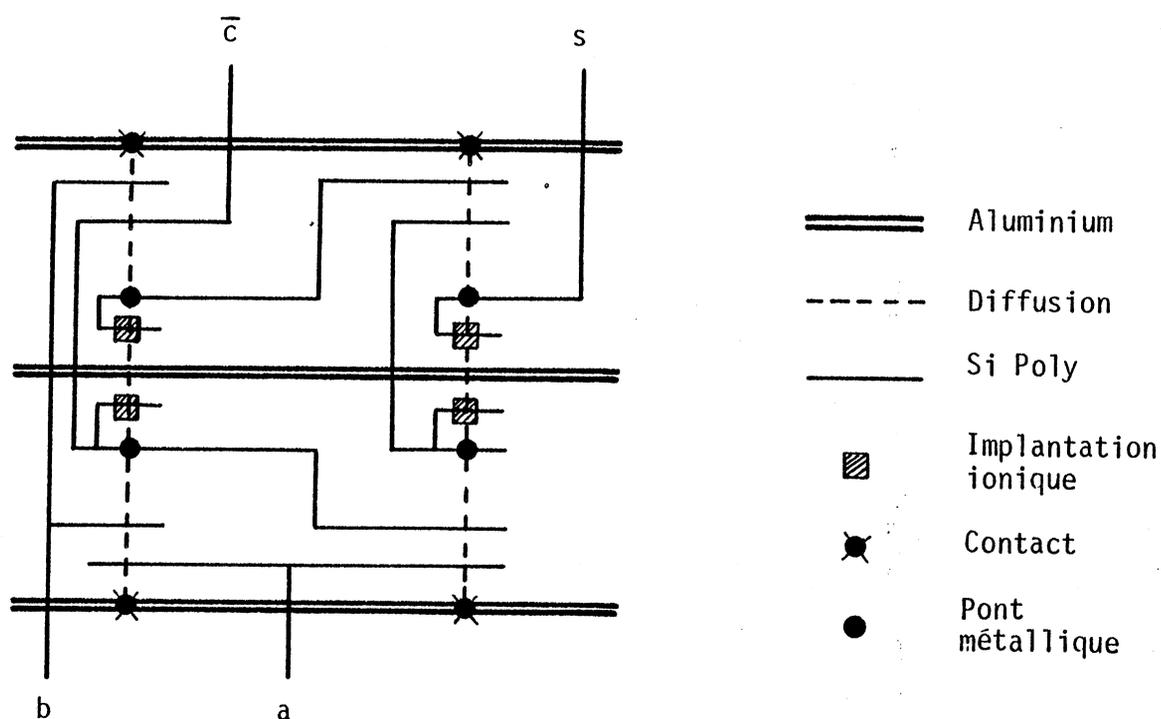


Figure 4. Demi additionneur

I - 2.5. Symbolisme par porte logique, sur grille

Le symbole de base est une porte logique, par exemple un inverseur pour le MD-MOS [Ma 78] ou l'I2L [Wi 77]. Les symboles sont disposés sur une grille et connectés entre eux par leurs points d'entrée/sortie. Ce type de symbolisme, le plus simple à assimiler pour l'utilisateur, est très proche du schéma logique si l'on prend, par exemple dans le cas du MD-MOS ou de l'I2L, la précaution de réaliser une synthèse en NAND du circuit. (Rappelons en effet, qu'en MD-MOS et en I2L, connecter deux sorties réalise un ET-câblé pour l'entrée suivante). Cela permet au concepteur de ne pas se préoccuper des détails fins liés à l'implantation : même la notion de transistor lui est, à ce niveau, transparente. De plus, ce symbolisme est le mieux adapté pour une implantation automatisée. Une étude approfondie du MD-MOS et de son application à l'implantation automatisée fait l'objet de la suite du travail.

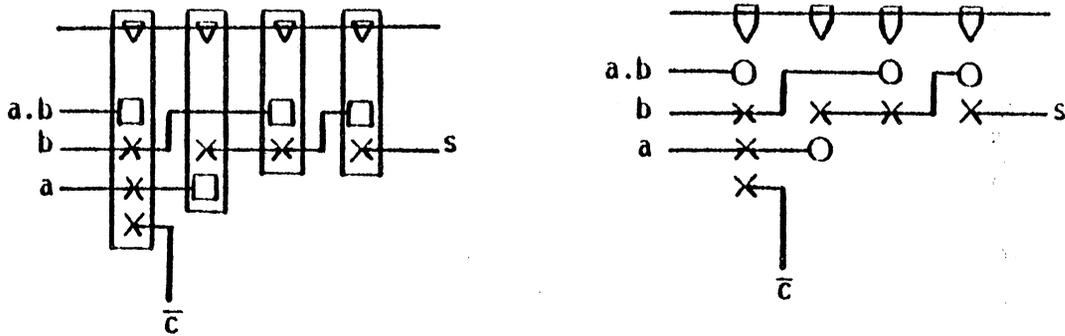
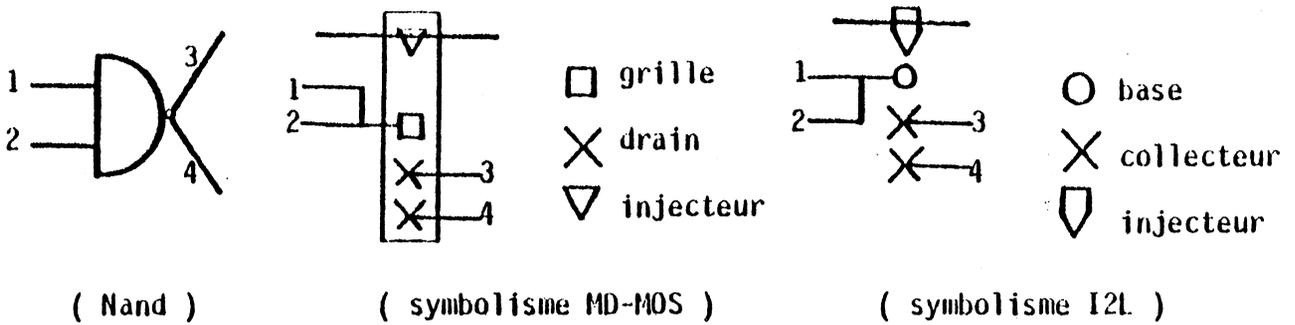


Figure 5. Demi additionneur (a) MD-MOS (b) I2L

1 - 2.6. Symbolisme par fonctions logiques complexes

Le symbolisme par fonctions logiques complexes est apparenté au symbolisme par connexion et site avec ou sans grille, avec la possibilité supplémentaire pour le concepteur d'utiliser des blocs précédemment implantés (symboliquement ou au micron), mémorisés dans une bibliothèque.

Le circuit est décrit à l'aide de ces blocs (enveloppe et points d'entrée/sortie du bloc) et des symboles de base définis pour la technologie; la rapidité de conception des masques est alors considérablement accrue. Des systèmes sont développés utilisant ce type de symbolisme et comprenant des programmes de compactage, parmi lesquels SLIM [Du 80] de Bell (implantation ou initiale sans grille) et TRICKY [Ha 81] de EFCIS (sur grille).

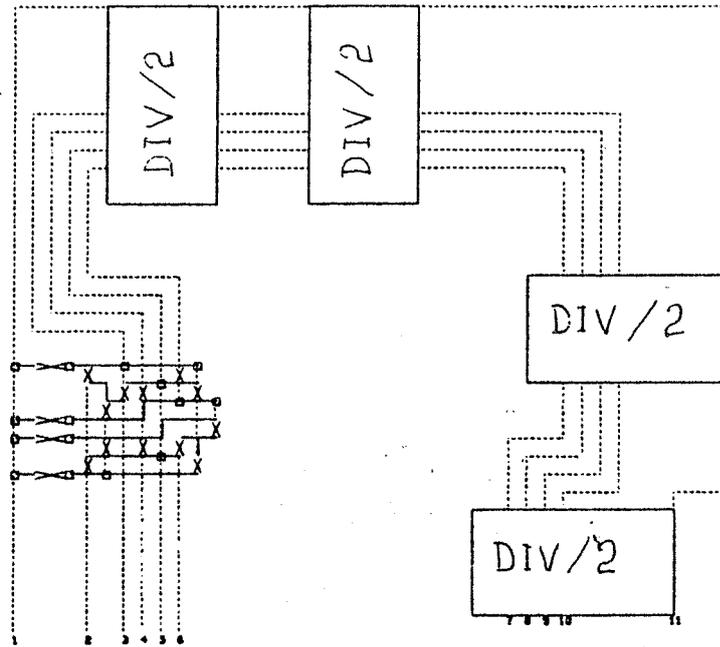


Figure 6.

I - 3. CONCLUSION

Dans le symbolisme par site, chronologiquement le plus ancien, la simplification porte essentiellement sur les règles d'implantation par rapport à celles au micron. Les symbolismes plus récents (par site et connexion sans grille ou "sticks") mettent en jeu des règles de dessin bien moins contraignantes et reflètent mieux la connectivité du circuit. Le concepteur peut alors se préoccuper plus efficacement de réaliser son circuit de manière correcte et prend en compte très tôt dans la conception l'aspect physique du circuit. De plus, les règles de dessin symbolique, peu liées à la technologie, assurent une meilleure indépendance au processus technologique et permettent une adaptation rapide à toute évolution, la modification des règles au micron influant peu ou prou sur les règles symboliques.

T A B L E A U R E C A P I T U L A T I F

SYMBOLISME	UTILISATION	TECHNOLOGIE	PROGRAMME DE COMPACTAGE
1	SLIC - Amer micro syst inc	P-MOS	non
	Bell Labs	C-MOS	non
2	Centre Norbert Ségard	C-MOS	non
3	Bell Labs	MOS - C-MOS	oui MULGA
4	Université de Berkeley	N-MOS	oui CABBAGE
	Caltech	N-MOS	oui
5	CNS	MD-MOS	non
	Siemens	12L	non
	Thomson Sescosem	12L	non
6	Bell Labs	N-MOS	oui SLIM
	EFCIS	N-MOS	oui TRICKY
	RCA	C-MOS	oui FLOSS

II - LA PORTE MD-MOS ET LES RÈGLES D'IMPLANTATION SYMBOLIQUE

II - 1. L'INVERSEUR EN TECHNOLOGIE MOS

Deux types de transistors MOS peuvent être distingués

MOS à *enrichissement* : ce transistor est bloqué lorsque la tension entre grille et source V_{GS} est nulle. Un canal de conduction est induit par la tension de grille ($|V_{GS}| > |V_{seuil}|$).

MOS à *déplétion* : ce transistor conduit même lorsque V_{GS} est nul. Il existe un canal préformé par dopage superficiel sous la grille.

Pour chacun de ces types, il existe des transistors canal N (conducteur par électrons) ou canal P (conducteur par porteurs positifs).

Un inverseur peut être réalisé, en technologie MOS, par l'association d'un transistor de signal à enrichissement et d'un transistor de charge à enrichissement (figure 7-a) ou à déplétion (figure 7-b).

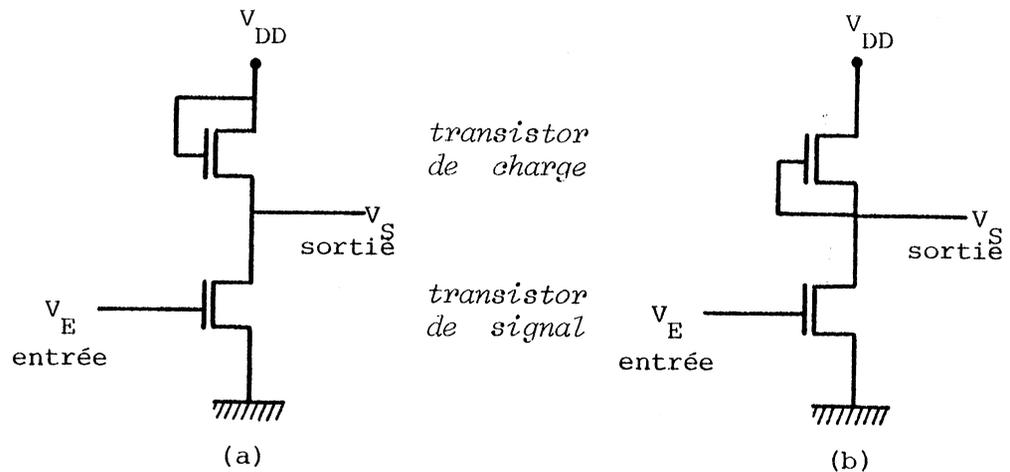


Figure 7. Inverseurs en technologie MOS

Fonctionnement de l'inverseur

Le transistor de charge est conducteur dans les deux cas.

- . Si l'entrée est au niveau haut (1 logique), le transistor de signal est alors conducteur et la tension de sortie est voisine de la masse (0 logique).
- . Si l'entrée est au niveau bas (0 logique), le transistor de signal est bloqué et la tension de sortie est voisine de V_{DD} (1 logique).

Deux raisons ont principalement justifié le choix de la technologie N-MOS pour réaliser la porte multi-drains MOS (MD-MOS) :

- utilisation de tensions d'alimentation basses (en général + 5 V).
- à densité d'intégration égale, la vitesse des transistors N-MOS est plus grande que celle des transistors P-MOS (la mobilité des électrons est supérieure à celle des porteurs positifs).

II - 2. PRINCIPE DE LA PORTE MD-MOS [Ma 78]

Considérons les deux inverseurs en cascade de la figure 8-a. Les deux transistors tels que le transistor de charge à déplétion du premier inverseur dont grille et source sont connectées à la grille du transistor de signal de l'inverseur suivant, constituent une porte MD-MOS, matérialisée en trait fort figure 8-a. L'entrée de la porte MD-MOS est alors $e_1 = \bar{a}$ et sa sortie est $s_1 = a$. Cette porte, prise isolément, se comporte comme un inverseur et le potentiel auquel est porté le drain dépend des caractéristiques de l'injecteur de courant que constitue le transistor de charge.

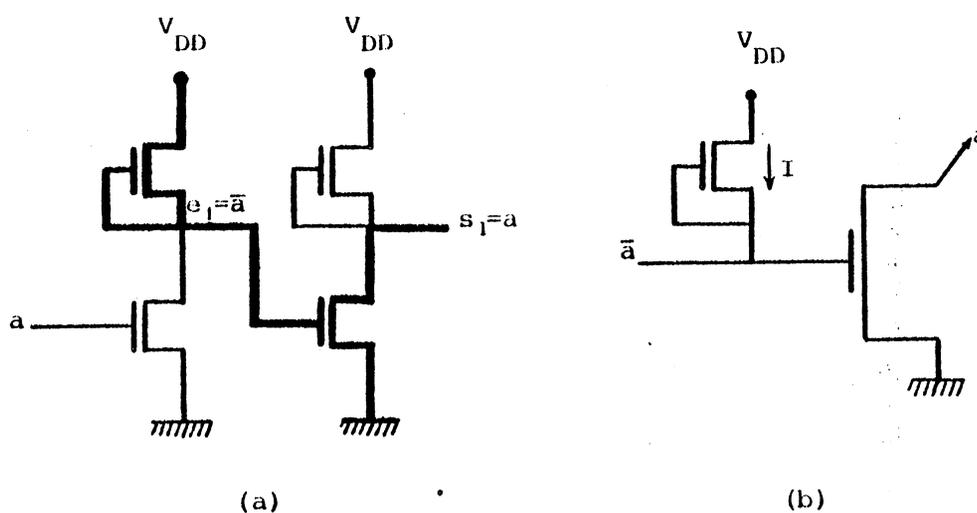


Figure 8. Structure de la porte MD-MOS (un seul drain) prise isolément

La figure 8-b présente le schéma électrique de cette porte MD-MOS dans le cas où elle ne comprend qu'un seul drain.

Une porte MD-MOS comportant deux drains est représentée en trait fort figure 9-a. Les grilles des transistors de signal sont toutes reliées au contact grille-source du transistor de charge à déplétion. Cet ensemble se comporte comme un inverseur; l'entrée vaut $e_1 = \bar{a}$ et les sorties valent $a_1 = a_2 = a$, si elles ne sont pas reliées à d'autres drains.

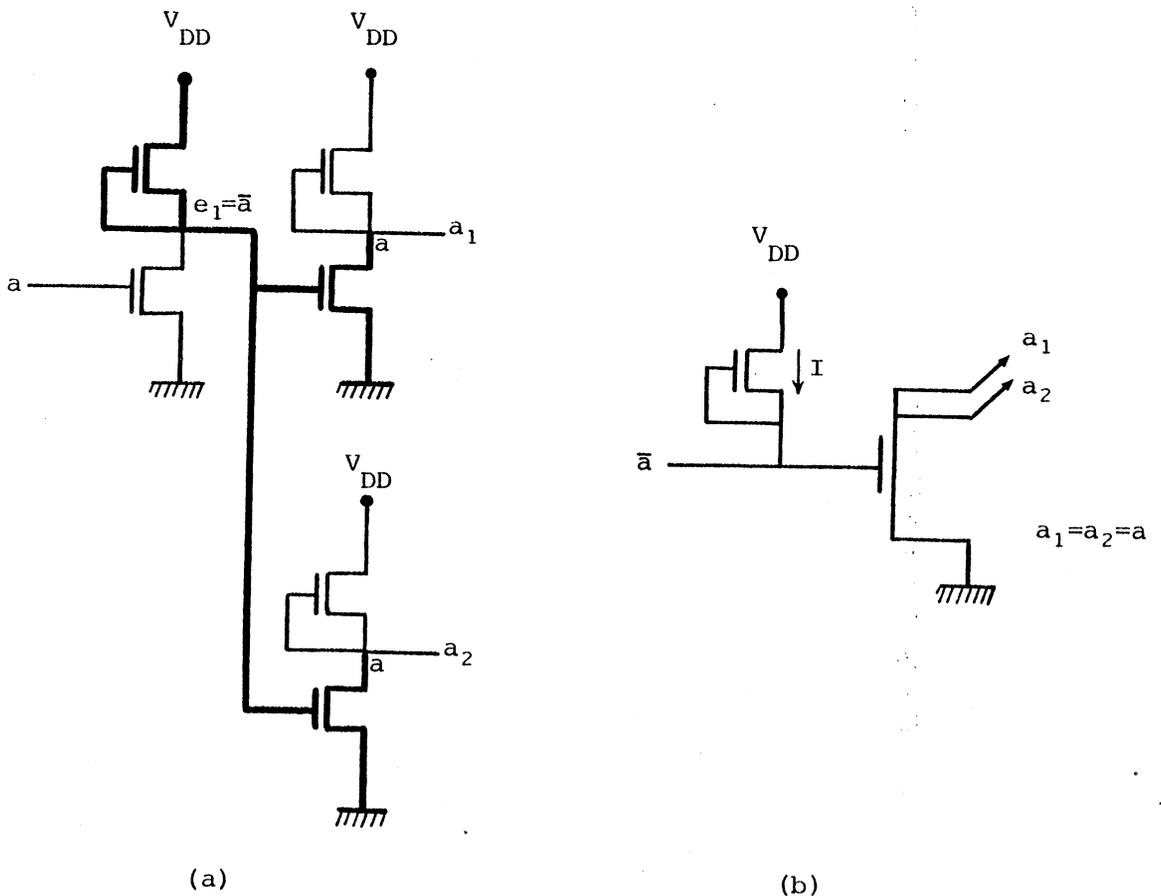


Figure 9. Structure de la porte MD-MOS (plusieurs drains) prise isolément

La figure 10-a montre comment la fonction non-ET est simplement réalisée à l'aide d'une seule porte MD-MOS pour laquelle la grille du transistor de signal est reliée à deux sorties (drains) de l'étage précédent.

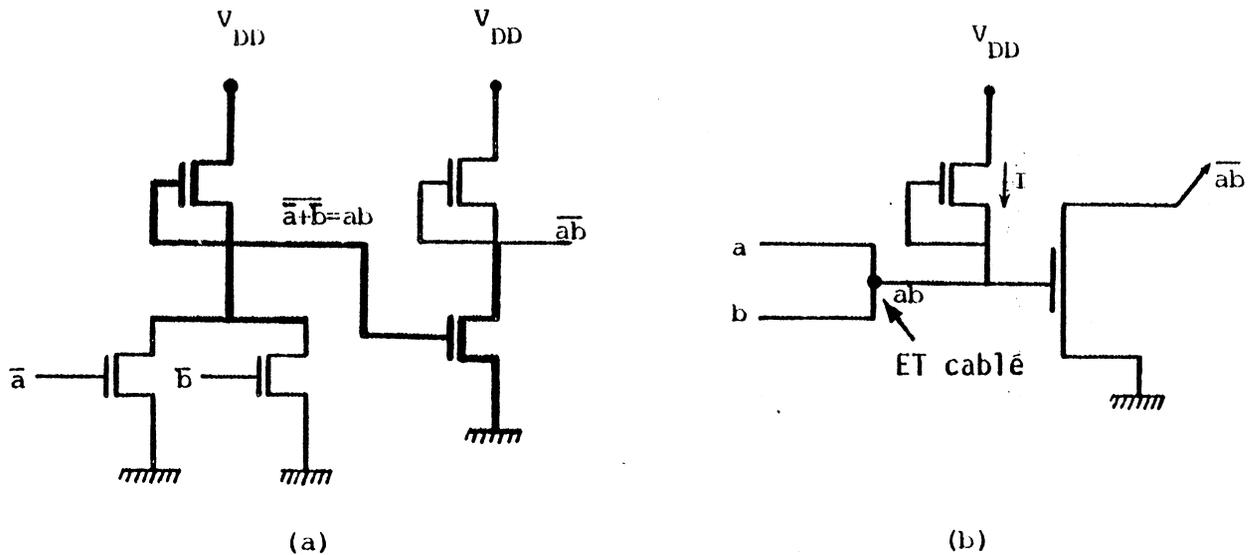


Figure 10 Réalisation d'un non-ET câblé à l'aide d'une porte MD-MOS

Sur les figures 8,9 et 10, les valeurs des sorties sont indiquées pour les cas où les portes considérées sont prises isolément. En effet, ces valeurs dépendent également des sorties d'autres portes auxquelles elles sont éventuellement connectées (à cause du ET câblé). Ainsi, une porte MD-MOS peut voir ses drains portés à des potentiels différents (figure 10').

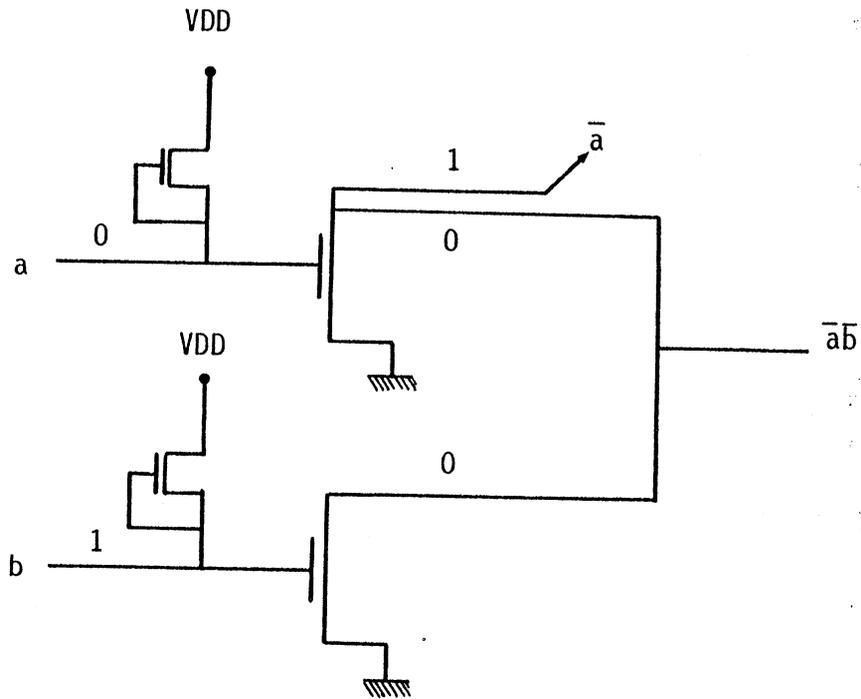


Figure 10'.

La représentation simplifiée des fonctions logiques de base (inverseur, NOR, NAND) à l'aide de portes MD-MOS est donnée figure 11.

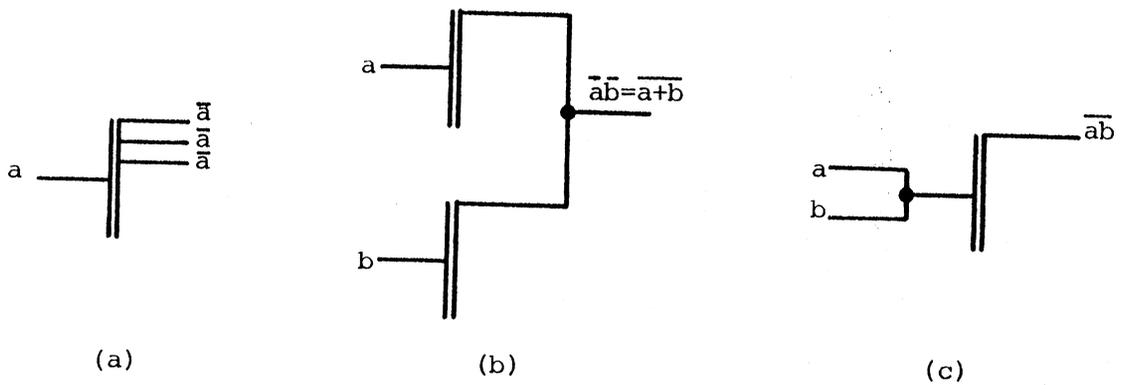


Figure 11 (a) Inverseur
 (b) NOR
 (c) NAND

II - 3. REALISATION TECHNOLOGIQUE

La zone grille (Si polycristallin) du transistor inverseur entourant chaque zone de drain est superposée aux canaux jointifs de grande largeur entre les zones de drain et la zone de source (Figure 12). Des portes logiques ayant des rapidités de commutation élevée et une densité d'intégration du transistor de signal voisine de la densité maximale de la technologie N-MOS utilisée sont ainsi réalisées. Dans un circuit MD-MOS, la géométrie des transistors de signal est fixée, et seules les dimensions du transistor de charge (injecteur) doivent être ajustées en fonction du temps de propagation à réaliser.

Lorsqu'une sortie d'une porte P attaque les grilles de plusieurs portes P_j , la taille du transistor de signal étant fixée à priori, le niveau zéro en sortie de P, ainsi que le temps de descente du signal sont dégradés. De plus, les différents signaux de sortie d'une porte peuvent être utilisés avec d'autres sorties et réaliser au moyen de ET câblés des fonctions logiques ne devant attaquer qu'une seule porte. Pour ces raisons, on utilisera autant de drains en sortie de P que de grilles des portes P_j à attaquer.

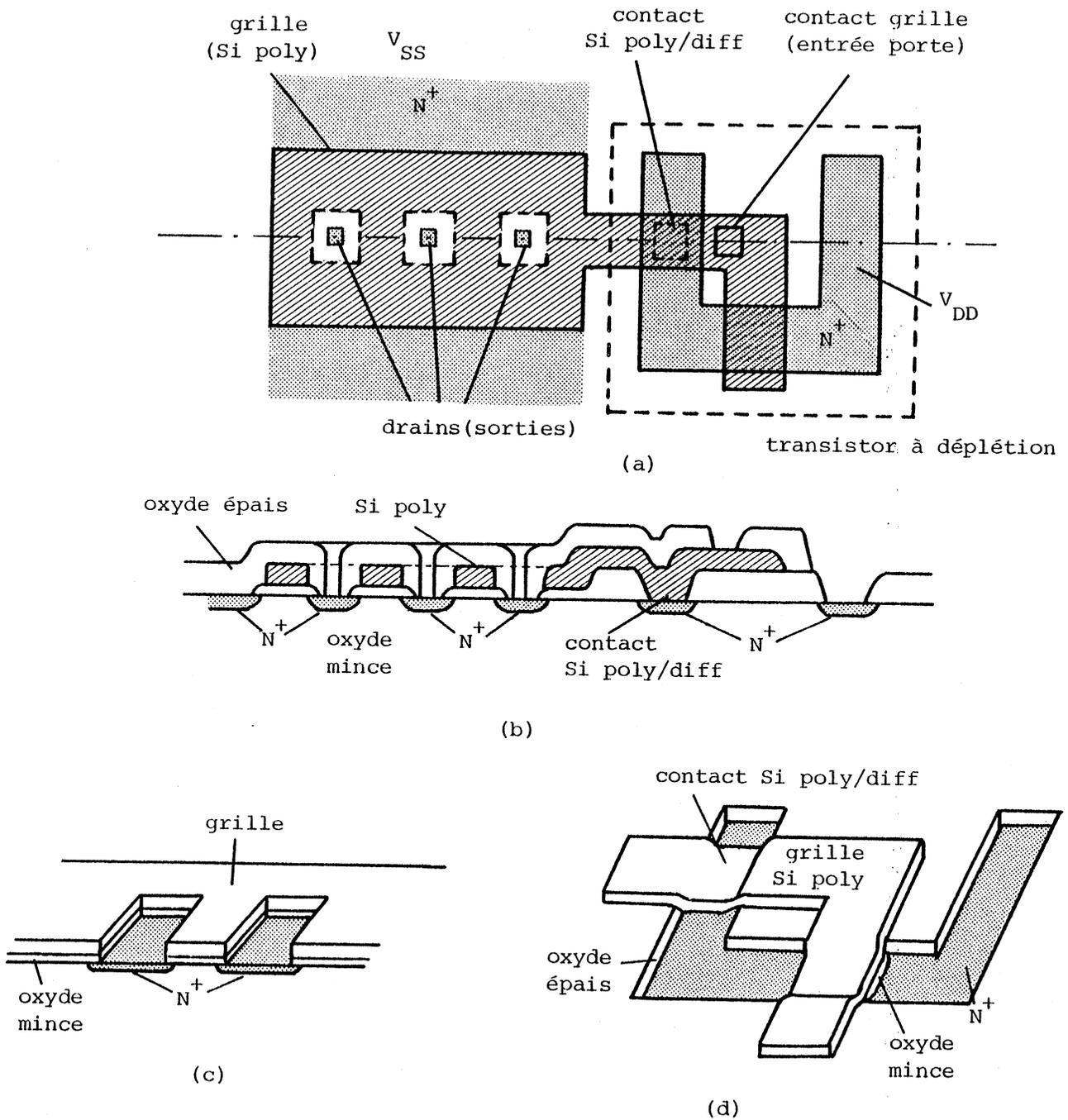


Figure 12. (a) et (b) Vue superficielle et coupe d'une porte MD-MOS

(c) Détail drains

(d) détail injecteur (MOS de charge)

II - 4. MODELE SYMBOLIQUE UTILISE POUR L'IMPLANTATION

La porte MD-MOS, de par sa structure graphique, est très bien adaptée à une représentation symbolique qui simplifie l'implantation des circuits MD-MOS (figure 13).

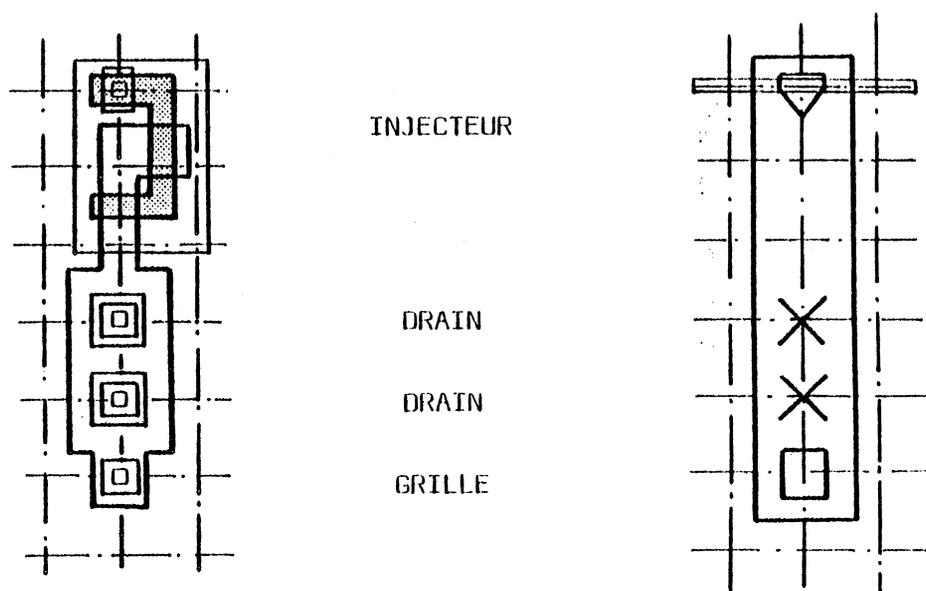


Figure 13. Symbolisme de la porte MD-MOS

Les éléments actifs (portes) et les éléments passifs (connexions en Alu ou connexions diffusées) sont disposés sur une grille rectangulaire de pas constant.

La porte MD-MOS est symbolisée par un rectangle de largeur fixe et de longueur variable, dépendant du nombre de points de contact et de connexions la traversant.

Longueur et contacts de la porte

- . La longueur de la porte influe sur ses performances et n'excèdera en général pas 12 pas de grille.
- . Une porte peut ou non présenter un injecteur, suivant qu'il y ait ou non un transistor de charge. Quand il existe, l'injecteur, à cause de son encombrement, impose au premier drain d'en être situé sur la porte à au moins 3 pas de grille. En revanche, on pourra disposer les drains sur n'importe quel site d'une porte démunie d'injecteur.
- . Il est possible de multiplier, sur une même porte, les points d'entrée (grille). La porte est en effet considérée comme une équipotentielle à la résistance série près, d'environ $50\Omega/\square$. Cependant, on assurera que la distance entre l'injecteur et la grille la plus éloignée soit inférieure ou égale à 8 pas de grille.

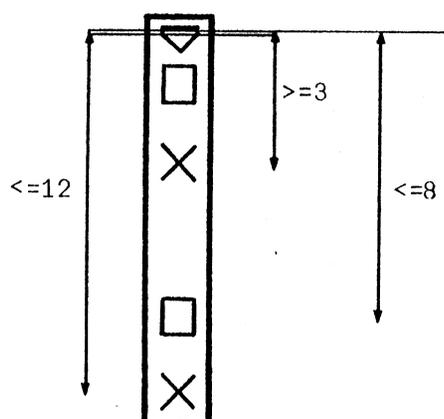


Figure 14. Position des contacts sur la porte MD-MOS

Les exemples de la figure 11 sont représentés symboliquement figure 15. Le ET-câblé (figure 15-c) peut être réalisé à l'aide de deux grilles sur la porte MD-MOS.

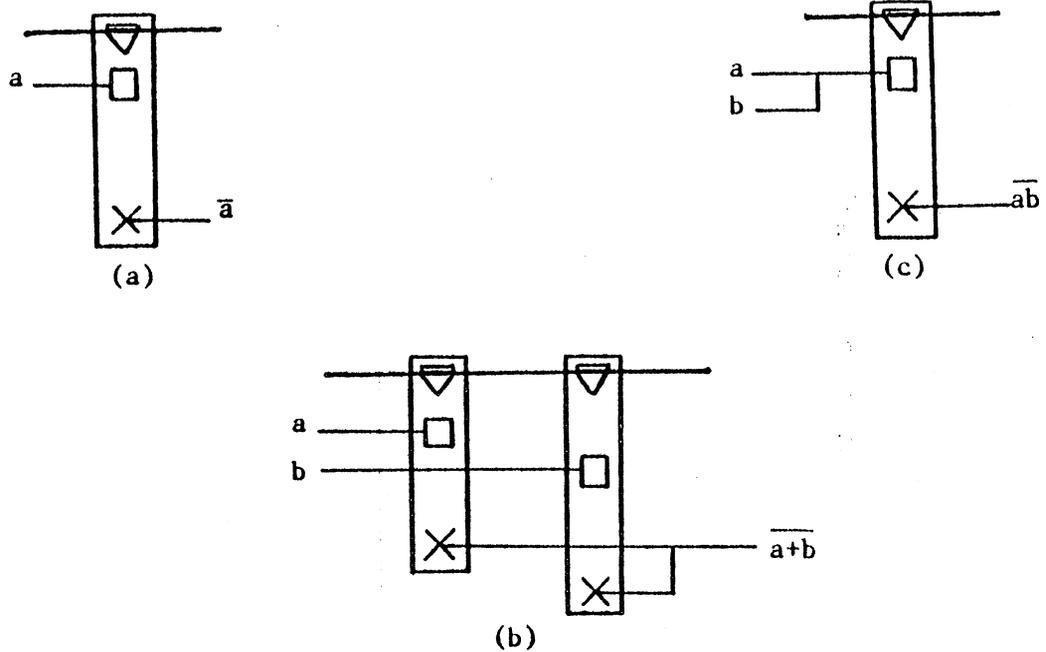


Figure 15. Inverseur, NOR et NAND en symbolisme MD-MOS

11 - 5. DIMENSIONNEMENT DU TRANSISTOR DE CHARGE

Considérons une porte MD-MOS dans un circuit. Les transistors de signal (drains) sont de taille fixe et imposent à la porte sa largeur. La largeur minimale W_c du canal du transistor de charge est fixée à $6\mu\text{m}$ pour une technologie N-MOS à déplétion en charge $6\mu\text{m}$. Dans la plupart des cas, seule la longueur L_c du canal de transistor de charge varie : elle est calculée en fonction du temps de propagation du signal désiré. Ce temps $t(\text{ns})$ est donné par

$$t = K \frac{C}{\gamma_c}$$

$$\text{où } \left\{ \begin{array}{l} C(\text{pF}) \text{ capacité à calculer} \\ K \quad \text{constante déduite des caractéristiques de la technologie} \\ \quad \quad \quad (\text{valeur moyenne } K = 100) \\ \gamma_C = W_C/L_C \end{array} \right.$$

La capacité C est la somme de trois quantités, soit :

- C_{con} , capacité de la connexion attaquant la porte considérée
($C_{\text{con}} = C_{\text{alu}} + C_{\text{diff}}$ où C_{alu} capacité de la portion Alu et C_{diff} capacité de la portion diffusion éventuelle),
- C_E , capacité de drain des sorties attaquant la porte considérée,
- C_S , capacité de grille des transistors de signal (drains de la porte considérée).

Le calcul de C_E et C_S est facile car les transistors de signaux sont de dimensions fixes.

Le concepteur doit donc simplement, lorsqu'il calibre un transistor de charge, calculer :

- N_{pas} , le nombre de pas de grille de la connexion attaquant la porte considérée,
- N_{DE} , le nombre de drains attaquant la porte considérée,
- N_{DS} , le nombre de drains de la porte considérée.

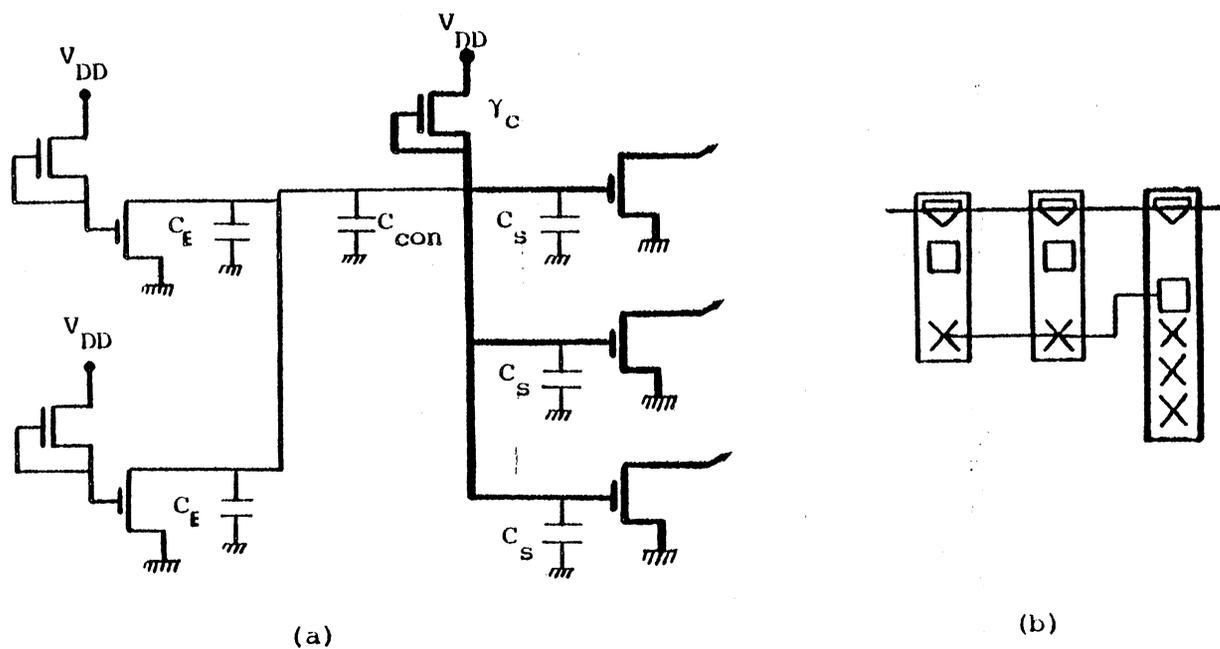


Figure 16. (a) Capacités à considérer
 (b) $N_{pas}=5$, $N_{DE}=2$, $N_{DS}=3$

Pour une technologie $6\mu\text{m}$ et un pas de grille de $21\mu\text{m}$, les valeurs nominales des capacités donnent :

$$\begin{aligned}
 C_{Alu/N_{pas}} &= 8 \cdot 10^{-3} \text{ pF} && (\text{largeur connexion égale à } 9\mu\text{m}) \\
 C_{diff/N_{pas}} &= 12 \cdot 10^{-3} \text{ pF} \\
 C_{E/N_{DE}} &= 3 \cdot 10^{-2} \text{ pF} \\
 C_{S/N_{DS}} &= 9 \cdot 10^{-2} \text{ pF}
 \end{aligned}$$

Exemple

Calcul de la dimension du transistor de charge de la porte (en trait fort) de la figure 16.

$$C_{\text{CON}} = 4 \cdot 10^{-2} \text{ pF}$$

$$C_{\text{E}} = 6 \cdot 10^{-2} \text{ pF}$$

$$C_{\text{S}} = 27 \cdot 10^{-2} \text{ pF}$$

$$\text{Soit } C = 37 \cdot 10^{-2} \text{ pF}$$

$$\gamma_{\text{C}} = \frac{100 \cdot C}{t} = \frac{37}{t}$$

Pour un temps de propagation $t = 60 \text{ ns}$, on obtient

$$\gamma_{\text{C}} = \frac{W_{\text{C}}}{L_{\text{C}}} - \frac{1}{2} \text{ soit } \begin{array}{l} W_{\text{C}} = 6 \mu\text{m} \\ L_{\text{C}} = 12 \mu\text{m} \end{array}$$

Par conséquent, outre la facilité d'implantation des circuits à l'aide du symbolisme MD-MOS, le calibrage des transistors de charge est une tâche aisée au vu du schéma d'interconnexion des portes.

Dans la pratique, les dimensions moyennes du transistor de charge sont calculées en fonction des performances désirées. Les dimensions peuvent, après implantation, être ajustées pour les parties critiques du circuit (connexions longues, portes ayant un grand nombre d'entrées ou de sorties...).

II - 6. CELLULE DE PORTES MD-MOS

Une cellule est constituée d'un ensemble de portes MD-MOS connectées entre elles, c'est-à-dire d'un ensemble d'éléments actifs et passifs. Deux pas de grille au moins doivent séparer les éléments actifs et les connexions diffusées. Cette séparation est réduite à un pas de grille entre connexions Alu, entre éléments actifs et connexions Alu, ou entre éléments passifs et connexions diffusées. Les connexions Alu sont tracées sur un seul niveau. Des passages diffusés de connexions permettent de résoudre les cas de croisements inévitables. Cependant, un passage diffusé est plus encombrant qu'une connexion Alu (à cause des contacts alu/diff), et induit une capacité parasite plus grande (1 pas de grille d'une connexion diffusée et 1.5 pas de grille d'une connexion Alu induisent la même capacité). Il faudra donc réaliser un compromis entre la surface et les performances souhaitées.

Dans le but de ne pas compliquer outre mesure le problème déjà difficile du placement et de l'interconnexion des portes, nous nous limiterons à des structures implantées simples comme la cellule à une ligne et la cellule à deux lignes qui sont par ailleurs très largement utilisées pour l'implantation manuelle.

II - 6.1. Cellule à 1 ligne

Les portes sont disposées tous les 2 pas de grille le long d'une alimentation rectiligne. L'espacement entre deux portes consécutives est augmenté si besoin est pour permettre le passage de connexions supplémentaires.

Nous nous intéresserons à des cellules de longueur inférieure à cent pas de grille, c'est-à-dire d'une cinquantaine de portes environ.

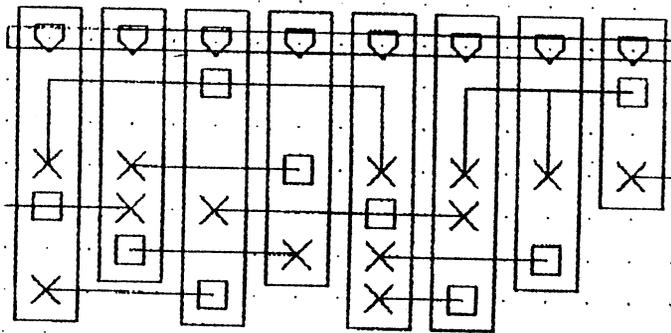


Figure 17. Cellule à une ligne

II - 6.2. Cellule à 2 lignes

La cellule à 2 lignes est constituée de deux cellules à 1 ligne disposées tête bêche, de manière à permettre le passage de connexions entre les deux. Elle comprend environ cent portes.

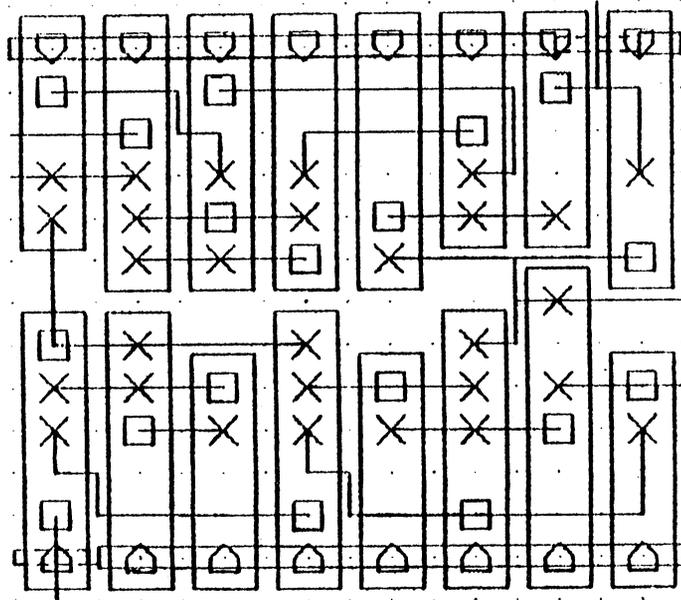


Figure 18. Cellule à deux lignes

Dans le cas de cellules adjacentes, une ligne d'alimentation est utilisée pour 2 rangées de portes. Certaines portions des lignes d'alimentation sont diffusées et permettent le passage de connexions entre les cellules (à travers l'alimentation). Ces diffusions ne sont pénalisantes ni en surface, ni en performances : il ne sera donc pas nécessaire d'en minimiser le nombre.

La figure 19 montre l'organisation des cellules entre elles.

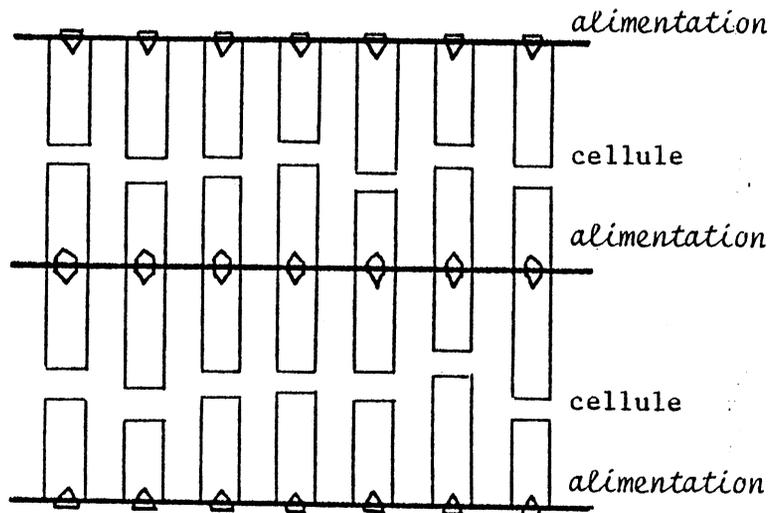


Figure 19. Cellules adjacentes

A partir d'une description logique du circuit, le schéma de portes MD-MOS est déduit et permet le passage à la réalisation du schéma implanté sous forme symbolique puis au dessins des différents masques technologiques.

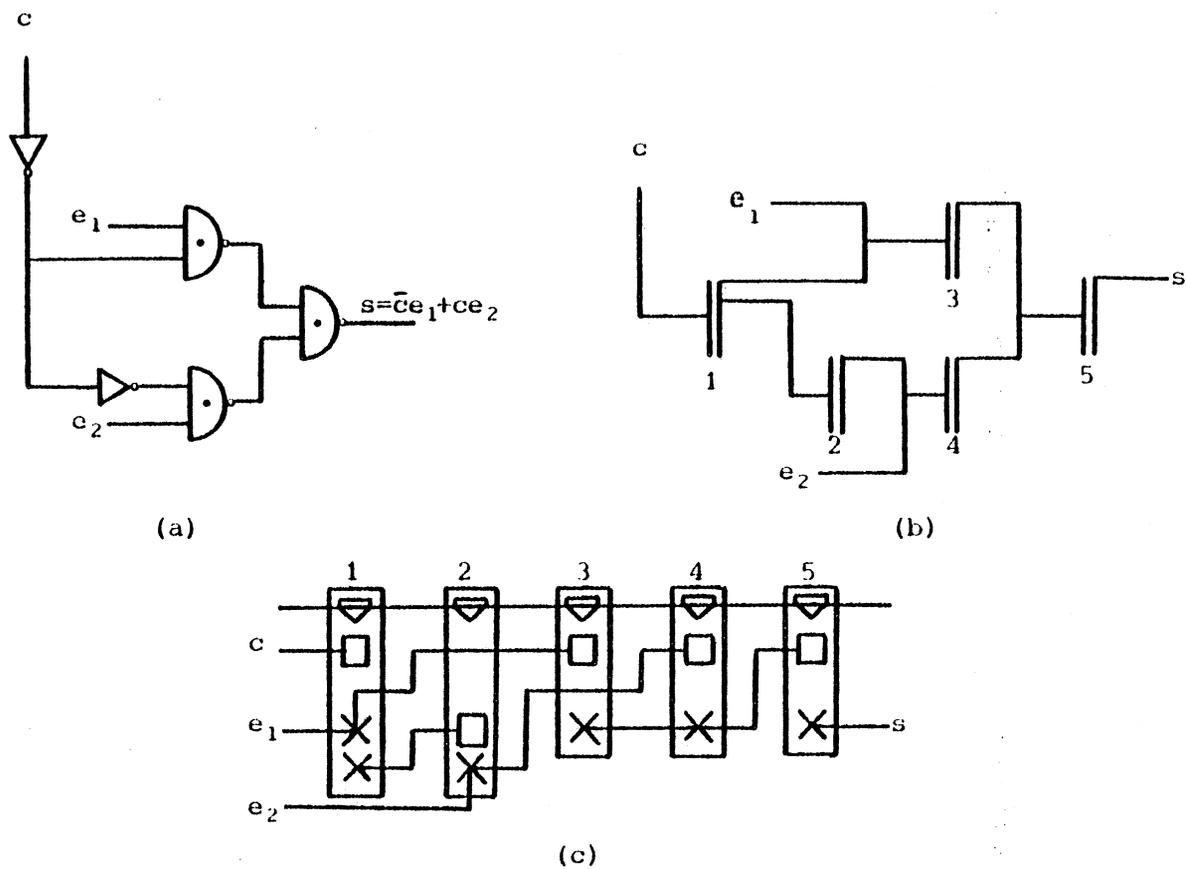


Figure 20. Multiplexeur 2 voies vers 1

(a) schéma logique (inverseurs et Nand)

(b) schéma MD-MOS équivalent

(c) schéma implanté sous forme symbolique

CHAPITRE III

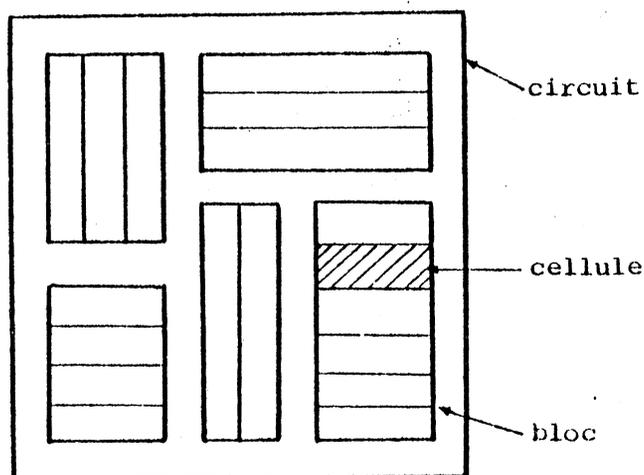
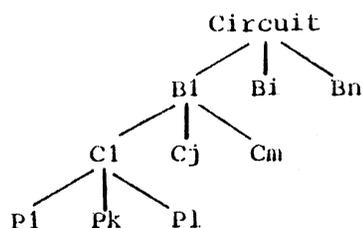
A. OASIS

OUTIL AUTOMATISE DE SYSTÈMES
EN IMPLANTATION SYMBOLIQUE

Nous nous proposons dans ce paragraphe de définir la structure générale d'un outil permettant l'implantation hiérarchisée et automatisée de circuits décrits sous forme symbolique. Son utilisation offre au concepteur de nombreux avantages, parmi lesquels :

- (i) gain du temps de conception,
- (ii) représentation symbolique,
- (iii) possibilité d'étudier différentes solutions,
- (iv) optimisation itérative suivant plusieurs critères, impossible à l'aide de méthodes manuelles.

L'étude fonctionnelle permet généralement de décomposer le circuit en blocs et les blocs en cellules. Ces blocs correspondent à des partitions du circuit formées d'éléments fortement connectés, les blocs étant peu connectés les uns aux autres. Les cellules sont déterminées de manière analogue dans les blocs. La décomposition est ainsi réalisée de manière à obtenir à chaque niveau un nombre moyen d'éléments; le circuit est alors défini à trois niveaux : bloc, cellule, porte.



La décomposition n'est en fait pas limitée, et un circuit très complexe pourra être décomposé en autant de niveaux nécessaires à une implantation facile. Le nombre de niveaux étant fixé pour une application, deux démarches multiniveaux sont possibles.

Démarche descendante

La taille et les dimensions des blocs (rectangulaires) sont estimées et un pré-placement des blocs est effectué. Le tracé des connexions entre les blocs définit alors les points d'entrée/sortie sur chaque bloc. Les cellules sont ensuite implantées finement (placement puis interconnexion), les points d'entrée/sortie des connexions sur les cellules étant fixés par le niveau précédent. Une remise en cause du placement et de l'interconnexion entre les blocs termine l'implantation du circuit.

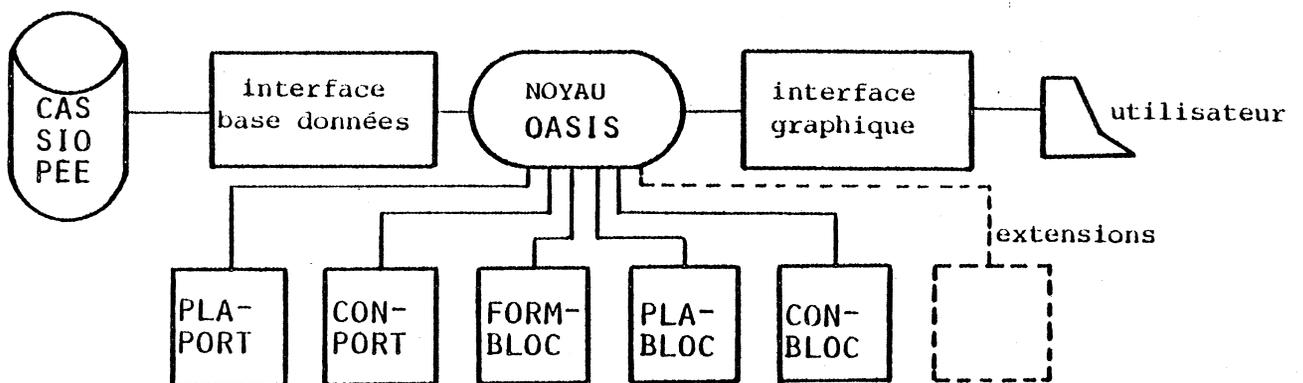
Démarche ascendante

Le placement et l'interconnexion des portes sont réalisés à l'intérieur des cellules, ce qui fixe les points d'entrée/sortie des connexions sur les cellules. Les blocs, formés de ces cellules, sont ensuite définis géométriquement (taille, dimensions), puis implantés, les points d'entrée/sortie de chaque bloc étant fixés par l'étape précédente.

Le choix entre ces deux démarches peut difficilement être fait à priori et dépend de l'application à traiter. La première démarche permet un traitement souple des blocs, la deuxième démarche un traitement souple des cellules. Cette souplesse dépend du fait que les

points d'entrée/sortie des connexions sur les blocs ou les cellules sont fixés ou non durant l'étape de placement et de tracé des connexions.

En fonction des paramètres d'entrée donnés par le concepteur, OASIS permet l'une ou l'autre de ces deux démarches.



Le "noyau" de OASIS permet

- (i) d'interpréter les commandes de l'utilisateur et ainsi générer la séquence d'opérations souhaitées dans OASIS;
- (ii) les communications entre les modules logiciels développés pour l'implantation, la base de données de CASSIOPEE à travers son interface et l'utilisateur;
- (iii) l'acquisition des données à traiter, à partir de l'utilisation directement ou de la base de données.

CASSIOPEE est le système de CAO intégré pour la conception de circuits que développe le centre Norbert Segard à Grenoble, et dont OASIS est un des éléments.

PLAPORT et CONPORT réalisent le placement et l'interconnexion automatisés des portes dans les cellules à partir d'une description logique (en NAND), la cellule implantée obtenue étant décrite à l'aide du symbolisme MD-MOS. Ces deux modules, fortement liés au MD-MOS, font l'objet dans la suite du chapitre, d'une étude détaillée, quant aux méthodes et algorithmes développés.

FORMBLOC construit les blocs à partir des cellules implantées, de sorte que deux cellules adjacentes partagent une même ligne d'alimentation. Les points d'entrée/sortie des cellules permettant le passage des connexions entre cellules sont fixés par PLAPORT et CONPORT ou le concepteur de manière à assurer la continuité des équipotentielles.

PLABLOC et CONBLOC réalisent le placement et l'interconnexion de blocs rectangulaires de tailles variables. Le travail concernant PLABLOC est présenté dans [Ma 82] et nous nous contenterons, dans le cadre de cette étude, d'en rappeler les grandes lignes.

A partir de leur description logique, PLABLOC génère un placement des blocs en deux étapes [Se 81]. Une solution initiale est tout d'abord construite de manière à définir les positions relatives des blocs les uns par rapport aux autres et à ce que le nombre de connexions courtes soit aussi grand que possible. Des transformations topologiques appliquées sur les blocs permettent ensuite d'améliorer le placement :

- (i) échange par paire de tous les blocs géométriquement interchangeables, une paire à la fois,
- (ii) rotation, quand elle est possible, de tous les blocs, un à la fois,
- (iii) symétrie, quand elle est possible, de tous les blocs, un à la fois.

Plusieurs critères sont améliorés lors de ces transformations :

- (i) surface du circuit (surface des canaux d'interconnexion),
- (ii) nombre de croisement des connexions,
- (iii) longueur des connexions,
- (iv) forme du circuit (on essaie en général d'approcher un carré).

CONBLOC, en cours de définition, réalise le tracé des connexions entre les blocs dans les canaux d'interconnexion déterminés lors du placement.

Nous nous intéressons donc, dans la suite de ce travail, aux méthodes d'implantation des portes MD-MOS organisées en cellules à une ligne et à deux lignes. Les points d'entrée/sortie des connexions sur les cellules sont ou non imposés à priori par le concepteur, suivant la démarche d'implantation appliquée.

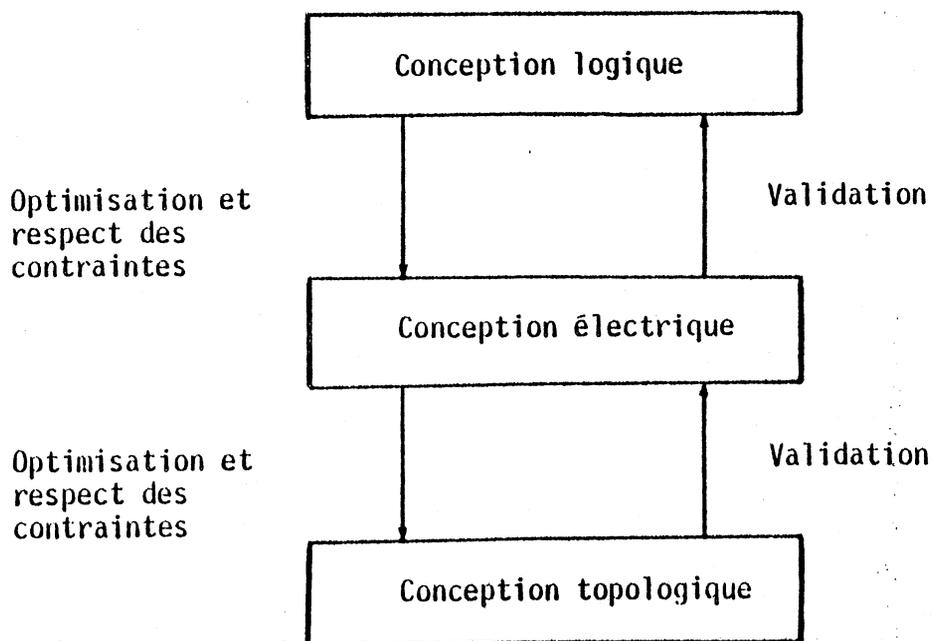
B. ETUDE DE CAS

METHODES DE PLACEMENT ET D'INTERCONNEXION AUTOMATISÉS
DE CIRCUITS MD-MOS DÉCRITS SYMBOLIQUEMENT

I - MÉTHODE GÉNÉRALE

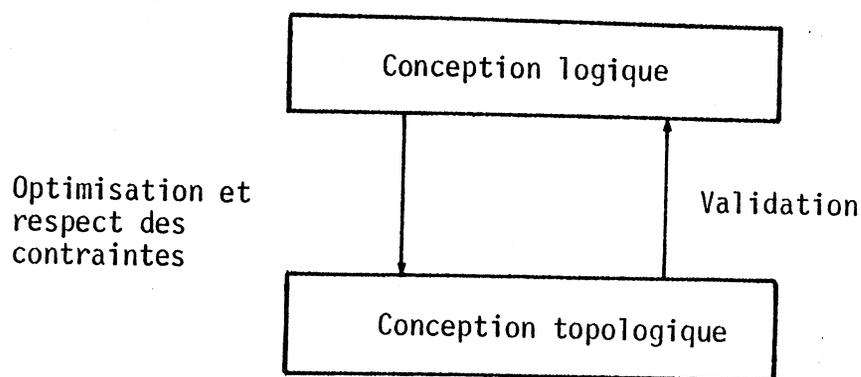
I - 1. OBJECTIFS ET METHODE GENERALE

Nous considérons des circuits ou portions de circuits MD-MOS représentés symboliquement et décrits par les interconnexions entre les portes. Nous proposons une méthode qui, à partir de la description logique, permet l'implantation automatique de tels circuits, c'est-à-dire réalisant automatiquement le placement des portes MD-MOS ainsi que le tracé des connexions. Rappelons que dans une approche de conception descendante classique, les étapes sont :



Dans notre approche par représentation symbolique, nous avons vu que la conception électrique (dimensionnement des transistors) n'interférerait pas sur l'implantation symbolique du circuit. En effet, les dimensions des transistors de charge varient dans les limites permises par la définition de l'enveloppe de la porte symbolique. L'ajustement de ces dimensions est réalisé de manière précise après que le schéma topologique ait été défini, c'est-à-dire après implantation symbolique du circuit.

La démarche de conception est donc simplifiée



Il s'agit donc de passer d'un schéma logique à un schéma topologique en s'assurant :

- (i) que l'on reste conforme aux spécifications logiques,
- (ii) que l'on respecte les contraintes de niveau inférieur,
- (iii) que l'on optimise des critères de coût.

Dans notre démarche, la conformité au niveau supérieur est assurée par le principe même de la conception descendante automatisée. Le passage au niveau topologique par implantation automatique est sûr, par définition; aucune erreur de description ne peut être introduite entre les deux étapes.

1 - 2. RESPECT DES CONTRAINTES

Il s'agit de respecter un certain nombre de règles introduites par l'étude électrique et topologique. Rappelons ici l'essentiel de ces règles, présentées dans le chapitre précédent.

- Structure du circuit hôte. Les portes sont disposées sur une grille carrée le long des lignes d'alimentation. Nous distinguerons des circuits ou cellules à une ligne ou deux lignes d'alimentation. L'espacement minimal entre deux portes est fixé à deux pas de grille. Les connexions sont tracées le long de la grille sur un seul niveau.
- Règles concernant la porte symbolique elle-même. Ces règles concernent la répartition des points de contact sur la porte, la longueur maximale de la porte. Rappelons la plus contraignante : les drains doivent être situés à au moins trois pas de grille de l'injecteur.

I - 3. OPTIMISATION

Une implantation, tout en respectant les contraintes, doit minimiser le coût global du circuit qui dépend essentiellement de sa surface et de la longueur des connexions. En fait, notre approche se décompose ici en deux étapes :

- le placement des portes influence, dans un premier temps, la longueur du circuit;
- le tracé des connexions influence la largeur du circuit; les portes sont plus ou moins longues suivant le tracé choisi des connexions.

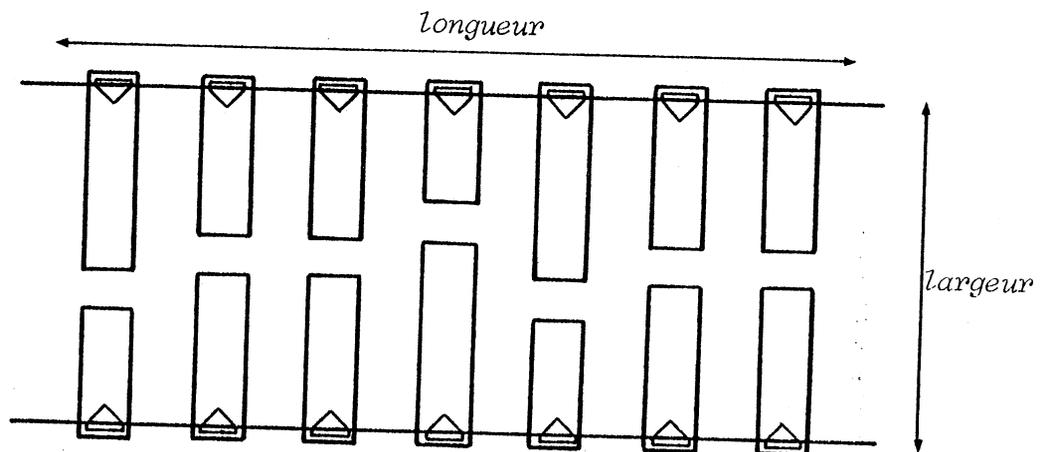


Figure 1.

Nous montrerons dans la suite comment la largeur du circuit peut être prise en compte dès l'étape de placement des portes.

1 - 4. ORGANISATION GENERALE

Globalement, le système d'implantation peut être vu comme le passage d'un niveau de description d'un circuit à un autre niveau, avec :

- respect des règles,
- minimisation de coût.

Les méthodes mises en oeuvre pour le placement sont des méthodes constructives fondées sur des coûts partiels, suivies d'améliorations itératives.

Pour le tracé, des classes de connexions sont définies et dans chaque classe, les connexions sont traitées séquentiellement.

L'organisation générale des différentes procédures utilisées pour l'implantation est donnée figure 2.

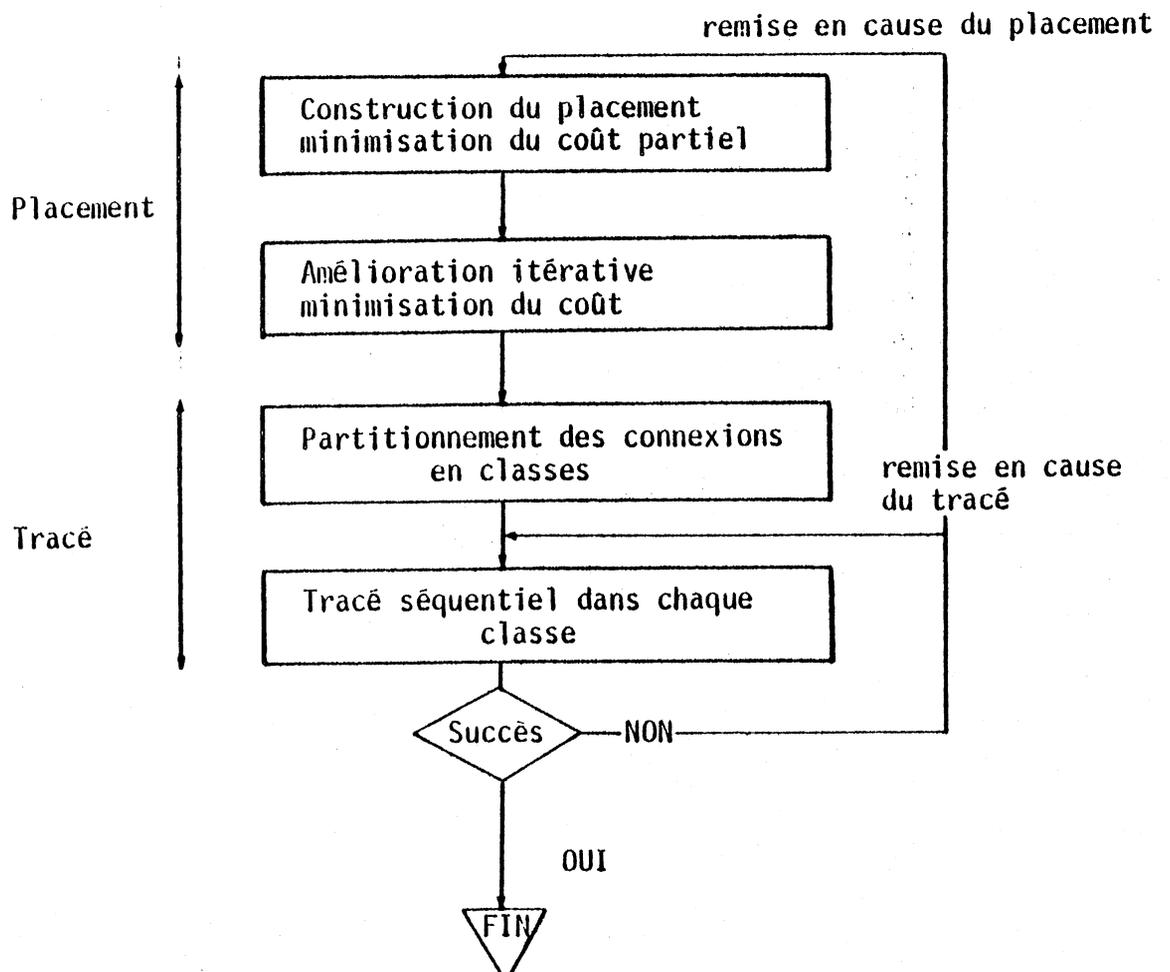


Figure 2. Organisation générale de l'implantation

Nous trouverons dans la suite du chapitre les méthodes proposées et les critères utilisés dans les procédures de placement et de tracé pour

- les cellules à une ligne d'alimentation,
- les cellules à deux lignes d'alimentation.

II - ÉTUDE DES CELLULES À UNE LIGNE D'ALIMENTATION

II - 1. ETUDE PRELIMINAIRE

II - 1.1. La fonction de PIC : définition

Soit un ensemble de segments parallèles à un axe (figure 3). Le pic en un point de l'axe est défini comme le nombre de segments traversés par une droite perpendiculaire passant par ce point. Ce pic définit alors une fonction (appelée fonction de pic) dont on note P_m la valeur du maximum.

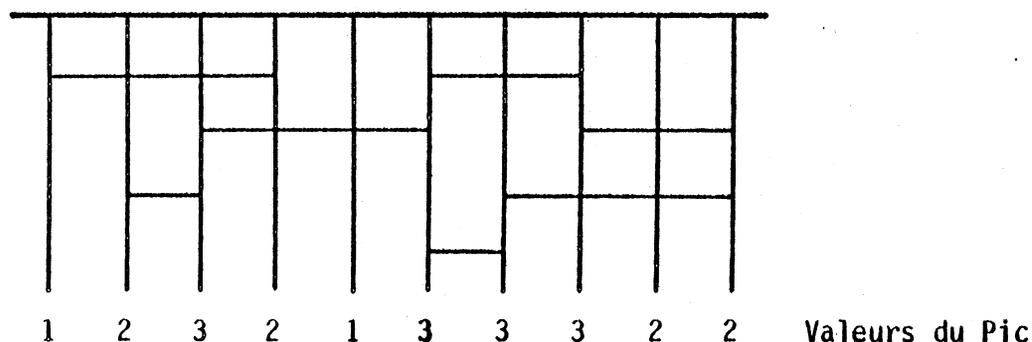


Figure 3. Fonction de pic associée à un ensemble de segments

II - 1.2. Intérêt du pic

Les dimensions minimales d'une cellule à une ligne peuvent facilement être estimées. La longueur minimum L_m est atteinte lorsque les portes sont disposées le long de la ligne d'alimentation et séparées les unes des autres par seulement deux pas de grille. L_m dépend donc du nombre de portes (n) dans la cellule et on a $L_m = 2n$ pas de grille.

La technique MD-MOS ne nécessite pas de fixer à priori les positions exactes des contacts grilles et drains sur la porte. Nous pouvons donc les fixer de telle manière que les connexions soient considérées comme des segments horizontaux (parallèles à la ligne d'alimentation). La largeur minimum l_m occupée par les segments est donc P_{m+1} , P_m étant le maximum de la fonction de pic calculée en tout point de grille de la ligne d'alimentation.

Les algorithmes d'implantation devront réaliser un circuit de dimensions aussi proches que possible de L_m et l_m . De plus, L_m étant fixé par le nombre de portes, il est important de construire une cellule dont l_m , c'est-à-dire P_m est le plus petit possible.

Propriété : La fonction de pic, et par conséquent P_m , dépendent, pour une cellule donnée, du placement des portes.

Démonstration : La disposition des portes détermine les positions des extrémités des connexions et donc leur longueur. Pour un placement donné, la fonction de pic est alors parfaitement définie. Une autre répartition des portes modifie la longueur de connexions et donc la fonction de pic associée.

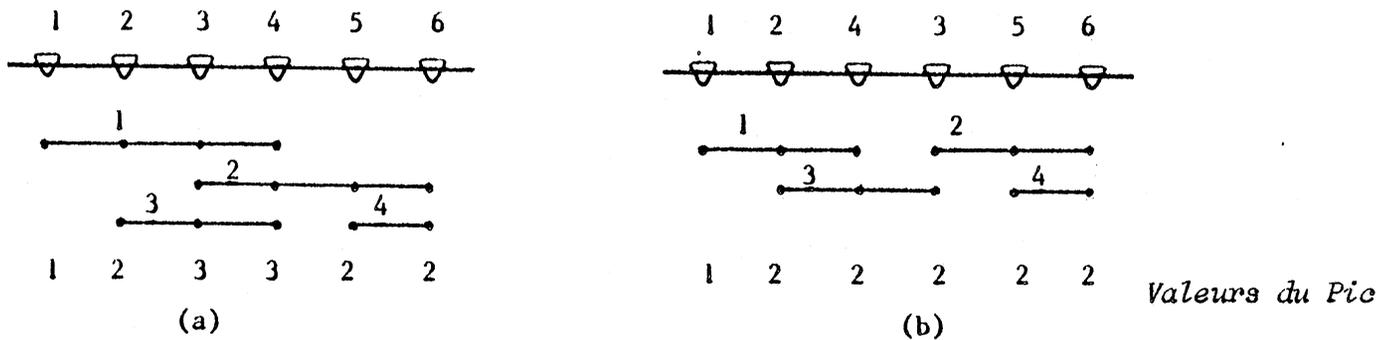


Figure 4. (a) $P_m = 3$
 (b) $P_m = 2$ après permutation des portes 3 et 4

Il est par conséquent important de minimiser P_m lors de l'étape de placement des portes, ce qui permet de minimiser la surface de la cellule. Cette fonction de pic sera également utilisée lors du tracé des connexions entre les portes.

II - 1.3. Algorithme de répartition des segments

Il s'agit de répartir un ensemble de segments parallèles, sur un nombre de lignes le plus petit possible. Les segments ne doivent pas se recouvrir. Pour cela, nous proposons l'algorithme RS suivant : les lignes sont remplies par les segments de gauche à droite, et ce l'une après l'autre à partir de la première. Pour le remplissage d'une ligne, choisir la connexion la plus à gauche (en cas de conflits, une des connexions possibles est choisie de manière aléatoire). Puis, choisir le segment le plus à gauche à partir de l'extrémité droite du segment précédemment choisi sur la ligne. Ceci est répété jusqu'à ce qu'aucun segment ne puisse être placé sur la ligne. La figure 5 montre les résultats obtenus par l'algorithme RS à l'exemple de la figure 3.

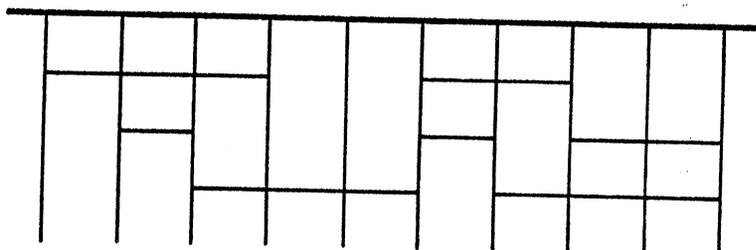


Figure 5. Répartition des segments de la figure 3 après RS

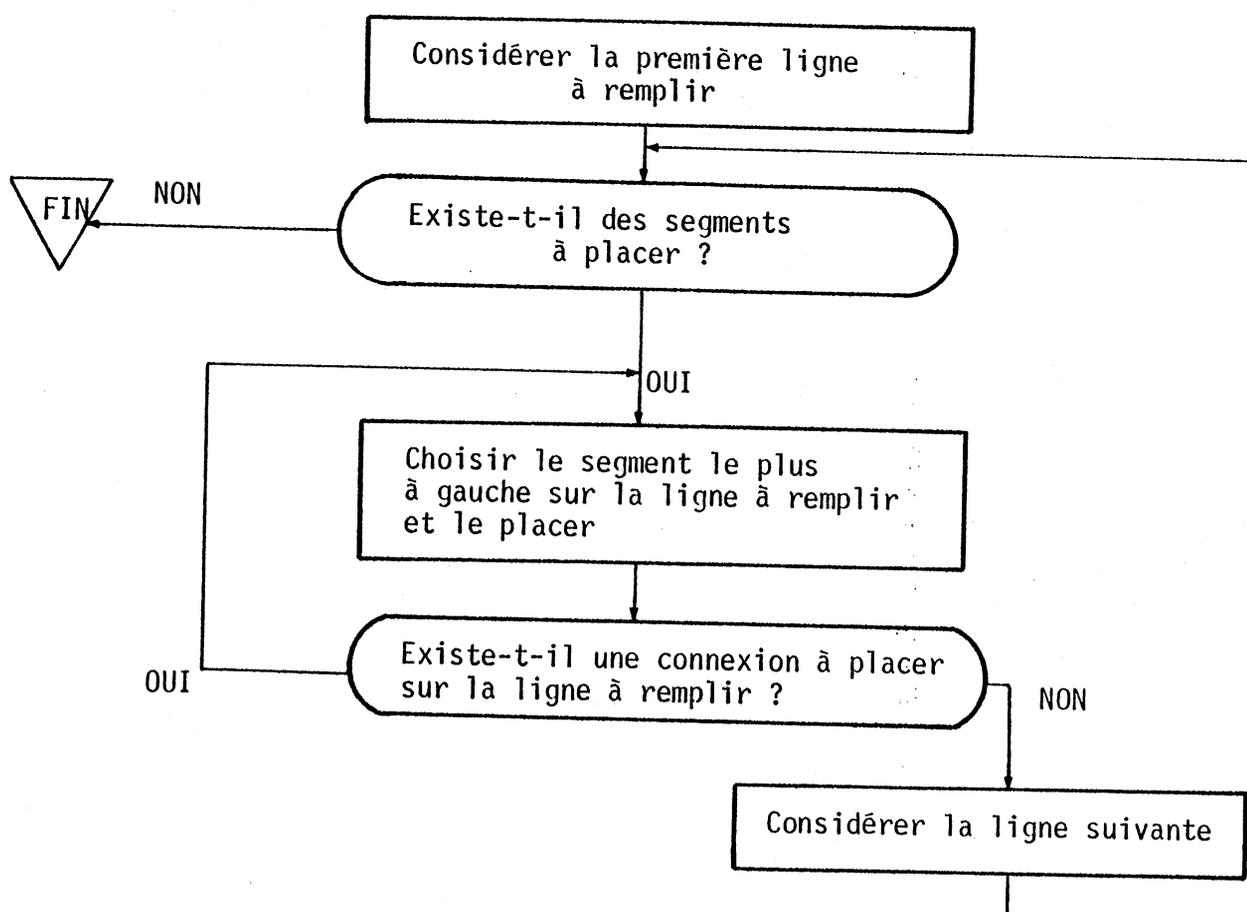


Figure 6. Algorithme RS de répartition des segments

Propriété : Cet algorithme place les segments en un nombre minimum de lignes : P_m

Démonstration : Considérons un ensemble de segments dont le maximum de la fonction de pic associée est P_m . Supposons que ces segments soient placés sur P_m+1 lignes : le segment AB (A étant son extrémité gauche) est placé sur la P_m+1 ème ligne (figure 7).

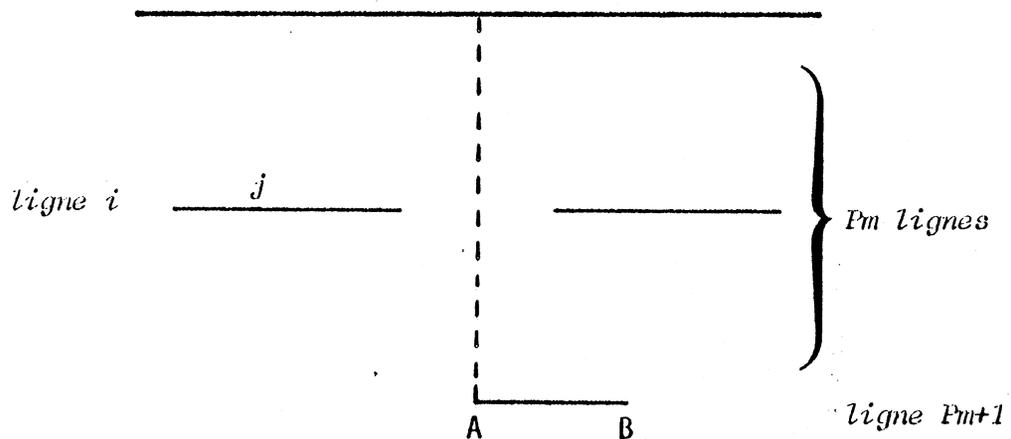


Figure 7. Il existe une position libre sur la verticale passant par A (ligne i)

Par hypothèse, le pic sur la verticale passant par A est inférieur ou égal à P_m . Il y a donc au moins une position vide sur cette verticale parmi les P_m premières lignes (ligne i sur la figure 7). D'après la définition de l'algorithme RS proposé, AB ou un segment ayant son extrémité gauche de même abscisse aurait dû être

placé sur la ligne i après le segment j . Cette contradiction nous permet de conclure que l'algorithme RS place les segments sur P_m lignes, P_m étant le plus petit nombre de lignes possible.

II - 2. PLACEMENT

II - 2.1. Placement initial constructif

a) Coût

Pour chaque candidat C au placement, deux valeurs sont calculées par rapport au sous-réseau déjà placé :

- N_1 , nombre de connexions de C avec les portes déjà placées.
- N_2 , nombre de portes connectées à C dans le sous-réseau déjà construit.

Ces deux quantités nous permettent de définir un vecteur coût partiel $\begin{pmatrix} N_1 \\ N_2 \end{pmatrix}$ que l'on veut minimiser et qui traduit bien la connectivité au sous réseau déjà placé du candidat au placement.

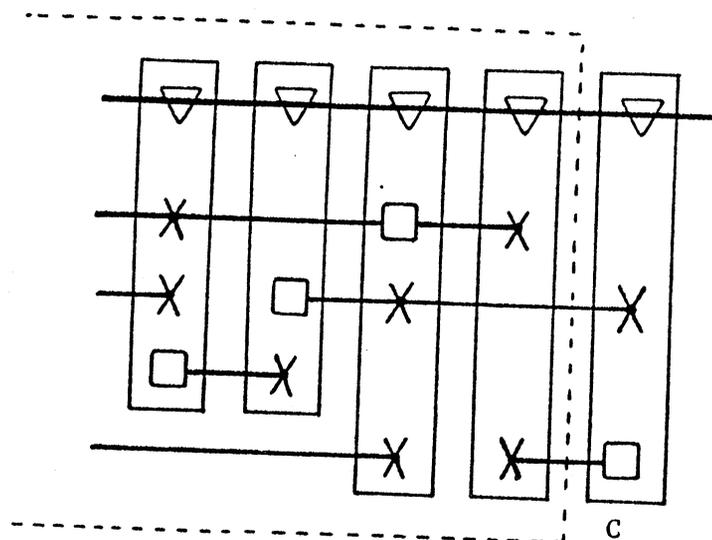


Figure 8. $N_1 = 2$, $N_2 = 3$

Les connexions d'entrée/sortie à gauche et à droite de la cellule sont prises en compte pour évaluer N_1 et N_2 . En effet, si la construction est effectuée de gauche à droite, N_1 et N_2 sont incrémentées ou décrémentées en fonction du nombre de connexions d'entrée/sortie à gauche ou à droite. Ceci a pour effet de privilégier les positions les plus à gauche ou à droite pour des portes fortement connectées respectivement à gauche ou à droite.

b) Algorithme

A chaque pas de l'algorithme constructif, une porte est sélectionnée en fonction de ses connexions avec les portes du sous-réseau déjà placé. On choisit, parmi les portes non placées, celle qui minimise le coût partiel. Lorsque plusieurs portes ont des coûts minimum égaux, on en sélectionne une parmi elles de manière aléatoire. Ce placement constructif permet d'obtenir un placement initial dont la longueur totale des connexions est minimale. Ce minimum n'est évidemment pas absolu puisque l'évaluation des critères pour choisir une porte n'est faite qu'en fonction des seules portes déjà placées et non pas en fonction de toutes les portes de la cellule.

Choix de la première porte

Le sens de construction n'est pas déterminé à priori mais après le choix de la première porte à palcer. Pour chaque porte, on calcule un nombre N tel que N est incrémenté ou décrémenté suivant qu'il existe respectivement une connexion d'entrée/sortie à gauche ou à droite. La première porte choisie est celle dont la valeur absolue de N est maximum et le signe N détermine le sens de construction : gauche à droite si positif, droite à gauche si négatif. La construction se poursuit dans le sens déterminé après le placement de cette porte.

L'algorithme général PC1 (placement constructif 1) pour la construction de gauche à droite d'une cellule est décrit figure 9.

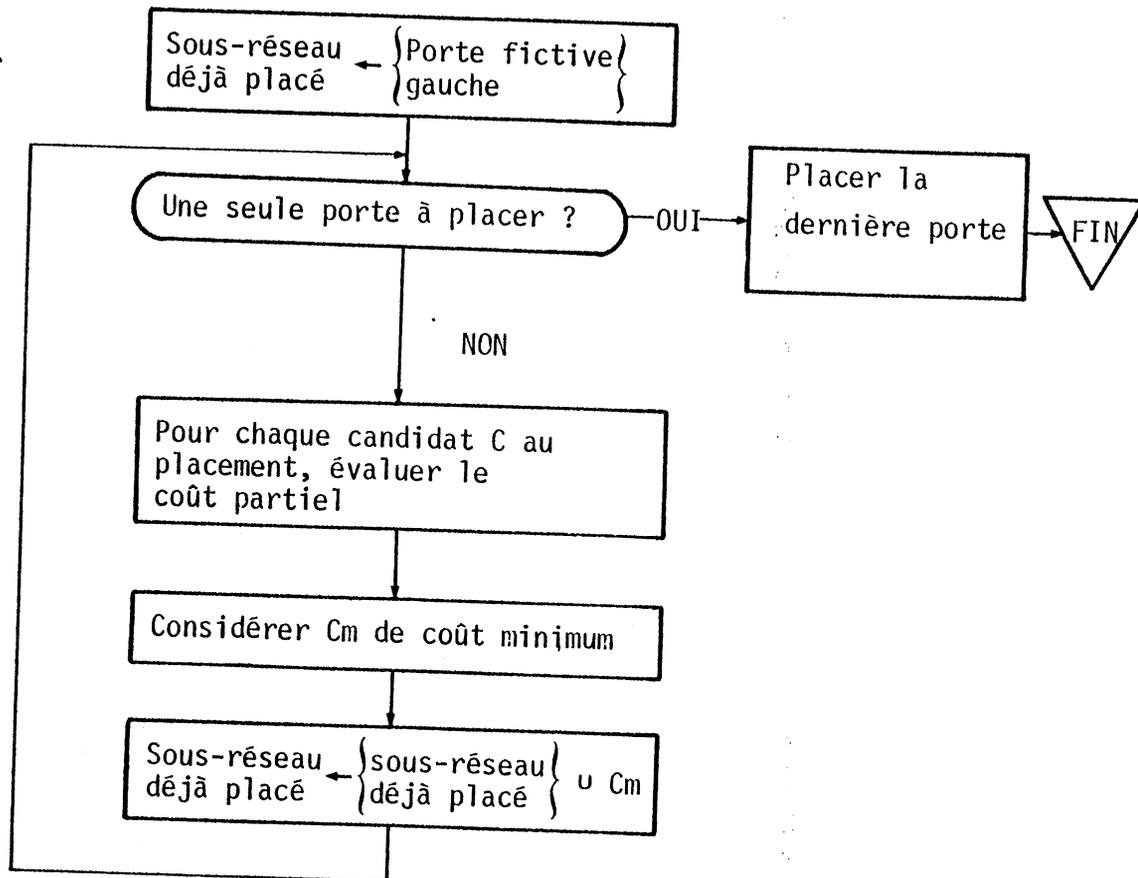


Figure 9. Algorithme PC1, placement constructif

c) Complexité

Considérons le pas de l'algorithme où, i portes étant placées, on cherche à placer la $(i+1)$ ème porte. Pour déterminer cette $(i+1)$ ème porte, $(n-i) * i$ opérations élémentaires (comparaisons, calculs des coûts..) indépendantes du nombre n de portes doivent être effectuées.

A la fin du placement constructif, $\sum_{i=1}^{n-1} (n-i) \times i = \frac{n(n^2-3n+2)}{6}$
opérations élémentaires ont été effectuées.

Cet algorithme constructif est donc polynomial d'ordre n^3 et fournit très rapidement une solution (2 à 3 secondes pour une centaine de portes).

II - 2.2. Amélioration itérative

a) Coût

La solution initiale obtenue par l'algorithme de placement constructif est améliorée en effectuant des échanges par paire des portes. Les trois composantes du vecteur coût à minimiser durant cette étape sont :

- P_m , le maximum de la fonction de pic, ce qui minimise la surface de la cellule.
- N_p , le nombre de points où P_m est atteint. Minimiser N_p permet de réduire plus efficacement P_m : en effet, pour une valeur fixée de P_m , le nombre de points où le maximum de la fonction de pic est atteint est minimisé jusqu'à la valeur 0 et P_m est alors décrémente de 1. En outre, la prise en compte de ce critère facilitera, comme on le montrera dans la suite, le tracé des connexions dans la cellule.
- L_c , longueur totale des connexions.

Pour pouvoir agir sur la solution finale, le concepteur choisit l'ordre dans lequel ces critères sont évalués, c'est-à-dire qu'il impose le vecteur coût en fonction de l'application.

- $\left\| \begin{array}{c} P_m \\ N_p \\ L_c \end{array} \right\|$ permet d'accorder un poids important au pic (surface) devant la longueur des connexions.
- $\left\| \begin{array}{c} L_c \\ P_m \\ N_p \end{array} \right\|$ permet, au contraire, d'accorder une plus grande importance à la longueur des connexions.

b) Algorithme

L'algorithme général consiste à faire l'échange systématique des $\frac{n(n-1)}{2}$ paires (n étant le nombre de portes) et de conserver la meilleure solution parmi les placements essayés, jusqu'à ce qu'il n'y ait plus d'amélioration.

Deux algorithmes d'itérations sont proposés :

- le premier (figure 10) effectue tous les échanges par paire possibles pour rechercher une "meilleure" configuration et réitère ce procédé à partir de cette "meilleure" configuration tant qu'il existe une amélioration;
- le deuxième (figure 11) effectue les échanges jusqu'à la première configuration améliorée et réitère ce procédé à partir de cette configuration jusqu'à ce qu'aucun échange ne soit plus accepté.

Nous noterons CS la configuration solution, CC la configuration courante à partir de laquelle sont effectués les échanges et CE la configuration après échange.

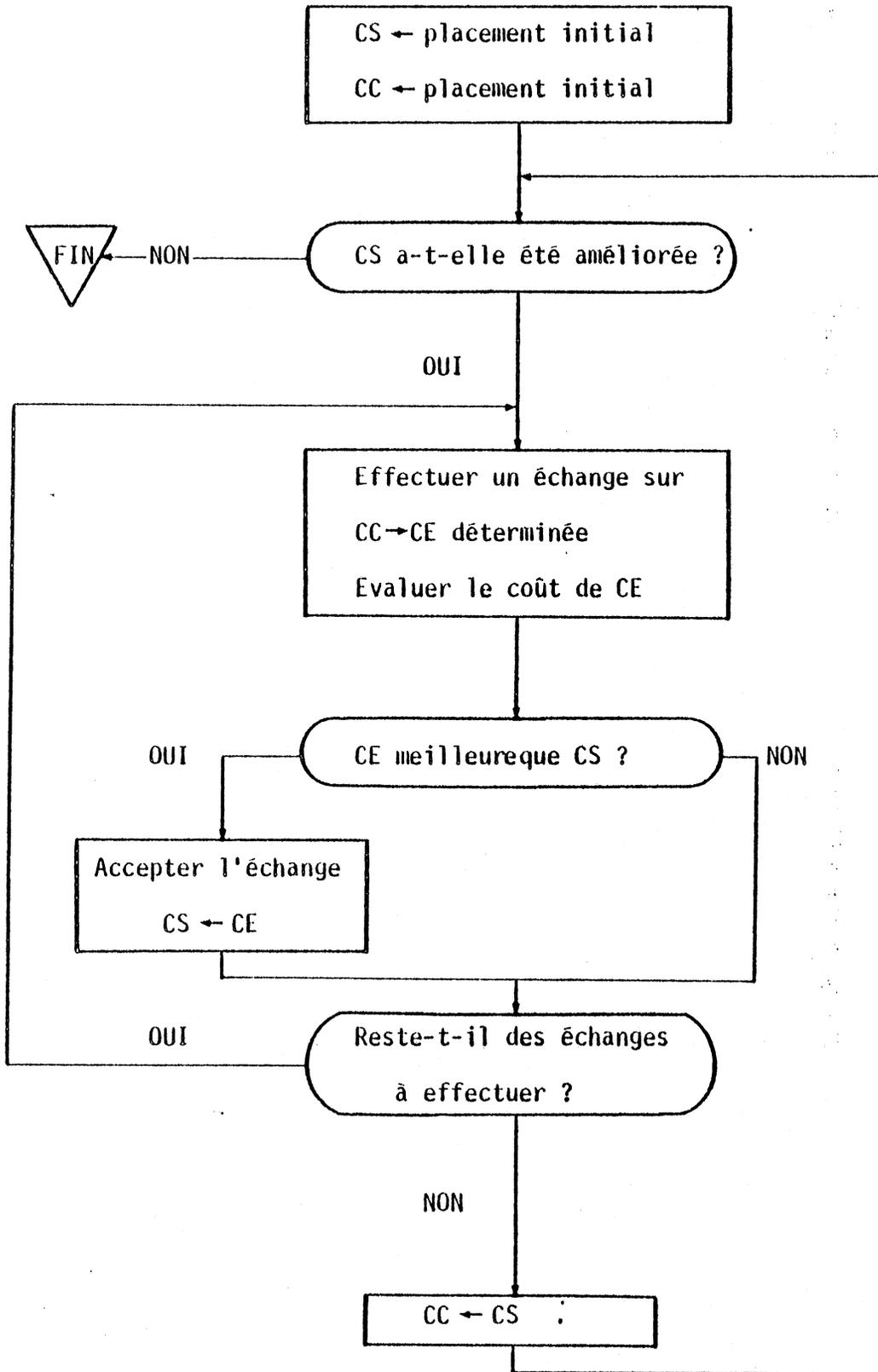


Figure 10. Algorithme II, d'échanges par paire

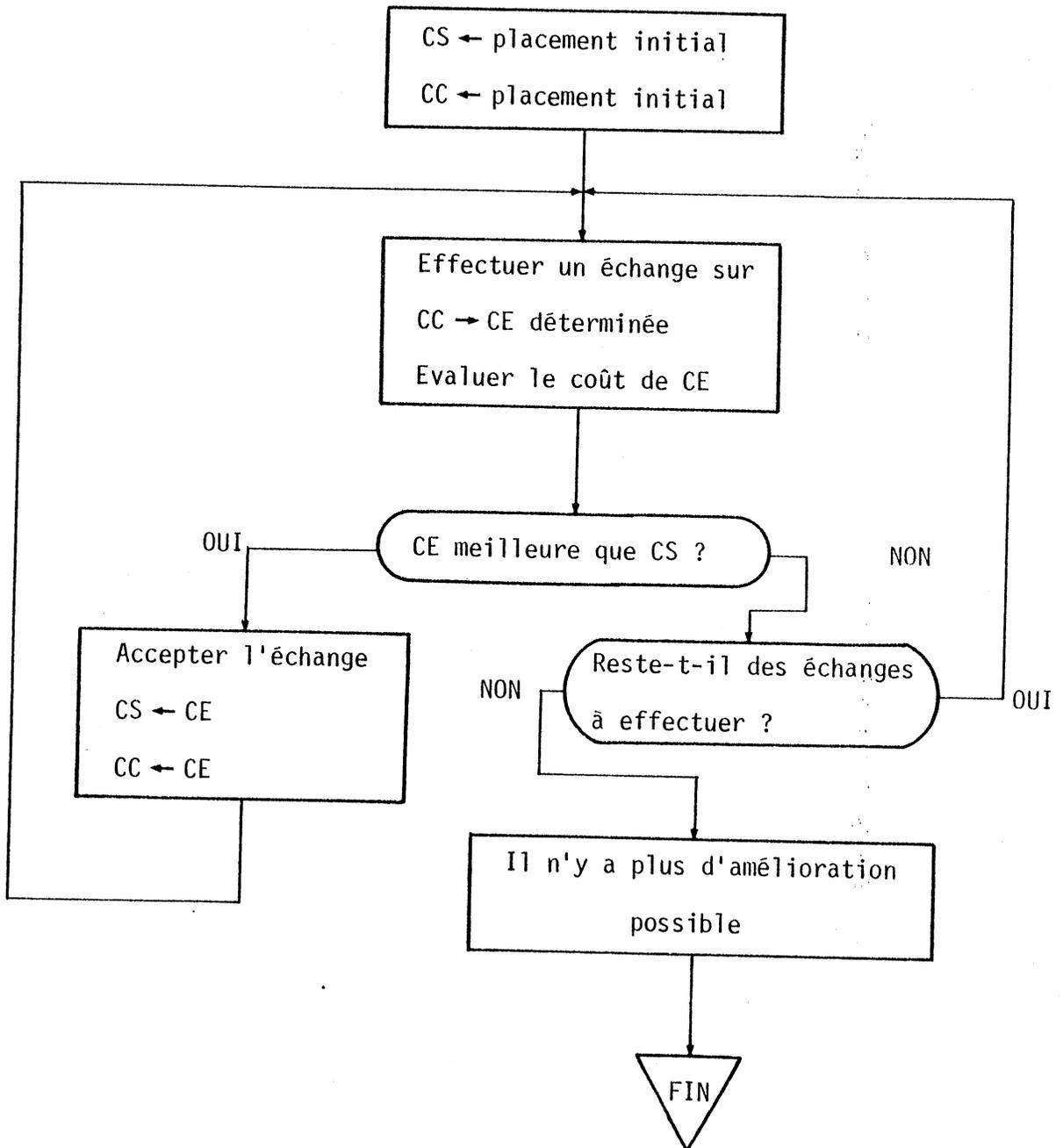


Figure 11. Algorithme I2, d'échanges par paire

c) Complexité

Ces deux algorithmes n'explorent pas le même ensemble de configurations parmi les $n!$ possibles (figure 12). On ne peut à priori pas assurer que l'un des algorithmes délivre une meilleure solution que l'autre : leur qualité dépend du placement initial à améliorer.

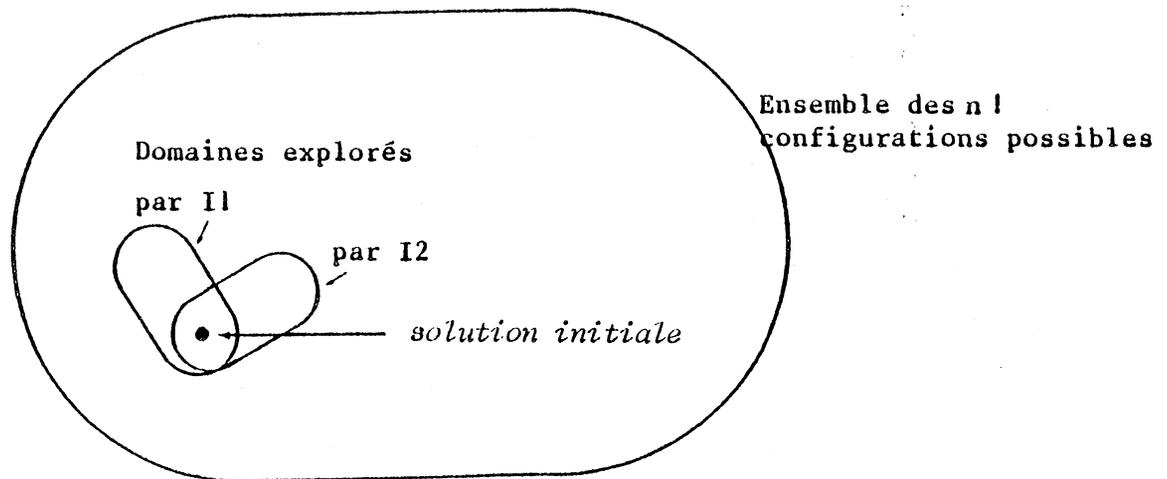


Figure 12. Les domaines explorés par I1 et I2 sont différents

Le nombre de configurations explorées par ces algorithmes itératifs est d'ordre n^2 . Pour chaque échange, le calcul du coût est effectué suivant un algorithme d'ordre n . Ces algorithmes I1 et I2 sont donc polynômiaux d'ordre n^3 . Leur programmation facile et leur exécution rapide permet de les implémenter tous les deux pour leur soumettre un même placement initial et choisir la meilleure solution obtenue.

Un meilleur placement peut aussi être déterminé si l'on soumet aux procédures itératives d'amélioration plusieurs placements initiaux équivalents d'une même cellule. Cette solution est d'autant plus réaliste que la faible complexité des algorithmes (polynomiale d'ordre n^3) permet d'assurer une grande rapidité d'exécution.

II - 3. TRACE DES CONNEXIONS

A ce stade de l'implantation de la cellule, le placement des portes est déterminé, mais l'emplacement des points de contact grille et drain des connexions sur chaque porte n'est pas fixé. L'étape de tracé doit donc (i) disposer les différents contacts sur la porte, (ii) choisir le chemin de connexion entre ces contacts.

II - 3.1. Partitionnement des connexions en deux classes

Le moyen le plus simple de traiter les connexions, pour un placement donné, serait de les considérer comme un ensemble de segments parallèles à la ligne d'alimentation. Cette solution n'est pas acceptable : les deux premières lignes, à partir de la ligne d'alimentation, sont inutilisées à cause de la contrainte de positionnement des drains. Cette méthode conduirait à une cellule de largeur maximale.

Nous sommes alors conduits à définir deux zones dans la cellule, et par conséquent deux classes de connexions :

- La zone 1 ou zone de cou dage correspond aux 3 premières lignes de la cellule. Dans cette zone, les connexions sont coudées de manière à ce qu'aucun drain n'occupe les deux premières lignes. La troisième ligne est occupée par des segments, dans les espaces libres de drains.
- La zone 2, à partir de la quatrième ligne, permet le tracé des connexions sous forme de segments.

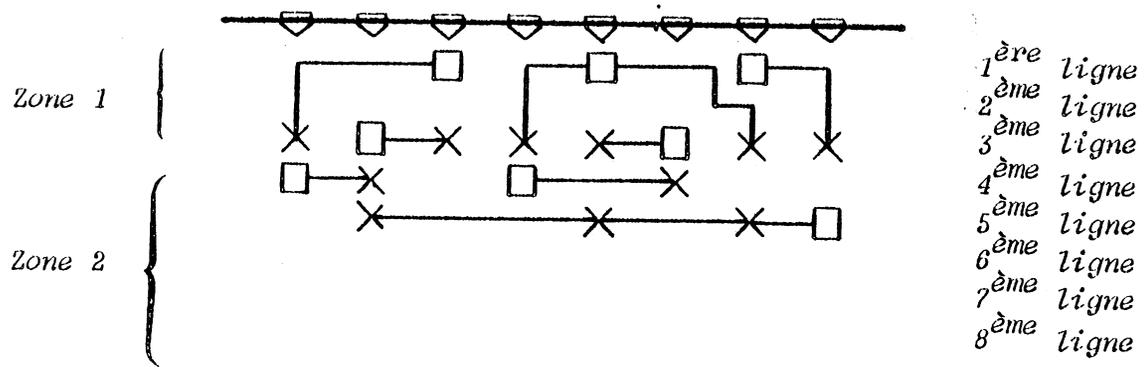


Figure 13. Les 2 zones

II - 3.2. Méthode de tracé

Nous noterons :

- . f la fonction de pic calculée sur l'ensemble des segments et P_m son maximum,
- . f' la fonction de pic calculée après élimination des connexions coudées et $P'm$ son maximum,
- . f'' la fonction de pic calculée sur l'ensemble des segments de la zone 2 et $P''m$ son maximum.

Les connexions sont tracées séquentiellement dans la zone 1, puis dans la zone 2. La méthode proposée a pour but d'obtenir une cellule de largeur inférieure à la largeur maximale P_{m+3} .

a) Principe

Les connexions à couder dans la zone 1 sont déterminées de façon à ce qu'elles recouvrent tous les points de pic maximum, et ce en des sites différents de drains. Ceci permet de tracer sur les deux premières lignes des connexions ou parties de connexions où le maximum du pic est atteint. On comprend l'intérêt de minimiser lors du placement le nombre de points où P_m est atteint, même dans le cas où P_m n'est pas lui-même réduit.

L'ensemble des segments couvés détermine sur la troisième ligne des intervalles limités par les drains et les côtés gauche et droit de la cellule (GA, AB, BD).

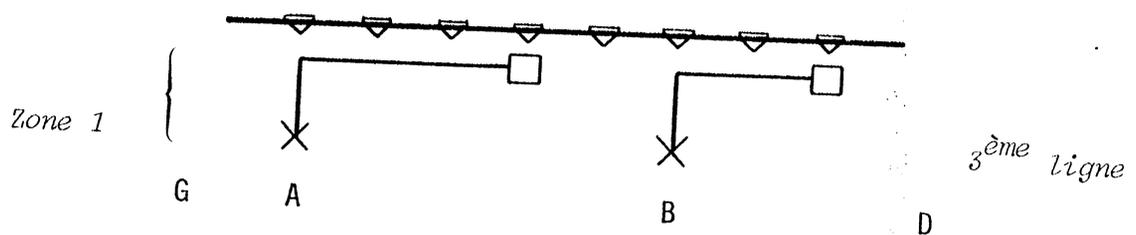


Figure 14. Intervalles de la 3^{ème} ligne

On trace alors dans ces intervalles des segments dont l'union recouvre tous les points où P'_m est atteint. Quand cela est possible, le maximum P''_m de f'' est au plus égal à P_{m-2} . Donc, au plus P_{m-2} lignes sont nécessaires pour tracer les connexions de la zone 2 en appliquant l'algorithme RS de répartition des segments.

La largeur de la cellule est alors P_{m+2} .

b) Conditions sur les connexions de la zone 2 pour obtenir une cellule de largeur P_{m+2}

- Les connexions coudées doivent recouvrir, en des points différents des drains, les pics maximum de $f(P_m)$. Nous avons alors $P'_m \leq P_{m-1}$.
- L'ensemble des segments à placer dans les intervalles de la troisième ligne doit recouvrir les pics maximum de f' (P'_m). Nous avons alors $P''_m \leq P_{m-2}$.

Lorsque les deux conditions $P'_m = P_{m-1}$ et $P''_m = P_{m-2}$ sont remplies, la largeur de la cellule est égale à P_{m+2} .

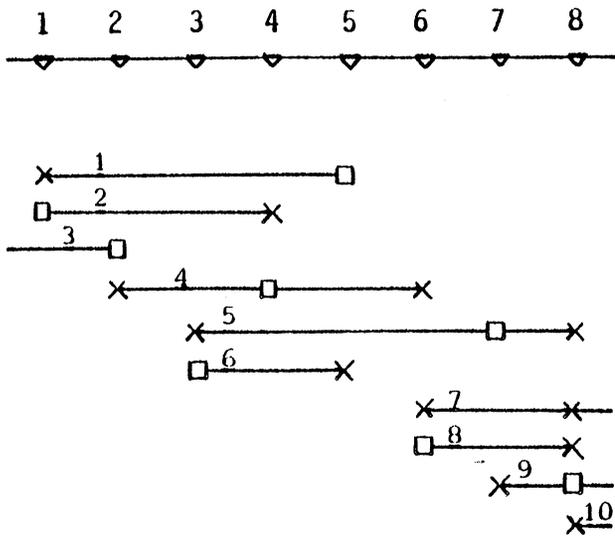
Il n'est en général pas possible, dans les cas pratiques étudiés de réduire à nouveau la largeur : en effet, il faudrait pouvoir, pour cela, couder des connexions sur la deuxième ligne de manière à libérer d'autres intervalles sur la troisième ligne, ce que ne permet pas l'encombrement du circuit.

c) Exemple

Considérons un ensemble de portes MD-MOS placées et la liste des segments correspondants (figure 15-a). Le pic des points 3,4 et 8 est maximum ($P_m=5$). Les connexions 1 et 2 recouvrent ces points et sont donc coudées dans la zone 1 (figure 15-b). Les segments 3 et 4, recouvrant les points 3, 4, 6 et 8 où P'_m est atteint sont ensuite placés dans les intervalles de la troisième ligne (figure 15-c).

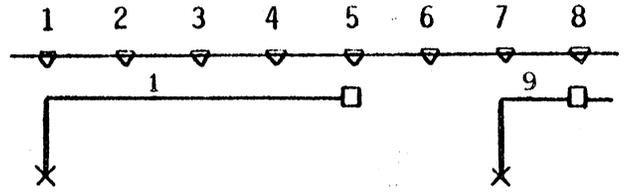
La figure 15-d montre le tracé final obtenu, après avoir réparti les segments sur trois lignes de la zone 2.

La cellule obtenue a une largeur de 7 pas de grille (P_m+2).

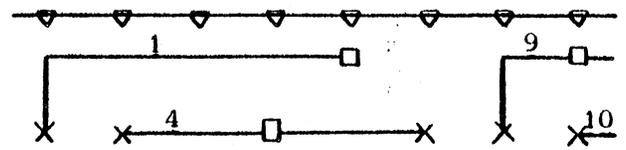


Pic 3 4 5 5 4 4 4 5

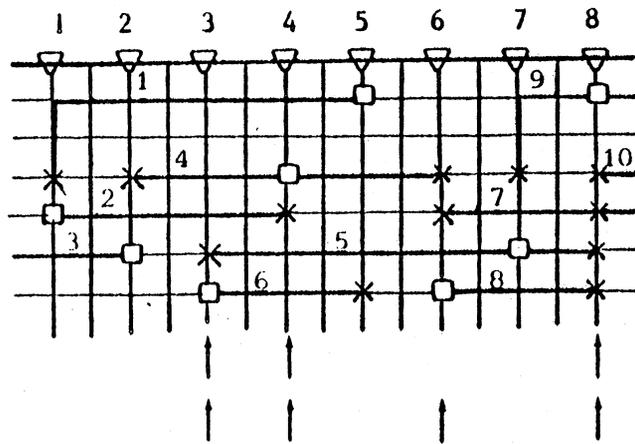
(a)



(b)



(c)



(d)

Figure 15. Tracé des connexions dans une cellule

II - 3.3. Connexions traitées dans la zone 1

La proportion des circuits dont la largeur n'excède pas P_{m+2} est d'autant plus élevée que le nombre de connexions susceptibles d'être coudées ou d'être tracées sur la troisième ligne est grand.

Dans la suite, nous dirons qu'une connexion est tracée avec retour arrière si elle recouvre l'une de ses extrémités (figure 16).

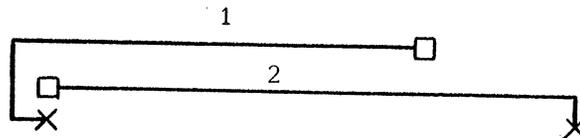


Figure 16. Tracé de 1 avec retour arrière

a) Choix des connexions à couder

Nous explorons trois types de configurations de connexions à couder.

Sans recouvrement

Les segments considérés ne se recouvrent pas. Leurs coudage est simple (figure 17).

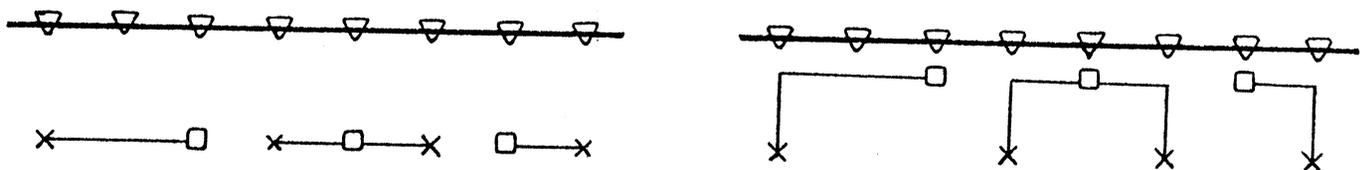


Figure 17. Coudage sans recouvrement

. Avec recouvrements, sans retour arrière

Les segments considérés se recouvrent. Leur tracé est possible sans retour arrière (figure 18).

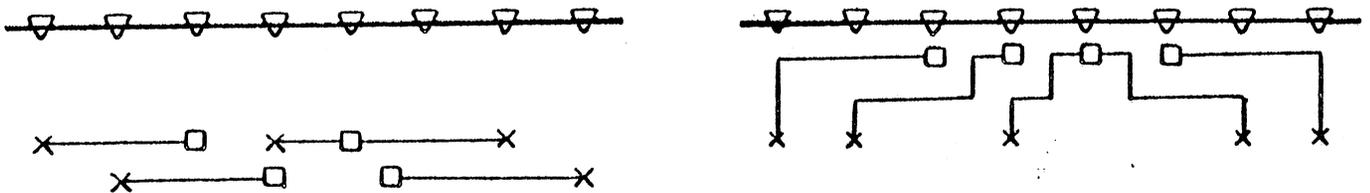


Figure 18. Coudage avec recouvrement, sans retour arrière

Lors de la recherche de ces segments, deux conditions doivent être respectées :

- seuls les recouvrements de deux connexions (jamais trois) sont autorisés. La troisième ligne est partout libérée sauf aux points comportant un drain.
- Sur chaque zone de recouvrement, l'une des connexions au moins ne comporte pas de drain. Elle sera placée au "dessus" de l'autre (c'est-à-dire le plus près possible de la ligne d'alimentation).

Un ensemble de segments vérifiant ces conditions étant déterminé, le segment est réalisé : les connexions, ne comportant pas de drain dans les zones de recouvrement, sont placées près de la ligne d'alimentation. En règle générale, on "repousse" le plus possible les connexions vers la ligne d'alimentation.

. Avec recouvrements et retours arrière

Les segments considérés se recouvrent et des retours arrière d'un pas de grille sont autorisés pour le tracé. Cette possibilité supplémentaire est introduite pour résoudre les cas où dans un recouvrement, les deux connexions comportent un drain (figure 19). Nous noterons REC la zone de recouvrement de deux connexions, et REC' la zone REC moins les extrémités.

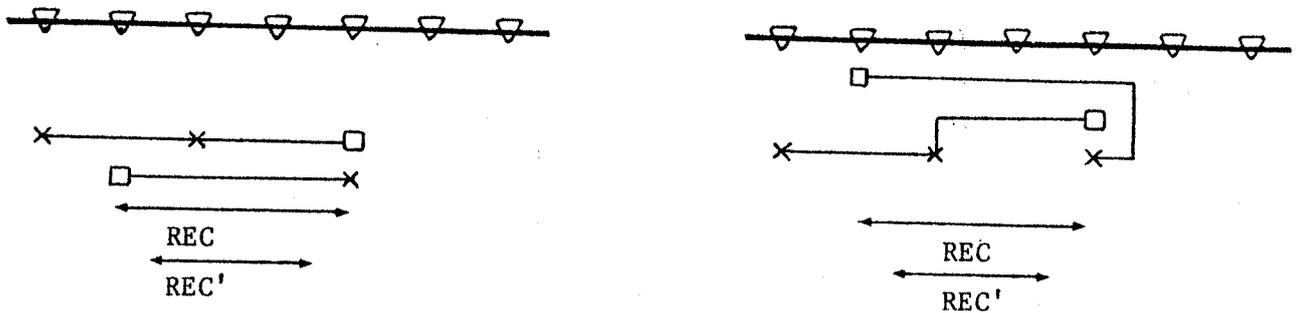


Figure 19. Zones de recouvrement

Plusieurs points sont examinés lors de la recherche des segments à couder.

- α) Ne pas considérer plus de deux connexions se recouvrant.
- β) Les connexions se recouvrant ne doivent pas comporter de drains superposés (figure 20).

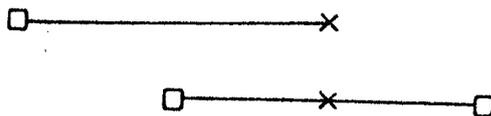
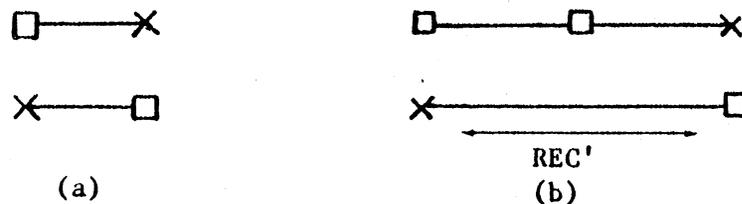


Figure 20. Recouvrement interdit : drains superposés

γ) En cas de recouvrement, l'une au moins des connexions ne doit pas comporter de drain dans REC'. Cette connexion sera coudée le plus près de la ligne d'alimentation et son tracé contiendra un retour arrière. Lorsque les deux connexions conviennent ($REC' \neq \emptyset$ ou pas de drain dans REC'), retenir, si possible, celle permettant un retour arrière à gauche. En effet, la suite des segments à couder étant examinée de gauche à droite, on évite ainsi les retours arrière à droite qui pourraient introduire des blocages.



δ) Les connexions étant numérotées 1 et 2, supposons que 1 vérifie γ). Cette connexion contient un drain à l'une des extrémités, droite ou gauche de REC.

Si ce drain est à gauche de REC, le retour arrière à gauche sera possible si : $borninf(1) \leq borninf(2)$ où $borninf(I)$ désigne l'abscisse de l'extrémité gauche (borne inférieure) du segment I (figure 22).

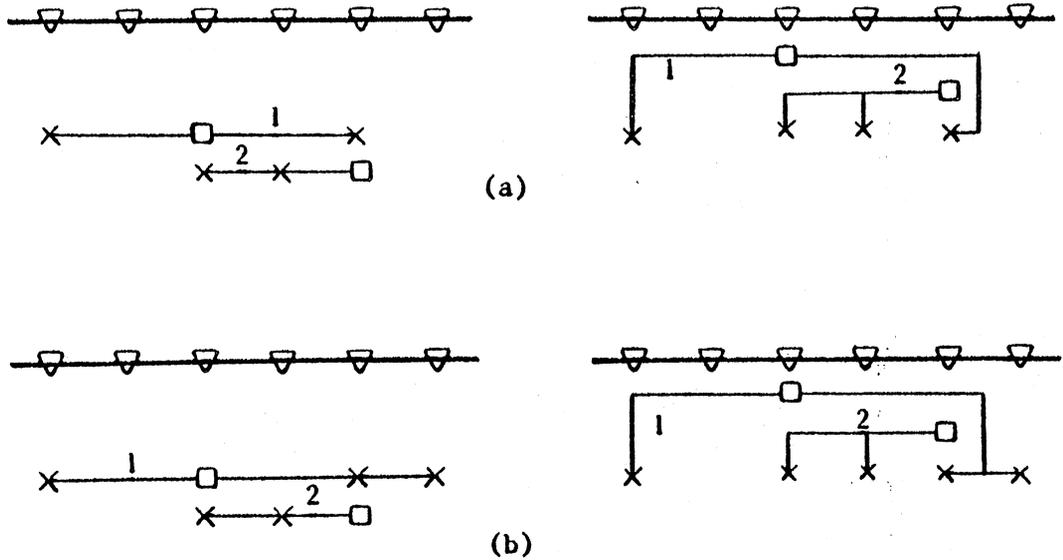


Figure 23. (a) $\text{bornsup}(1) = \text{bornsup}(2)$

(b) $\text{bornsup}(1) > \text{bornsup}(2)$

e) Remarquons, en outre, qu'un retour arrière occupe une verticale comprise entre deux portes. S'il existe un retour arrière à droite d'une porte, on ne pourra pas effectuer un retour arrière à gauche de la porte suivante. Il faut par conséquent s'assurer, avant d'effectuer un retour arrière à gauche, que la verticale à utiliser n'est pas déjà occupée.

Il faut donc, pour tracer deux connexions se recouvrant avec retour arrière :

- . Que α) et β) soient respectées
- . Que l'une au moins des deux connexions vérifie γ). On s'assure ensuite que δ) et γ) sont bien respectées.

La recherche des segments à couder est effectuée de manière arborescente parmi les trois classes ainsi définies. Ces segments, une fois couvés, libèrent sur la troisième ligne de la place (suite d'intervalles limités par des drains) utilisée pour le tracé des segments recouvrant les pics maximum de f' , ce qui permet de réduire d'un pas la largeur du circuit.

b) Choix des segments à placer sur la troisième ligne

Soit AB un intervalle libre, de la troisième ligne pour lequel une suite de segments est déterminée :

- *segments sans recouvrements, inclus dans AB (figure 24)*



Figure 24.

- *segments avec recouvrements, inclus dans AB.*

Les connexions dans ce cas ne sont pas tracées sous forme de segments sur la troisième ligne mais déformées sur la deuxième ligne pour respecter la deuxième condition énoncée précédemment (§ II.3.2b).

Les recouvrements de plus de deux segments sont exclus. De plus, le tracé est effectué sans retour arrière. Notons REC la zone de recouvrement des deux connexions, REC''_g et REC''_d les zones REC étendues à gauche ou à droite d'un pas de grille (figure 25).

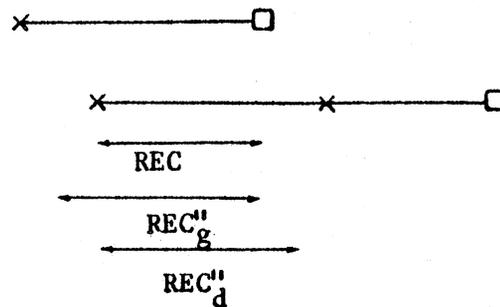


Figure 25. Zones de recouvrement

Pour permettre le tracé de ces segments sur les troisième et deuxième lignes, on s'assurera que :

- dans le cas d'un recouvrement à gauche, la deuxième ligne est inoccupée dans la zone REC''_g (figure 26a),
- dans le cas d'un recouvrement à droite, la deuxième ligne est inoccupée dans la zone REC''_d (figure 26b).

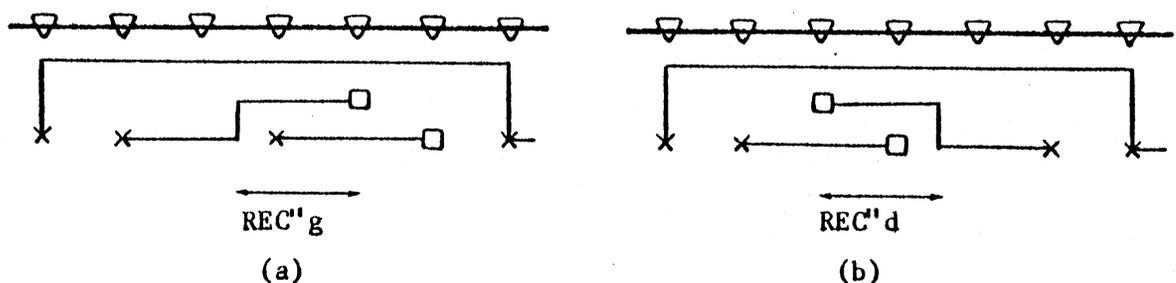


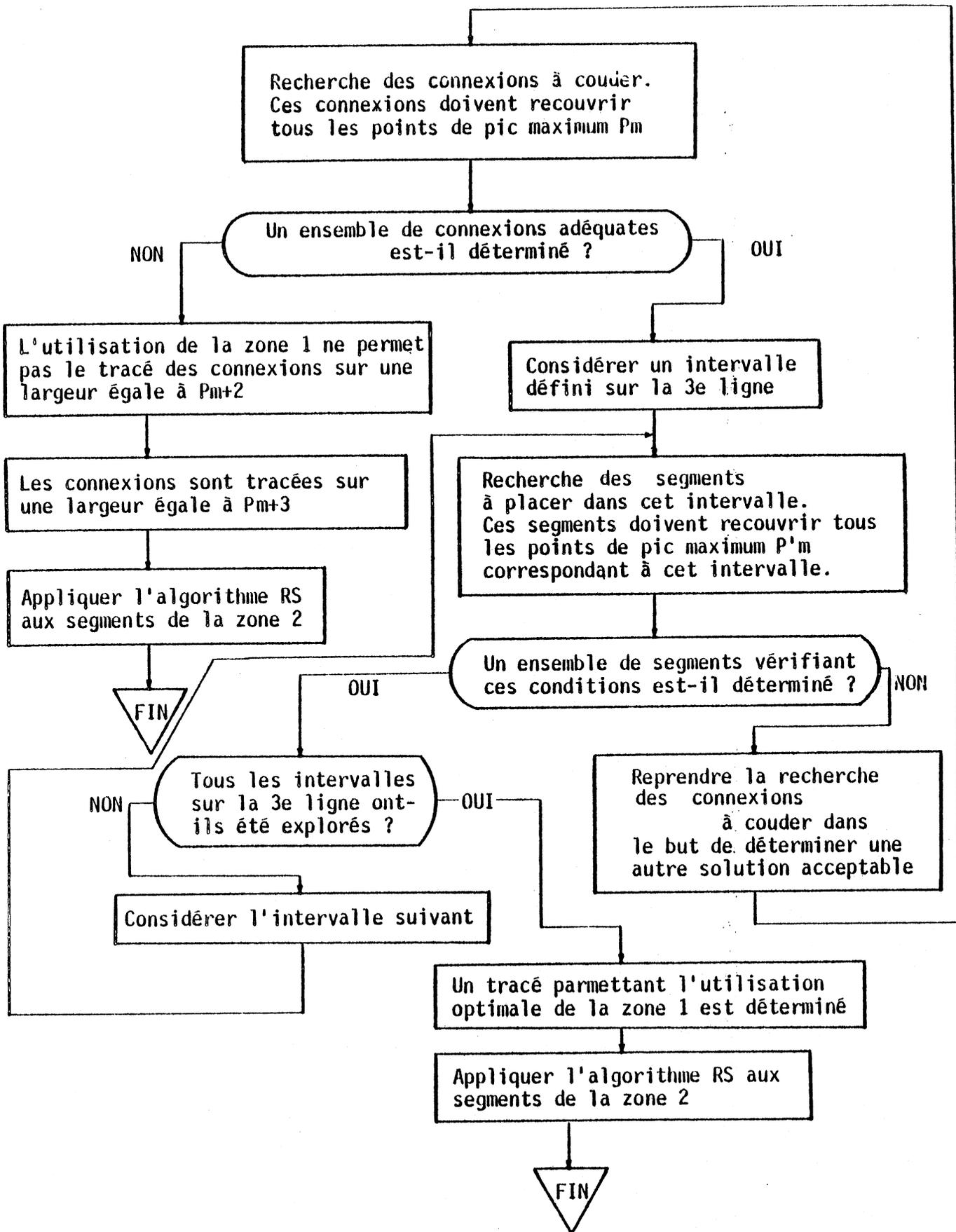
Figure 26. Coudage de connexions de la 3e ligne

Les segments tracés sur la troisième ligne sont déformés de manière à ce que les connexions obtenues permettent une utilisation optimale de cette troisième ligne. Ainsi, on augmente le nombre de cellules pour lesquelles une réduction de surface est possible.

II - 3.4. Algorithme TC de tracé des connexions

Dans le cas général, le tracé est décomposé en deux étapes successives :

- recherche des connexions de la zone 1 et tracé,
- tracé des connexions de la zone 2.



Soit m le nombre de connexions à traiter. La complexité de chacune des étapes est de l'ordre de m^2 . Cet algorithme est par conséquent du même ordre et donc très rapide. De plus, il délivre dans la plupart des cas une solution de largeur P_{m+2} .

Un exemple de cellule à une ligne traité par l'algorithme TC est donné figure 27.

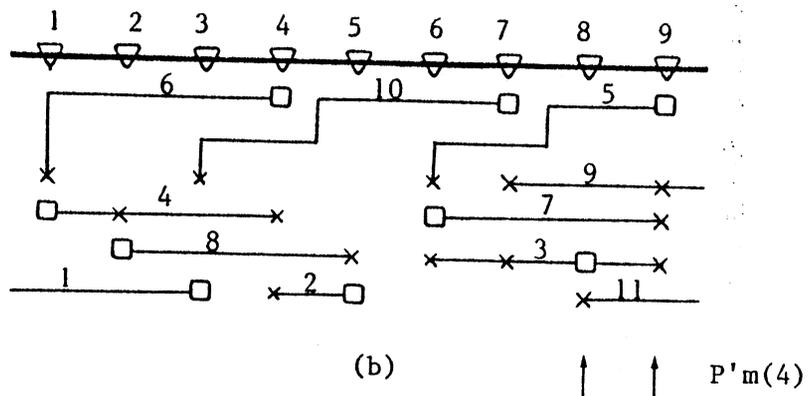
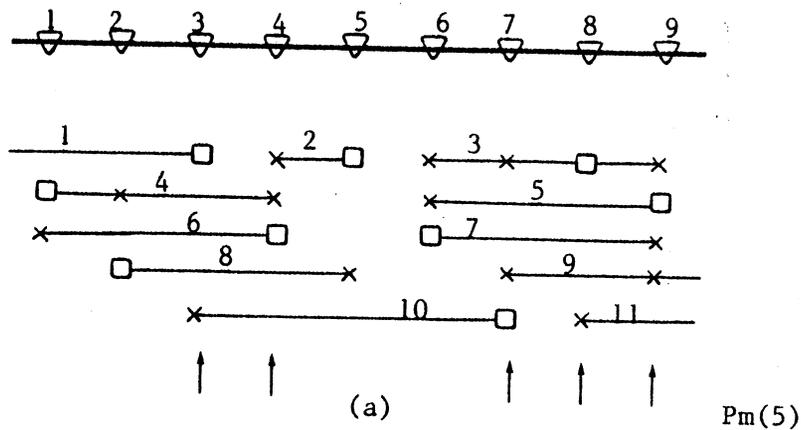
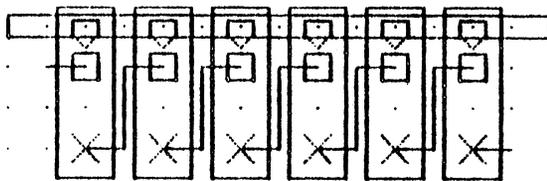


Figure 27. Exemple de cellule à une ligne traitée par TC

II - 4. EXEMPLES - REMARQUES

Exemple 1

Solution automatique

Dans ce cas, pas d'amélioration manuelle possible.

Exemple 2

$$\left\{ \begin{array}{l} 9 \text{ portes} \\ 11 \text{ équipotentiellles} \end{array} \right.$$
Version a

Solution obtenue par application des méthodes de placement et de tracé automatiques.

Surface du rectangle englobant $S_R = 126$

Surface occupée $S = 114$

Version b

Solution manuelle (même placement).

Surface de rectangle englobant $S_R = 108$

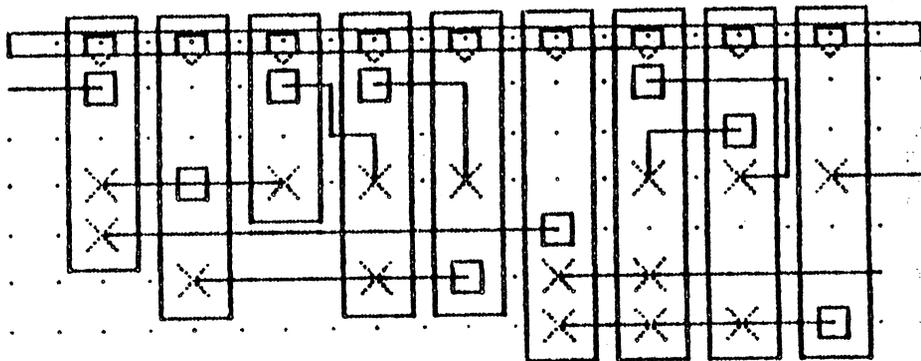
Surface occupée $S = 106$

Amélioration en surface de la version b par rapport à la version a :

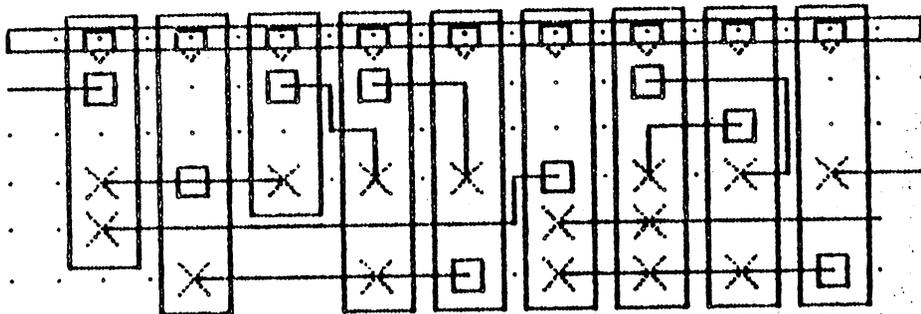
$$A_R = 14,6\%$$

$$A = 7\%$$

Exemple 2



version a



version b

Exemple 3

Additionneur 4 bits avec retenue anticipée

{ 48 portes
 { 57 équipotentiels

Version a

Solution obtenue par application des méthodes de placement et de tracé automatiques.

Surface du rectangle englobant : $S_R = 1248$

Surface occupée : $S = 1039$

Version b

Solution manuelle (même placement).

Surface du rectangle englobant : $S_R = 1248$

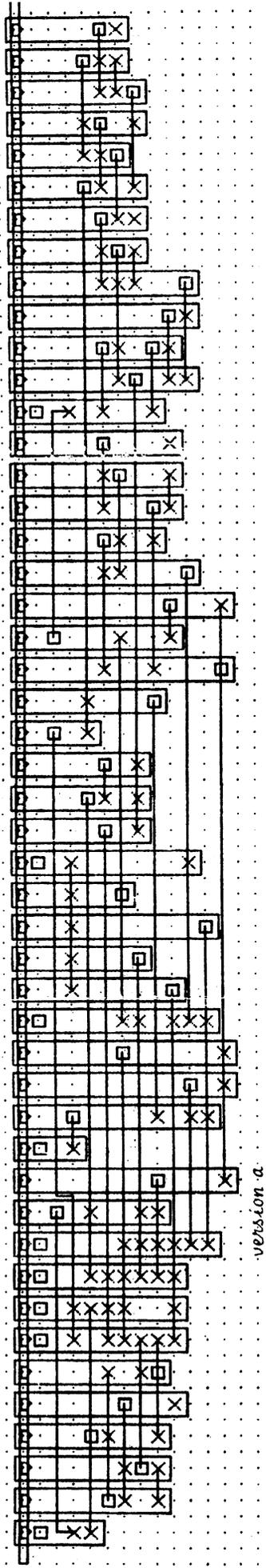
Surface occupée : $S = 908$

Amélioration

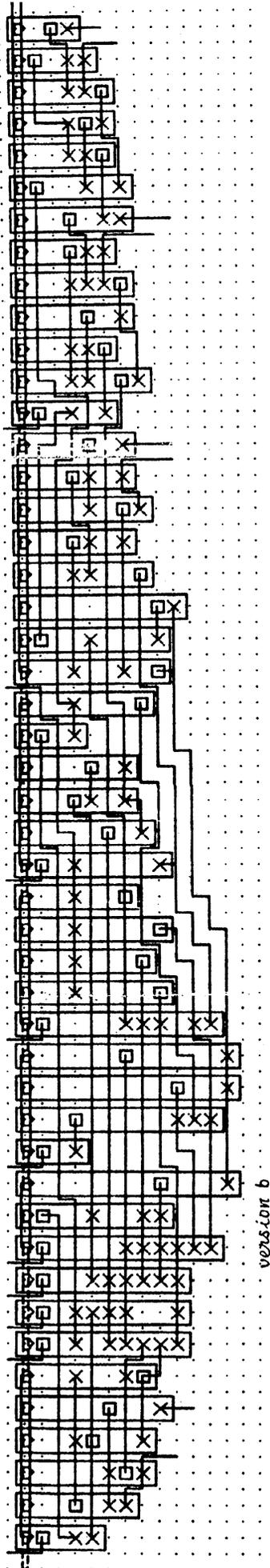
$$A_R = 0\%$$

$$A = 12,6\%$$

Example 3



version a



version b

La cellule illustrée dans l'exemple 1 est en fait un "cas d'école" qui montre cependant bien que l'algorithme construit dans ce cas la solution optimale.

Les exemples 2-a et 3-a illustrent les méthodes exposées pour le placement des portes et le tracé des connexions. Des solutions manuelles sont également proposées (2-b et 3-b); elles sont en fait déduites des solutions 2-a et 3-a : le placement est inchangé et les segments ont été déformés pour permettre des gains en surface occupée de 7% et 12,6% respectivement.

Les méthodes retenues pour le placement et l'interconnexion des portes dans les cellules à une ligne présentent plusieurs avantages :

- elles sont simples à mettre en oeuvre,
- les méthodes sont efficaces et permettent l'implantation des cellules en un nombre réduit de pas. Le coudage, par l'utilisation des deux premières lignes, permet en général un gain d'un pas de grille. La largeur des circuits obtenue est alors P_m+2 .

Il est possible d'accroître encore l'efficacité des algorithmes de tracé sans modifier la structure générale de la méthode. Il suffit en effet d'augmenter le nombre de connexions susceptibles d'être coudées ou placées sur la troisième ligne en introduisant des configurations supplémentaires. Par exemple, on pourra choisir une connexion et la placer sur la troisième ligne sans qu'elle soit entièrement incluse dans l'intervalle correspondant : on permet alors un recouvrement entre une connexion de la troisième ligne et une connexion coudée (figure 27').

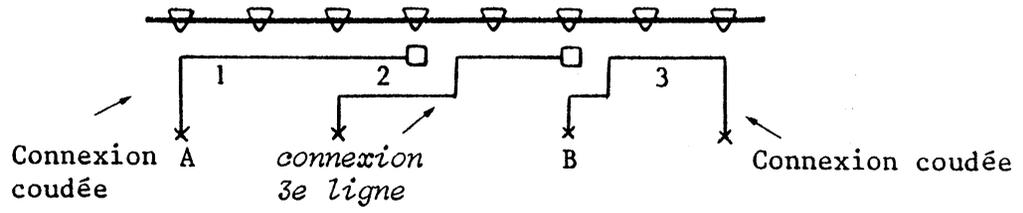


Figure 27'. Configuration supplémentaire:
recouvrement entre 2 et 3

III - ÉTUDE DES CELLULES À DEUX LIGNES D'ALIMENTATION

Les méthodes précédemment étudiées sont modifiées et adaptées aux problèmes de placement et d'interconnexion de portes dans les cellules à deux lignes. La fonction de pic f , calculée sur l'ensemble des deux lignes, est définie de la même manière que pour la cellule à une ligne. La valeur maximum de f , P_m est encore liée à la largeur minimale de la cellule, et sera par conséquent minimisée lors du placement.

Les points d'entrée/sortie de la cellule peuvent être disposés sur les quatre côtés : gauche, haut, droit, bas (dans le cas des aiguilles d'une montre). Deux types de contraintes peuvent être imposés :

- *contrainte de positionnement d'un point d'entrée/sortie :*
côté imposé, intervalle imposé, position exacte fixée;
- *rang ou ordre imposé entre les points d'entrée/sortie.*

Notations

Dans ce chapitre, les cellules C considérées ont des lignes d'alimentation horizontales, ce qui ne restreint évidemment pas la généralité de l'étude. La cellule est constituée de deux cellules à une ligne C_H (située en haut) et C_B (située en bas).

La séquence de blocs (ensemble de portes) C^1, C^2, \dots, C^k résulte d'un partitionnement vertical suivant les lignes de coupe V_1, V_2, \dots, V_{k-1} (figure 28-a).

Le partitionnement horizontal d'un bloc C^i définit C_H^i et C_B^i (figure 28-b).

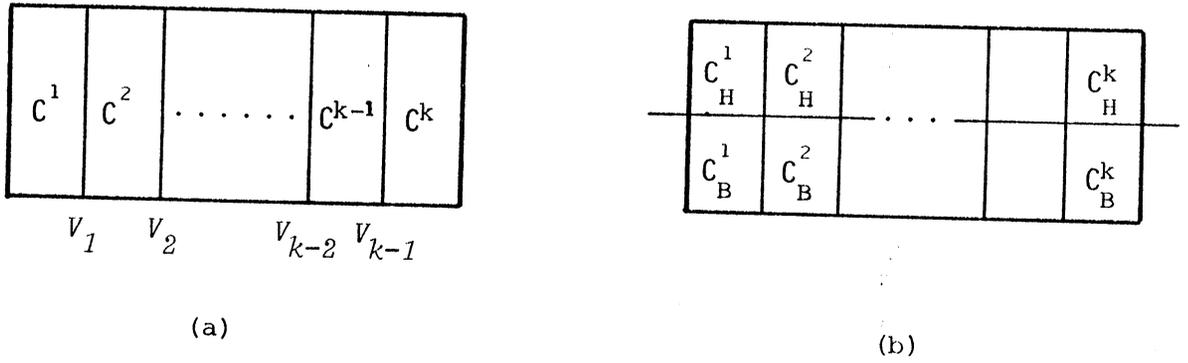


Figure 28. (a) Partitionnement vertical
(b) Partitionnement horizontal

Définitions

Une connexion interne relie les portes de C_H ou de C_B .

Une connexion transverse relie les portes de C_H aux portes de C_B .

Dans la cellule à deux lignes, une équipotentielle peut donc se décomposer en connexions internes et transverse (figure 29).

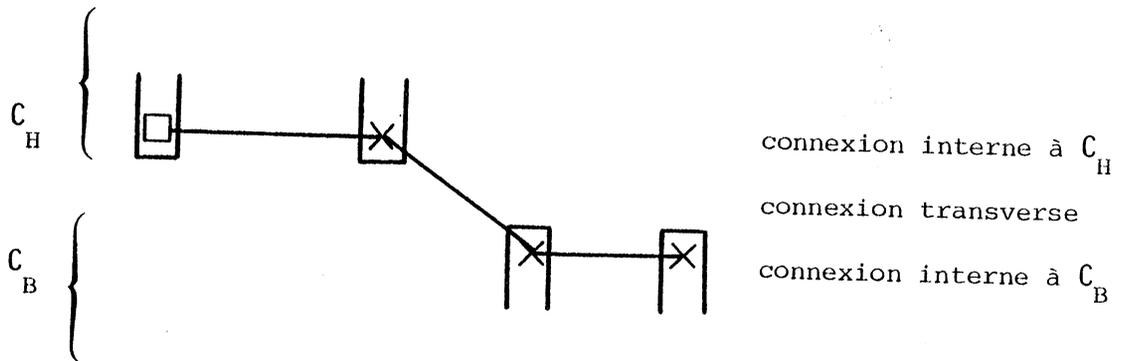


Figure 29.

III - 1. PLACEMENT

III - 1.1. Placement initial

La construction d'un placement initial est réalisée en quatre étapes distinctes :

- affectation de certaines portes à C_H ou C_B ,
- partitionnement vertical en $C^1, C^2 \dots C^k$,
- partitionnement horizontal
- positionnement des portes dans chaque bloc.

a) Affectation de certaines portes à C_H ou C_B

On veut affecter certaines portes à C_H ou C_B de manière à ce que le nombre des connexions transverses soit le plus petit possible. Pour cela, on étudie les positions des points d'entrée/sortie des connexions de la cellule.

. Connexion à 1 point d'entrée/sortie en haut ou en bas de la cellule

- * Si toutes les portes appartenant à cette connexion n'ont pas d'autres points d'entrée/sortie, on leur impose d'être positionnées du même côté que le point d'entrée/sortie.
- * S'il existe une porte appartenant à cette connexion et comprenant d'autres points d'entrée/sortie, deux cas sont envisagés :
 - Tous les points d'entrée/sortie sont d'un même côté. On impose aux portes d'être positionnées sur ce côté.

- Les points d'entrée/sortie sont situés sur les deux côtés.

La porte ainsi déterminée est appelée *module critique* (figure 30). Une répartition des modules critiques de la cellule est déterminée, minimisant le nombre de connexions transverses d'une ligne à l'autre.

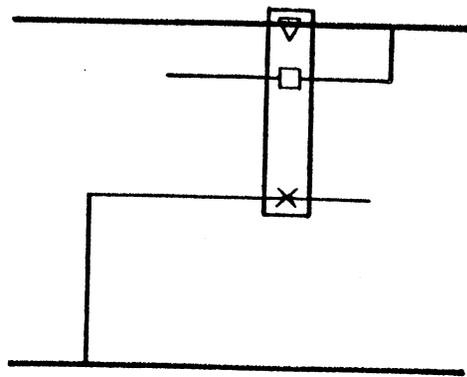


Figure 30. Module critique

- Connexion à 2 points d'entrée/sortie en haut ou en bas de la cellule
 - * Si les deux points d'entrée/sortie sont situés d'un même côté, on est ramené au cas précédent (1 seul point d'entrée/sortie).
 - * Si les deux points d'entrée/sortie sont situés sur les deux côtés, on n'imposera aucune position aux portes correspondantes : en effet, le passage de la connexion de C_H à C_B est obligé, indépendamment de la position des portes.

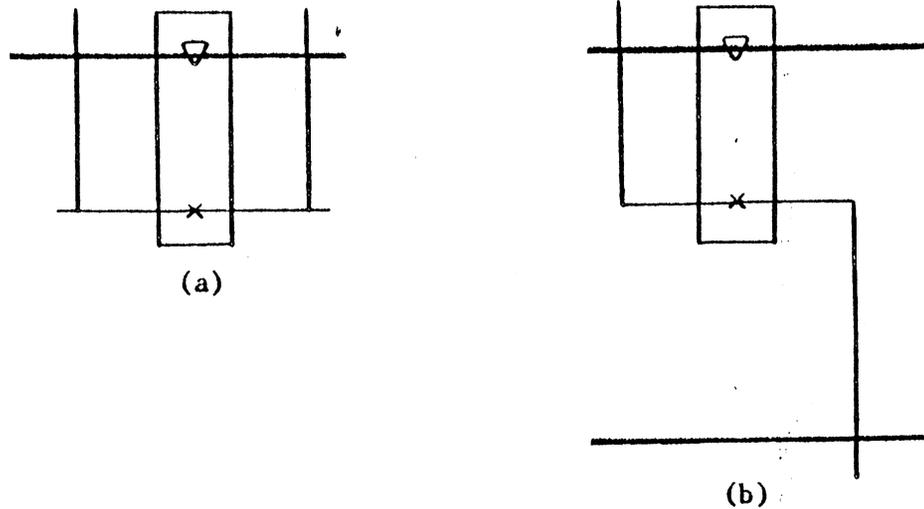


Figure 31. (a) les deux points d'E/S du même côté
 (b) passage de C_H à C_B obligé

A la fin de cette étape, les portes ont l'un de ces trois états :

- . pas de contrainte
- . haut
- . bas.

La répartition des portes ainsi déterminée assure que le nombre de connexions transverses d'entrée/sortie est minimal. Cependant, si cette répartition est pénalisante pour les autres critères (pic maximum par exemple), on peut ne pas considérer ces contraintes dans le placement constructif et minimiser alors le nombre de connexions transverses durant l'amélioration itérative du placement.

b) Partitionnement vertical

On cherche à déterminer une séquence de blocs C^1, C^2, \dots, C^k telle que le nombre de connexions coupant les verticales V_i , autrement dit le pic en V_i , soit le plus petit possible.

α) Coût

Lors de la recherche de chaque partition C^i , le vecteur

coût minimisé est $\begin{pmatrix} N \\ P_i \end{pmatrix}$ où

. N représente le nombre de connexions du candidat à placer dans C^i aux portes des blocs déjà formés C^1, C^2, \dots, C^{i-1} et aux points d'entrée/sortie situés à gauche de V_i . Les portes déjà sélectionnées pour constituer la partition C^i interviennent également dans le calcul de N .

. P_i représente le pic en V_i . Le calcul de cette composante du coût est effectué de manière à prendre en compte les contraintes de positionnement (intervalle ou point fixé) des points d'entrée/sortie en haut et en bas de la cellule (connexions traversant les lignes d'alimentation).

La contribution au pic d'une connexion d'entrée/sortie est calculée en fonction de la position de l'intervalle contenant le point d'entrée/sortie (pour le cas d'un point fixé, l'intervalle est réduit à ce point). Se reporter à la figure 32 où a, b, c sont les trois positions possibles de l'intervalle d'entrée/sortie de la connexion i_0 .

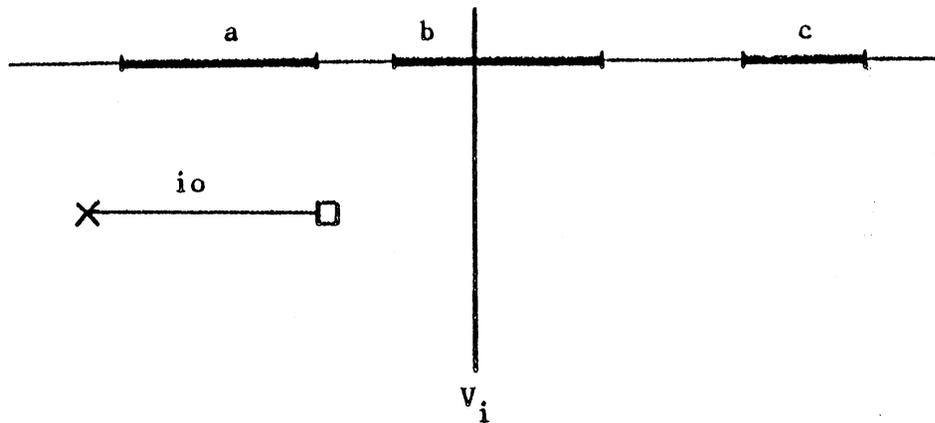


Figure 32. Contribution de i_o au calcul du pic en V_i

- (a) $pic + pic$
- (b) $pic + pic$
- (c) $pic + pic+1$

β) Algorithme

Il nous faut dès à présent fixer le nombre de portes constituant chaque partition. Ce nombre doit être assez petit pour permettre un placement ultérieur efficace en un temps de calcul raisonnable. Des partitions de 6 portes nous semblent le mieux convenir pour la suite de la méthode proposée. Ce choix fournit une quinzaine de partitions lors du placement constructif d'une cellule d'une centaine de portes.

Construction du bloc C^i

Durant cette étape de construction des blocs C^i , le coût considéré est en fait le coût partiel calculé sur le sous-réseau constitué des portes appartenant aux partitions déjà construites. Pour construire le premier bloc C^1 , on considère la connectivité aux points d'entrée/sortie de la cellule sur C^1 . L'algorithme constructif est analogue à celui développé

pour la cellule à une ligne. Dans le cas présent, cependant, l'emplacement exact des portes dans le bloc n'est pas fixé durant cette étape. On s'assurera que chaque bloc ne comporte pas plus de trois portes devant appartenir à C_H^i et C_B^i .

. Itération sur le bloc C^i

Lors de cette itération sur le bloc C^i , on minimise le coût réduit $\|P_i\|$ au moyen d'échanges par paire des portes de C^i et de celles n'appartenant pas encore à une partition.

L'algorithme de partitionnement en blocs consiste donc à déterminer successivement par la méthode constructive suivie de l'amélioration itérative, les blocs C^1, C^2, \dots puis C^k , sans fixer à l'intérieur de chaque bloc les positions des portes.

c) Partitionnement horizontal

Le partitionnement horizontal consiste à affecter dans chaque bloc C^i , trois portes à C_H^i et trois portes à C_B^i .

. Construction d'un partitionnement initial

Cette étape est initialisée avec l'ensemble des portes affectées à C_H et C_B durant III-1.1.a, et avec l'ensemble des points d'entrée/sortie de la cellule.

Lors du traitement d'un bloc C^i , on affecte alternativement une porte de C^i à C_H^i , puis une porte à C_B^i . On sélectionne ainsi sur chaque ligne de la cellule la porte la plus connectée aux portes déjà placées sur la ligne et aux points d'entrée/sortie situés sur ce même côté.

Ce procédé construit des lignes C_H et C_B très connectées, ce qui réduit le nombre de connexions transverses.

• Amélioration itérative

Lors de l'amélioration itérative du partitionnement

horizontal, le vecteur coût $\begin{pmatrix} L_{mt} \\ L_t \end{pmatrix}$ est minimisé.

L_{mt} représente le maximum des "longueurs" des connexions transverses, L_t la "longueur" totale des connexions transverses.

Cette "longueur" n'est pas la longueur réelle de la connexion transverse puisque dans chaque bloc les positions exactes des portes ne sont pas encore déterminées. C'est une fonction du nombre de verticales V_i entre les points les plus proches des deux parties internes de l'équipotentielle transverse. Soient AB et CD les parties internes d'une connexion transverse (figure 33), la "longueur" est alors $1 + \text{nombre de verticales } V_i \text{ entre AB et CD}$, soit 3 dans l'exemple proposé.

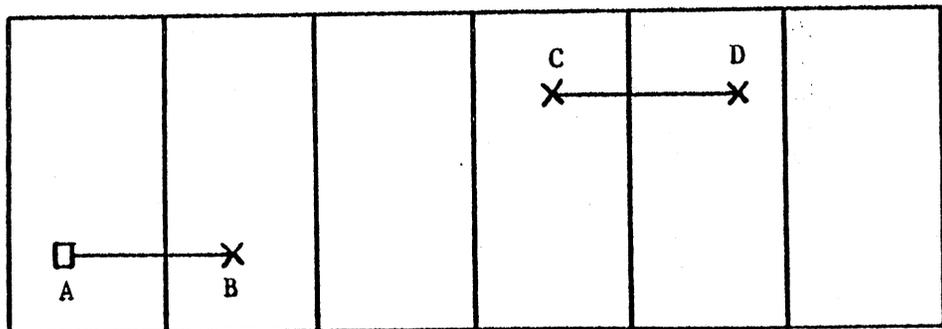


Figure 33. La longueur de la connexion transverse AB - CD est 3

Un algorithme d'échanges par paire est appliqué pour minimiser ce coût, ce qui a pour effet de réduire le nombre de connexions transverses longues et de conserver les courtes. Ces échanges sont effectués dans chaque bloc C^i entre les portes de C_H^i et C_B^i , et ce séquentiellement dans C^1, C^2, \dots, C^k . Le procédé est réitéré jusqu'à ce qu'il n'y ait plus d'amélioration.

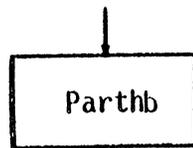
d) Positionnement des portes

Dans cette dernière étape du placement constructif, les portes sont positionnées de manière exacte dans la cellule. Les blocs C^1, C^2, \dots, C^k sont examinés successivement. Pour chaque bloc C^i , les $3! \times 3! = 36$ configurations possibles des portes de C_H^i et C_B^i sont formées et la longueur totale des connexions internes entre les portes des blocs C^1, C^2, \dots, C^i est évaluée. La configuration minimisant cette longueur est retenue. Le placement initial est parfaitement déterminé lorsque le dernier bloc a été examiné.

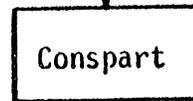
e) Méthode générale - complexité

Le placement constructif initial est donc réalisé, à partir de la description logique du circuit, au moyen de quatre procédures correspondant aux quatre étapes de la méthode.

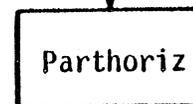
Description logique du circuit
(liste des équipotentielles)



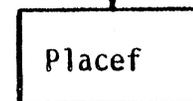
Certaines portes appartiennent à C_B ou C_H pour minimiser le nombre de connexions transverses



La cellule est partitionnée en blocs C^i . La position exacte des portes n'est pas définie.



Les blocs C^i sont partitionnés en C_H^i et C_B^i .



Fin du placement constructif

En fait, la procédure parthb ne concerne pas directement le placement constructif des portes. Elle permet simplement, si le concepteur le souhaite, d'imposer, avant placement, à certaines portes d'appartenir à C_H ou à C_B . Elle ne concerne qu'un nombre petit de portes et pourra ne pas être appliquée lors de la construction du placement.

La cellule considérée comprend n portes. Le nombre moyen des équipotentielles est alors $m = 3n$. Il convient d'examiner les ordres de complexité de Conspart, Parthoriz et Placef pour en déduire la complexité de l'algorithme général de placement constructif.

Conspart : évaluons le nombre d'opérations élémentaires effectuées lors de la recherche de la partition C^{j+1} ($j \in [0, n \text{ div } 6 - 1]$), c'est-à-dire au moment où $6j$ portes ont été affectées aux blocs C^1, C^2, \dots, C^j .

Lors de la construction de C^{j+1} , la $(i+1)^{\text{ème}}$ porte ($i \in [0, 5]$) est choisie parmi les $n - 6j - i$ portes non encore affectées à un bloc, et l'évaluation du vecteur coût nécessite $6j + i$ opérations. Remarquons que le nombre d'opérations nécessaires pour sélectionner la première porte ($i = j = 0$) est égal à n . Dans le cas général ($i \neq 0$ et $j \neq 0$), nous obtenons $\sum_{i=0}^5 (n - 6j - i)(6j + i)$ opérations lors de la construction du bloc C^{j+1} .

Lors de l'amélioration itérative du bloc C^{j+1} , $6(n - 6(j+1))$ échanges sont effectués pour chacun desquels $6(j+1)$ opérations permettent d'évaluer le vecteur coût, soit donc $6(n - 6(j+1)) \times 6(j+1)$ opérations.

Nous obtenons alors pour la recherche de toutes les partitions de

la cellule

$$S = \sum_{j=0}^{ndiv6-1} \left(\sum_{i=0}^5 (n-6j-i)(6j+i) + 36(n-6(j+1))(j+1) \right).$$

Faisons l'hypothèse simplificatrice (pour le calcul de cette expression) que n est multiple de 6, i.e. $n = 6N$. Cela ne modifie en rien le degré du polynôme recherché, et par conséquent l'ordre de complexité de la procédure. En effet, dans le cas où $n \neq 6N$, les $n - 6n \text{ div } 6 = n \text{ reste } 6$ éléments "supplémentaires" forment la dernière partition sans calculs dépendant du nombre total d'éléments.

On obtient finalement

$$S = 72 N^3 - 37 N \quad \text{ce qui porte à } n^3 \text{ l'ordre de complexité du partitionnement vertical.}$$

Parthoriz : évaluons le nombre d'opérations élémentaires effectuées lors du partitionnement horizontal d'un bloc C^{j+1} .

Lors de la construction de C_H^{j+1} , le premier élément est choisi parmi 6, le second parmi 4 et le troisième parmi 2, le nombre d'opérations nécessaires à l'évaluation du vecteur coût étant $3j+i$ ($i \in [0,2]$), ce

qui donne $\sum_{i=0}^2 (6-2i)(3j+i)$ opérations.

Lors de la construction de C_B^{j+1} , le premier élément est choisi parmi 5, le second parmi 3 et le troisième sans calculs, ce qui donne

$\sum_{i=0}^1 (5-2i)(3j+i)$ opérations.

L'amélioration itérative sur C^{j+1} est réalisée au moyen de $3 \times 3 \times 6(j+1)$ opérations.

Le calcul, effectué avec la même hypothèse que précédemment, don-

ne pour l'ensemble de toutes les partitions

$$S = \sum_{j=0}^{ndiv6-1} \left(\sum_{i=0}^2 (6-2i)(3j+i) + \sum_{i=0}^1 (5-2i)(3j+i) + 54(j+1) \right)$$

$$= 57 N^2 + 8 N$$

ce qui donne un ordre de complexité n^2 pour le partitionnement horizontal.

Placef : Le placement final des portes est déterminé, dans chaque bloc C^{j+1} , après $36 \times 6 (j+1)$ opérations élémentaires, soit pour l'ensemble de la cellule (c'est-à-dire les N blocs)

$$S = \sum_{j=0}^{N-1} 6 \times 36(j+1) = 108 N^2 + 108 N$$

ce qui donne un ordre n^2 de complexité du placement final.

Procédure	Nombre d'opérations	Complexité
Conspart	$72 N^3 - 37 N$	n^3
Parthoriz	$57 N^2 + 8 N$	n^2
Placef	$108 N^2 + 108 N$	n^2
Algo.général	$72 N^3 + 165 N^2 + 79 N$	n^3

L'ordre de complexité n^3 de cet algorithme constructif de placement assure une grande rapidité d'exécution des programmes correspondants pour quelques centaines de portes.

III - 1.2. Amélioration itérative du placement

a) Coût

Des transformations sont effectuées sur le placement initial pour minimiser un vecteur coût C qui reflète bien les caractéristiques d'un bon placement, c'est-à-dire d'un placement permettant un tracé facile des connexions. Divers critères sont minimisés durant cette étape.

- . P_m , le maximum de la fonction de P_{ic} . Comme pour la cellule à une ligne d'alimentation, minimiser ce critère revient à minimiser la largeur, et par conséquent la surface de la cellule.

- . N_p , nombre de points où ce maximum est atteint. Ce critère permet de bien préparer le tracé, qui sera d'autant plus facile que N_p est petit.

- . N_c , nombre de croisements des connexions transverses. Les passages diffusés utilisés pour résoudre les croisements inévitables sont coûteux en surface et induisent des capacités importantes. Minimiser N_c permet donc un gain en surface et en performance.

- . L_{mt} , maximum des longueurs des connexions transverses. Les connexions transverses sont d'autant plus difficiles à tracer qu'elles sont longues. Il est donc important de réduire les longueurs des plus longues. En revanche, les connexions transverses entre portes rapprochées sont conservées et privilégiées.

- . L_t , longueur totale des connexions transverses.

- . N_t , nombre des connexions transverses.

- . L_c , longueur totale des connexions.

Plusieurs ordres de priorité pourront être établis entre ces critères pour constituer le vecteur coût à minimiser. Cependant, le vecteur $\|P_m, N_p, L_{mt}, L_t, N_t, L_c\|$ nous semble le plus approprié pour fournir un placement pour lequel la méthode de tracé proposée (§III-2) sera d'une grande efficacité.

Evaluation du nombre de croisements des connexions transverses

Pour évaluer N_c , il nous faut considérer deux à deux les équipotentielles entre C_H et C_B . Chacune de ces équipotentielles e se décompose en e_H (partie interne à C_H), e_B (partie interne à C_B) et e_t (partie transverse).

Remarquons que e_H et e_B peuvent être réduites à un seul point.

Il existe un croisement entre e_1 et e_2 si les conditions suivantes sont remplies :

$$\text{borninf}(e_{1H}) < \text{borninf}(e_{2H})$$

$$\text{bornsup}(e_{1H}) < \text{bornsup}(e_{2H})$$

$$\text{borninf}(e_{2B}) < \text{borninf}(e_{1B})$$

$$\text{bornsup}(e_{2B}) < \text{bornsup}(e_{1B})$$

Ces conditions caractérisent les quatre seuls cas de croisement entre e_1 et e_2 (figure 34).

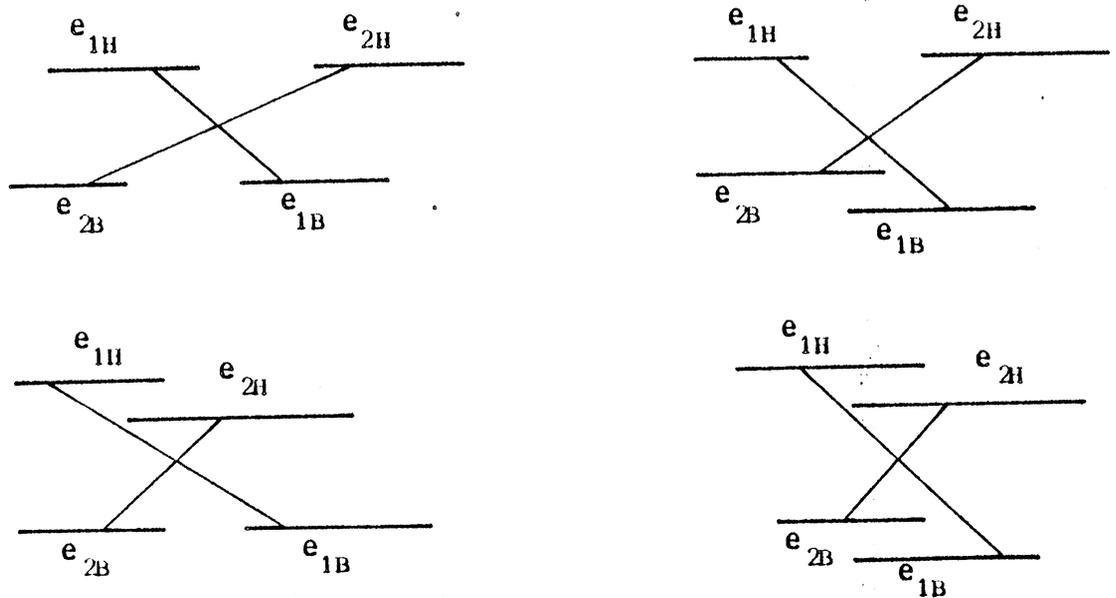


Figure 34. Les quatre cas de croisements entre e_1 et e_2

Calcul de la longueur d'une connexion transverse

Deux cas peuvent se présenter :

- . les deux parties internes à chaque ligne de la connexion transverse se recouvrent (figure 35-a);
- . les deux parties internes à chaque ligne de la connexion transverse ne se recouvrent pas (figure 35-b).

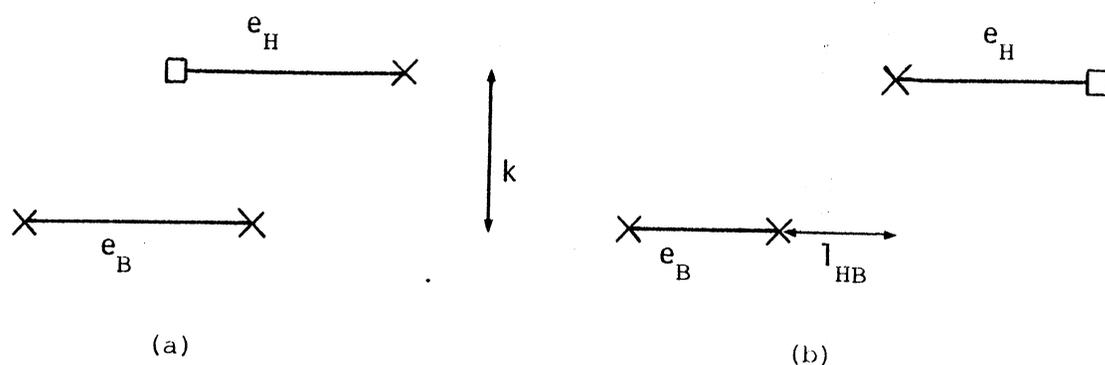


Figure 35. Calcul de la longueur des connexions transverses
 (a) $l_{HB} = 0$
 (b) $l_{HB} \neq 0$

La longueur de la connexion transverse est donnée par $l_{HB} + k$, où l_{HB} représente la différence des abscisses des deux extrémités les plus proches dans le cas où e_H et e_B ne se recouvrent pas ($l_{HB} = 0$ sinon) et k une valeur moyenne et pondérée de la distance séparant verticalement les deux parties de l'équipotentielle et égale à $p(P_m - P_{moy})$. P_{moy} est le pic moyen calculé entre les deux extrémités les plus éloignées de e_H et e_B et p un réel supérieur à 1 fixé suivant l'importance donnée aux connexions transverses par rapport aux connexions internes.

b) Algorithme et complexité

L'algorithme utilisé est similaire à celui défini pour l'amélioration itérative du placement dans la cellule à une ligne. L'échange systématique de toutes les paires de portes est effectué dans le but de minimiser le vecteur coût. Cependant, pour la cellule à deux lignes, deux types d'échanges sont considérés : l'échange entre deux portes de C_H ou de C_B , l'échange entre une porte de C_H et une porte de C_B .

L'algorithme revient alors à appliquer successivement trois algorithmes d'itérations

- sur les portes de C_H ,
- sur les portes de C_H et C_B simultanément,
- sur les portes de C_B .

L'algorithme s'arrête lorsqu'une amélioration du coût n'est plus possible après échange systématique des

$$2 \times \frac{1}{2} \left(\frac{n}{2} \left(\frac{n}{2} - 1 \right) \right) + \frac{n \times n}{2 \times 2} = \frac{n}{2} (n-1)$$

paires de portes. Le vecteur coût étant évalué suivant un algorithme d'ordre n , cet algorithme d'amélioration itérative est donc polynomial d'ordre n^3 , ce qui assure une exécution rapide des programmes correspondants.

III - 2. TRACE DES CONNEXIONS

Le problème du tracé des connexions dans la cellule à deux lignes est décomposé en deux. On se préoccupera tout d'abord des connexions internes à C_H et à C_B , puis des connexions transverses.

III - 2.1. Connexions internes

Pour le tracé des connexions internes, la cellule à deux lignes est décomposé en trois zones :

- . la zone IH correspondant aux trois premières lignes de C_H à partir de l'alimentation,

- . la zone 1B correspondant aux trois premières lignes de C_B à partir de l'alimentation
- . la zone 2, comprise entre l_1 et l_2 , les quatrièmes lignes de C_B et C_H à partir de l'alimentation.

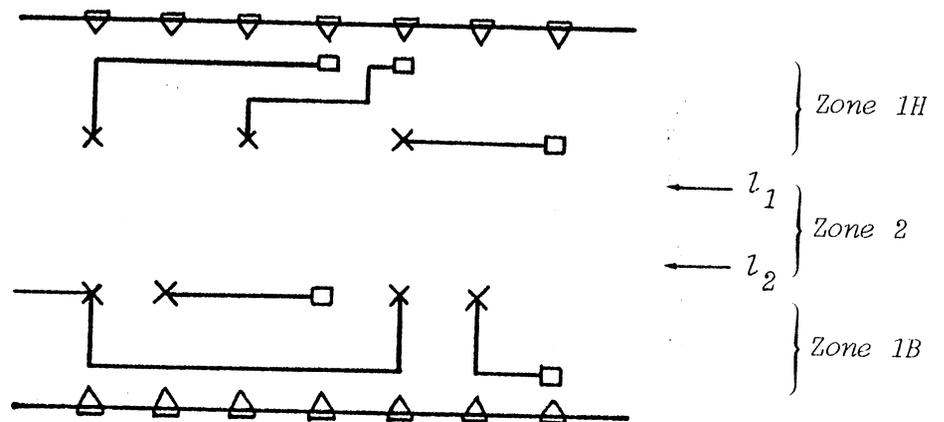


Figure 36. l_1 et l_2 , quatrième ligne à partir de l'alimentation

a) Tracé des connexions dans les zones 1H et 1B

Les zones 1H et 1B sont étudiées séparément. Dans chacune de ces zones, les connexions sont traitées de la même manière que pour la cellule à une ligne, et ce en considérant le pic total de la cellule : coudage des connexions et tracé des segments dans les intervalles de la troisième ligne.

b) Tracé des connexions dans la zone 2

Les connexions de la zone 2 sont traitées dans un premier temps sous forme de segments parallèles aux alimentations.

Considérons la fonction de pic f' définie en supprimant les connexions tracées dans les zones 1H et 1B, et P'_m son maximum.

Au moins P'_m lignes seront donc nécessaires pour le tracé des segments dans la zone 2. Cependant, dans une cellule à deux lignes, un segment appartient à l'une ou l'autre des lignes. On ne peut donc plus assurer que les segments compris entre ℓ_1 et ℓ_2 pourront être tracés sur P'_m lignes (comme dans le cas de la cellule à une ligne). L'algorithme RS défini précédemment est modifié pour tenir compte de cette nouvelle contrainte.

Ces segments pourront effectivement être tracés sur P'_m lignes entre ℓ_1 et ℓ_2 s'il existe au plus :

- 0 "trou" sur les verticales de pic P'_m ,
- 1 "trou" sur les verticales de pic P'_m-1 ,
- 2 "trous" sur les verticales de pic P'_m-2 ,

et ainsi de suite, où les "trous" sont définis comme étant des positions non occupées par les segments à l'intérieur de chaque ligne. La figure 37 décrit l'exemple d'un tracé comprenant 0 "trou" en A, B et C, 1 "trou" en D, E et F.

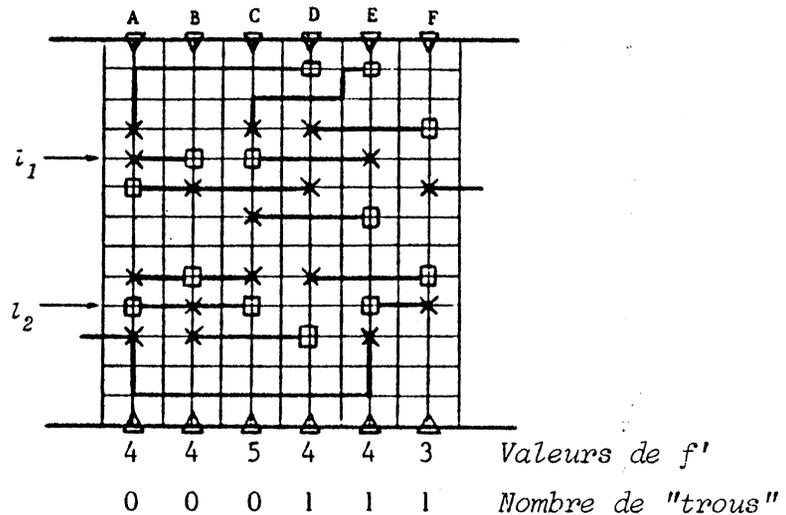


Figure 37. Nombre de "trous" dans la cellule

Une fonction "trous autorisés" est donc définie à partir de la fonction de pic, et recalculée après le remplissage de chaque ligne. On placera donc sur une ligne un ensemble de segments disjoints couvrant tous les points à 0 "trou autorisés" (ou le plus possible), et le plus possible de points à 1, 2, ... "trous autorisés".

Dans l'exemple de la figure 38, où seules les connexions entre l_1 et l_2 ont été représentées, le tracé ne satisfait pas à ces conditions. Notons la présence d'un "trou" sur la verticale 8, alors que la valeur autorisée est 0. Le tracé est effectué sur $P_m^i + 1$ lignes (c'est-à-dire 10 dans l'exemple).

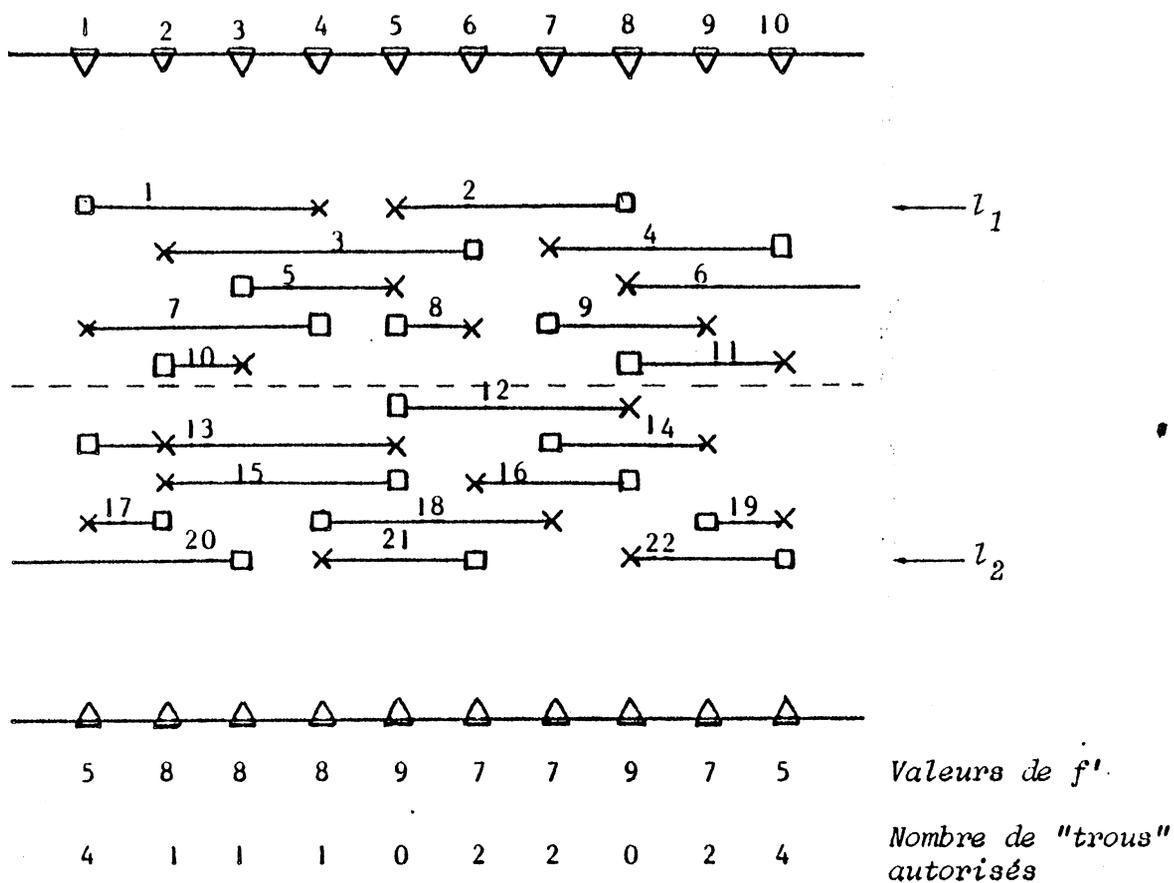


Figure 38. Les segments sont tracés sur P'_m+1 lignes

Un algorithme de tracé TS2 des segments dans la zone 2 est proposé (figure 39). Il traite alternativement une ligne de C_H , puis une ligne de C_B , et ainsi de suite d'une partie à l'autre de la zone 2.

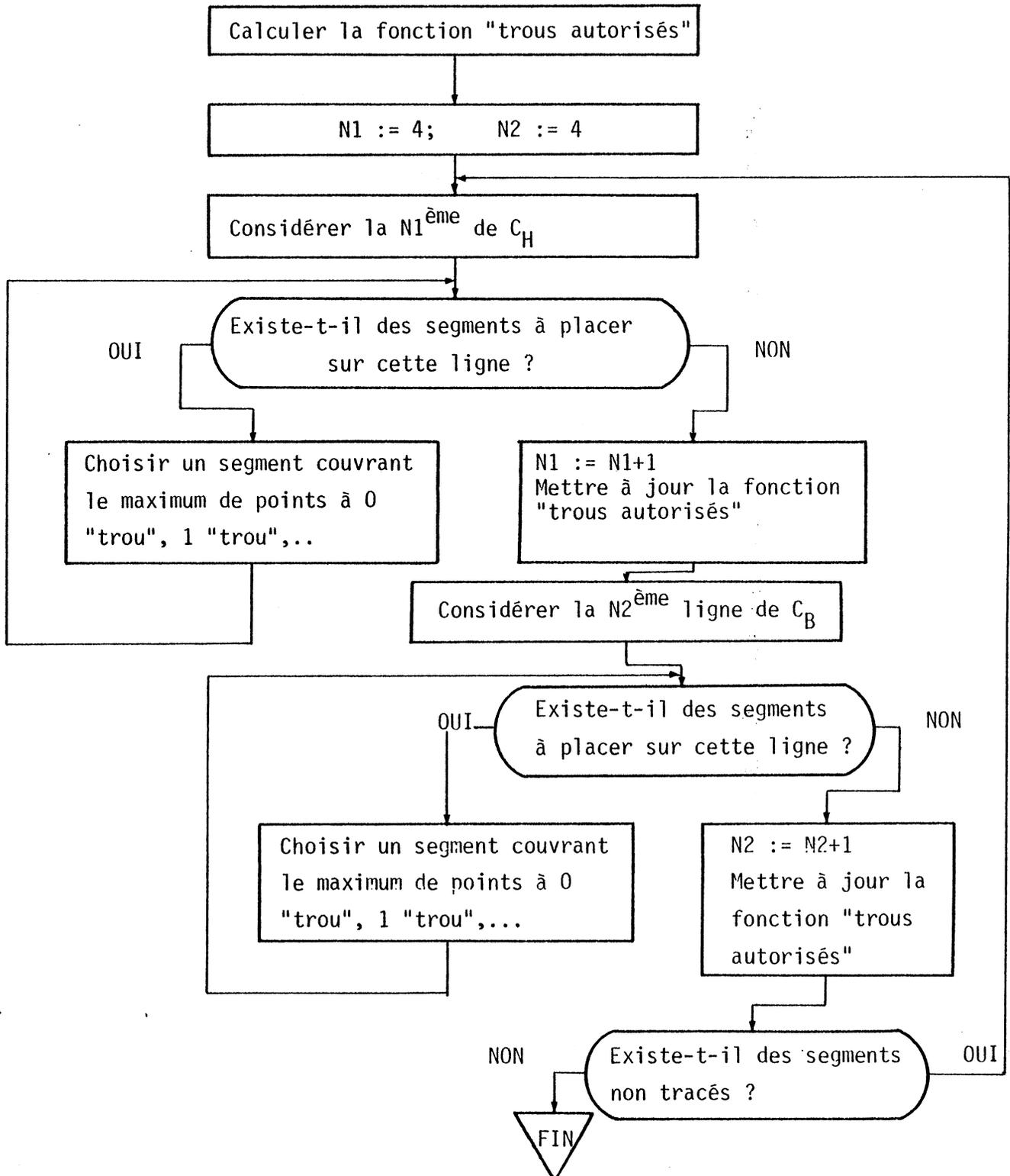


Figure 39. Algorithme TS2, tracé des segments dans la zone 2

Cet algorithme appliqué à l'exemple de la figure 38, permet le tracé des segments de la zone 2 sur P'_m lignes (figure 40).

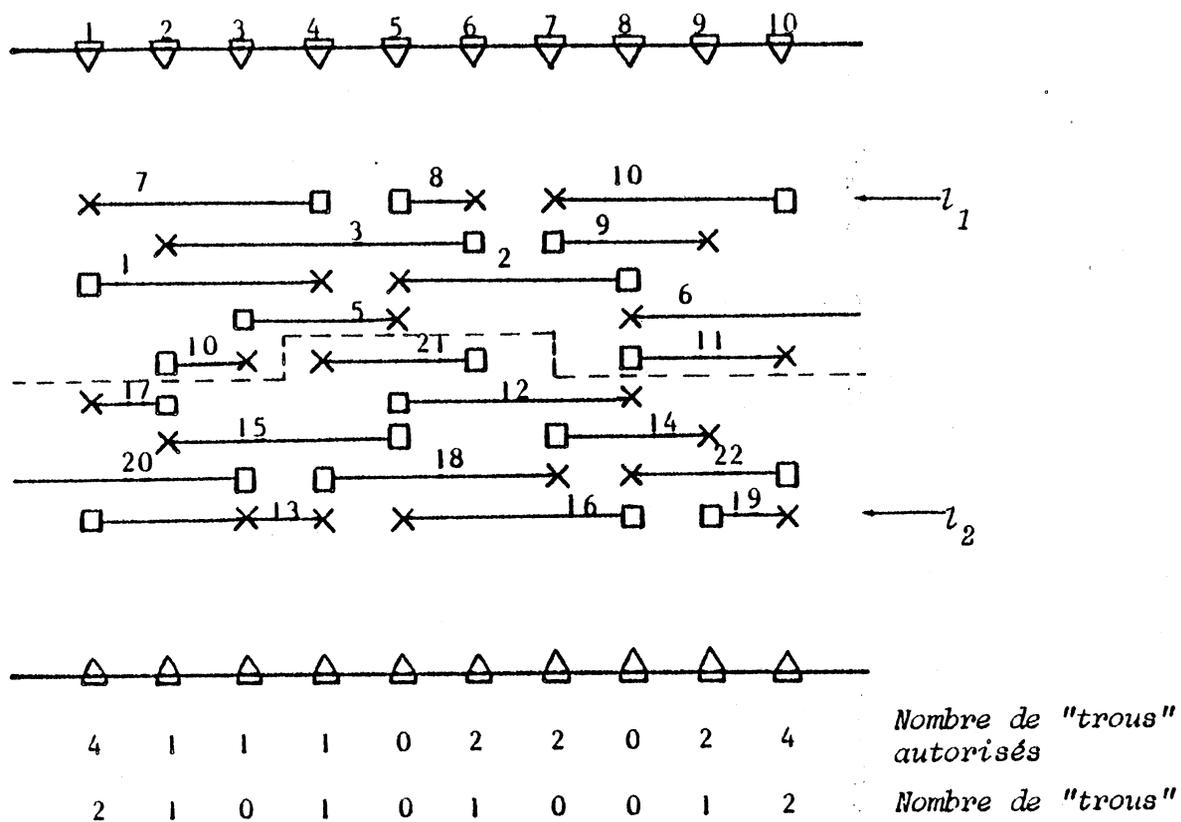
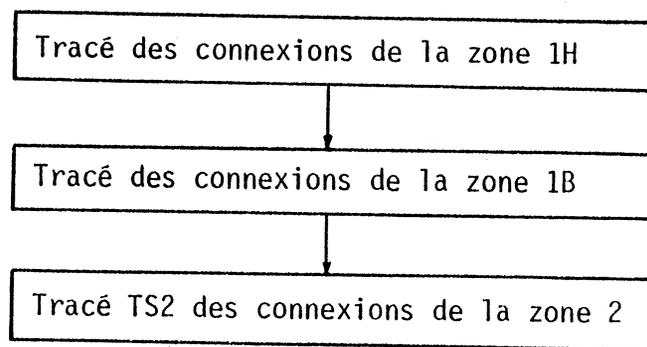


Figure 40. Exemple de la figure 39 après l'algorithme TS2

Lorsque les conditions sur le nombre de trous ne sont pas remplies pour permettre un tracé sur P'_m lignes, les segments sont déformés en utilisant ces trous, ce qui permet le plus souvent de diminuer la largeur du circuit.

c) Algorithme de tracé des connexions internes

L'algorithme de tracé des connexions internes est donc composé de trois procédures indépendantes.



La faible complexité de cet algorithme, de l'ordre de m^2 (m étant le nombre de connexions internes à traiter), assure une exécution rapide des programmes correspondants.

Par cette méthode, toutes les connexions internes sont tracées de manière automatique; la qualité de la solution dépend alors uniquement du nombre de lignes nécessaires au tracé, et par conséquent de la largeur du circuit.

III - 2.2. Connexions transverses entre C_H et C_B

Lors de l'étape de placement des portes, le nombre des connexions transverses a été minimisé. De plus, les connexions transverses les plus longues ont été supprimées. Il ne reste donc à tracer qu'un nombre réduit de connexions transverses relativement courtes. Pour ce faire, les deux parties de la cellule sont écartées pour permettre le passage de toutes les connexions.

A ce jour, les programmes de traitement automatique de cette étape n'ont pas été réalisés. En effet, des connexions de forme très irrégulière doivent être établies et un concepteur humain parvient plus facilement à les dessiner qu'un programme aussi sophistiqué soit-il.

Cependant, un traitement interactif de cette étape est envisagé. Il est justifié par le petit nombre de connexions à traiter et le fort encombrement de la cellule à cette étape. Il sera utile, en particulier, d'exploiter la propriété de dédoublement de la grille sur la porte MD-MOS. Ainsi, le tracé de certaines connexions transverses sera grandement facilité.

III - 3. Exemples - RemarquesExemple 1. Partie de circuit logique

{ 38 portes
40 équipotentielles

Version a

Solution obtenue par les méthodes automatiques de placement des portes et de tracé de segments. Tracé manuel des équipotentielles d'Entrée/sortie et transverses.

Surface S = 722.

Version b

Solution manuelle (même placement).

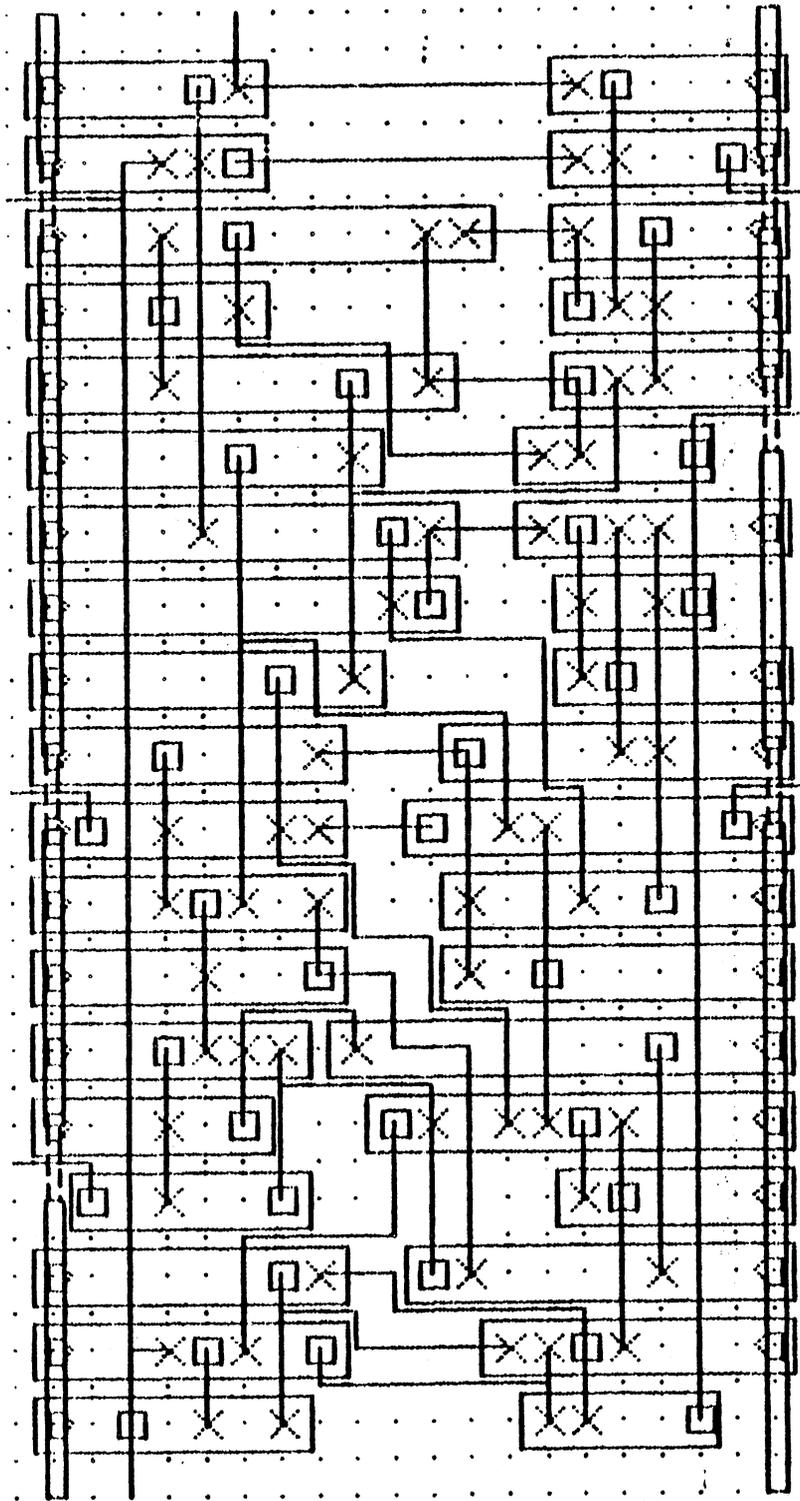
Surface S = 508.

Amélioration en surface

A = 29,6%.

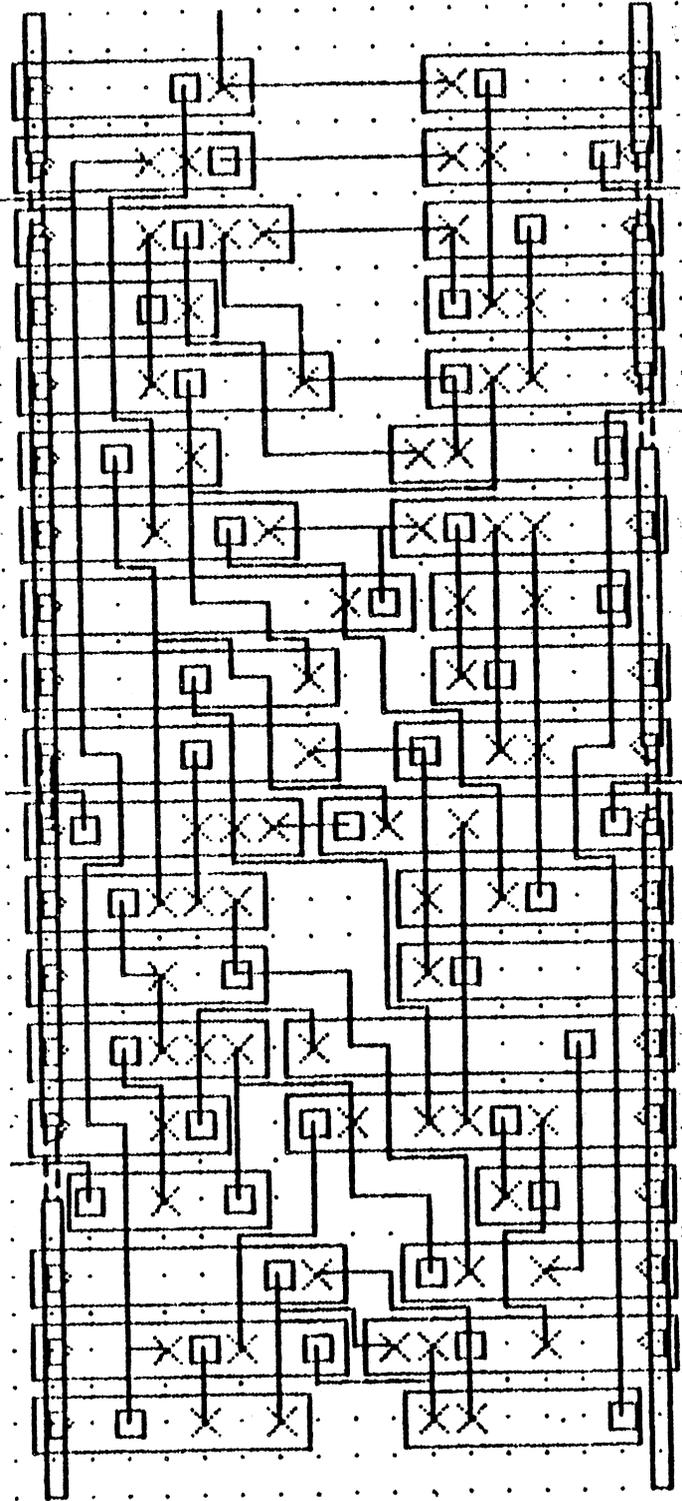


Exemple 1



version a.

Example 1



version b

Exemple 2

Additionneur 4 bits avec retenue anticipée

{ 48 portes
{ 57 équipotentielles

Version a

Solution obtenue par les méthodes automatiques de placement des portes et de tracé de segments. Tracé manuel des équipotentielles d'Entrée/sortie et transverses.

Surface S = 1020.

Nombre de croisements 2

Version b

Solution manuelle (même placement)

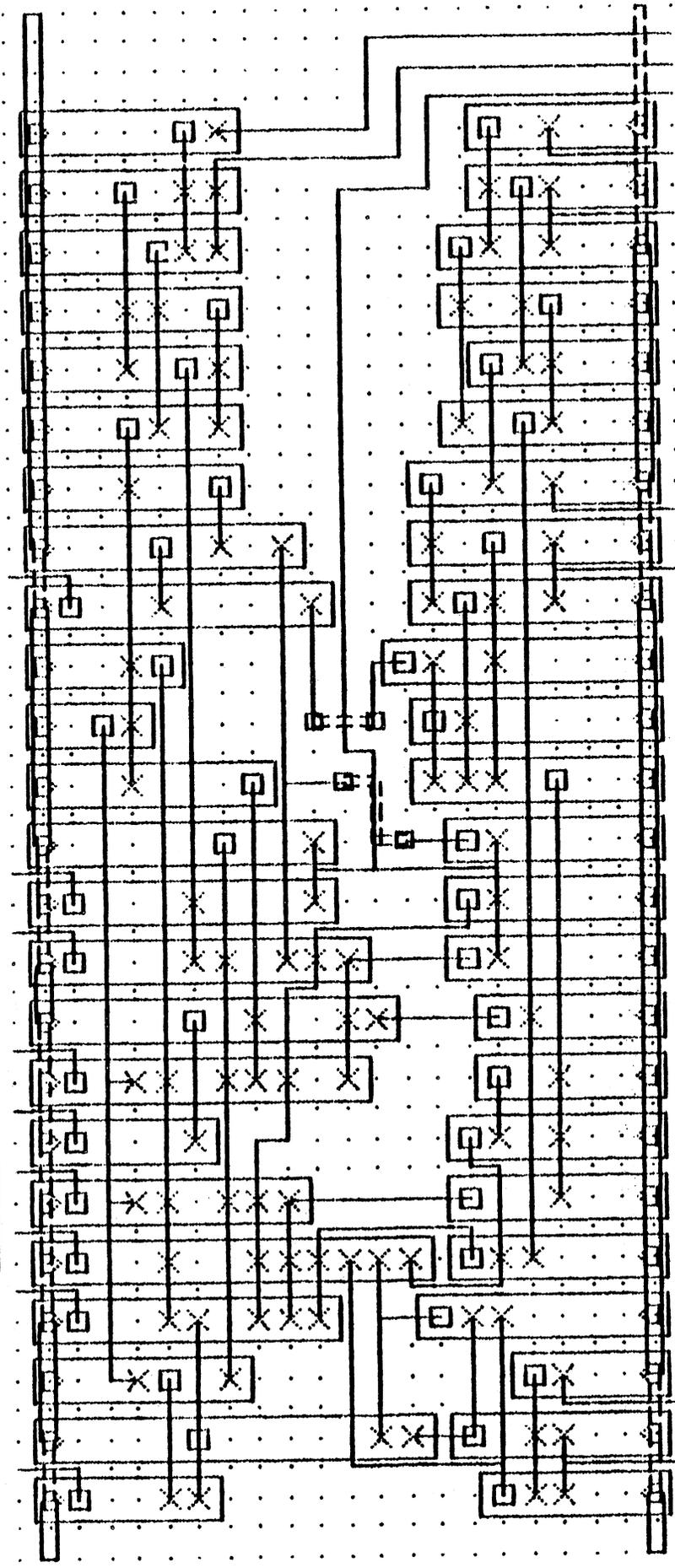
Surface S = 918.

Nombre de croisement 1

Amélioration en surface

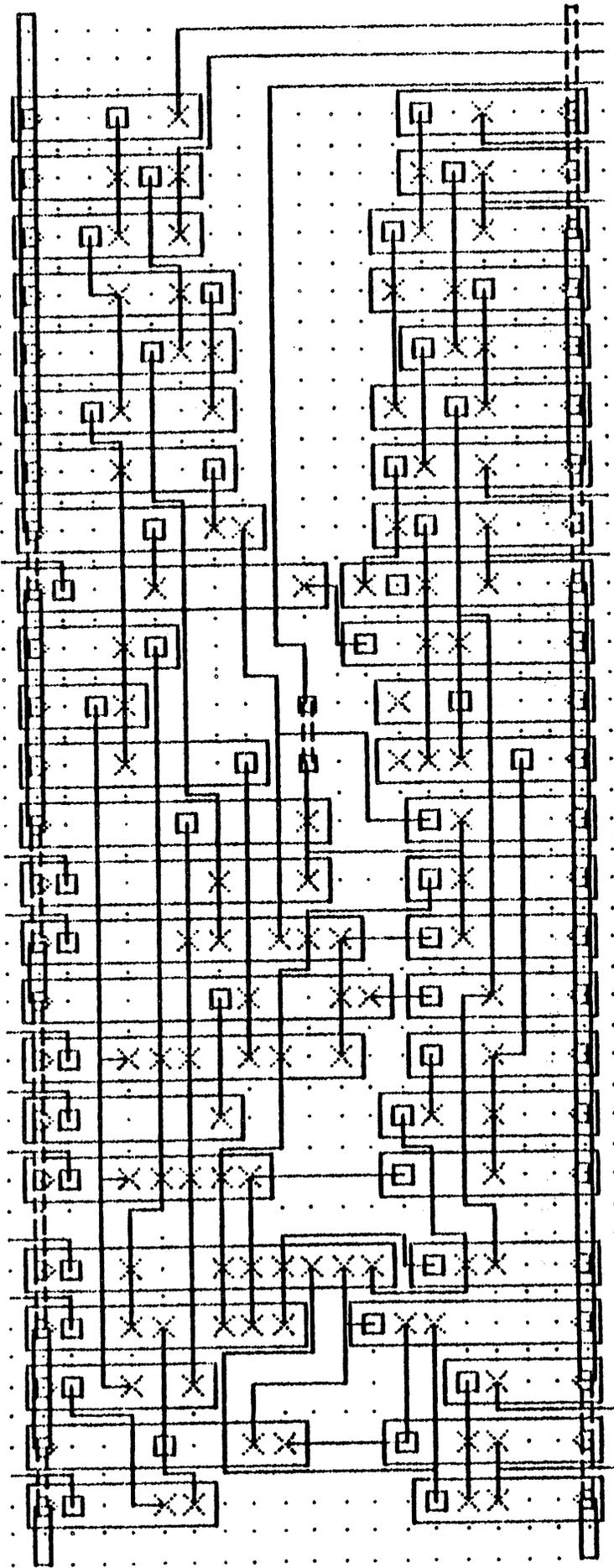
A = 10%.

Exemple 2.



version a

Exemple 2



version b

Deux exemples illustrent les méthodes développées pour l'implantation automatisée des cellules à deux lignes. Les *versions a* ont été obtenues par application de ces méthodes sauf pour le tracé des équipotentielles d'entrée/sortie et transverses réalisé manuellement. Les solutions manuelles proposées (*versions b*) occupent une surface moindre (29.6% et 10% suivant l'exemple) et réduisent le nombre de croisements des connexions.

La possibilité de dédoubler les grilles sur la porte MD-MOS n'a pas été envisagée lors des méthodes proposées. Cependant, cette facilité pourrait être prise en compte lors du placement des portes et du tracé des connexions pour le traitement automatique des connexions transverses. L'emplacement de la grille, non fixé à priori, peut être choisi à l'extrémité de la porte, ce qui permet un tracé aisé de la connexion transverse (figure 41).

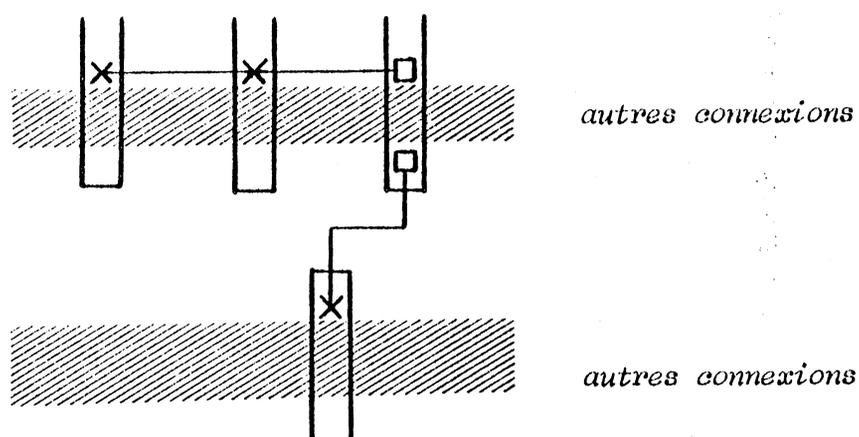


Figure 41. Dédoublage d'une grille

IV - CONCLUSION

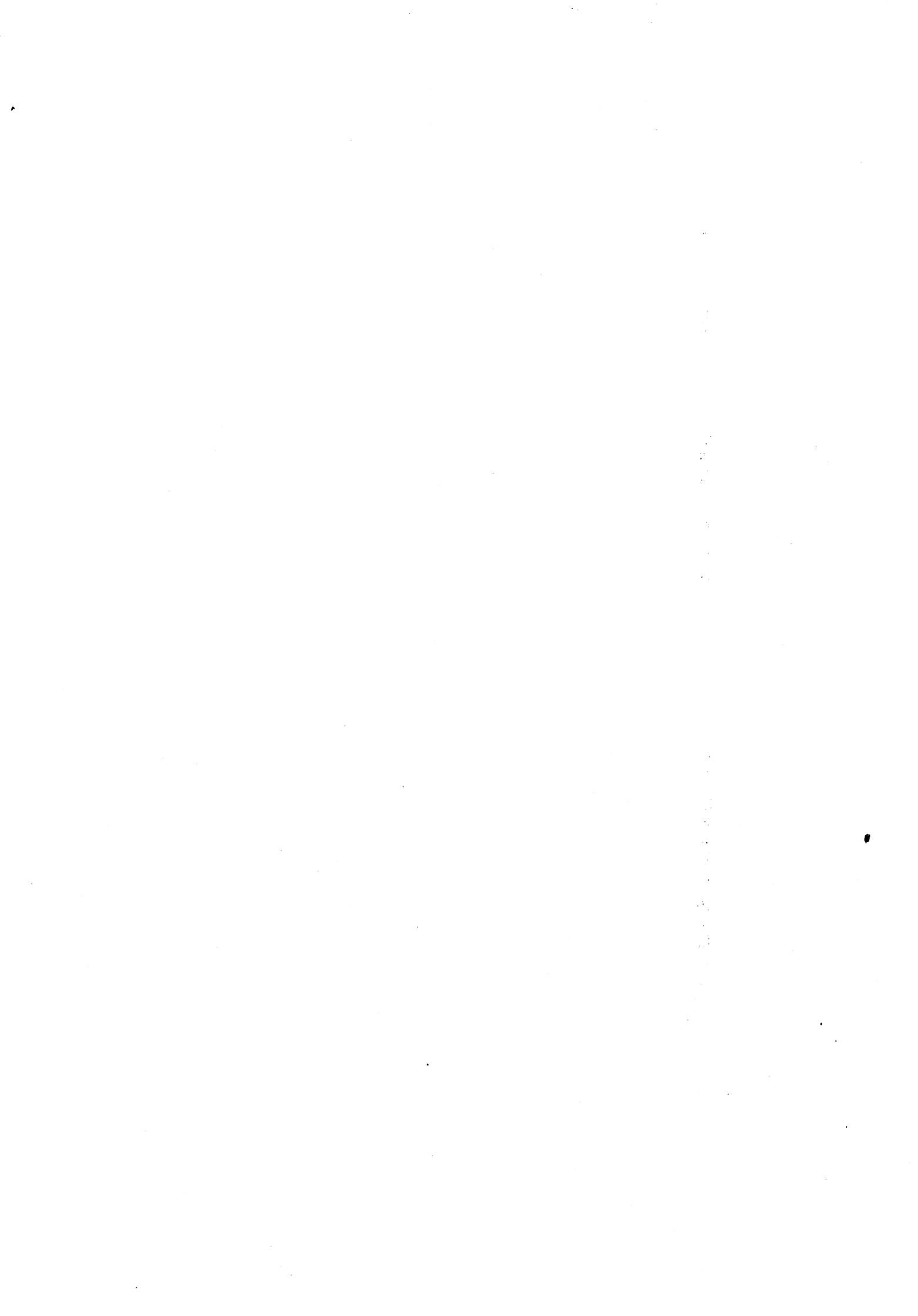
Diverses méthodes de placement et d'interconnexion dans les cellules MD-MOS à une et deux lignes ont donc été définies et validées sur quelques exemples de circuits. Ces méthodes permettent de construire des procédures spécialisées pour :

- le placement sur une ligne,
- le placement sur deux lignes,
- le tracé de segments sur une ligne,
- le tracé de segments sur deux lignes,
- le coudage des connexions,
- le tracé des transverses.

Ces procédures s'inscrivent dans un système interactif (OASIS) pour permettre une implantation efficace des circuits. En effet, une solution entièrement automatique ne manquerait pas de se "bloquer" (en général lors du tracé) et conduirait alors à des circuits inachevés qu'on ne pourrait pas manuellement terminer. L'utilisation interactive de ces procédures apporte donc souplesse et aide considérables lors de l'implantation. Le concepteur suit l'évolution du circuit à implanter et est mieux à même d'agir en imposant des choix.

De plus, une interface avec le système CASSIOPEE du C.N.S. (Grenoble) est en cours de développement et permettra de traiter, à l'aide de ces procédures, un grand nombre de circuits réels. Une évaluation statistique de l'efficacité de ces méthodes pourra alors être proposée.

Les quelques exemples traités à ce jour ont permis de faire apparaître une augmentation de surface de 10 à 30% par rapport aux méthodes manuelles pour des temps de traitement (codage des circuits, placement et interconnexion, lecture des résultats) dix fois moindre. L'interface avec CASSIOPEE permettra un gain supplémentaire en temps.



CONCLUSION

L'évolution vers des circuits structurés est indispensable pour maîtriser la conception des circuits futurs. Le type de symbolisme utilisé au cours de ce travail permet une démarche vers cette structuration et a rendu possible des méthodes de conception sûre, en particulier au niveau de l'implantation automatisée.

Il nous semble que l'élaboration du système OASIS a permis de noter la nécessité des méthodes heuristiques proches de celles utilisées en intelligence artificielle et d'une forte interactivité entre le concepteur et les outils CAO. Dans l'avenir, il est vraisemblable que de tels systèmes seront du type "système expert" composés :

- d'une base de données comportant les descriptions, les règles d'utilisation de la structure hôte; en particulier, les interdictions relatives aux emplacements des modules et au tracé des connexions y figureront ainsi que les coûts à optimiser,
- d'un ensemble d'algorithmes et stratégies dûment répertoriés et évalués.

Le concepteur réalisera alors l'implantation du circuit en "collaboration" avec ce système expert qui, au vu de la stratégie choisie saura proposer un placement des modules ou un tracé des connexions en respectant les règles d'utilisation de la base de données et en optimisant par rapport à des critères donnés.

Cette approche donnera sans doute un bon résultat sur les réseaux prédéfinis dont les structures et règles figées doivent précisément permettre une conception structurée et assistée par ordinateur.

B I B L I O G R A P H I E

- [Ab 72] ABEL L.
"On the automated layout of multi-layer planer wiring and a related graph coloring problem". Coordinated Sc. Lab. Rep. n° 546, University of Illinois, Janvier 1972.
- [Ak 67] AKERS S.B.
"A modification of Lee's path connection algorithm".
IEEE Trans. Electro. Comp., Vol. EC-16, Fev.1967, pp.97-98.
- [AK 72] AKERS S.B.
"Routing". Design Automation of Digital Systems, Edited by M.A. Breuer, Vol.1, Prentice Hall, 1972, pp.283-333.
- [Be 79] BENTLEY J.L. et OTTMAN T.
"Algorithms for reporting and counting geometric intersections",
IEEE Tr. Comp, Vol. C-28, Sept. 1979, pp.643-647.
- [Br 77] BREUER M.A.
"A class of min-cut of Placement Algorithms". 14th Design Automation Conference, New Orleans, Louisiana, Juin 1977, pp.284-290.
- [Br 81] BREUER M.A., FRIEDMAN A.D., IOSUPOVICZ A.
"A survey of the State of the Art of Design Automation".
Computer, Octobre 1981, pp.58-75.
- [Do 80] DONATH W.E.
"Complexity theory and design automation". 17th Design Automation Conference, Minneapolis, Minesota, June 1980, pp.412-419.

- [Du 80] DUNLOP A-E.
"SLIM - The Translation of Symbolic Layouts into Mask Data".
17th Design Automation Conference, Juin 1980, pp.595-602.
- [Fish 78] FISHER R.S.
"A multi-pass, multi-algorithm approach to PCB routing".
15th Design Automation Conference, Las Vegas, Nevada,
June 1978, pp.82-91.
- [Fisk 67] FISK C.J., CASKEY D.L., WEST L.I.
"ACCEL : Automated Circuit Card Etching Layout". Proc. IEEE
Vol. 55, n° 11, 1967, pp.1971-1982.
- [Gi 76] GIBSON D, NANCE S.
"SLIC - Symbolic Layout of Integrated Circuits", 13th
Automation Conference, San Francisco, Juin 1976, pp.434-440.
- [Gl 59] GLASER R.H.
"A quasi-simplex method for designing suboptimal packages
for electronic building blocks". Proc. 1959 Comp. Appl. Symp.
Armour Research Foundation, Illinois Inst. of Tech., pp.100-111.
- [Go 78] GOTO S., KUH E.S.
"An approach to the two-dimensional placement, problem in
circuit layout". IEEE Tr. on Circ. and Syst., Vol. Cas-25,
n°4, Avril 1978, pp.208-21.
- [Go 79] GOTO S.
"A two-dimensional algorithm for the master slice LSI layout
problem". 16th D.A. Conf., San Diego, California, June 1979,
pp.11-17.
- [Ha 80] HANCZAKOWSKI A.
"TRICKY : Symbolic Layout System for Integrated Circuits",
6th ESSCIRC, European Solid State Circuits Conference,
Sept. 1980, Grenoble, pp.123-129.

- [Han 66] HANAN M.
"On Steiner's problem with rectilinear distance". SIAM
Journal of Applied Math., Vol.14, n°2, March 1966, pp.255-265.
- [Han 72] HANAN M., KURTZBERG J.M.
"Placement techniques". Edited by M.A. BREUER. Vol.1,
Prentice Hall, 1972, pp.211-282.
- [Has 71] HASHIMOTO A., STEVENS J.
"Wire routing by optimizing channel assignment within large
apertures". 8th Design Automation Workshop, 1971, pp.155-169.
- [He 80] HEYNS W., SANSEN W., BEKE H.
"A line-expansion algorithm for the general routing problem
with a guaranteed solution". 17th Design Automation Conference,
Minneapolis, Minnesota, Juin 1980, pp.243-249.
- [Hi 69] HIGHTOWER D.W.
"A solution to line-routing problems on the continuous plane".
6th Design Automation Workshop 1969, pp.1-24.
- [Hi 80] HIGHTOWER D.W., BOYD R.L.
"A generalized Channel router", 17th Design Automation
Conference, Minneapolis, Minnesota, Juin 1980, pp.12-21.
- [Hs 80] HSUEH M.Y.
"Symbolic Layout Compaction", NATO Advanced Study Institute
Sogesta - Urbino, Italy, Juillet 1980.
- [Kh 77] KHOKHANI K.H., PATEL A.M.
"The Chip Layout Problem : a 14th Design Automation Conference
1977, pp.291-297.
- [Kr 56] KRUSKAL J.B.
"On the shortest spanning subtree of a graph and the traveling
salesman problem", Proc. Amer. Math. Soc. Vol.7, 1956,
pp.48-50.

- [Ku] 79] KULKARNI K.G., JAYAKUMAR V.
"Ordering of connections for Automatical Routing". IEEE Tr.
on Computers, Vol., C-28, n°10, Oct. 1979, 791-794.
- [Kun 72] KUNTZMANN J.
"Theorie des Réseaux (graphes)". Edité par Dunod, Paris 1972.
- [Kur 65] KURTZBERG J.M.
"Algorithms for Blackplane Formation". Microelectronics in
Large Systems. Spartan Books, 1965, pp.61-76.
- [La 79] LAUTCHER V.
"A min-cut placement algorithm for general cell assemblies
based on a graph representation". 16th D.A. Conf. San Diego,
California, Juin 1979, pp.1-9.
- [Lee 61] LEE C.Y.
"An Algorithm for Path Connections and its Application".
IRE Trans. Electro. Comp., Vol.EC-10, Sept.1961, pp.346-365.
- [Li 63] LITTLE J.D.C. et al
"An Algorithm for the Traveling Salesman Problem". J. Oper.
es., Vol.11, 1963, pp.972-989.
- [Lop 80] LOPEZ A.D. et LAW H.S.
"A Dense Gate Matrix Layout Method for MOS VLSI". IEEE Tr.
el. Dev., Vol. ED 27, n°8, Août 1980.
- [Lor 80] LORENZETTI M.J., SMITH II R.J.
"An implementation of a saturated zone multilayer printed
circuit board router". 17th Design Automation Conference,
Minneapolis, Minnesota, Juin 1980, pp.255-262.
- [Lu 80] LUEBBERT F., ULREY M.
"Gate assignment and pack placement : two approaches compared".
17th DA Conference, Minneapolis, Minnesota, Juin 1980,
pp.472-482.

- [Ma 78] MAJOS J., LARDY J.C.
"The Multidrain MOS transistor". 4th ESSIRC, European Solid State Circuit Conference, 18-21- Sept. 1978, Amsterdam, pp.133-135.
- [Ma 82] MALLADI R.
"Conception électrique et implantation de circuits intégrés".
Thèse de Docteur-Ingénieur, INP de Grenoble, Janvier 1982.
- [Me 80] MEAD C., CONWAY L.
"Introduction to VLSI System". Reading, Massachusetts, Addison-Wesley, 1980.
- [Mi 68] MIKAMI K., TABUSHI K.
"A computer program for optimal routing of printed circuit connectors". IFIPS 1968, pp.1475-1478.
- [Ni 80] NISHIOKA I et al
"An automatic routing system for high density multilayer printed wiring board". 17th Design Automation Conference, Minneapolis, Minnesota, Juin 1980, pp.520-527.
- [Pr 78] PREAS B.T., GWYN C.W.
"Methods for Hierarchical Automatic Layout of Custom LSI Circuit Masks". Design Automation Conference, 1978, pp.206-212.
- [Pr 79] PREAS B.T., VAN CLEEMPUT W.M.
"Routing algorithms for hierarchical IC layout". Proc. ISCAS, Tokyo, 1979.
- [Re 77] REINGOLD E.M., NIEVERGELT J., DEO N.
"Combinatorial Algorithms, Theory and Practice". Prentice Hall, 1977, pp.121-130.
- [Ru 74] RUBIN F.
"An iterative technique for printed wire routing".
11th Design Automation Workshop 1974, pp.308-313.

- [Rub 74] RUBIN F.
"The Lee path connection algorithm". IEEE Trans. on Comp.
Vol. C-23, n°9, Sept. 1974, pp.907-914.
- [Sc 76] SCHWEIKERT D.G.
"A 2-dimensionnal placement algorithm for the layout of
electrical circuits". 13th Design Automation Conference,
San Francisco, California, pp.408-416.
- [Se 80] LEBLOND A., SERRERO G., VERDILLON A.
"Automatic Layout of Symbolic MD-MOS Circuits". 1st ICC Conf.
Port Chester, New-York, Oct. 1980, pp.772-776.
- [Se 81-1] MALLADI R., SERRERO G.
"Implantation automatique des circuits : placement et
interconnexion". Rapport de Recherche n° 238, Laboratoire
IMAG, Mars 1981.
- [Se 81-2] SERRERO G. VERDILLON A.
"Connectique pour la conception des circuits". Colloque
International sur les Nouvelles Orientations des Circuits
Intégrés, Avril 1981, Paris, pp.196-200.
- [Se 81-3] MALLADI R., SERRERO G., VERDILLON A.
"Automatic Placement of Rectangular Blocks with the Intercon-
nection Channels". 18th Design Automation Conference,
Nashville, Tennessee, Juin 1981, pp.419-425.
- [SE 81-4] MALLADI R., SERRERO G.
"OASIS - An Automatic Design Tool for Symbolic Layout of
Circuits", 7ème ESSCIRC, Sept. 1981, Freiburg (RFA), pp.31-33.
- [Se 82] MALLADI R., SERRERO G.
"Placement et interconnexion pour l'implantation automatique
des circuits". L'Onde électrique, Vol.62, n°1, Janvier 1982,
pp.75-87.

- [Sm 80] SMITH V.K., SMITH II R.J., PRESTON P.A.
"COMET - A fast component placer". 17th D.A. Conference,
Minneapolis, Minnesota, Juin 1980, pp.465-471.
- [St 61] STEINBERG L.
"The Blackboard Wiring Problem : A Placement Algorithm".
SIAM Rev. Vol.3, n°1, 1961, pp.37-50.
- [Ta 80] TADA F. et Al
"A fast maze router with iterative use of variable search
space restriction". Design automation Conference, Minneapolis
Minnesota, Juin 1980, pp. 250-254.
- [Tr 81] TRIMBERGER S. et Al
"A structured design methodology and associated software
tools". IEEE Tr. Circ. and Syst. Vol. Cas-28, n°7, Juillet 1981,
pp.618-634.
- [We 81] WESTE N.H.E.
"MULGA - an interactive symbolic layout system for the design
of integrated circuits". The Bell System Techn., J. Vol. 60,
n°6, Juillet-Août 1981, pp.823-857.
- [Wi 74] WILSON D.C., SMITH II R.J.
"An experimental comparison of Force Directed Placement
Techniques". Design Automation Workshop, 1974, pp.194-199.
- [Wi 77] WITTENZELINER E.
"Computer Aided Design of Large-Scale Integrated I2L Logic
Circuits". IEEE J. of Solid State Circ., Vol. SC-12, n°2,
Avril 1977, pp.199-204.
- [Wu 76] WU W. SCHMIDT D.C.
"A floating via router on a saturated zone strategy". IEEE
Tr. on Circ. and Syst., Decembre 1976, pp.830-835.

ANNEXE

—

DESCRIPTION DES CIRCUITS A TRAITER

Le circuit à traiter, pour les programmes d'implantation développés, est décrit sous la forme d'un ensemble de ses équipotentiels. Les informations concernant la nature du point de contact sur la porte (drain, grille), le nombre et la nature des contraintes d'entrée/sortie de l'équipotentielle sont mémorisés.

Cette structure peut être directement déduite de la description du circuit dans la base de données de CASSIOPEE et est codée de la manière suivante en PASCAL.

Le tableau EQUIP contient alors toutes les données nécessaires au problème d'implantation.

```
TYPE ROP = (Rien, Ordre, Posit); %nature de la contrainte d'E/S%
```

```
ZUD = 0..2; %Nombre d'E/S%
```

```
PGDHB = (P,G,D,H,B); %côté de la contrainte d'E/S%
```

```
POINTES = record
```

```
  case COTE : PGDHB of
```

```
    P : ( );
```

```
    G,D,H,B : (case CONTR:ROP of
```

```
      Rien : ( );
```

```
      Ordre : (Rang : integer);
```

```
      Posit : (Inf, Sup : integer))
```

```
  end;
```

```
EQUIPOT = record
```

```
  Num : array [1..10] of integer;
```

```
  PLOT : array [1..10] of (grille, drain);
```

```
  Case NES : ZUD of
```

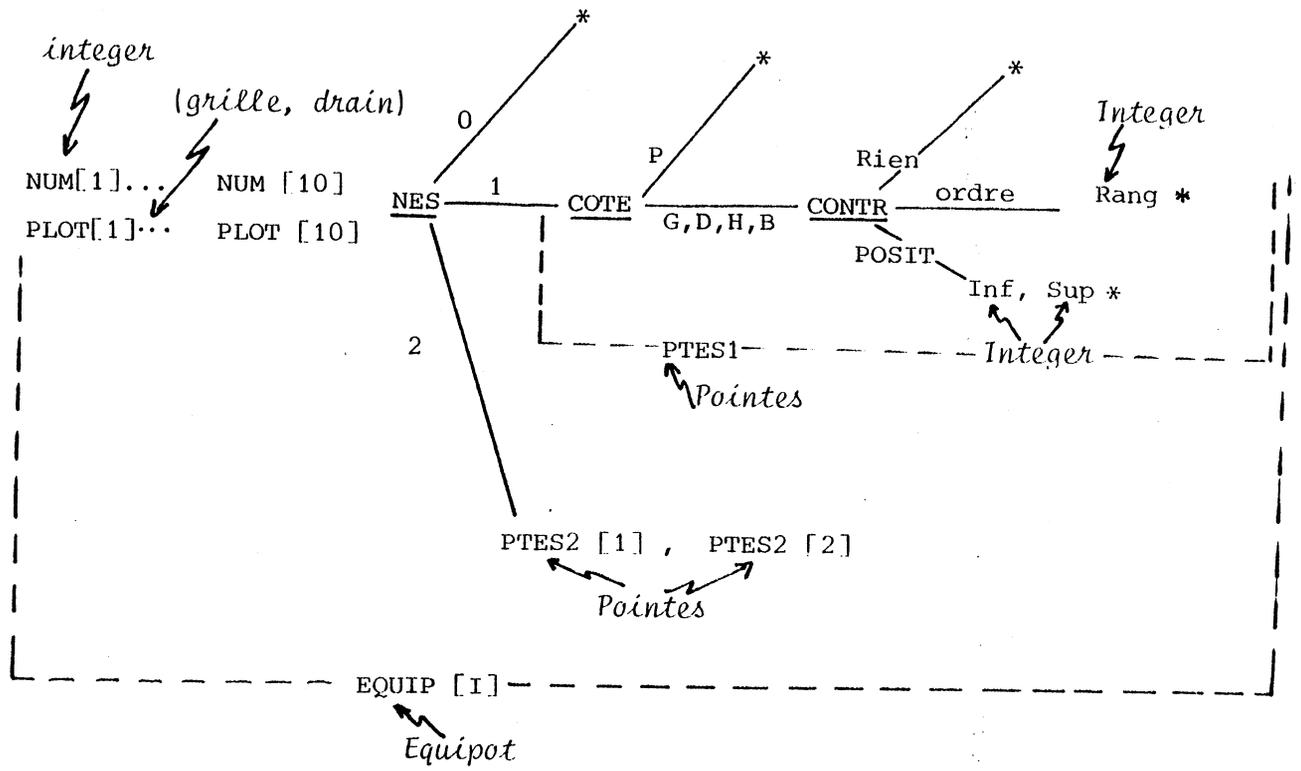
```
    0 : ( );
```

```
    1 : (PTES1 : Pointes);
```

```
    2 : (PTES2 : array [1..3] of Pointes)
```

```
  end;
```

```
VAR EQUIP : Array [1..80] of Equipot;
```



Structure d'une équipotentielle

La suite d'entiers NUM [I] indiquant le numéro des portes connectées à l'équipotentielle est terminée par la valeur sentinelle 0.

A cause des procédures de lecture de données en PASCAL tel qu'il existe au CNS, les quantités non entières de EQUIP [I] seront codées sous forme d'entier de la manière suivante :

Contact { Grille : 1
 { Drain : 2

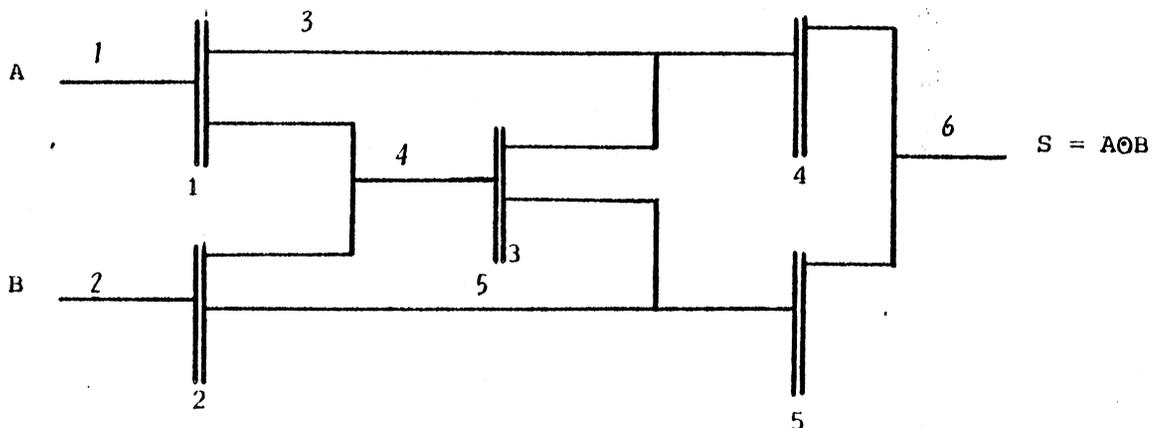
COTE { P : 0
 { G : 1
 { D : 2
 { H : 3
 { B : 4

CONTR { Rien : 1
 { Ordre : 2
 { Position : 3

NES { pas d'E/S : 0
 { 1 E/S : 1
 { 2 E/S : 2

Exemple :

Codage de la disjonction



Disjonction en MD-MOS

Si nous imposons aux entrées A et B d'être sur le côté gauche, A au dessus de B (c'est-à-dire A en 1ère position) et à la sortie S d'être sur le côté droit, le codage de ce circuit sera le suivant :

```

1 1 0 1 1 2 1
2 1 0 1 1 2 2
1 2 3 2 2 4 1 0 0
1 2 2 2 3 1 0 1
2 2 3 2 5 1 0 0
4 2 5 2 0 1 2 1

```

Cette structure permet un traitement simplifié du placement et du tracé dans la cellule car elle reflète parfaitement les différents critères liés à la connectivité du circuit et utilisés dans les procédures de placement et de tracé.

De plus, quand il s'avère nécessaire de travailler sur une structure du circuit par portes plutôt que par équipotentielles, le passage à une telle structure est aisé et permet de fournir pour chaque porte, la liste des équipotentielles qui lui sont rattachées.

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 avril 1974,

VU les rapports de présentation de

- . Madame G. SAUCIER, Professeur
- . Monsieur G. MAZARE, Professeur

Monsieur Gilles SERRERO

est autorisé à présenter une thèse en soutenance pour l'obtention du diplôme de
DOCTEUR-INGENIEUR, spécialité "Génie Informatique".

Fait à Grenoble, le 2 mars 1982

Le Président de l'I.N.P.-G.

Président
de l'Institut National Polytechnique
de Grenoble

