



PAOLA : un système d'optimisation topologique de P.L.A

Tomas Perez Segovia

► To cite this version:

Tomas Perez Segovia. PAOLA : un système d'optimisation topologique de P.L.A. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1985. Français. NNT : . tel-00316330

HAL Id: tel-00316330

<https://theses.hal.science/tel-00316330>

Submitted on 3 Sep 2008

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

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

THESE

présentée à

l'Institut National Polytechnique de Grenoble

pour obtenir le grade de
DOCTEUR DE 3ème CYCLE
«Informatique»

par

PEREZ SEGOVIA Tomás



PAOLA: UN SYSTEME D'OPTIMISATION TOPOLOGIQUE DE P.L.A.



Thèse soutenue le 25 octobre 1985 devant la commission d'examen.

G. MAZARÉ	}	Président
F. ANCEAU		Examineurs
J.L. LARDY		
G. NOGUEZ		

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1982-1983

Président de l'Université : D. BLOCH

Vice-Président : René CARRE

Hervé CHERADAME

Marcel IVANES

PROFESSEURS DES UNIVERSITES :

ANCEAU François	E.N.S.I.M.A.G.
BARRAUD Alain	E.N.S.I.E.G.
BAUDELET Bernard	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	U.E.R.M.C.P.P.
CHERUY Arlette	E.N.S.I.E.G.
CHIAVERINA Jean	U.E.R.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.M.A.G.

.../...

LESSIEUR 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.M.A.G.
MOREAU René	E.N.S.H.G.
MORET Roger	E.N.S.I.E.G.
MOSSIERE Jacques	E.N.S.I.M.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.
PIAU Jean-Michel	E.N.S.H.G.
POLOUJADOFF Michel	E.N.S.I.E.G.
POUPOT Christian	E.N.S.E.R.G.
RAMEAU Jean-Jacques	E.N.S.E.E.G.
RENAUD Maurice	U.E.R.M.C.P.P.
ROBERT André	U.E.R.M.C.P.P.
ROBERT François	E.N.S.I.M.A.G.
SABONNADIÈRE Jean-Claude	E.N.S.I.E.G.
SAUCIER Gabrielle	E.N.S.I.M.A.G.
SCHLENKER Claire	E.N.S.I.E.G.
SCHLENKER Michel	E.N.S.I.E.G.
SERMET Pierre	E.N.S.E.R.G.
SILVY Jacques	U.E.R.M.C.P.P.
SOHM Jean-Claude	E.N.S.E.E.G.
SOUQUET Jean-Louis	E.N.S.E.E.G.
VEILLON Gérard	E.N.S.I.M.A.G.
ZADWORNY François	E.N.S.E.R.G.

PROFESSEURS ASSOCIES

BASTIN Georges	E.N.S.H.G.
BERRIL John	E.N.S.H.G.
CARREAU Pierre	E.N.S.H.G.
GANDINI Alessandro	U.E.R.M.C.P.P.
HAYASHI Hirashi	E.N.S.I.E.G.

PROFESSEURS UNIVERSITE DES SCIENCES SOCIALES (Grenoble II)

BOLLIET Louis
Chatelin Françoise

PROFESSEURS E.N.S. Mines de Saint-Etienne

RIEU Jean
SOUSTELLE Michel

CHERCHEURS DU C.N.R.S.

FRUCIART Robert
VACHAUD Georges

Directeur de Recherche
Directeur de Recherche

.../...

ALLIBERT Michel	Maître de Recherche
ANSARA Ibrahim	Maître de Recherche
ARMAND Michel	Maître de Recherche
BINDER Gilbert	
CARRE René	Maître de Recherche
DAVID René	Maître de Recherche
DEPORTES Jacques	
DRIOLE Jean	Maître de Recherche
GIGNOUX Damien	
GIVORD Dominique	
GUELIN Pierre	
HOPFINGER Emil	Maître de Recherche
JOUD Jean-Charles	Maître de Recherche
KAMARINOS Georges	Maître de Recherche
KLEITZ Michel	Maître de Recherche
LANDAU Ioan-Dore	Maître de Recherche
LASJAUNIAS J.C.	
MERMET Jean	Maître de Recherche
MUNIER Jacques	Maître de Recherche
PIAU Monique	
PORTESEIL Jean-Louis	
THOLENCE Jean-Louis	
VERDILLON André	

CHERCHEURS du MINISTERE de la RECHERCHE et de la TECHNOLOGIE (Directeurs et Maîtres de Recherches, ENS Mines de St. Etienne)

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

PERSONNALITES HABILITEES à DIRIGER des TRAVAUX de RECHERCHE (Décision du Conseil Scientifique)

ALLIBERT Colette	E.N.S.E.E.G.
BERNARD Claude	E.N.S.E.E.G.
BONNET Rolland	E.N.S.E.E.G.
CAILLET Marcel	E.N.S.E.E.G.
CHATILLON Catherine	E.N.S.E.E.G.
CHATILLON Christian	E.N.S.E.E.G.
COULON Michel	E.N.S.E.E.G.
DIARD Jean-Paul	E.N.S.E.E.G.
EUSTAPOPOULOS Nicolas	E.N.S.E.E.G.
FOSTER Panayotis	E.N.S.E.E.G.

.../...

DELHAYE Jean-Marc
DUPUY Michel
JOUVE Hubert
NICOLAU Yvan
NIFENECKER Hervé
PERROUD Paul
PEUZIN Jean-Claude
TAIEB Maurice
VINCENDON Marc

C.E.N.G. (STT)
C.E.N.G. (LETI)
C.E.N.G. (LETI)
C.E.N.G. (LETI)
C.E.N.G.
C.E.N.G.
C.E.N.G. (LETI)
C.E.N.G.
C.E.N.G.

LABORATOIRES EXTERIEURS

DEMOULIN Eric
DEVINE
GERBER Roland
MERCKEL Gérard
PAULEAU Yves
GAUBERT C.

C.N.E.T.
C.N.E.T. (R.A.B.)
C.N.E.T.
C.N.E.T.
C.N.E.T.
I.N.S.A. Lyon

*«Caminante, son tus huellas
el camino, y nada más;
caminante, no hay camino,
se hace camino al andar.»*

Antonio Machado

*a Tatiana
a Carininda
a Marta
a Rodolfo*

RESUME

Lors de la conception des circuits intégrés VLSI, les Réseaux Logiques Programmables (P.L.A.) permettent le dessin automatique des masques à partir d'une description logique. La surface occupée par ces PLAs peut, dans certains cas, s'avérer prohibitive; d'où l'intérêt des méthodes d'optimisation topologique de ceux-ci.

Après avoir défini les différentes représentations possibles des PLAs, nous présentons l'état de l'art en ce qui concerne l'optimisation topologique des PLAs.

La méthode des "Lignes Brisées" est ensuite détaillée en insistant sur les heuristiques choisies ainsi que sur les interactions qui existent entre l'étape d'optimisation et l'étape de tracé des connexions internes.

On termine par une présentation globale du système PAOLA d'optimisation topologique et dessin de PLAs.

MOTS - CLES :

Programmable Logic Array, Conception Assistée par Ordinateur, Conception de Circuits Intégrés, V.L.S.I., Heuristiques, Optimisation Topologique.

RERCIEMENTS

Je voudrais exprimer toute ma reconnaissance à :

- Monsieur **F. ANCEAU** professeur à l'I.N.P.G. qui a bien voulu m'accueillir au sein de l'équipe de Recherche en Architecture des Ordinateurs et qui est à la base du projet PAOLA,

- Monsieur **G. MAZARE**, professeur E.S.I.M.A.G, qui m'a fait l'honneur de présider le jury de cette Thèse,

- Monsieur **J.L. LARDY** chef du département M.C.C au C.N.E.T. Grenoble pour l'aide qu'il m'a donné pendant la rédaction de ce mémoire, ainsi que pour avoir accepté de faire partie de ce jury,

- Monsieur **G. NOGUEZ** professeur à l'université PARIS VI pour avoir accepté de faire partie de ce jury,

- Monsieur **S. CHUQUILLANQUI** avec qui ce travail à été réalisé,

- Messieurs **A. GUYOT** et **J. P. SCHOELLKOPF** pour les nombreuses discussions que nous avons eu ensemble,

- toute l'équipe de Recherche en Architecture des Ordinateurs pour leur soutien pendant si longtemps,

- au **CNET** Grenoble qui à mis à ma disposition tous les moyens nécessaires à la rédaction de cette thèse,

- et enfin à Monsieur **IGLESIAS**, et avec lui toute l'équipe de reprographie de l'IMAG, pour son travail soigné et sa gentillesse.

TABLE DES MATIERES

TABLE DES MATIERES

TABLE DES MATIERES

CHAPITRE I

INTRODUCTION

I.A	LA CONCEPTION DES CIRCUITS INTEGRES	3
I.A.1	Optimisation Locale Et Optimisation Globale	7
I.B	LES PLAS	9
I.B.1	Les Différentes Représentations Des PLAs	9
I.B.1.a	Représentation Fonctionnelle.	9
I.B.1.b	Représentation Logique.	10
I.B.1.c	Représentation Symbolique.	12
I.B.1.d	Représentation Structurale.	13
I.B.1.e	Représentation Topologique.	14

CHAPITRE II

OPTIMISATION DES PLAS

II.A	INTRODUCTION	19
II.B	MINIMISATION LOGIQUE	20
II.C	OPTIMISATION TOPOLOGIQUE	20
II.D	LE PLA CLASSIQUE	21
II.E	MODIFICATION DES CONNECTEURS	23
II.F	PARTITIONNEMENT DES MATRICES	24
II.G	PLACEMENT LATERAL DES CONNECTEURS	25
II.H	OPTIMISATION PAR TRIANGULARISATION	27
II.I	OPTIMISATION PAR LINEARISATION DE LA MATRICE OU	29
II.J	OPTIMISATION PAR PLIAGE (FOLDING)	30
II.K	OPTIMISATION EN LIGNES BRISEES	35
II.L	OPTIMISATION PAR MONOMES ARBORESCENTS	38

CHAPITRE III

LA METHODE DES LIGNES BRISEES

III.A	INTRODUCTION	45
III.B	DEFINITIONS ET CONVENTIONS	46
III.B.1	Les Monômes.	46
III.B.2	Couverture D'un Segment	46
III.B.3	Segment D'Entrée, Segment De Sortie.	48
III.C	ORDONNACEMENT DES MONOMES	49
III.D	DUPLICATION DES MONOMES	51
III.D.1	Choix Du Monôme à Dupliquer.	52
III.D.1.a	Nombre De Transistors Controlés (NBT).	52
III.D.1.b	DELTA (Longueur De La Couverture Du Monôme).	53
III.D.1.c	Distance Barycentre Moyenne (B-M).	53
III.D.1.d	Facteur D'Occupation (FO).	53
III.D.1.e	Facteur D'Utilisation (FU).	54
III.D.1.f	Choix Par Taille De Trou.	54

TABLE DES MATIERES

III.D.1.g	Choix Par Distance Bord-Trou Vertical Pondérée.	55
III.D.2	Comparaison Des Critères De Choix.	56
III.D.3	Duplication Des Monômes.	57
III.E	COMPACTAGE DES MATRICES	58
III.E.1	Regroupement Des Sorties.	60
III.F	TRAITEMENTS COMPLEMENTAIRES	61
III.F.1	Conflits Topologiques Dans La Matrice OU.	62
III.F.2	Conflits Topologiques Dans La Matrice ET.	64
III.F.3	Connecteurs En Haut Et En Bas.	66
III.F.4	Problèmes Liés Au Tracé Des Connexions Internes.	66
III.G	GENERATION DU DESSIN DES MASQUES ET TRACE INTERNE DES CONNEXIONS	68
III.H	RESULTATS-CONCLUSION	71

CHAPITRE IV PRESENTATION DU SYSTEME PAOLA

IV.A	DESCRIPTION GENERALE	77
IV.B	MODULE D'OPTIMISATION TOPOLOGIQUE	78
IV.C	MODULE D'EVALUATION ELECTRIQUE	79
IV.D	MODULE DE TRACE DES CONNEXIONS ET DE DESSIN DES MASQUES	79
IV.E	MODULE DE GESTION DU FICHIER TECHNOLOGIQUE	80
IV.F	EXTENSIONS POUR D'AUTRES TECHNOLOGIES	80
IV.G	CONCLUSION	81

BIBLIOGRAPHIE

REFERENCES BIBLIOGRAPHIQUES	85
---------------------------------------	----

CHAPITRE I
INTRODUCTION

I.A LA CONCEPTION DES CIRCUITS INTEGRES

La réalisation des circuits intégrés comporte deux étapes bien distinctes : la conception et la fabrication.

La conception des circuits intégrés consiste à fournir un jeu de dessins des masques à partir de spécifications, qui peuvent être plus ou moins rigoureuses, décrivant la fonctionnalité du circuit.

Ces dessins de masques décrivent entièrement le circuit. Ils servent à la création des masques, le circuit étant fabriqué à partir de ces masques.

L'introduction de l'automatisation (par le biais de l'informatique) dans la conception était considéré, il y quelques années comme un moyen d'accélérer celle-ci ainsi que d'augmenter sa fiabilité. Aujourd'hui cette informatisation est le seul moyen par lequel peuvent être conçus des circuits comportant des centaines de milliers de transistors.

La complexité atteinte dans les nouveaux circuits intégrés à très grande échelle (VLSI) impose, pour leur conception, l'utilisation de méthodes strictes et rigoureuses pour pouvoir appréhender cette complexité, ainsi que l'utilisation d'outils de plus en plus puissants d'aide à la conception.

Ces méthodes de conception s'appuient sur la croissance exponentielle de la complexité d'un problème. Autrement dit il est plus facile de résoudre N problèmes simples qu'un seul problème dont la taille est N fois plus grande.

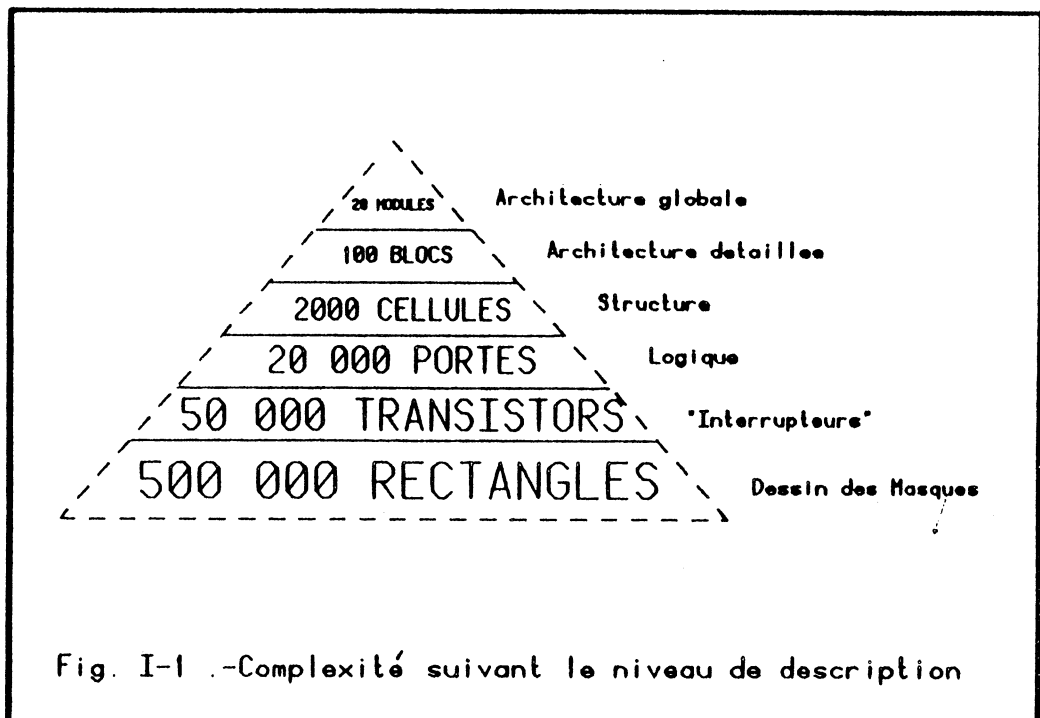
La première devise des concepteurs de circuits VLSI est DIVISER POUR REGNER.

Par ailleurs, il est impossible d'appréhender ensemble tous les aspects d'un circuit VLSI. Il faut donc faire des abstractions, et pratiquement les niveaux sur lesquels les problèmes se traitent sont les suivants :

- le niveau fonctionnel
- le niveau architectural
- le niveau comportemental
- le niveau structurel
- le niveau logique
- le niveau «interrupteur» (pour les circuits MOS)
- le niveau électrique

- le niveau topologique

Leurs frontières, leur ordre ou même leur non recouvrement ne sont pas toujours très bien établis. Mais ce découpage permet de réduire la complexité des problèmes à manipuler (fig. I.1), et surtout de manipuler des aspects différents du même circuit sans les mélanger. Ainsi lorsqu'on spécifie l'architecture d'un circuit on se soucie peu des problèmes électriques de celui-ci.



Ainsi la conception d'un circuit intégré est la transformation d'une description fonctionnelle en une description du dessin des masques. Cette transformation se fait par étapes successives, et cette manière de procéder s'appelle la conception descendante ou conception hiérarchique.

Deux types de traitement s'effectuent à chaque niveau :

- le passage d'un niveau au niveau suivant comme le passage d'un schéma logique à un réseau de transistors. Ce passage consiste à ajouter de l'information puisque chaque niveau comporte toute l'information des niveaux supérieurs plus des informations qui lui sont propres. Chaque niveau doit donc satisfaire toutes les contraintes des niveaux supérieurs plus celles particulières à ce niveau.
- des transformations au sein de ce niveau,

- * des prises en compte de contraintes propres à ce niveau comme le dimensionnement des transistors ou la minimisation logique
- * des préparations pour passer au niveau suivant tels que la synthèse en portes inverseuses.

Parmi les contraintes qui doivent être respectées on peut citer

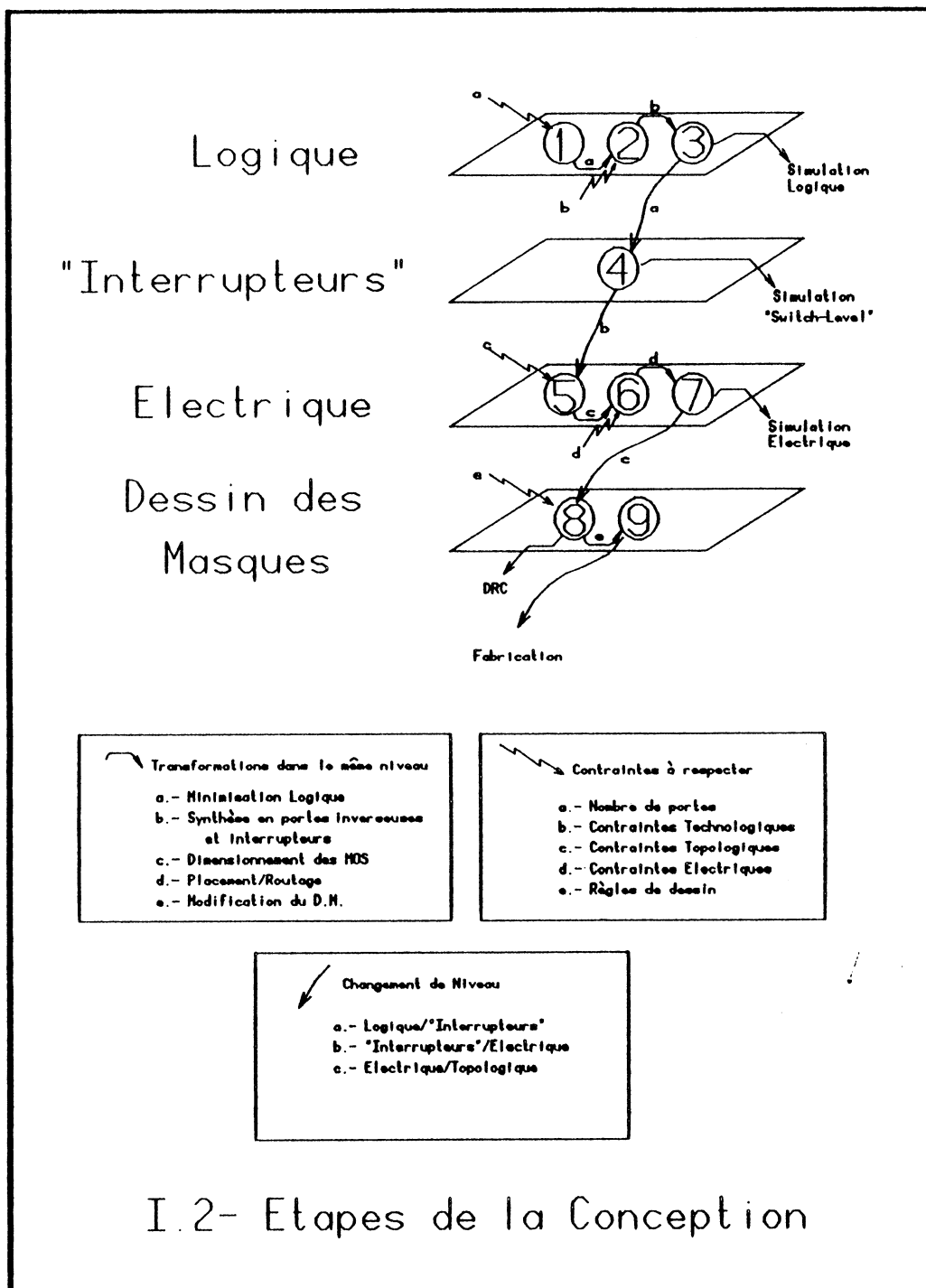
- la conformité fonctionnelle du circuit.
- la conformité comportementale du circuit (consommation, vitesse, etc..).
- le respect des règles de dessin d'une technologie particulière.
- la surface du circuit doit être la plus faible possible (pb. de rendement en production).
- les coûts de conception, de correction et de test en production doivent être les plus faibles possibles.

Lorsqu'il est impossible de satisfaire des contraintes à un niveau donné il faut les faire «remonter» au(x) niveau(x) supérieur(s) pour y trouver une solution. Dans la conception de circuits performants il est courant de délaissier momentanément la conception descendante.

Une troisième méthode indispensable dans la conception des systèmes VLSI est la REGULARITE qui consiste à utiliser autant que possible un petit nombre d'objets qui, disposés judicieusement, permettent par un jeu d'assemblage la réalisation du plus grand nombre possible d'éléments fonctionnels. Cette régularité gagne à être utilisée avec des outils informatiques d'assemblage. Une «bibliothèque» de cellules doit être définie ainsi que leur règles d'assemblage et éventuellement un logiciel qui se chargera de l'assemblage suivant ces règles.

Le fait d'utiliser la régularité permet de diminuer fortement le temps de conception car celle-ci est réduite à l'assemblage des cellules. La fiabilité en est augmentée du fait de la vérification implicite ou explicite des règles d'assemblage.

Ceci se traduit, dans le cas de la conception descendante, par une automatisation des niveaux inférieurs comme le dessin des masques [JER-83], la génération entière de blocs fonctionnels de grande taille tels que les parties opératives [SCH-83], [DON-84] et même la synthèse complète d'automates à partir d'une description fonctionnelle [FOR-83] et [FLA-84].



Cette méthode est très puissante, mais elle entraîne dans la plupart des cas une augmentation importante de la surface finale du circuit ainsi qu'une réduction des performances électriques. Pour pallier à ces inconvénients on utilise souvent des optimiseurs qui réalisent des transformations au sein d'un même niveau.

Pour ceci une myriade d'outils d'aide à la conception ont été créés ou adaptés. Ces outils vont des éditeurs graphiques aux «compilateurs» de silicium en passant par les vérificateurs, les optimiseurs et les bases de données.

On appelle Conception Assistée par Ordinateur l'ensemble de ces méthodes et de ces outils.

Cette thèse présente un de ces outils de C.A.O. spécialisé dans l'optimisation de certaines structures régulières : les Réseaux Logiques Programmables ou PLA.

I.A.1 Optimisation Locale Et Optimisation Globale

La réduction de la complexité par découpage se traduit dans la conception des C.I. par la division du circuit en «blocs». Pour que ce découpage puisse se maintenir aux différents niveaux il est généralement réalisé sur les descriptions de haut niveau. Il faut relier les différents blocs entre eux par ce que nous appellerons des connexions, reliant des connecteurs appartenant à des blocs différents.

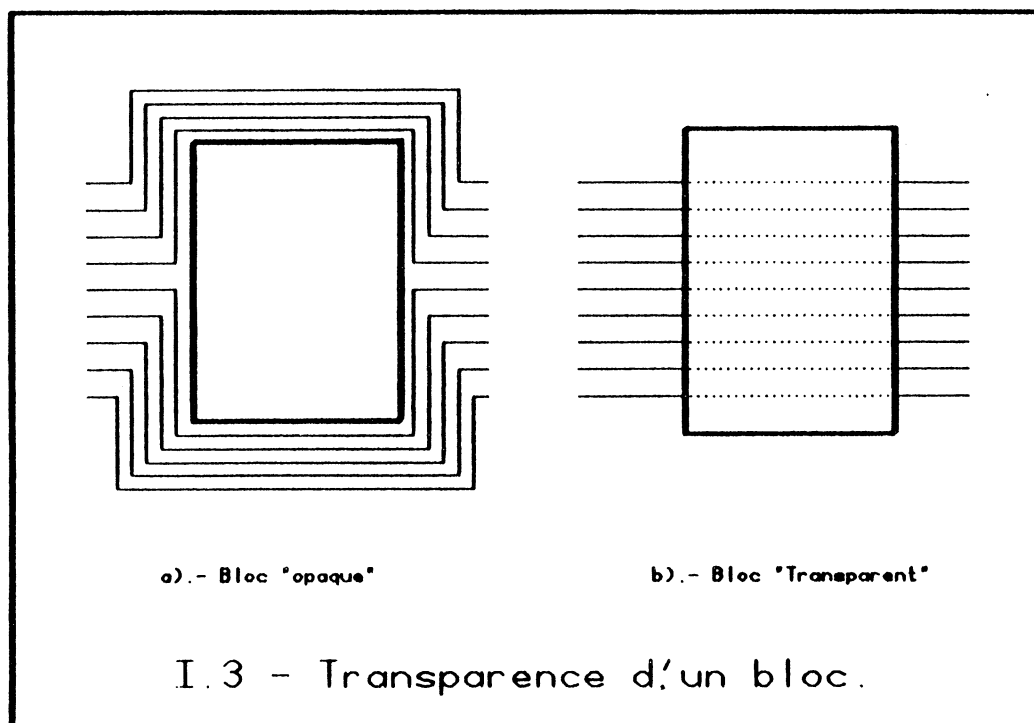
La surface des circuits détermine, entre autres, le rendement en production. Il est donc très important de faire des optimisations pour réduire la surface, tant au niveau des blocs comme au niveau des connexions. Pour ceci il faut donc réaliser :

- un bon placement de ces blocs pour réduire les connexions
- une adaptation topologique de ces blocs [VAN-79].
- une distribution adéquate des points de connexion (connecteurs) de chaque bloc

D'où la nécessité de faire un plan de masse du circuit s'appuyant sur une évaluation de la surface, de la forme et de la position des connecteurs de chaque bloc. Dans la génération du plan de masse on doit tirer profit de la «déformabilité» (ou de la «dureté») et de la «transparence» des blocs [REI-82].

La déformabilité caractérise la faculté de choisir la forme (et donc la surface) d'un bloc. Elle dépend de la structure interne de celui-ci. Les blocs générés automatiquement sont plus «durs» que les blocs dessinés manuellement (pour peu que la forme soit déterminée avant de faire le dessin).

La transparence d'un bloc est la capacité de celui-ci à se laisser traverser par des connexions externes. Elle est utilisée pour réduire la longueur des connexions entre blocs (fig. I-3).



L'emplacement des connecteurs doit être fait d'après le plan de masse et doit permettre une réduction de la longueur des connexions ainsi que du nombre de croisements entre celles-ci.

Suivre cette méthodologie d'optimisation globale suppose que :

- les évaluations, tant de forme que de surface, soient réalistes. Ceci ne peut se faire qu'avec l'aide de concepteurs très expérimentés ou des outils qui «contiennent» cette expérience.
- chaque bloc doit être conçu pour un circuit particulier, ce qui limite l'utilisation des bibliothèques de blocs.
- la conception doit se faire d'une manière hiérarchique.

Le résultat donne des circuits très performants dont le rendement en production est élevé.

Néanmoins, il est courant, soit pour des raisons de coût de conception soit par manque d'outils adaptés, d'utiliser l'optimisation locale au niveau des blocs. Les blocs sont alors conçus plus ou moins indépendamment les uns des autres, placés dans le circuit et interconnectés par des programmes de tracé automatique de connexions. Généralement cette méthode se traduit par une grande perte de surface en connexions.

I.B LES PLAS

Les Reseaux Logiques Programmables (PLAs) sont des structures régulières qui, générés par des logiciels particuliers, permettent le passage automatique d'une description logique au dessin des masques. Ils implémentent des fonctions logiques combinatoires, et ils présentent les caractéristiques suivantes :

- une structure logique simple
- une structure électrique simple
- et une structure topologique très simple.

Donc des règles simples de transformation. Ce qui rend la programmation des «transformateurs» ou générateurs (de dessins de masques) de PLAs extrêmement simple. C'est pourquoi ils sont devenus très vite des éléments largement utilisés dans la conception des circuits intégrés.

Il ne faut pas les confondre avec les FPLAs (Field Programmable Logic Array) ou avec les PALs (Programmable Array Logic) qui eux sont des circuits complets, en boîtiers MSI, réalisés à base de PLAs.

I.B.1 Les Différentes Représentations Des PLAs

Les PLAs étant des structures définies à plusieurs niveaux, ils ont des représentations particulières à chacun de ces niveaux. Il est important de bien les différencier car des traitements distincts peuvent y être associés.

Ces différentes représentations sont :

I.B.1.a Représentation Fonctionnelle.

Il est possible de décrire une fonction logique non pas par des équations mais par sa réponse à certaines valeurs des entrées. Dans ce cas on se sert d'un double tableau où on indique à chaque ligne la valeur des sorties face à chaque combinaison des entrées (fig. I-4).

Lorsque la valeur d'une entrée n'est pas spécifiée ou qu'elle est définie comme $\emptyset$ (phi) dans une ligne, les sorties ne dépendent pas de cette entrée. Une combinaison d'entrées absente implique que les sorties peuvent prendre n'importe quelle valeur. Ces caractéristiques permettront au niveau logique une certaine liberté d'action lors des transformations logiques.

Cette représentation fonctionnelle peut définir n'importe quel circuit logique combinatoire.

A	B	C	F1	F2
0	0	x	1	X
0	1	x	0	X
1	0	x	0	X
1	1	x	1	X
x	0	0	X	0
x	0	1	X	1
x	1	0	X	1
x	1	1	X	0

F1 : CONJUNCTION
 $F1 = A \odot B$

F2 : DISJUNCTION
 $F2 = B \oplus C$

I.4 - Description Fonctionnelle.

I.B.1.b Représentation Logique.

Il existe deux types de PLAs implémentant chacune une des deux formes canoniques (ou de Lagrange) des fonctions logiques.

Or toutes les fonctions logiques peuvent être décrites sous ces deux formes. Il en résulte que toutes les fonctions logiques peuvent être implémentées par des PLAs.

Ces formes canoniques sont :

- * la somme des produits des entrées et de leur complément.
- * le produit des sommes des entrées et de leur complément.

Dans le premier cas le PLA est composé de :

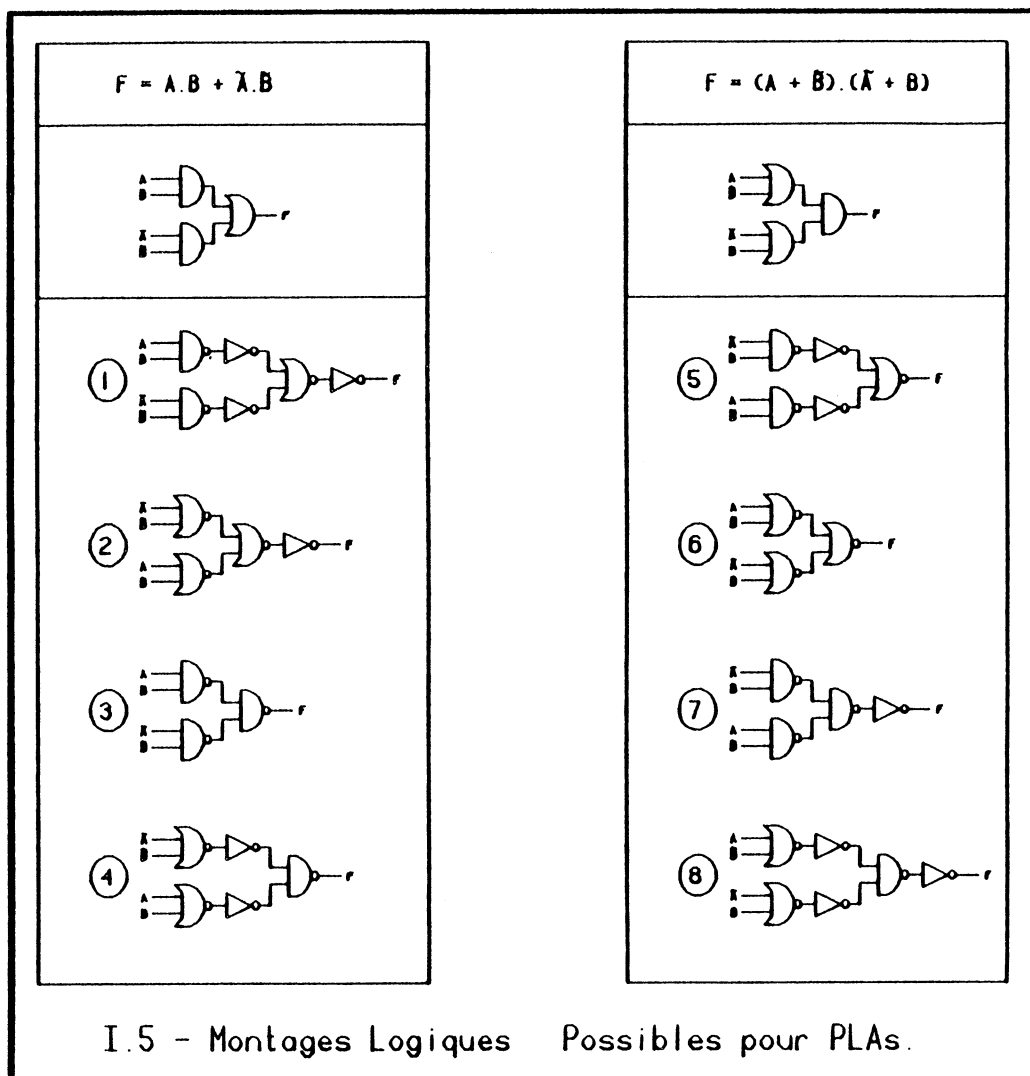
- un ensemble d'inverseurs pour générer le complément de chaque entrée
- d'un ensemble de portes réalisant la fonction ET (produit) entre les entrées et leur complément.
- d'un ensemble de portes réalisant la fonction OU (somme) entre les différentes sorties de l'étage précédent.

On appelle termes-produits ou plus simplement monômes les sorties du deuxième étage.

Dans le deuxième cas les deux ensembles ET et OU sont permutés.

Dans les technologies actuelles de fabrication des circuits intégrés il est impossible de réaliser directement des portes ET et des portes OU. Il est donc nécessaire de les transformer, au niveau logique, suivant les règles de De Morgan en portes inverseuses : soit en portes NON-ET (NAND) soit en portes NON-OU (NOR) plus des inverseurs.

Parmi les huit montages possibles (fig. I-5), deux sont pratiquement utilisés : le montage NOR-NOR et le montage NAND-NAND : ils présentent un nombre minimal de portes.



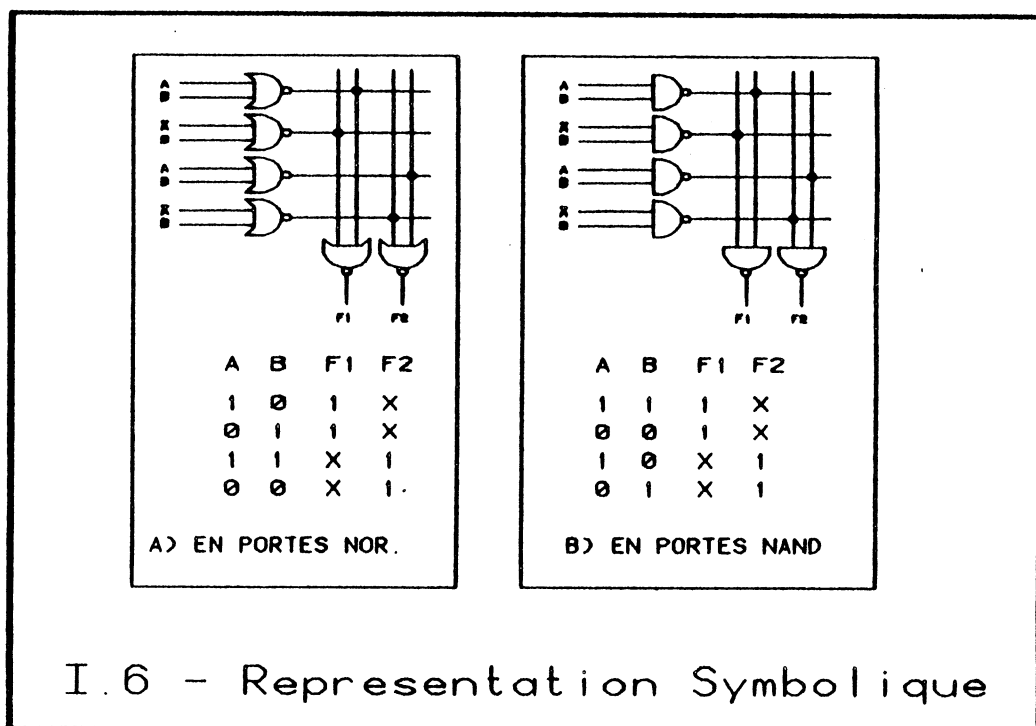
Le choix parmi ces deux structures est fait soit au niveau de la technologie utilisée pour la réalisation du circuit, soit par des contraintes électriques ou topologiques, soit par les performances souhaitées du PLA.

Les PLAs peuvent être décrits comme toute fonction logique par :

- un tableau de vérité
- un ensemble d'équations logiques (ou un schéma logique équivalent), minimisées ou non.

I.B.1.c Représentation Symbolique.

Lorsque la structure logique des PLAs est connue, il est possible de la représenter d'une manière symbolique par exemple à l'aide d'un double tableau (fig. I-6) où chaque ligne correspond à un monôme. Dans le premier sous-tableau chaque colonne correspond à une entrée et dans la deuxième chaque colonne correspond à une sortie. Si le monôme M est généré à partir de l'entrée E on placera la valeur de l'entrée dans l'élément [M, E] du premier sous-tableau, dans le cas contraire on placera la valeur $\emptyset$ (phi). L'utilisation du monôme M dans la génération de la sortie S sera représentée par une valeur particulière de l'élément [S, M] du deuxième sous-tableau.



Cette représentation utilise des valeurs ternaires dans le premier sous-tableau et binaires dans le deuxième. Dans le cas des PLAs elle peut être identique à la représentation fonctionnelle, mais elle suppose connue la structure logique de ceux-ci.

I.B.1.d Représentation Structurale.

Les PLAs sont réalisés par un réseau de décodeurs «un vers deux» (inverseurs) plus deux matrices de transistors soit en série (structure NAND-NAND), soit en parallèle (structure NOR-NOR).

L'utilisation de la valeur directe d'une entrée dans la génération d'un monôme correspond au placement d'un transistor entre cette entrée et le monôme. Si c'est la valeur inverse, le transistor sera placé entre le monôme et la sortie de l'inverseur correspondant.

Si la position des lignes des entrées et de leur inverse est détermininée on peut représenter la première matrice de transistors par un tableau à 2 * NE colonnes et NM lignes (fig. I-7). Chaque ligne correspondant à un monôme et chaque colonne à une entrée directe ou inverse. L'élément [M, E] du tableau aura la valeur VRAI si il existe un transistor entre l'entrée (directe ou inverse) E et le monôme M. Dans le cas contraire sa valeur sera FAUX.

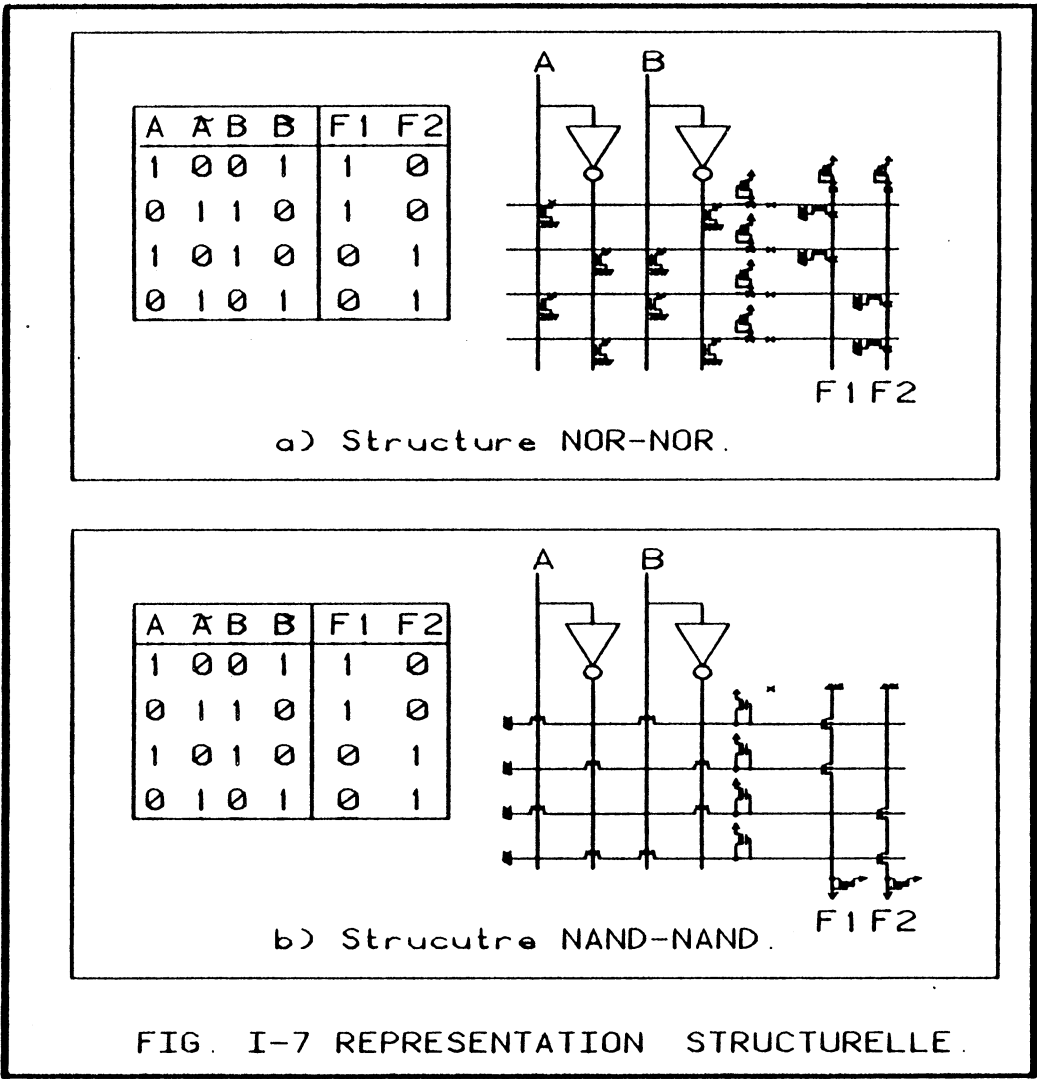


FIG. I-7 REPRESENTATION STRUCTURELLE.

De même la deuxième matrice de transistors sera représentée par un tableau associé au premier où sera codée la présence de transistors suivant le même schéma.

Cette représentation ne contient que les informations de connexion entre transistors, entrées, sorties et monômes.

I.B.1.e Représentation Topologique.

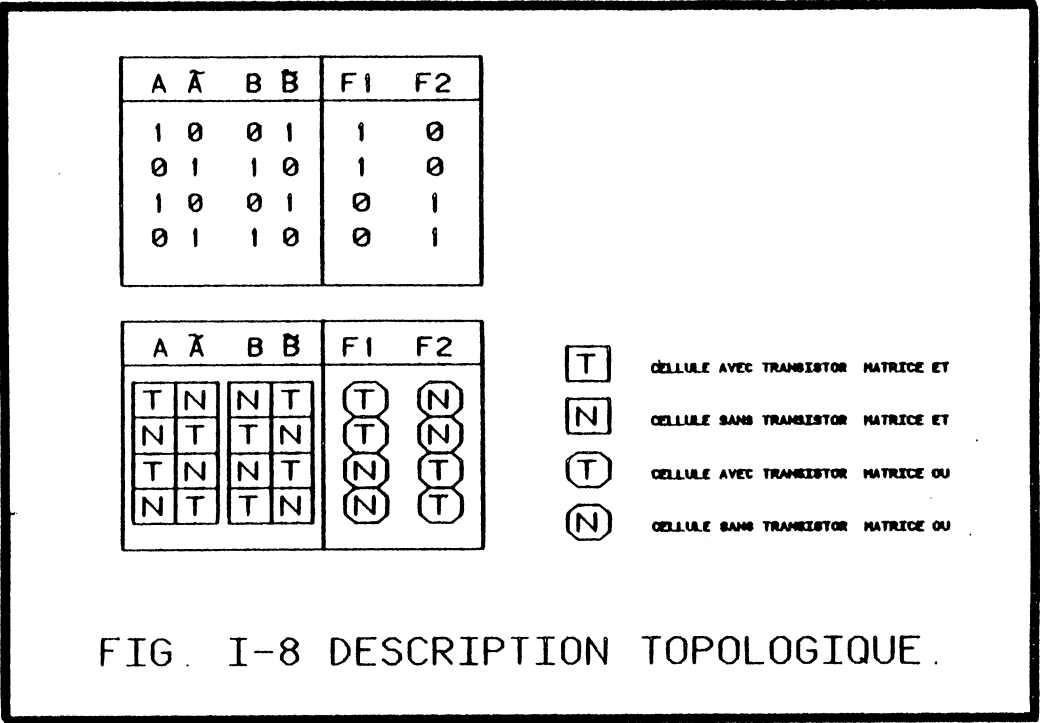
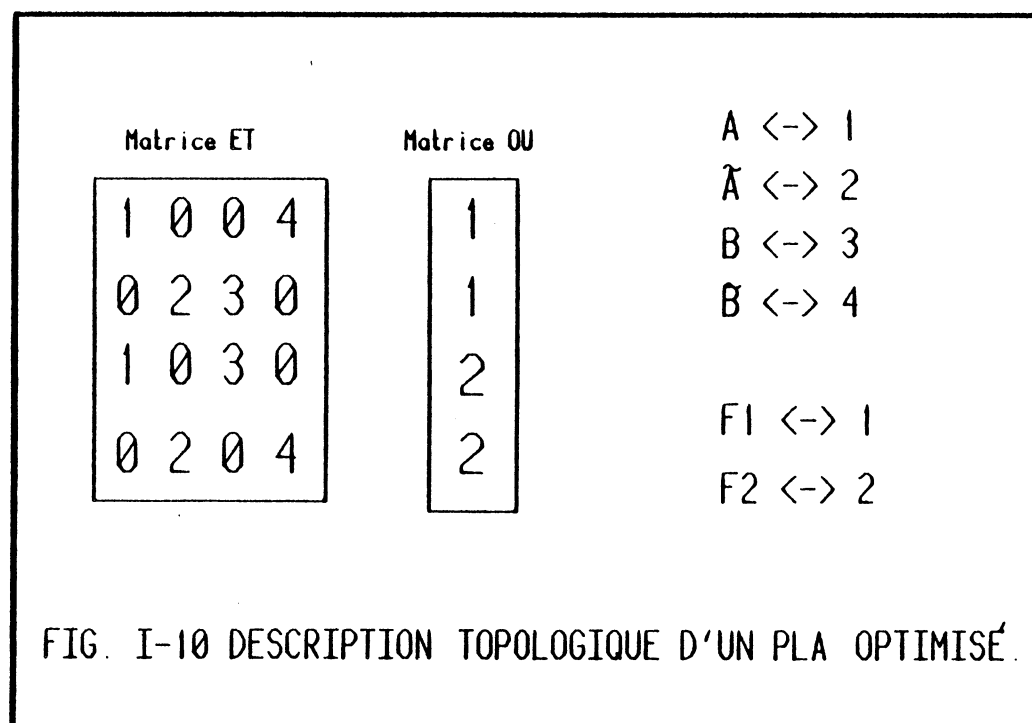
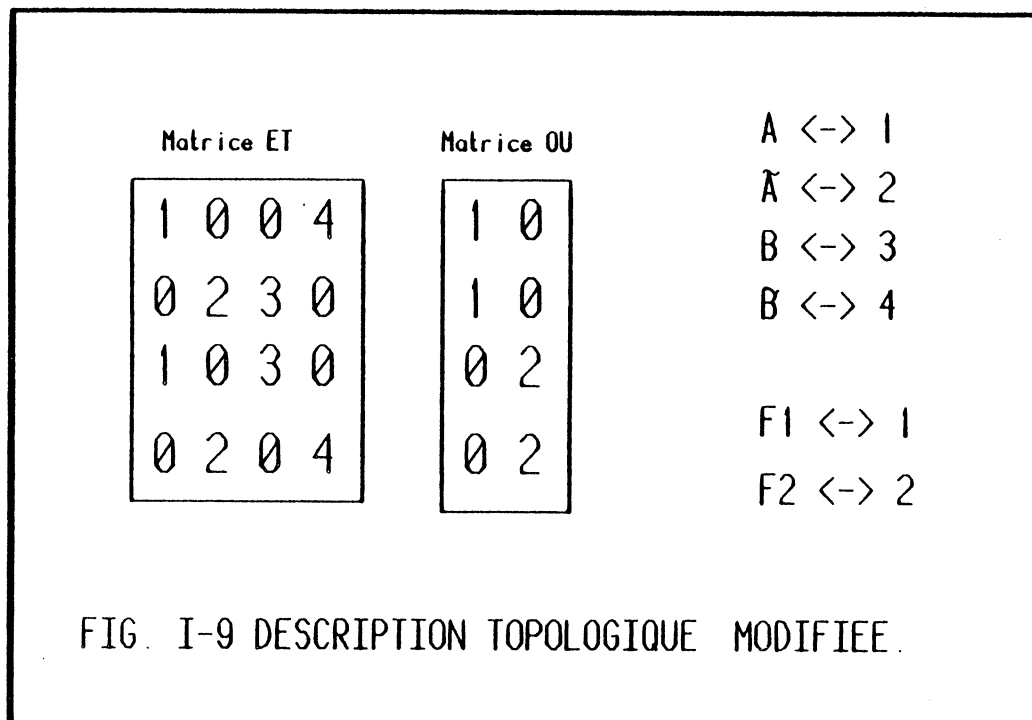


FIG. I-8 DESCRIPTION TOPOLOGIQUE.

On peut aussi représenter le dessin des masques des matrices à l'aide de deux tableaux binaires. Chaque élément du tableau représente une cellule (ou point PLA) contenant ou non un transistor. Les cellules des deux matrices peuvent être différentes ou identiques (à une rotation près). On sait alors que chaque cellule comprend (pour des PLAs à structure NOR-NOR en technologie N-MOS grille en silicium polycristallin) :

- une ligne en aluminium correspondant aux monômes (dans la matrice ET) ou aux sorties (dans la matrice OU).
- une ligne en silicium polycristallin correspondant aux entrées (matrice ET) ou aux monômes (matrice OU).
- une ligne en diffusion correspondant aux lignes de masse.
- éventuellement un transistor et un contact.

Il peut être intéressant dans certains cas d'ajouter un troisième type de cellule qui symbolise les lignes de rappel de masse.



Dans cette représentation chaque colonne est associée soit à une entrée (matrice ET) soit à une sortie (matrice OU).

Pour pouvoir représenter clairement un PLA optimisé suivant la méthode des lignes brisées (cf. Chapitre III), nous avons dû faire une modification à cette représentation. Elle consiste à définir dans chaque élément des tableaux contenant un transistor le numéro de l'entrée ou de la sortie associée (fig. I-9).

Ceci permet la représentation des PLAs qui contiennent plusieurs entrées ou plusieurs sorties dans la même colonne, les deux matrices étant disjointes.

Dans les technologies N-MOS la structure NOR-NOR occupe une surface plus importante de silicium, mais permet une réponse plus rapide que la structure NAND-NAND; elle est donc pratiquement la seule utilisée.

Par contre dans la technologie C-MOS statique la structure NAND-NAND est souvent utilisée dans des circuits qui ne nécessitent pas de grandes performances temporelles tels que les montres à quartz ou les calculatrices.

Toutes ces représentations, faites à base de deux tableaux, définissent les PLAs à des niveaux différents.

CHAPITRE II

OPTIMISATION DES PLA

II.A INTRODUCTION

Les principaux avantages des PLAs par rapport à la logique anarchique (ou aléatoire) sont les possibilités de génération et de dessin automatiques. Ils réduisent le temps de conception, facilitent les corrections éventuelles [FLE-75] et diminuent fortement les risques d'erreurs.

La dissociation des caractéristiques logiques, électriques et topologiques simplifie la vérification :

- La simulation logique peut se faire sur une description structurale (tableau binaire) sans tenir compte des caractéristiques électriques ou topologiques.
- La simulation électrique se réduit à la simulation du chemin le plus long à travers le PLA. Ce chemin critique peut se déduire directement de la description structurale du PLA puisque il existe une conformité complète entre la représentation structurale, le schéma électrique et le dessin des masques.
- Finalement la conformité des masques avec les règles de dessin technologiques est facilement obtenue avec une bonne définition des cellules de base et de leurs méthodes d'assemblage.

Néanmoins les PLAs présentent un inconvénient de taille : la surface qu'ils occupent est beaucoup plus importante que leur équivalent en logique aléatoire. Ceci est dû à leur structure régulière qui impose des contraintes topologiques au niveau du point PLA, donc au niveau du bloc tout entier.

Pour diminuer cet inconvénient on peut faire appel à deux traitements différents mais complémentaires :

- la minimisation logique qui consiste à trouver la forme logique minimale pour réaliser une fonction logique donnée.
- l'optimisation topologique qui s'attaque à la surface de silicium nécessaire à la réalisation de ces fonctions logiques.

Les méthodes de minimisation logique existent depuis longtemps [MCC-57] et ont été améliorées ou adaptées à la résolution de problèmes de grande taille. Par contre l'optimisation topologique n'est apparue qu'avec l'avènement des circuits intégrés de moyenne complexité, et plus particulièrement des microprocesseurs, vers les débuts des années 1970. A cette époque elle faisait partie de l'art des concepteurs, mais avec l'augmentation de la capacité d'intégration des C.I. cet art s'est transformé en une série de techniques et ces techniques en méthodes.

II.B MINIMISATION LOGIQUE

Une boîte noire définie par son comportement logique peut être réalisée par une quantité immense de fonctions logiques de manière équivalente. La minimisation logique consiste à trouver parmi ces fonctions équivalentes celle qui comprend le nombre le plus faible d'opérateurs logiques.

Les méthodes classiques de minimisation logique utilisent comme opérateurs les fonctions ET, OU et NON. Mais dans la pratique ces opérateurs logiques sont souvent différents. Ainsi le concepteur de systèmes a pour opérateurs les boîtiers du commerce, par exemple la famille de circuits TTL, des FPLA ou des PAL. Dans la conception des circuits intégrés les opérateurs peuvent être

- soit les portes inverseuses NON-ET (NAND) et NON-OU (NOR)
- soit les transistors (interrupteurs).

Les méthodes classiques, étant clairement définies et simples d'utilisation, sont utilisées universellement aujourd'hui avec des post-traitements pour les adapter à chaque problème particulier. Mais le besoin de nouvelles méthodes propres à chaque type d'opérateurs se fait sentir, particulièrement dans le cas des PLAs optimisés topologiquement.

II.C OPTIMISATION TOPOLOGIQUE

Une optimisation topologique performante, en réduisant considérablement la surface des PLAs, permet d'augmenter leur domaine d'utilisation. C'est pour ceci que plusieurs méthodes d'optimisation topologique ont été proposées.

Ces méthodes forment trois familles :

- les méthodes qui sans faire une réelle optimisation du PLA modifient l'emplacement des connecteurs pour améliorer le câblage externe.
- les méthodes qui optimisent localement le PLA
- les méthodes qui réalisent l'optimisation du PLA en tenant compte de leur environnement à l'intérieur d'un circuit.

Après avoir défini le PLA non optimisé que nous appellerons «PLA CLASSIQUE» et qui nous servira d'élément de comparaison, nous présenterons les différentes méthodes d'optimisation topologique connues à ce jour.

II.D LE PLA CLASSIQUE

Le PLA classique implémente une forme canonique (optimisée ou non) des fonctions logiques à réaliser. Nous ne nous intéresserons qu'à la forme NOR-NOR qui est pratiquement la seule utilisée dans les technologies N-MOS classiques [MEA-80].

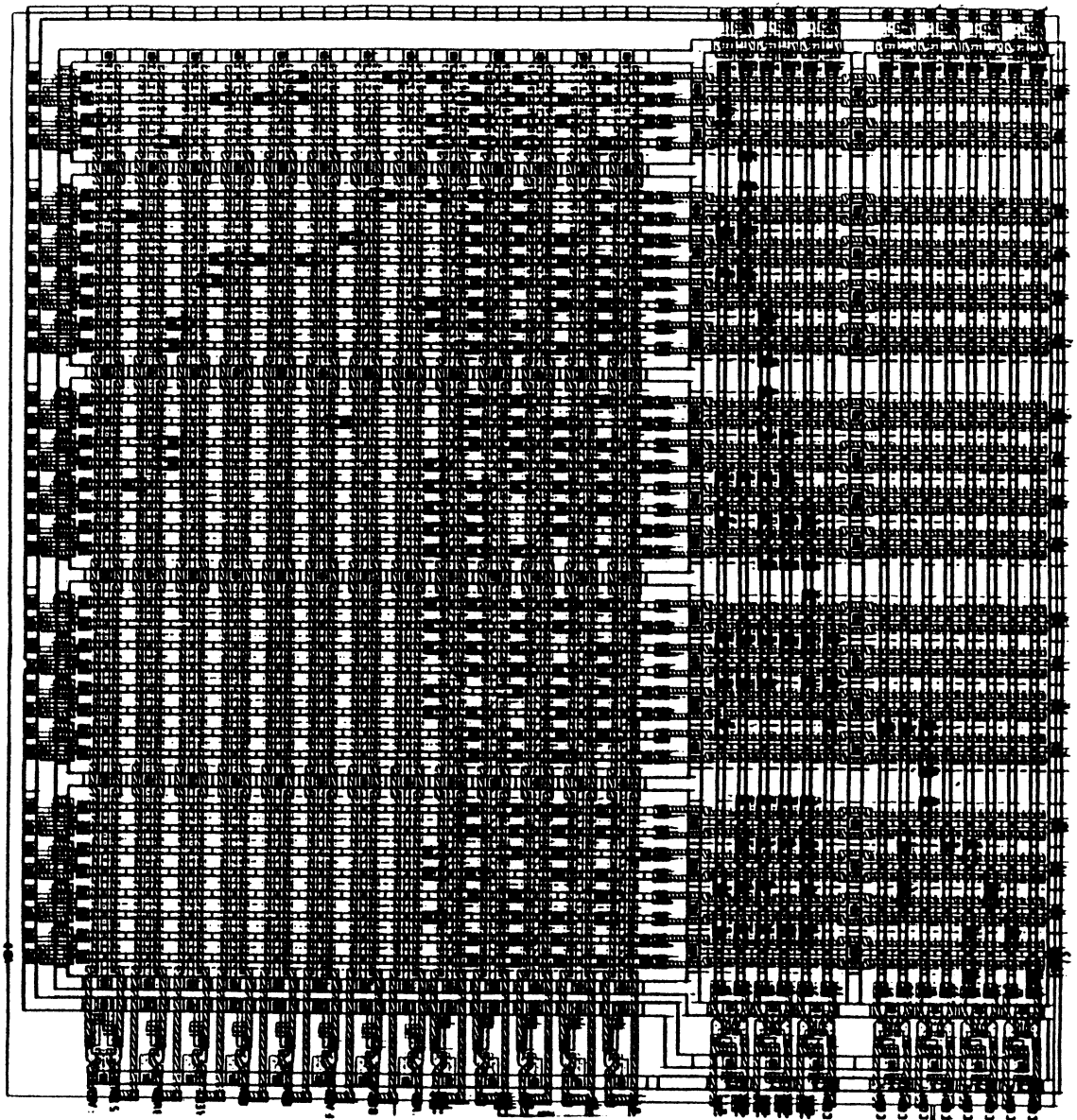


Fig. II - 1. - PLA Classique.

Il est formé de deux matrices rectangulaires (fig. II-1), les entrées étant parallèles aux sorties et perpendiculaires aux Termes-Produits. La surface du PLA classique est :

$$S = (KE * E + (KS + KR1/PO) * S) * (KM + KR2/PE) * M$$

avec

- E : nombre d'entrées
- S : nombre de sorties
- M : nombre de monômes
- KE : pas des entrées
- KS : pas des sorties
- KM : pas des monômes
- KR1 : pas des rappels de masse dans la matrice OU
- KR2 : pas des rappels de masse dans la matrice ET
- PE : nombre de monômes entre deux rappels de masse (matrice ET)
- PO : nombre de sorties entre deux rappels de masse (matrice OU)

Le PLA classique a toutes ses entrées et sorties du même côté. Les connecteurs d'entrée sont distants de KE et les connecteurs de sortie de KS. Il nécessite donc un cablage externe pour amener ces signaux au pas déterminé par le dessin du point PLA (fig. II-2).

L'ordre des connecteurs d'entrée et de sortie n'est pas fixé et il peut être utilisé pour simplifier le cablage externe, mais il ne modifie en rien la surface du PLA.

Les programmes de génération de masques de PLAs classiques sont simples et se limitent généralement à placer des cellules-point avec transistor et des cellules-point sans transistor, des cellules de rappel de masse et à gérer leurs symétries et rotations. Dans ce cas les lignes en silicium polycristallin traversent complètement les matrices et ajoutent des capacités superflues ce qui augmente inutilement le temps de propagation du PLA.

Il existe aussi des programmes plus astucieux qui génèrent des lignes de poly de longueur strictement nécessaire.

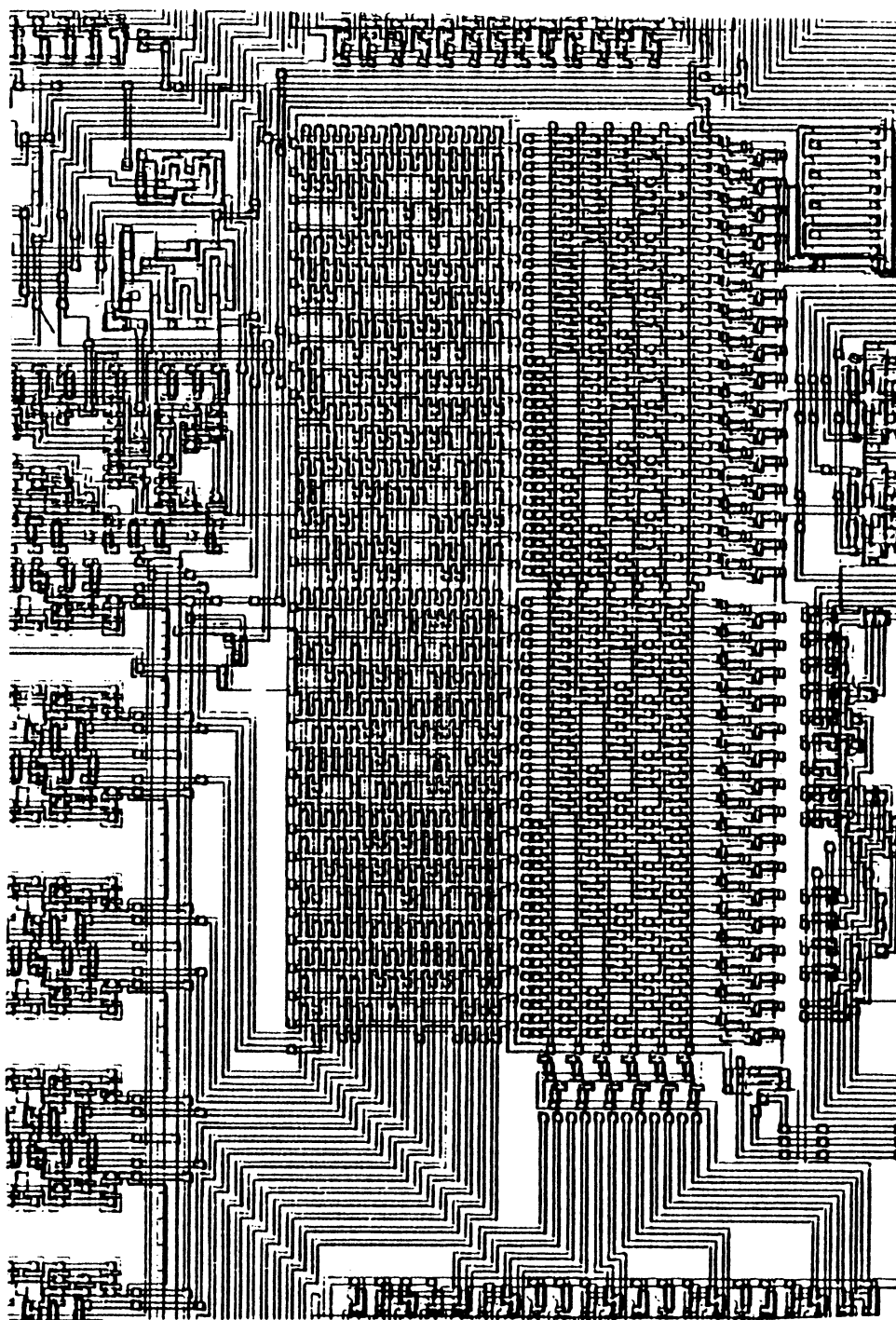
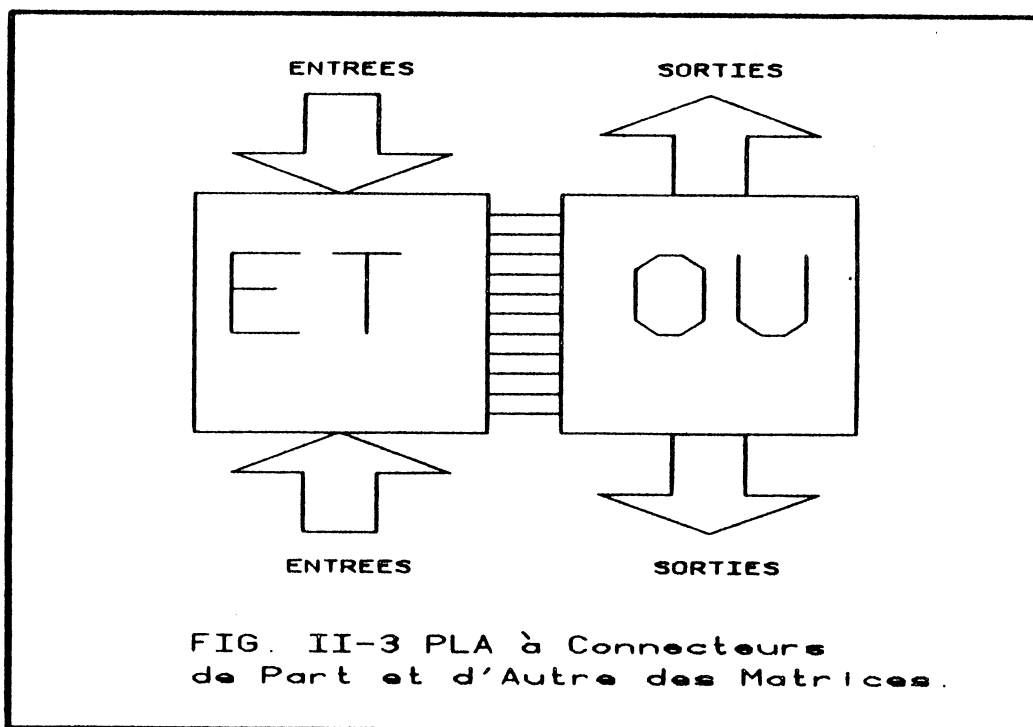


Fig. II - 2. - PLA Classique dans son environnement.

II.E MODIFICATION DES CONNECTEURS

Etant donné que les lignes d'entrée (en poly) et les lignes de sortie (en alu) traversent entièrement le PLA, il est facilement envisageable de placer les connecteurs soit en haut, soit en bas du PLA (fig. II-3).



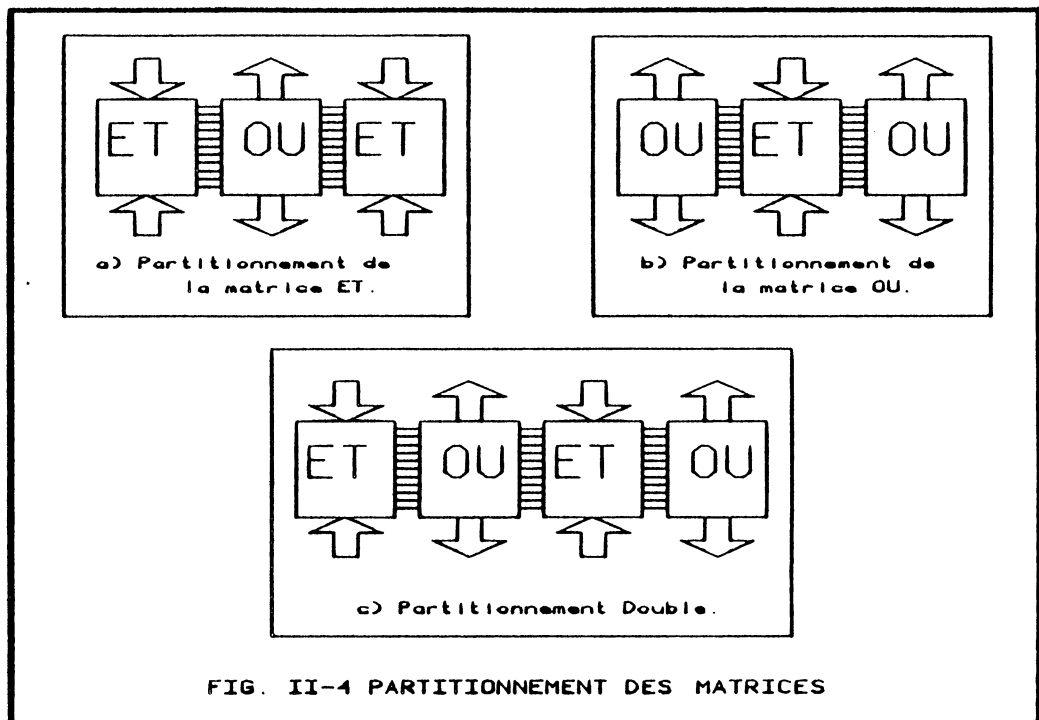
La surface des matrices n'est pas modifiée, mais l'enveloppe doit prendre en compte l'emplacement des éventuels amplificateurs d'entrée et/ou sortie. La surface globale peut alors être modifiée, et, en fonction de la forme de ces amplificateurs on peut soit gagner soit perdre en surface. De toutes façons le principal intérêt de cette méthode est de disposer judicieusement ces connecteurs pour réduire le tracé des interconnexions externes.

II.F PARTITIONNEMENT DES MATRICES

Cette méthode consiste à réaliser un PLA avec deux matrices OU, placées de part et d'autre de la matrice ET (fig. II-4). On obtient ainsi deux groupes de sorties à des emplacements géographiques différents, ce qui permet d'éviter un grand nombre de croisements de fils à l'extérieur du PLA.

De la même façon le partitionnement peut se faire sur la matrice ET ou même sur les deux. Le PLA résultant est à peine plus grand que l'équivalent «CLASSIQUE» à cause des interfaces supplémentaires entre les matrices (principalement une rangée de contacts). Les PLAs monomatrices [VAR-84] et [VAR-85] constituent un cas extrême du partitionnement des matrices.

On trouve un PLA à deux matrices OU dans le microprocesseur INTEL-8085 pour le décodage des instructions. La première matrice OU génère les commandes pour l'automate des temps, tandis que la deuxième définit à quelles classes appartient l'instruction en cours de décodage.



II.G PLACEMENT LATERAL DES CONNECTEURS

Dans toutes les méthodes précédentes l'accès aux entrées et sorties se fait sur le pas interne des lignes en poly ou alu, c'est à dire que les connecteurs sont pratiquement juxtaposés et, même si leur ordonnancement correspond avec celui des connexions externes, il faut une certaine surface pour amener ces dernières sur les connecteurs (fig. II-5). Ceci est principalement vrai lorsque le PLA génère un grand nombre de commandes pour un ou plusieurs blocs de grande taille. Il est alors intéressant de disposer les connecteurs de sortie sur le côté du PLA et non sur les extrémités (fig. II-6). Si le nombre de monômes est supérieur au nombre de sorties on dispose alors d'une grande latitude pour placer les connecteurs.

Un tracé interne des connexions doit alors être réalisé, et ce tracé interne impose :

- que les lignes de poly soient les plus courtes possible pour libérer de la place à l'intérieur du PLA.
- un réordonnement judicieux des segments de sortie (lignes en alu) pour faciliter le câblage interne.
- un allongement éventuel du PLA pour accéder aux segments de sortie qui restent inaccessibles.
- le tracé des connexions internes doit se faire soit en diffusion soit en poly (pour une technologie N-MOS classique), ce qui augmente la capacité et la résistance des lignes de sortie et augmente donc le temps de propagation du PLA.

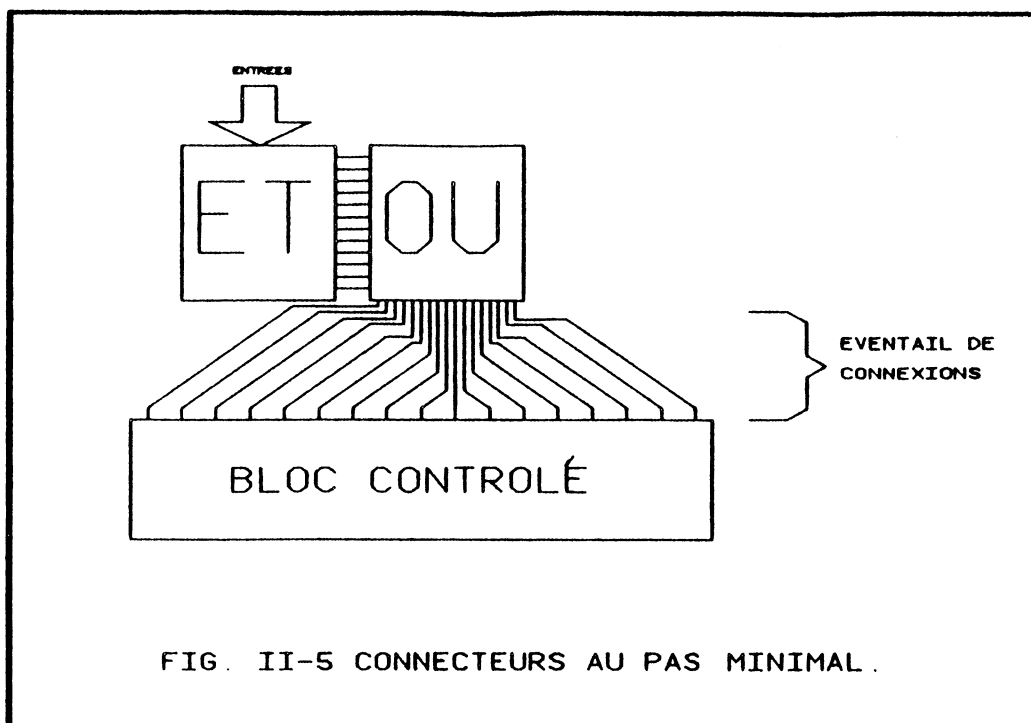


FIG. II-5 CONNECTEURS AU PAS MINIMAL.

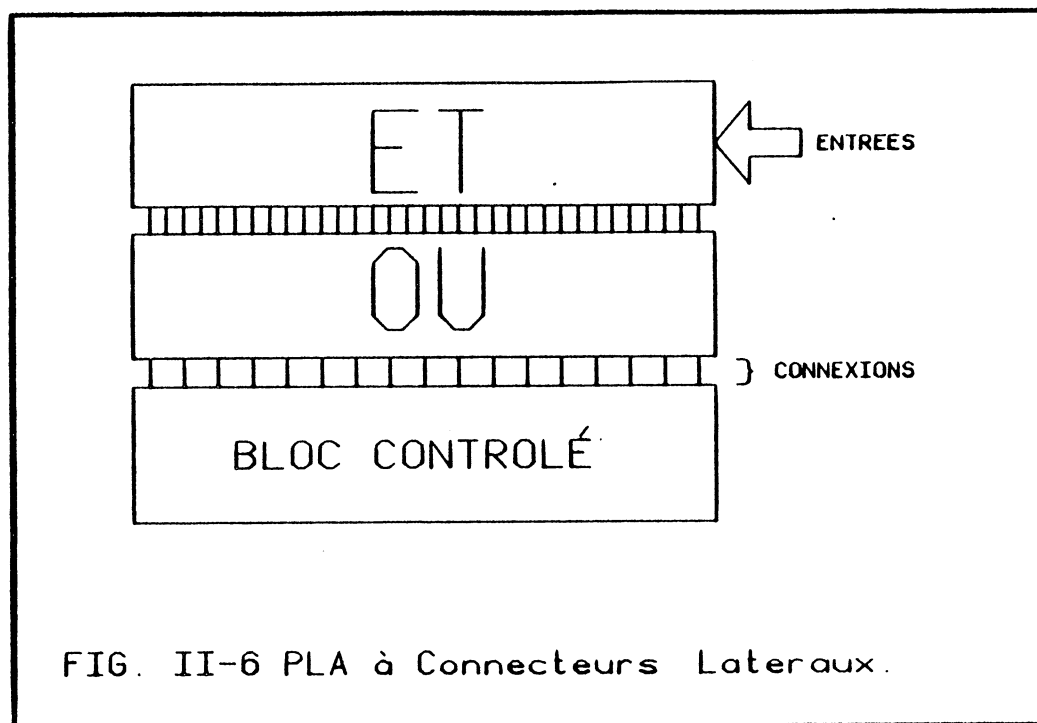
Cette méthode permet, en disposant les amplificateurs de sortie sur le côté du bloc PLA, d'avoir des contraintes topologiques moins importantes dans le dessin des premiers et donc de réduire un peu la surface du PLA.

Un exemple de ce type d'optimisation peut être trouvé dans le microprocesseur INS-8072 de National Semiconductors où l'emplacement des connecteurs de sortie correspond presque exactement à la position de leur utilisation dans la partie opérative du circuit.

Cette même méthode peut être aussi appliquée aux entrées d'un PLA, mais avec des contraintes plus importantes car :

- en général, la matrice ET a un taux de remplissage beaucoup plus important que la matrice OU,
- l'utilisation des lignes d'aluminium pour le câblage interne impose de placer les transistors de charge de la matrice ET entre les deux matrices augmentant un peu la surface globale.

Mais si le nombre d'entrées latérales est restreint ce type d'optimisation peut permettre une réduction importante de la longueur de certaines lignes.



II.H OPTIMISATION PAR TRIANGULARISATION

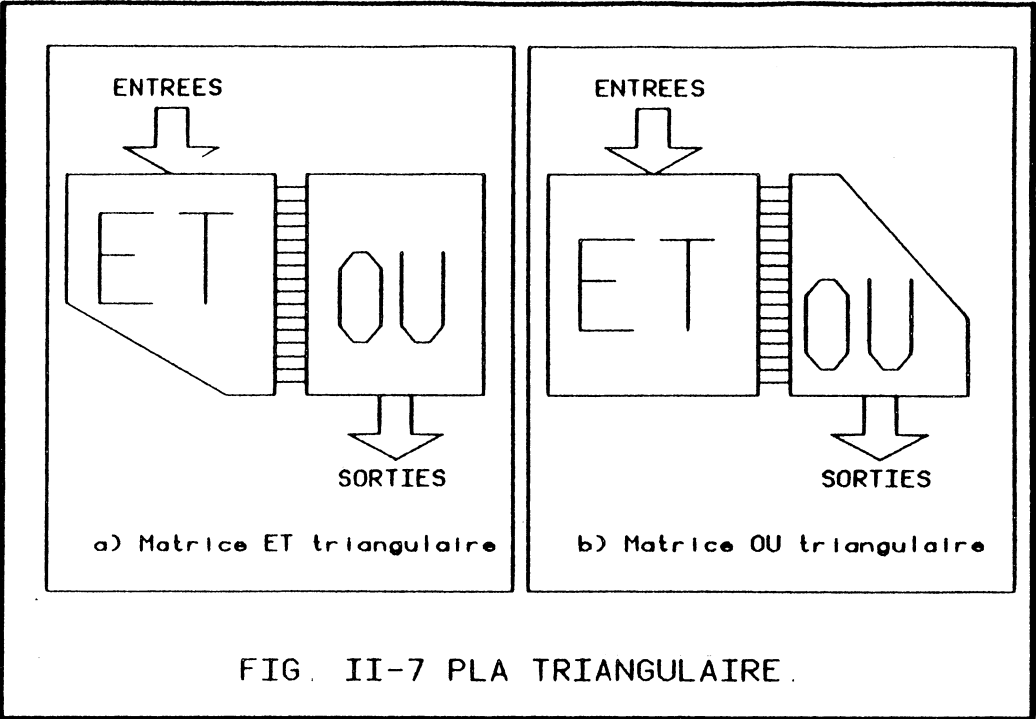
Cette méthode d'optimisation se base sur le fait qu'il existe toujours dans le pla des points sans transistors. Il est alors facile de permuter des entrées, des sorties ou des monômes pour repousser dans un coin le plus grand nombre possible de points vides (donc inutiles) qui seront supprimés sur le dessin des masques.

Trois algorithmes d'optimisation triangulaire ont été proposés dans [PER-79], mais leurs performances très médiocres et surtout l'impossibilité de fixer les connecteurs d'entrée et sortie nous ont fait abandonner cette voie.

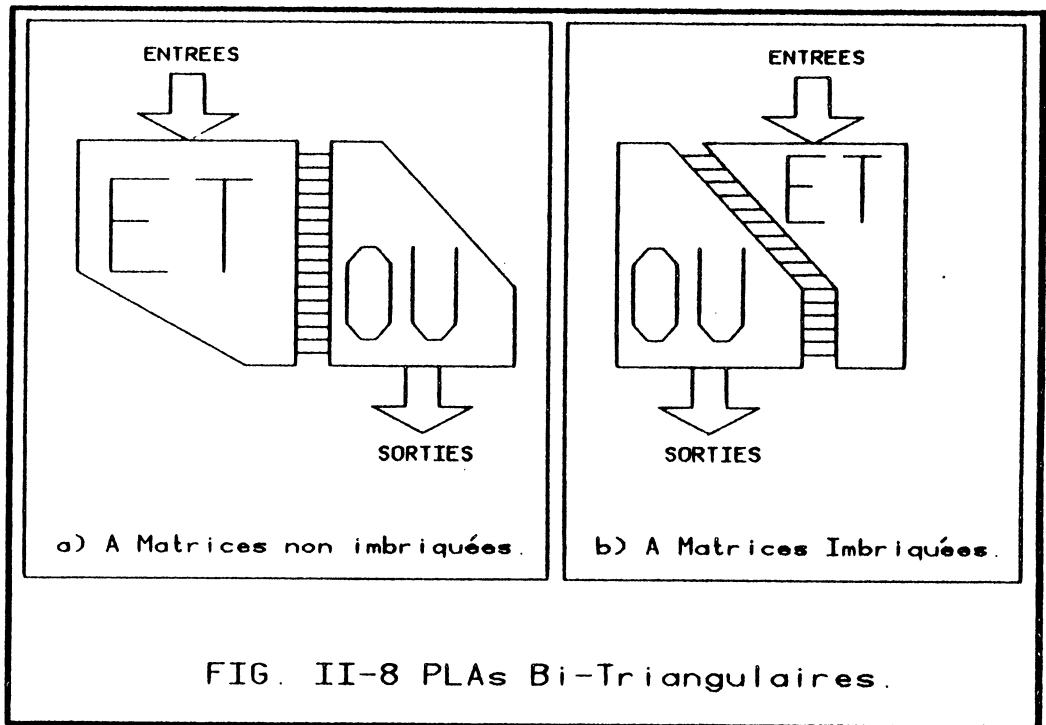
La génération de PLAs triangulaires se fait comme pour un PLA classique, la suppression des points inutiles se faisant «manuellement» sur un éditeur implanté. Le PLA résultant présente une enveloppe «biseautée» dans les coins où les points inutiles ont été regroupés et supprimés (fig.II-7).

Le gain en surface dépend du contenu du PLA et est rarement supérieur à 10 ou 20%. Mais le principal inconvénient réside dans la forme des zones libérées qui est irrégulière et donc n'est utilisable que pour y placer de la logique anarchique.

La position des connecteurs des entrées et des sorties est déterminée par l'optimisation et ne peut donc pas être utilisée pour améliorer le câblage externe. Il est toutefois possible de fixer l'emplacement des connecteurs d'entrée et/ou de sortie, mais l'optimisation y perd beaucoup en performance.



Dans certains cas la forme triangulaire des matrices peut être utilisée pour imbriquer les deux matrices, mais ceci suppose que la forme des zones libérées dans les deux matrices soient semblables (fig. II-8). Le résultat est alors similaire au pliage simple des lignes et colonnes (voir chapitre II.J).



L'optimisation triangulaire a été utilisée dans la conception du microprocesseur SC/MP de National Semiconductors. Dans ce circuit la surface libérée a été utilisée pour placer la gestion des interruptions qui est une des parties des microprocesseurs où une conception topologique régulière est la plus difficile à réaliser.

II.1 OPTIMISATION PAR LINEARISATION DE LA MATRICE OU

Lorsque la matrice OU comporte un très faible taux de remplissage et un nombre important de sorties, l'utilisation d'un PLA classique entraîne un gaspillage énorme de surface. Si, en plus, l'environnement du PLA permet la disposition latérale des connecteurs de sortie, l'optimisation par linéarisation de la matrice OU permet un gain très important de surface.

Cette optimisation consiste en :

- une duplication des monômes qui attaquent plusieurs sorties pour obtenir un PLA où chaque monôme n'attaque qu'une seule sortie.
- un réordonnancement des monômes pour faire en sorte que tous les monômes impliqués dans une même sortie soient contigus.
- le placement des segments de sortie en une seule colonne.
- et finalement, le positionnement latéral des connecteurs de sortie.

La hauteur du PLA (par rapport au PLA classique) augmente du nombre de duplications plus une fraction qui dépend de la distance entre rappels de masse dans la matrice ET. Il faut donc prévoir dans le plan de masse du circuit l'emplacement nécessaire au PLA.

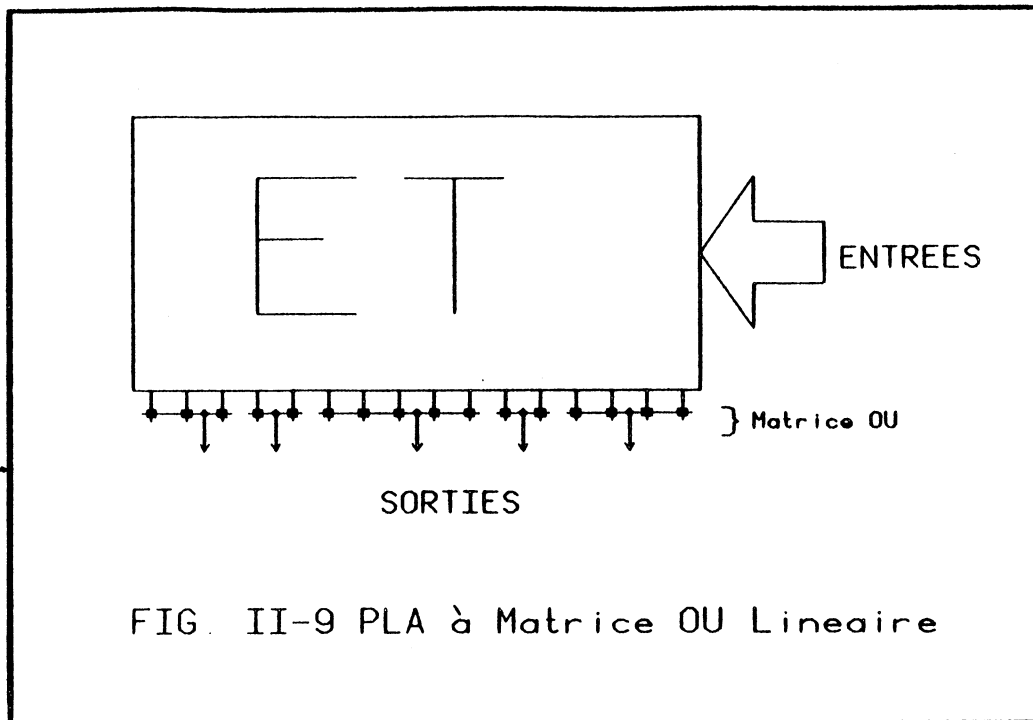
Par contre l'ordre des connecteurs de sortie n'est pas modifié lors de l'optimisation. Il comporte donc tous les avantages des PLAs à sorties latérales.

Le gain en surface dépend évidemment du contenu de la matrice OU, mais dans les cas les plus intéressants (sans duplication de monômes) le gain est de :

$$(KM + KR2/PE) * M * (KS + KR1/PO) * (S-1)$$

C'est à dire que la nouvelle matrice OU contient une seule colonne au lieu de S. La taille de la matrice OU étant réduite au minimum, le temps de propagation sera inférieur à celui du PLA classique équivalent.

Il est intéressant de coupler cette méthode avec des procédés de minimisation logique travaillant sortie par sortie et non globalement sur toutes les fonctions.



La programmation de cette méthode est très simple.

Un exemple de PLA linéaire se trouve dans le circuit MC2 de Hewlett-Packard où il décode l'état de l'automate de séquençage pour générer certaines commandes de la partie opérative.

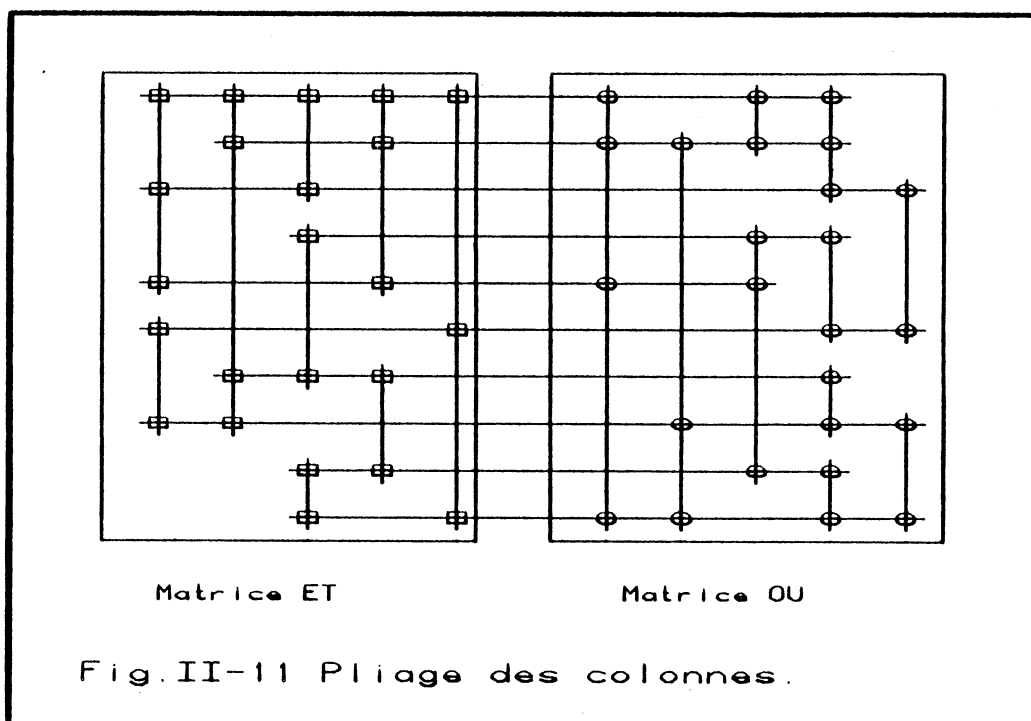
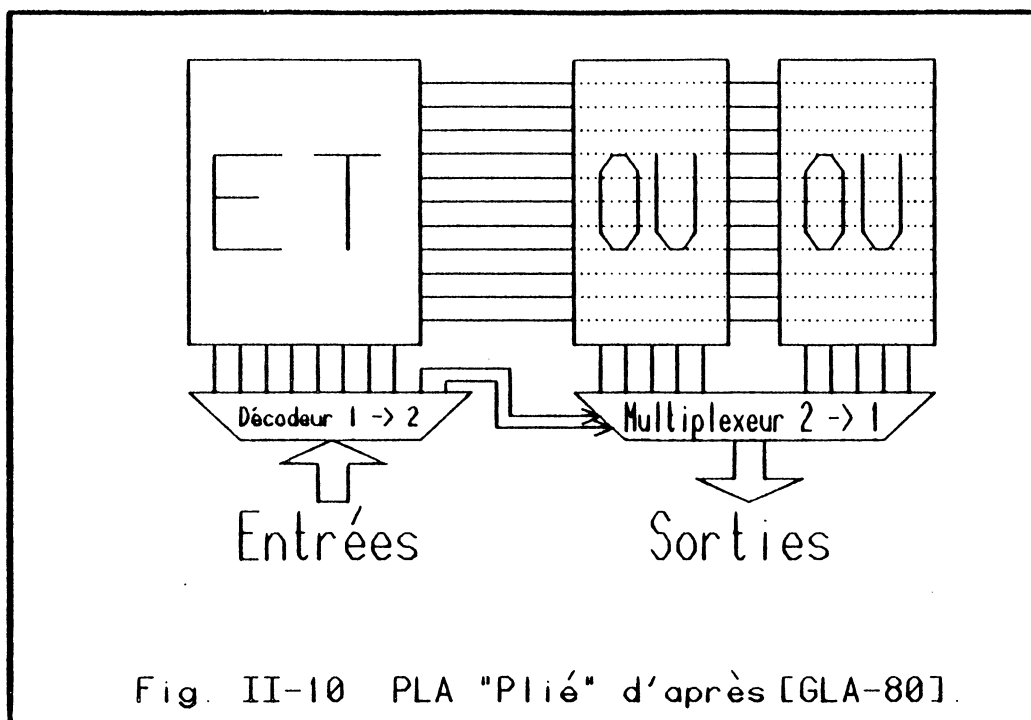
II.J OPTIMISATION PAR PLIAGE (FOLDING)

Il existe dans la littérature deux types d'optimisation désignées sous le nom de «FOLDING» (pliage).

La première [GLA-80] consiste à réaliser des PLAs contenant une matrice ET à NE - 1 entrées, deux matrices OU (à NS sorties chacune) plus un réseau de NS multiplexeurs deux vers un commandé par l'entrée non utilisée dans la matrice ET. Topologiquement les deux matrices OU sont imbriquées. La figure II-10 présente ce type de «pliage». D'après Glasser et Penfield [GLA-80], les avantages sont : la possibilité de modifier la forme du PLA et réduire son temps de réponse et sa consommation.

Le deuxième type de pliage (folding) profite du faible taux de remplissage (nb. de transistors/nb. de points) des matrices. Il consiste :

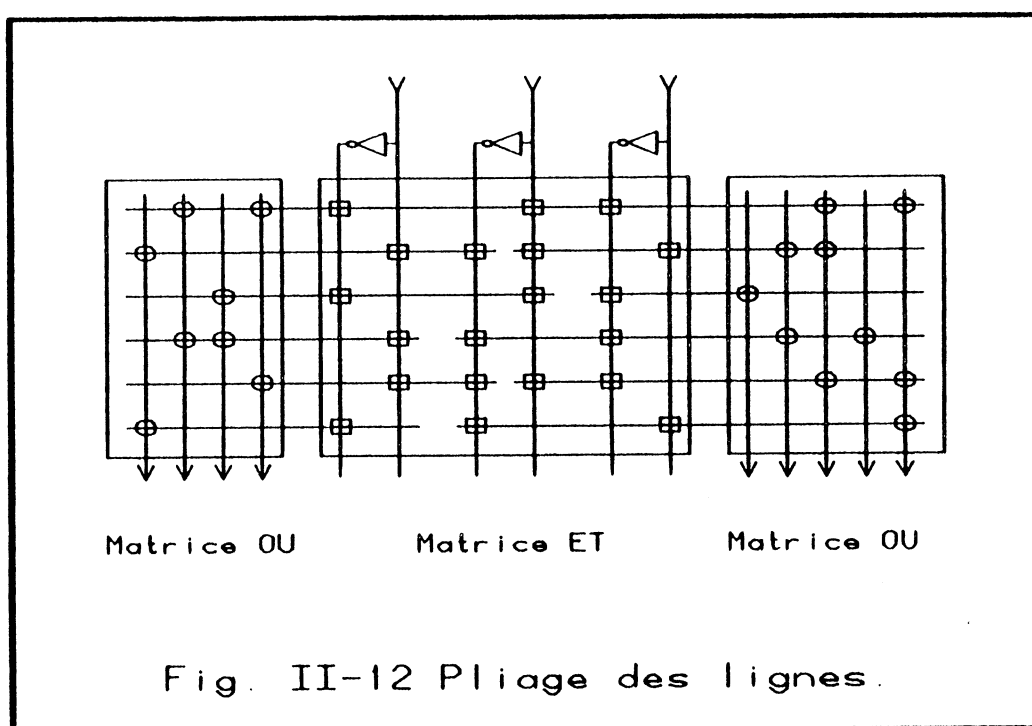
- Pour le pliage de colonnes à placer dans une même colonne deux ou plusieurs entrées et/ou sorties (fig. II-11).



- Pour le pliage de lignes à placer dans une même ligne deux ou plusieurs monômes (fig. II-12).

La première référence à ce type d'optimisation a été présentée par [GRE-76] sous le nom de «segmentation», Greer introduit aussi la notion de PLA monomatrice qui sera reprise ultérieurement [VAR-84]. Schmookler [SCH-80] l'utilise pour «compresser» des Unités Arithmétiques et Logiques réalisées à base de PLAs. Wood [WOO-79] est le premier à introduire le terme «folding» (pliage) qui désignait l'utilisation

1. d'une colonne pour placer deux entrées ou sorties.
2. d'une ligne pour placer deux monômes.



Actuellement cette méthode est appelée pliage simple (simple folding) par opposition au pliage multiple. Le pliage complet d'un PLA qu'il soit fait au niveau des lignes ou des colonnes réduit de moitié la surface du PLA. Le pliage mixte (lignes et colonnes) complet réduit la surface au quart de la surface initiale. D'où l'intérêt croissant que suscite cette méthode d'optimisation.

L'optimisation par pliage est une méthode d'optimisation locale.

Luby et al. [LUB-82] démontrèrent que le problème du pliage des PLAs est un problème NP-complet, ce qui implique qu'il est impensable de trouver un résultat optimal avec des temps de calcul réalistes et ceci sur des exemples de moyenne et grande taille. Les algorithmes existants résolvent donc :

1. soit des sous-problèmes dont la complexité n'est plus exponentielle, mais qui présentent de grandes restrictions; alors une résolution optimale peut être trouvée.
2. soit le problème complet mais avec des heuristiques particulières; alors l'optimalité des solutions ne peut pas être assurée.
3. soit un problème de taille limitée où une solution optimale peut être trouvée dans un temps important mais non prohibitif [GRA-82].

Dans le premier cas on trouve le pliage bipartite [EGA-82], [HU-83] où les «coupures» si elles existent sont placées dans la même ligne (colonne) voir fig. II-13.

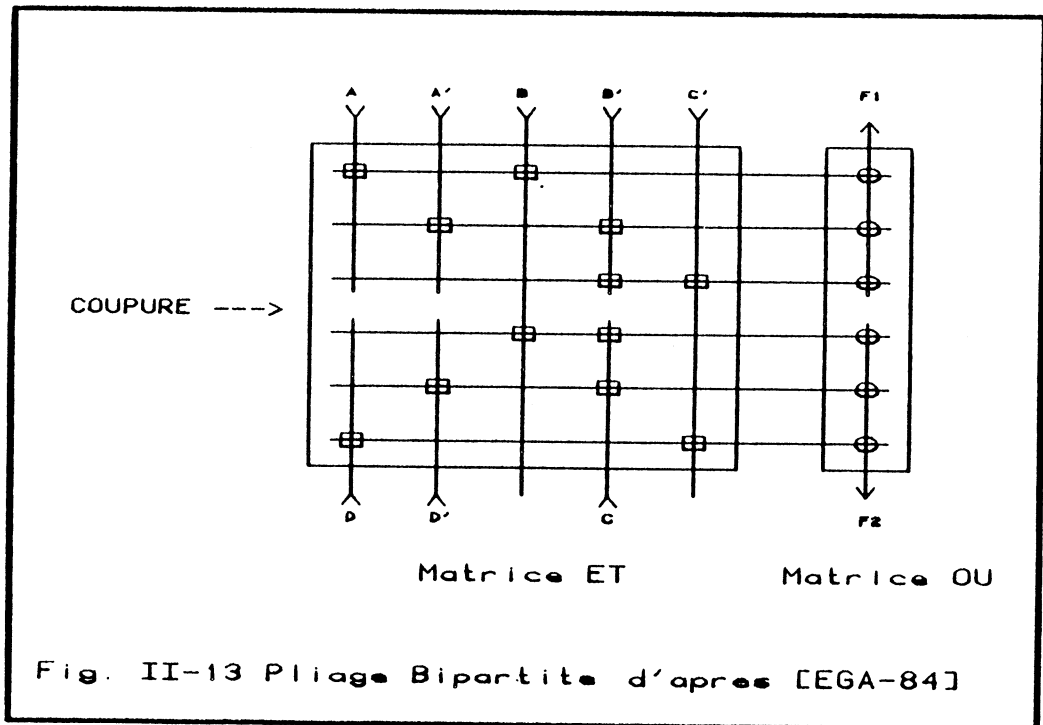


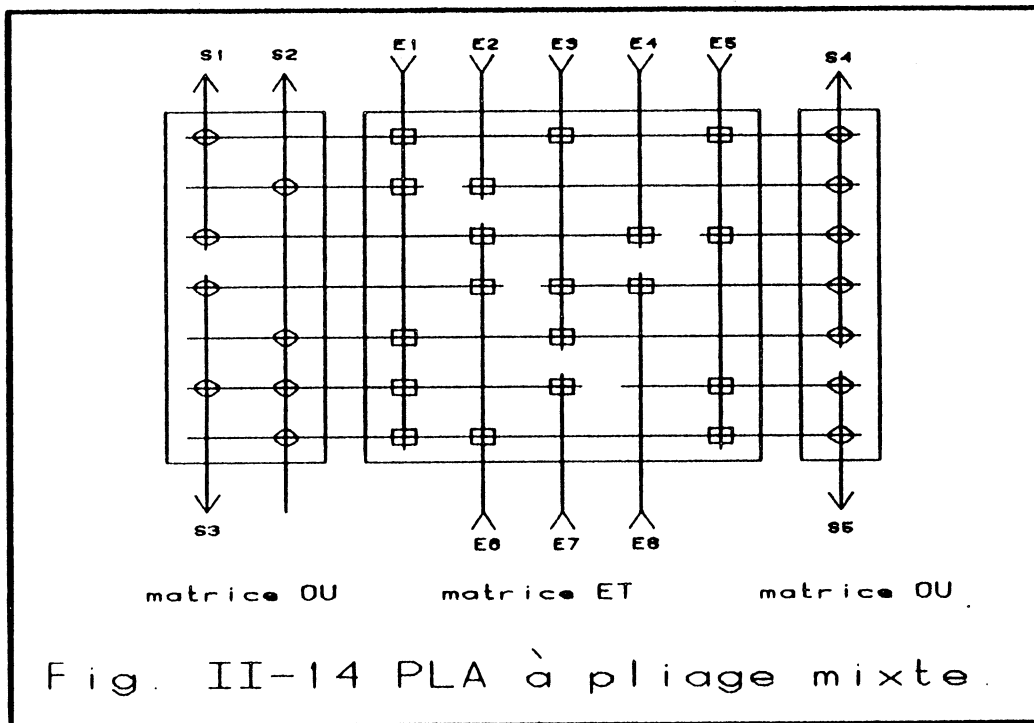
Fig. II-13 Pliage Bipartite d'après [EGA-84]

Dans ce cas la recherche des solutions peut être faite par un parcours d'arbre optimisé pour trouver des solutions rapidement [EGA-82].

Dans le deuxième cas plusieurs heuristiques ont été présentées avec une étude des limites de ces méthodes [LIU-83]. Certains découpent le problème en sous-problèmes en mettant en jeu plusieurs heuristiques [SUW-81], tandis que d'autres traitent le problème dans son ensemble [HAC-82/a], [LUB-82], [DE-83].

Le pliage mixte consiste à faire sur le même PLA du pliage de lignes et du pliage de colonnes (fig. II-14). Dans ce cas les résultats dépendent non seulement du contenu du PLA, mais de l'ordre

dans lequel les deux pliages ont été réalisés.

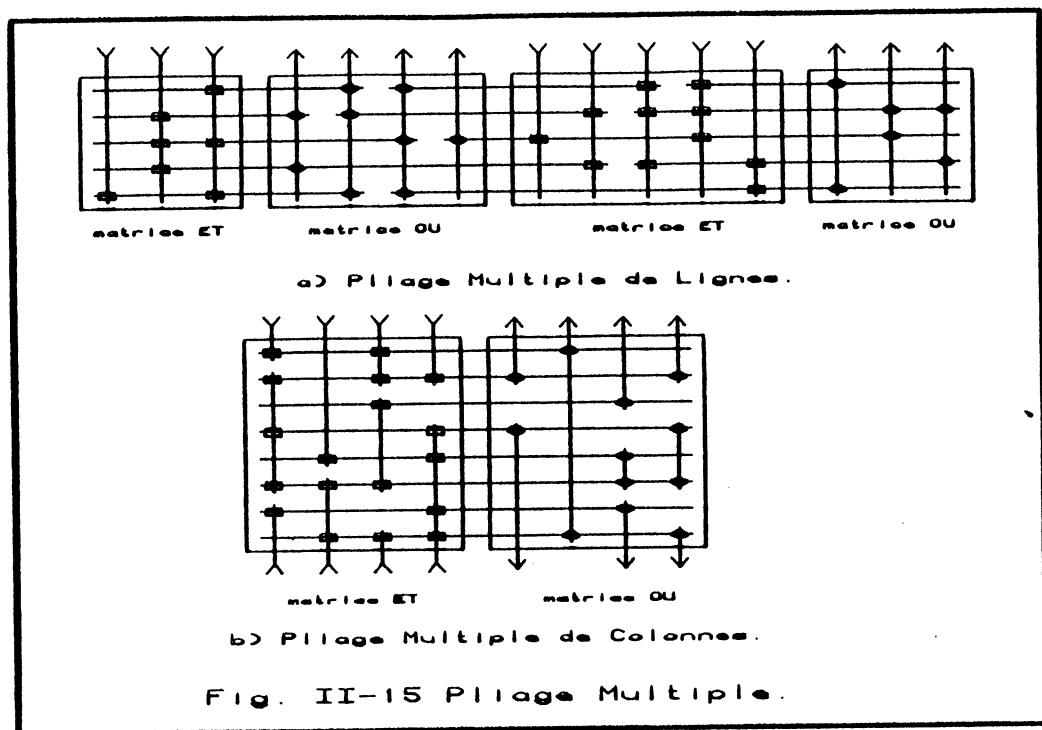


Le pliage multiple (multiple folding), consiste à placer non pas deux, mais plusieurs entrées et/ou sorties (monômes) dans la même colonne (ligne) fig. II-15. Ceci permet une réduction beaucoup plus importante de la surface. Néanmoins de nouvelles contraintes apparaissent :

- le fait d'introduire un pliage multiple rend plus complexes les programmes de dessin des masques.
- la nécessité d'un tracé interne de connexions pour amener les fils d'entrée ou de sortie sur le bord du PLA.
- la conformité au niveau des règles de dessin implique que l'optimisation et le tracé interne des connexions doivent se faire en tenant compte des règles de dessin. Ces contraintes doivent donc «remonter» aux niveaux supérieurs.

Dans un premier temps, ces problèmes ont été complètement ignorés [PAI-81], [HAC-81], puis, on a introduit des coupages du PLA pour le tracé interne [DEM-83/c] ce qui n'est pas une solution acceptable. Il est aussi possible de faire appel à un quatrième niveau d'interconnexion (deuxième metal ou polycide) [DEM-84].

Jusqu'à présent aucun dessin de masques d'un PLA optimisé par pliage multiple n'ayant été présenté (seulement des dessins à main levée), on peut supposer que ces problèmes n'ont pas été encore résolus.



Dans tous les algorithmes de pliage la position des connecteurs d'entrée et sortie est déterminée par les programmes d'optimisation ce qui implique un tracé externe pour relier les entrées et sorties du PLA aux blocs concernés. généralement ce cablage externe occupe toute la surface (sinon plus) qui a été gagnée par l'optimisation du PLA (fig. II-16).

Et même si certains résultats sont très intéressants [HAC-82/b], le tracé externe des connexions entraîne une augmentation globale de surface et non pas une réduction [STE-82].

Un algorithme de pliage avec contraintes à été présenté [DEM-83/b] mais les auteurs n'ont pas insisté sur ses performances...

II.K OPTIMISATION EN LIGNES BRISEES

L'optimisation par lignes brisées consiste à placer autant de segments que possible dans une même colonne.

La méthode peut être vue soit comme approche différente du pliage multiple de colonnes, soit comme une généralisation des méthodes suivantes :

- optimisation par accès latéral.
- optimisation linéaire.

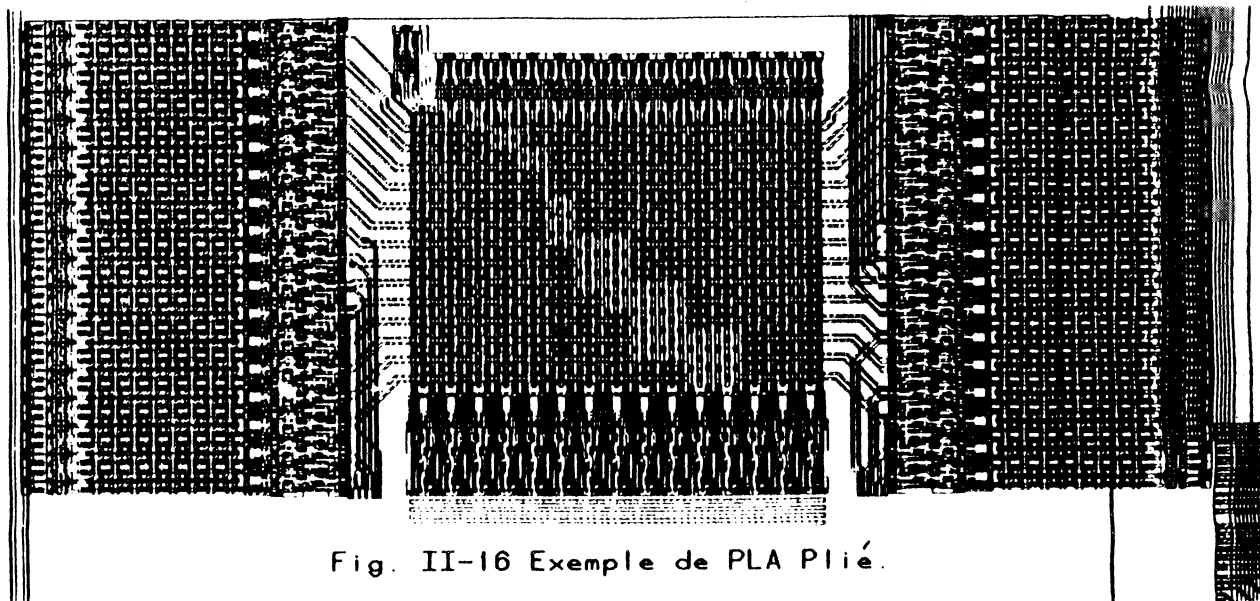


Fig. II-16 Exemple de PLA Plié.

- optimisation par partitionnement des matrices.

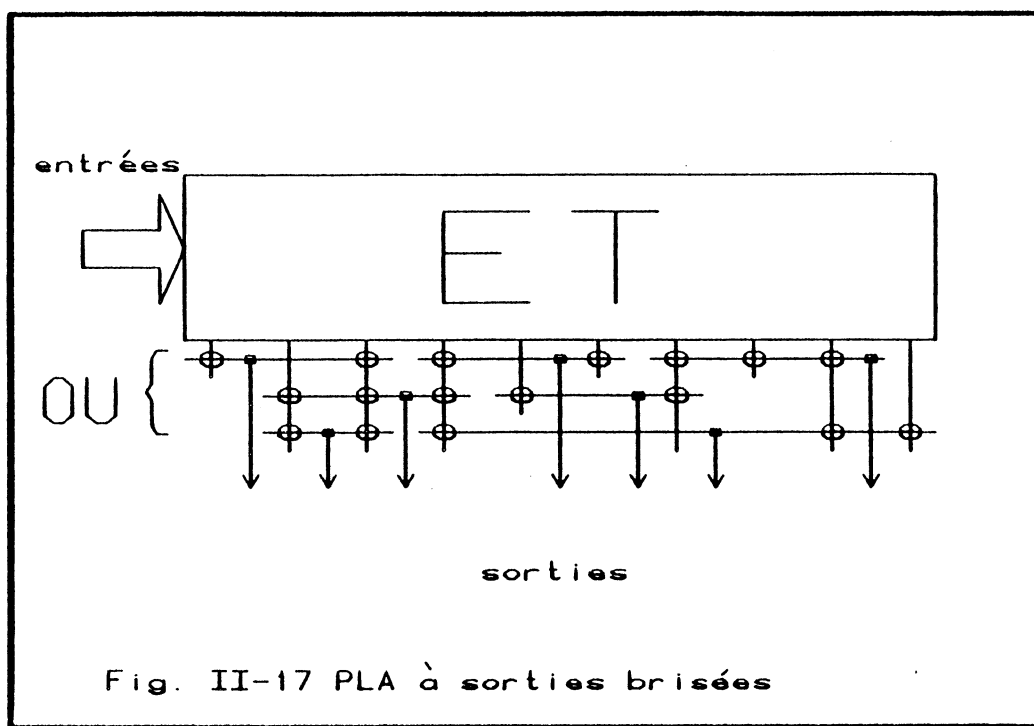
De la première méthode il utilise l'accès latéral aux entrées et aux sorties avec en plus le tracé des connexions internes. Elle ressemble à l'optimisation linéaire pour ce qui est de la possibilité de dupliquer des monômes et car plusieurs segments d'entrée (ou de sortie) peuvent partager une même colonne. En fait les PLAs linéaires sont un cas extrême des PLAs à lignes brisées. Le découpage des matrices permet, en plus, une optimisation particulière pour chaque groupe d'entrées ou de sorties.

Elle introduit des caractéristiques nouvelles :

- l'emplacement des connecteurs d'entrée et sortie (latéral ou non) est défini par l'utilisateur.
- l'optimisation se faisant à partir de ces emplacements, la connectabilité du PLA est assurée.
- la duplication des monômes est contrôlée.

Cette méthode est décrite dans le chapitre suivant.

Le PLA résultant a une hauteur au moins égale à son équivalent en PLA classique, mais une largeur qui peut être beaucoup plus faible. Les gains en surface dépendent du contenu du PLA, mais surtout du taux de remplissage des matrices. Il est donc possible de prévoir statistiquement, et avec une marge d'erreur de 5 à 10% la taille du



PLA optimisé en ne connaissant que le taux de remplissage de celui-ci [REI-83].

L'optimisation par lignes brisées comporte aussi quelques inconvénients :

- le nombre de solutions possibles augmente sous forme exponentielle avec le nombre d'entrées, de sorties et de monômes, il est donc impensable de trouver un optimum dans un temps polynomial. Les algorithmes devront donc être heuristiques et le résultat sera quasi-optimal.
- la régularité topologique étant «cassée», le dessin des PLAs à lignes brisées est beaucoup plus complexe qu'avec les autres méthodes d'optimisation.
- la nécessité d'un tracé interne de connexions entre chaque «sortie» optimisée et le connecteur associé.

De part ses caractéristiques cette méthode convient à l'optimisation des matrices OU des décodeurs ou générateurs de commandes ainsi qu'aux entrées de condition dans les PLAs de séquençement.

Le gain en surface de cette méthode varie en fonction du contenu du PLA entre :

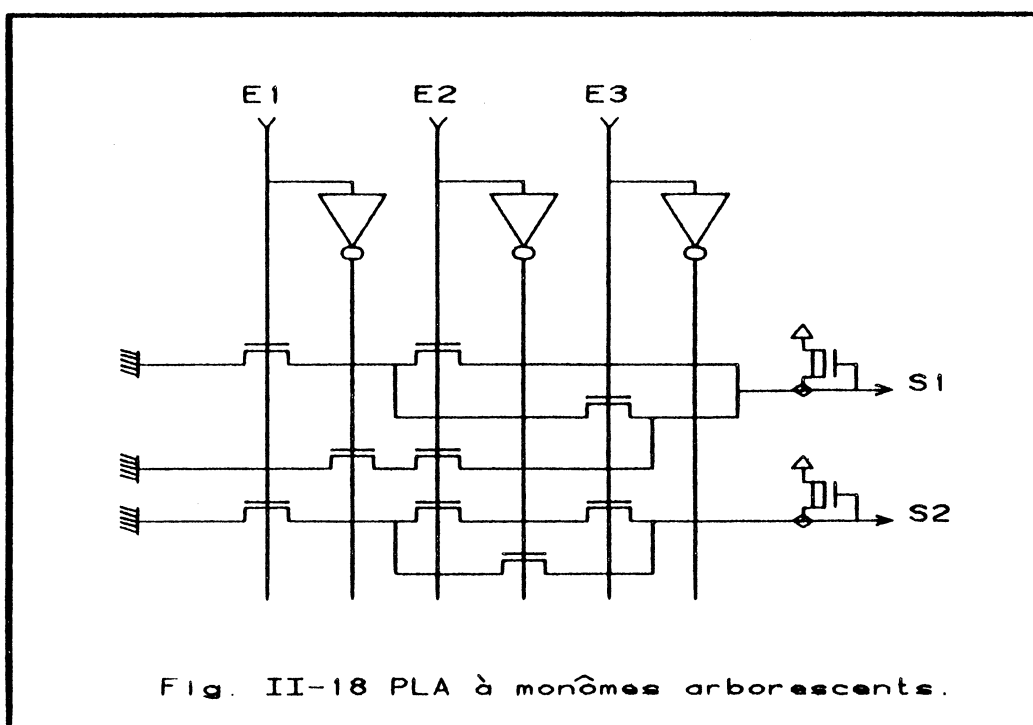
1. 30 et 50% pour des PLAs réalisant à la fois du séquençement et de la génération de commandes.
2. jusqu'à 70% pour des PLAs de décodage.

Dans le microprocesseur Z-80 de ZILOG le décodage des instructions est réalisé par un PLA à lignes brisées tant dans la matrice ET que dans la matrice OU. Avec toutefois une particularité : certains segments s'étendent sur plusieurs colonnes. Dans cet exemple le gain en surface est de 42% par rapport à son équivalent classique, sans compter le gain au niveau du câblage externe.

II.1 OPTIMISATION PAR MONÔMES ARBORESCENTS

L'optimisation par monômes arborescents tient une place particulière dans la famille des méthodes d'optimisation topologique des PLAs car elle modifie leur structure logique. Certains parlent de logique structurée [SHI-79] plutôt que de PLAs optimisés, et c'est en effet une étape intermédiaire entre les PLAs et la logique aléatoire.

Les PLAs à monômes arborescents sont réalisés par des portes logiques complexes (NON-OU-ET) au lieu des deux niveaux NOR-NOR ou NAND-NAND.



Ce type de PLA comporte une matrice unique avec des sorties latérales (fig. II-18), et est équivalent à une structuration d'un ensemble de portes complexes ayant des entrées communes. Leur

principal inconvénient est dû à la mise en série de transistors, ce qui provoque un accroissement du temps de réponse du PLA. Ils ne seront donc utilisés tels quels que dans des circuits ou dans des blocs fonctionnels où le temps de réponse peut être important [GUY-79].

Il existe toutefois certaines conditions où les PLAs à monômes arborescents peuvent présenter des avantages supérieurs aux autres types de PLAs. C'est particulièrement le cas de la validation temporelle de commandes par un ensemble de signaux de temps dans certaines architectures de microprocesseurs [OBR-81]. Dans ce cas le nombre de transistors à mettre en série ne dépasse pas trois pour des circuits à deux niveaux d'interprétation. Dans ce cas, les signaux temporels parcourent tout le PLA, les commandes à valider accèdent latéralement, et les commandes validées sortent du côté opposé (fig. II-19). Ceci impose que les commandes et les signaux temporels soient valides à l'état haut car il n'y a pas de décodeurs (inverseurs) en entrée.

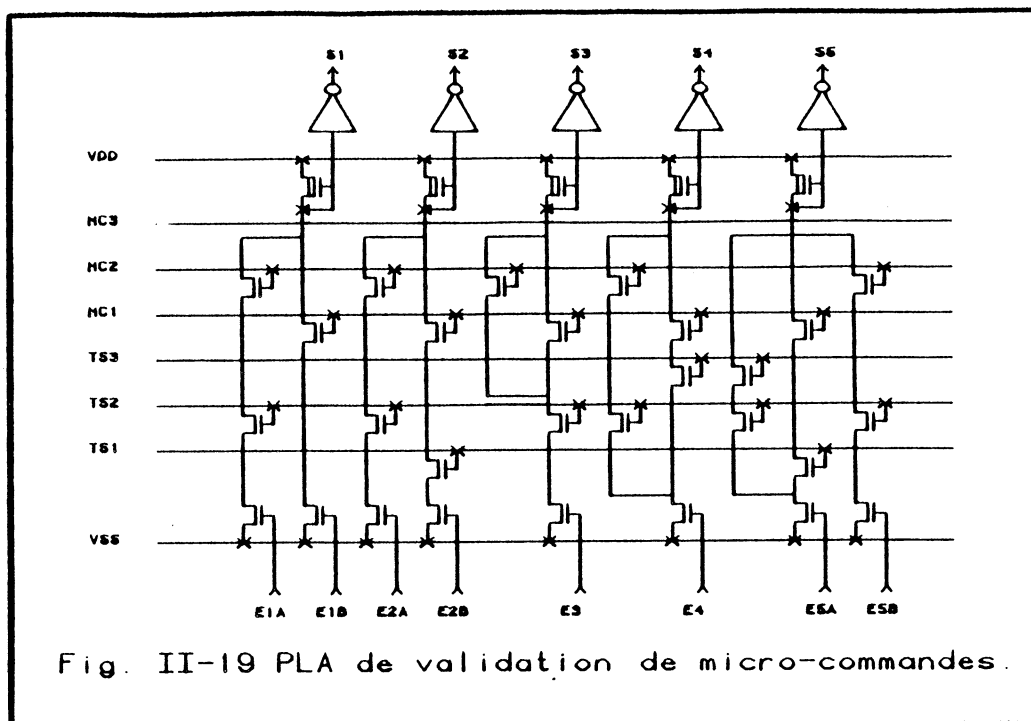


Fig. II-19 PLA de validation de micro-commandes.

Les PLAs à monômes arborescents ressemblent aux PLAs linéaires, mais le fait de fusionner les deux matrices en une seule réduit fortement la puissance consommée car il n'y a que NS portes au lieu de NM + NS, de plus, l'état actif des sorties étant à l'état bas, il n'y a que les sorties actives qui consomment à un instant donné.

Ce type d'optimisation a été utilisé dans la conception des deux PLAs du microprocesseur MOM.400 [GUY-79], [PER-79] et [MAR-78]. Le premier réalisant le décodage des instructions, le deuxième validant temporellement les commandes. les microprocesseurs Z-80 et Z-8000 de ZILOG présentent aussi des structures similaires quoique leurs

concepteurs les ont décrits comme une structuration de la logique anarchique [SHI-79].

OPTIMISATION DES PLAS
OPTIMISATION PAR MONOMES ARBORESCENTS

Page 11-23

CHAPITRE III
LA METHODE
DES LIGNES BRISEES

III.A INTRODUCTION

Dans ce chapitre nous présentons la méthode d'optimisation topologique appelée «PAR LIGNES BRISEES».

Le but de cette méthode est de réduire la surface des PLAs de manière globale dans un circuit. Il faut donc laisser au concepteur la liberté complète dans la disposition des connecteurs d'entrée/sortie, ceci en vue d'un cablage externe le plus simple possible ou pourquoi pas, de connecter le PLA avec les blocs voisins par simple aboutement. Ce principe impose que l'optimisation se fasse à partir des contraintes dûes aux emplacements de ces connecteurs.

Il faut aussi permettre une certaine souplesse dans la forme du PLA pour l'adapter topologiquement au sein du circuit auquel il appartient. L'optimisation devra donc se faire sous contrôle du concepteur.

Etant donné que les PLAs ont une structure régulière, l'optimisation topologique peut se faire sur une représentation symbolique du PLA. Nous avons choisi la représentation structurale qui permet la manipulation au niveau des transistors, et d'où il est aisé d'en déduire les equipotentiels. La génération des dessins des masques étant aussi relativement simple.

Nous avons donc abordé l'optimisation topologique en même temps que le dessin des PLAs optimisés et le tracé interne des connexions.

Certains problèmes concernant le dessin des masques et le tracé des connexions internes étant impossibles à résoudre dans la phase de dessin, nous avons dû introduire des contraintes supplémentaires au niveau de l'optimisation symbolique.

L'optimisation des PLAs se réalise en quatre phases :

- l'ordonnancement des monômes, qui est une étape primordiale pour l'optimisation.
- la duplication des monômes, qui permet de modifier la forme du PLA par un allongement et un rétrécissement de celui-ci.
- le compactage des matrices où l'optimisation proprement dite est réalisée.
- les traitements complémentaires qui préparent les matrices pour la génération des masques et pour le tracé des connexions internes.

L'optimisation peut être réalisée sur une matrice ou sur les deux.

III.B DEFINITIONS ET CONVENTIONS

Nous allons décrire ici les concepts utilisés dans l'application de la méthode des LIGNES BRISEES.

III.B.1 Les Monômes.

Un monôme est défini comme un double vecteur ayant une partie ET et une partie OU. La partie ET du monôme comporte NE éléments dont la valeur est :

- * «E» si ce monôme comporte un transistor associé à l'entrée «E».
- * ZERO pour les autres éléments.

De même la partie OU du monôme comporte NS éléments dont la valeur est :

- * «S» si le monôme est associé via un transistor à la sortie «S».
- * ZERO autrement.

par exemple le monôme suivant

0 2 0 0 5 6 / 1 2 0 0 5

est généré par les entrées 2, 5 et 6, et partage des transistors avec les sorties 1, 2 et 5. Une permutation des éléments de la partie ET ou des éléments de la partie OU ne modifie pas la fonction logique du monôme et donc ne modifie pas celui-ci. On peut dire que les monômes suivants sont identiques :

1 0 3 0 5 6 / 1 0 3 4 0

et

0 1 3 5 6 0 / 1 3 4 0 0

On ne peut avoir deux fois la même valeur du même côté du monôme. Ceci reviendrait à dire qu'il connecté deux fois à la même entrée ou à la même sortie.

Les monômes sont différenciés par leur ordre dans la matrice.

III.B.2 Couverture D'un Segment

La couverture ET (CET) d'un monôme est la partie du monôme délimitée par les deux transistors extrêmes de la partie ET du monôme,

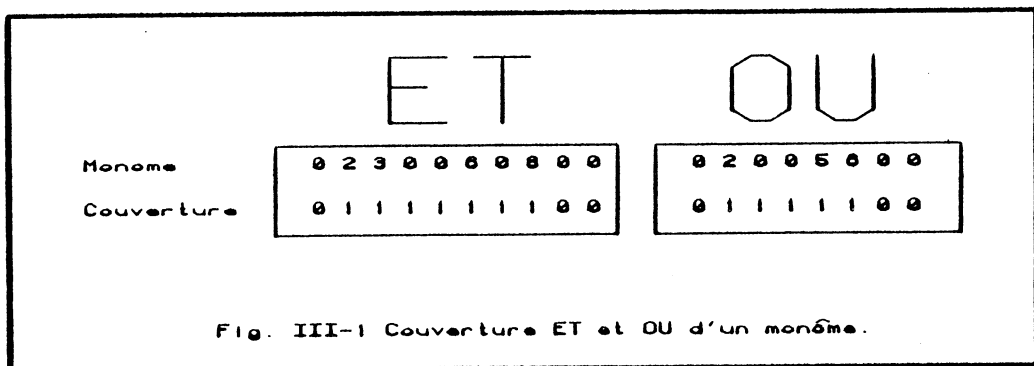
de même la couverture OU d'un monôme est l'ensemble de points entre le premier et le dernier transistor (bornes incluses) du monôme.

$$CET[m] = (CPT[m], CDT[m])$$

avec

$CPT[m]$ = numéro de la colonne de la matrice ET où le premier transistor du monôme m est placé.

$CDT[m]$ = numéro de la colonne de la matrice ET où le dernier transistor du monôme m est placé.



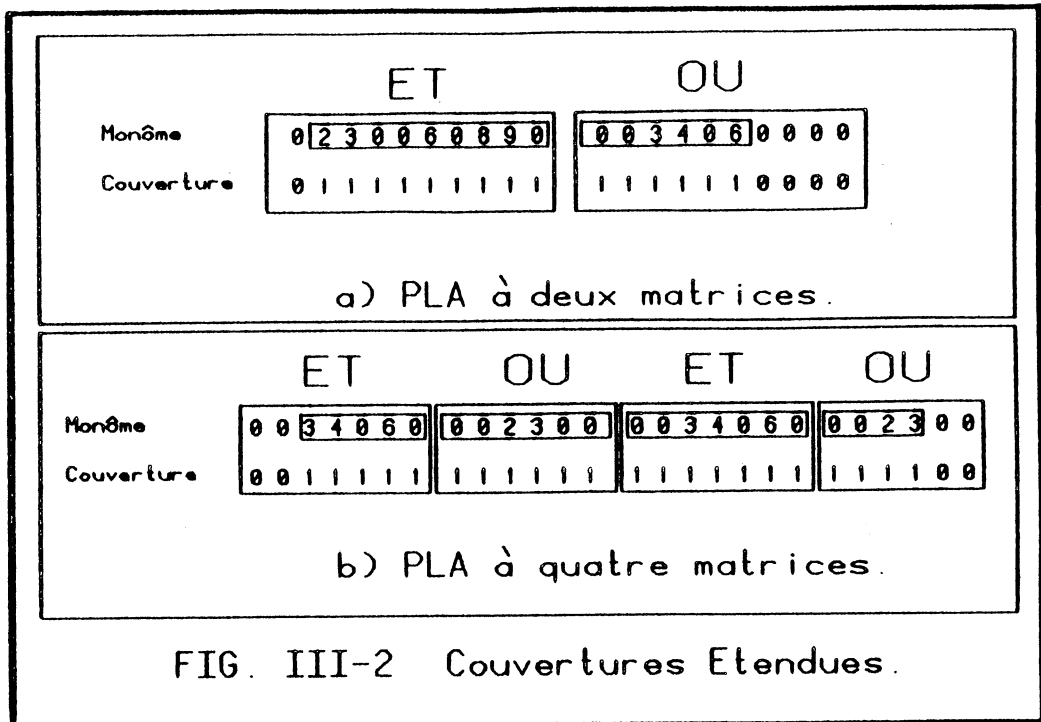
La couverture ET étendue d'un monôme pour un PLA à deux matrices est le segment qui part du premier transistor du monôme et qui finit sur le bord de la matrice ET qui juxtapose la matrice OU.

La couverture OU étendue d'un monôme pour un PLA à deux matrices est le segment qui part du bord intérieur de la matrice OU et qui finit sur le dernier transistor du monôme.

Dans un PLA à plusieurs matrices, la couverture étendue d'un monôme est celle déjà décrite pour les matrices externes; pour les matrices internes la couverture étendue d'un monôme est le segment du monôme qui relie les deux bords, droit et gauche, de la matrice.

Il faut souligner que la couverture d'un monôme comprend tous les transistors du monôme dans la matrice plus les points (cellules) intermédiaires. La couverture étendue comprend la couverture du monôme plus tous les points (vides) jusqu'au(x) bord(s) de la matrice.

Le nombre de points contenus dans la couverture d'un monôme est au minimum égal au nombre de transistors du monôme (dans la matrice considérée), et au maximum au nombre de colonnes de la matrice. Ce nombre dépend de la position (colonne) des entrées dans la matrice ET et des sorties dans la matrice OU.



III.8.3 Segment D'Entrée, Segment De Sortie.

Un segment d'entrée est le segment associé à une entrée, limitée par le premier et dernier transistor de cette entrée. Le segment d'entrée est égal à la couverture de l'entrée.

La même définition s'applique aux segments de sortie.

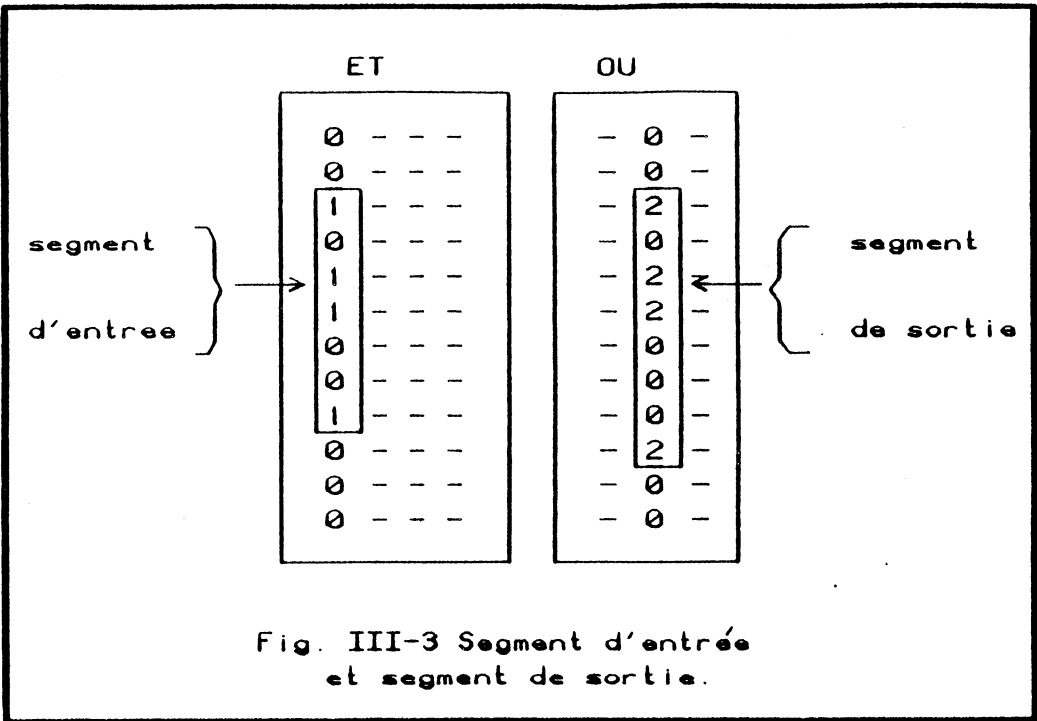
Physiquement un segment d'entrée correspond à une ligne en silicium polycristallin couvrant tous les transistors que cette entrée attaque ainsi que les points nécessairement vides entre ceux-ci.

Un segment de sortie correspond à une ligne d'aluminium qui relie tous les drains des transistors qui génèrent cette sortie.

Les segments d'entrée et de sortie couvrent au moins autant de points que de transistors leur sont associés, et au maximum ils couvrent toute une colonne de la matrice.

Nous parlerons de couverture d'une entrée ou d'une sortie pour désigner l'extension du segment :

- soit jusqu'à un des bords (supérieur ou inférieur) de la matrice.
- soit jusqu'aux deux bords.
- soit jusqu'à un point déterminé généralement face au connecteur de cette entrée ou sortie.



III.C ORDONNACEMENT DES MONOMES

C'est dans cette phase que la position des connecteurs est prise en compte, et c'est l'ordonnement des monômes qui conditionnera l'aptitude à compacter les matrices.

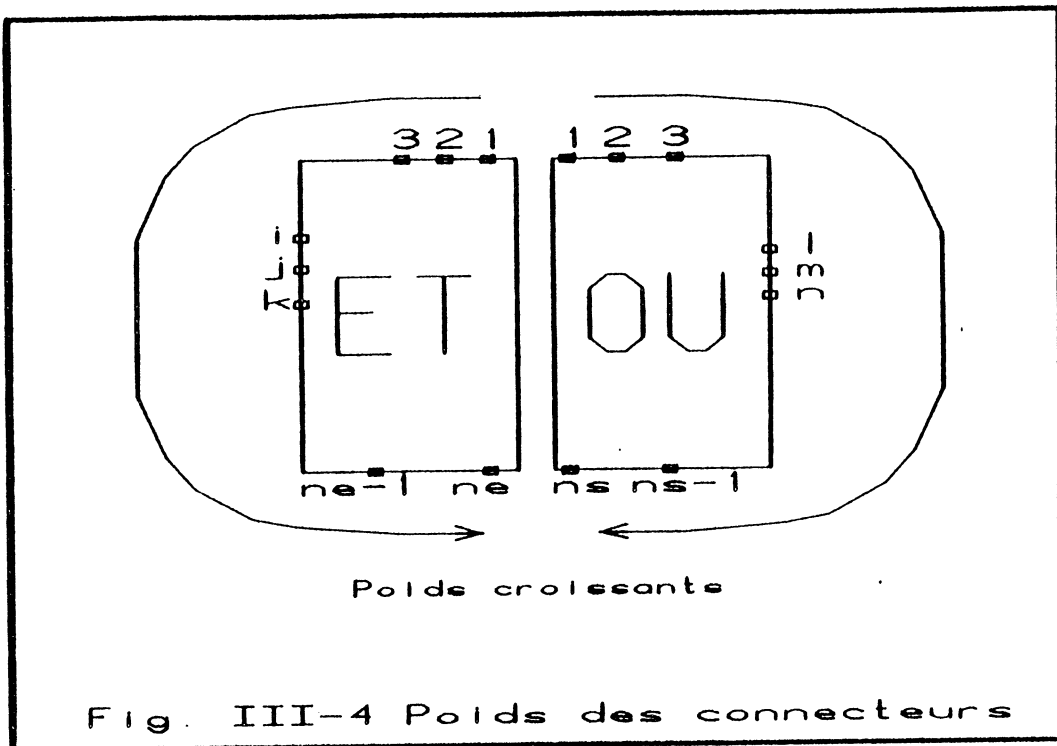
Les entrées d'un PLA attaquent des monômes, ces derniers attaquent à leur tour les sorties. Pour optimiser un PLA à partir des connecteurs il faut que les monômes attaqués par une entrée soient disposés près du connecteur de cette entrée. De même les monômes qui attaquent une sortie doivent être disposés près du connecteur de cette sortie.

Nous commençons donc par donner un poids à chaque sortie en fonction de l'ordre de son connecteur (fig. III-4). On détermine le barycentre de chaque monôme de la manière suivante :

$$B_i = \frac{\text{SOMME } (j \text{ si le monôme } i \text{ est connecté à la sortie } j)}{\text{SOMME } (1 \text{ si le monôme } i \text{ est connecté à la sortie } j)}$$

et on trie les monômes suivant leur barycentre.

Ainsi chaque monôme se trouve placé dans une position moyenne entre les sorties qui lui sont connectées.



La taille des segments de sortie est réduite globalement. Le but de cet ordonnancement est la «diagonalisation» de la matrice OU.

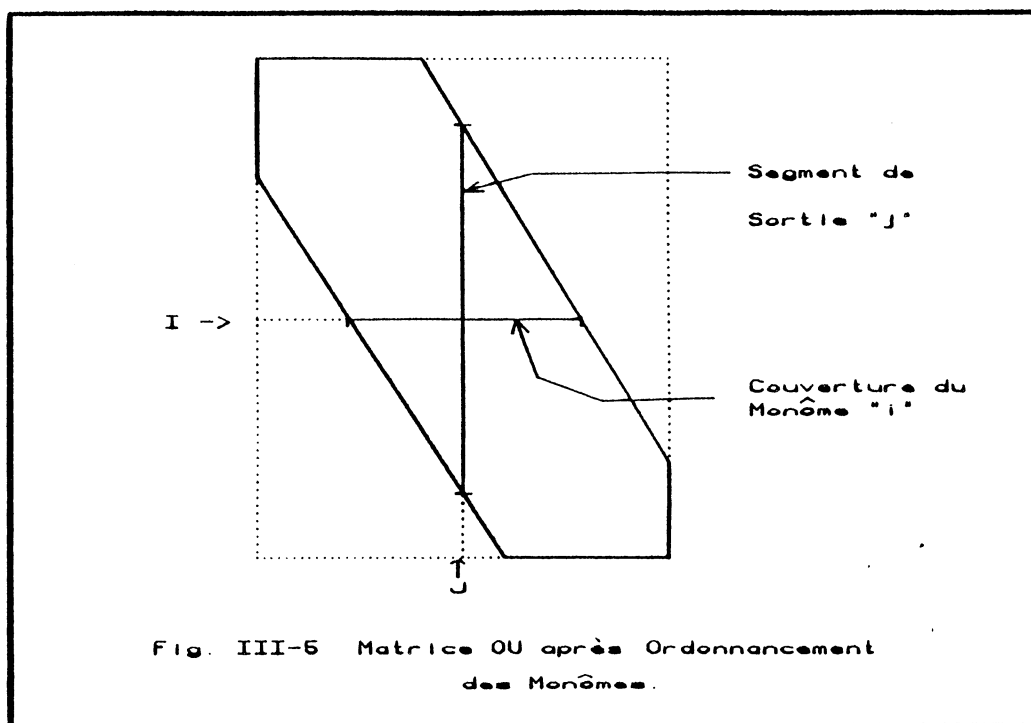
Le réordonnancement des monômes tel qu'il vient d'être présenté se réalise sur la matrice OU. Mais chaque monôme ayant aussi une partie dans la matrice ET, il faut que ce réordonnancement soit répercuté sur cette dernière. C'est à dire que si on déplace la partie OU d'un monôme, il faut faire de même pour sa partie ET, et vis-versa.

Ce réordonnancement peut aussi se faire en fonction de la position des entrées. pour ceci il faut donc calculer le barycentre en fonction de celles-ci au lieu des sorties, ou pourquoi pas, en tenant compte en même temps des entrées et des sorties.

Si on fait un ordonnancement simple, soit à partir des entrées, soit à partir des sorties, on privilégie une matrice par rapport à l'autre. On obtiendra alors une matrice bien optimisée, l'autre l'étant moyennement ou pas du tout.

Mais, si l'ordonnancement se fait en tenant compte à la fois des entrées et des sorties, on obtiendra un PLA dont les deux matrices sont mal optimisées. C'est pourquoi cette dernière option ne sera pratiquement pas utilisée.

Les PLAs ayant normalement une matrice OU beaucoup plus creuse que la matrice ET, il est donc nécessaire de pouvoir privilégier la première plutôt que la dernière.



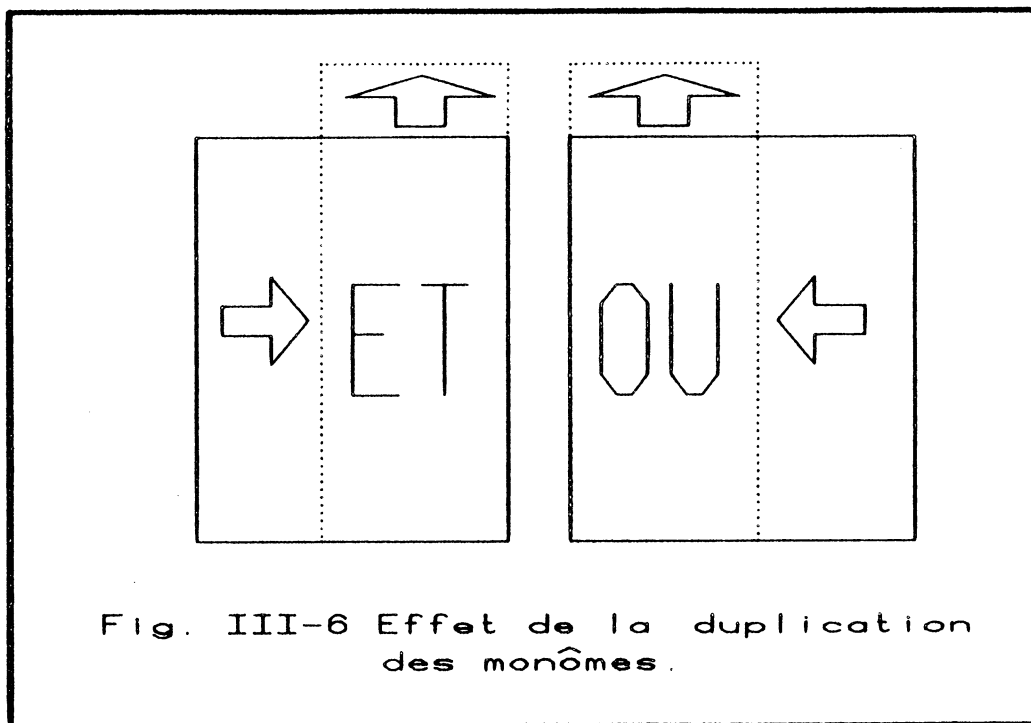
III.D DUPLICATION DES MONOMES

Il faut noter que le placement barycentrique regroupe les transistors autour de la diagonale de la matrice (Fig. I-5), et si jamais tous les transistors sont placés sur cette diagonale, le compactage pourra être complet (sur une seule colonne). Donc tous les moyens qui aident à la diagonalisation des matrices permettront un compactage plus important.

La diagonalisation de la matrice ET ne peut être améliorée sans modifier sa structure logique, par contre il est possible de rendre la matrice OU plus diagonale en dupliquant des monômes.

Ceci se réalise de la manière suivante :

- il faut d'abord choisir le monôme à dupliquer.
- on crée un nouveau monôme dont la partie ET est identique à la partie ET du monôme à dupliquer.
- on détermine un point de la couverture OU du monôme à dupliquer.
- la partie OU à gauche de ce point est dupliquée dans le monôme fils et mise à zéro dans le monôme père.
- on calcule le barycentre des deux monômes et les deux monômes complets (partie ET et partie OU) sont positionnés dans les matrices en fonction de leur barycentre.



Cette duplication allonge le PLA dans le sens de la hauteur, mais la couverture des nouveaux monômes ainsi que celle des sorties associées seront plus faibles. La réduction des segments de sortie va peut être permettre le placement de deux sorties sur une même colonne (ou niveau). L'allongement du PLA peut améliorer sa connexion avec le bloc contrôlé.

La difficulté de cette phase consiste à choisir le monôme qui une fois dupliqué permettra effectivement de gagner une colonne, ainsi que le point de coupure.

III.D.1 Choix Du Monôme A Dupliquer.

Le choix de ce monôme est crucial pour les performances du compactage. La détermination du monôme à dupliquer doit se faire sur un critère rapidement calculable car il doit être déterminé pour tous les monômes du PLA.

Nous avons testé plusieurs critères de choix que nous présentons ci-dessous.

III.D.1.a Nombre De Transistors Controlés (NBT).

Plus un monôme controle de transistors (sorties), plus sa couverture est importante, donc plus une duplication de celui-ci améliorera la diagonalisation de la matrice OU. On obtient donc le premier critère :

choisir le monôme i tel que

$$NBT[i] = \max_j (NBT[j])$$

pour j = [1..NM]

avec

$$NBT[j] = \sum_k (1 \text{ si le monôme } j \text{ est associé à la sortie } k, \\ 0 \text{ sinon})$$

pour k = [1..NS]

III.D.1.b DELTA (Longueur De La Couverture Du Monôme).

D'un autre côté il semble normal que la duplication d'un monôme qui a une grande couverture permette une meilleure diagonalisation qu'un monôme à faible couverture. On définit alors la longueur de la couverture d'un monôme de la manière suivante :

$$DELTA[m] = CDT[m] - CPT[m]$$

Le choix se fait alors sur le monôme qui a la couverture la plus importante.

III.D.1.c Distance Barycentre Moyenne (B-M).

On peut définir le facteur d'équilibre d'un monôme comme la distance entre son barycentre et le point milieu de sa couverture.

$$EQ[m] = |(CDT[m] - CPT[m])/2 - BAR[m]|$$

Les monômes équilibrés (ou centrés) auront donc un facteur d'équilibre proche de zéro, tandis que les monômes déséquilibrés auront une valeur plus importante. Le choix du monôme à dupliquer peut se porter sur le monôme le plus déséquilibré.

III.D.1.d Facteur D'Occupation (FO).

Il peut exister deux monômes ayant même couverture et même facteur d'équilibre, mais avec un nombre différent de transistors. Par exemple :

monôme [i] 0 2 3 4 5 0 0

monôme [j] 0 2 0 0 5 0 0

Dans ce cas les deux monômes ont la même couverture et le même barycentre, mais la duplication du deuxième entrainera une réduction plus importante des segments de sortie 2 et 5.

On définit alors le facteur d'occupation comme la somme de la longueur de la couverture et du nombre de transistors.

$$FO[m] = DELTA[m] + NBT[m]$$

Le choix se fait alors sur le monôme qui présente le facteur d'occupation le plus important.

III.D.1.e Facteur D'Utilisation (FU).

On peut trouver le cas où le facteur d'occupation ne donne pas satisfaction. Soient les monômes

monôme[i] = 0 0 3 0 0 0 0 8 0

et

monôme[j] = 0 0 0 4 5 6 7 0 0

Ils ont tous les deux le même facteur d'occupation, mais manifestement la duplication du monôme i rendra la matrice plus diagonale que celle du monôme j.

On définit alors le facteur d'utilisation comme :

$$FU[m] = NBT[m]/DELTA[m]$$

Ce qui représente le «remplissage» de la couverture du monôme. Le choix doit alors se faire sur le monôme à plus faible facteur d'utilisation.

III.D.1.f Choix Par Taille De Trou.

On désigne par «TROU HORIZONTAL» un ensemble de points vides contigus à l'intérieur de la couverture du monôme.

Monôme	0	<table><tr><td>2</td><td>0</td><td>0</td><td>5</td><td>0</td><td>7</td><td>0</td><td>9</td><td>10</td><td>0</td><td>0</td><td>0</td><td>14</td></tr></table>													2	0	0	5	0	7	0	9	10	0	0	0	14	0
2	0	0	5	0	7	0	9	10	0	0	0	14																
Trou	0	0	<table><tr><td>2</td><td>2</td></tr></table>		2	2	0	<table><tr><td>1</td></tr></table>	1	0	<table><tr><td>1</td></tr></table>	1	0	0	<table><tr><td>3</td><td>3</td><td>3</td></tr></table>			3	3	3	0	0						
2	2																											
1																												
1																												
3	3	3																										

Fig. III-7 "Trou" Horizontal.

On définit alors :

$TROU[m] = \text{Nb. de points vides contigus dans la couverture de } m$

Le choix par taille de trou consiste à prendre le monôme qui comporte le trou le plus grand et qui donnera par la duplication deux monômes fils très distants.

III.D.1.g Choix Par Distance Bord-Trou Vertical Pondérée.

On peut aussi considérer les trous des sorties qui sont connectées à chaque monôme. Dans ce cas on peut se limiter aux trous contigus au premier ou au dernier transistor de la sortie.

On calcule alors pour chaque monôme et pour chaque sortie du monôme le «bord-trou» correspondant. On pondère cette valeur en fonction de la distance de la sortie avec le barycentre du monôme.

On calcule la matrice des trous verticaux :

$TV[m,s] = \text{nb. de points vides contigus verticalement au point } M, S \text{ si ce point est vide et si il appartient à la couverture (verticale) de la sortie } s, \text{ zéro autrement.}$

à partir de celle-ci on détermine la matrice des bords-trous-verticaux

$BTV[m, s] = TV[m, s+1]$ si le point $[M, S]$ contient le premier transistor de la sortie s ,
 $= TV[m, s-1]$ si le point $[M, S]$ contient le dernier transistor de la sortie s ,
 $= \text{zéro autrement.}$

finalement, pour chaque monôme

$DBTVP[m] = \text{SOMME } (BTV[m,s] * |(bary[m] - s)|)$
pour $s = 1 \text{ à } S$

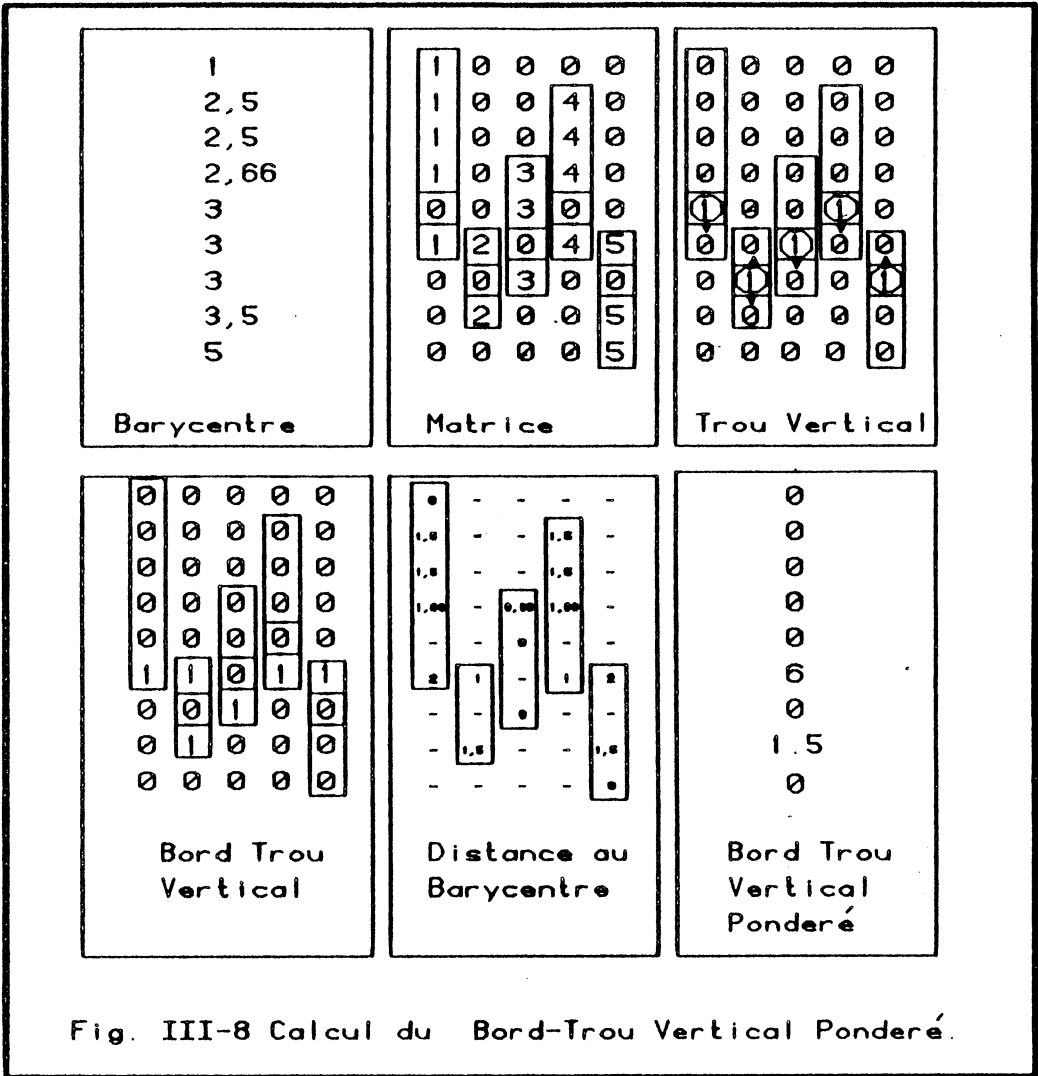


Fig. III-8 Calcul du Bord-Trou Vertical Ponderé.

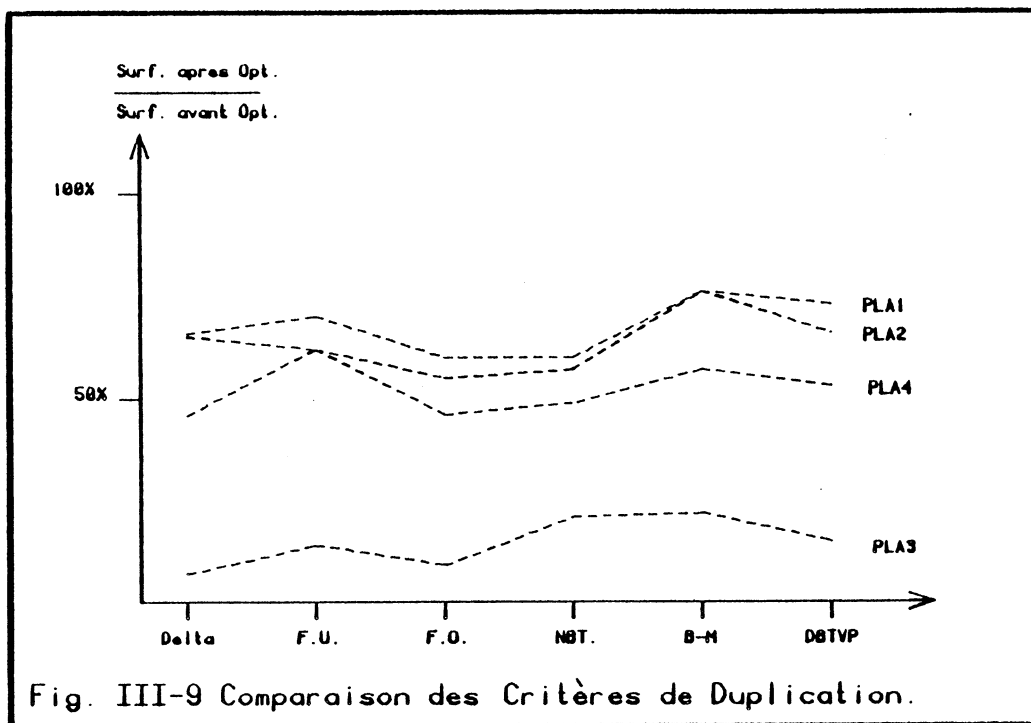
Le monôme à dupliquer est alors celui qui a le DBTVP plus important.

Ce critère favorise les monômes, qui étant connectés à des sorties très éloignées, allongent beaucoup les segments de ces sorties.

Ce critère doit être évalué entièrement à chaque duplication de monôme. Il est donc très coûteux en temps de calcul.

III.D.2 Comparaison Des Critères De Choix.

Ces différents critères ont été testés séparément sur plusieurs exemples et les résultats après compactage sont décrits dans la figure. III-9, le facteur d'occupation donnant les meilleurs résultats dans tous les cas.

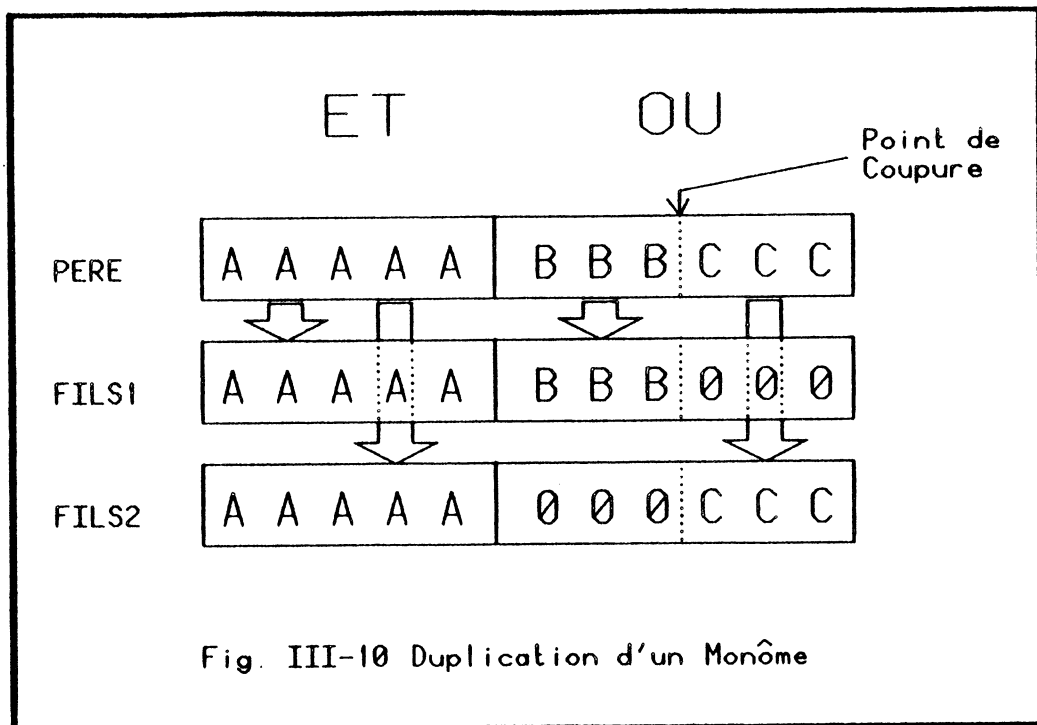


Ces critères favorisent des caractéristiques différentes des monômes et, sur le nombre restreint d'exemples que nous avons pu tester, il est difficile de cerner exactement les particularités du PLA d'après lesquelles il convient mieux de réaliser la duplication suivant tel critère. De même il pourrait être intéressant de faire les premières duplications suivant un critère particulier et d'utiliser ensuite d'autres critères.

III.D.3 Duplication Des Monômes.

Une fois le monôme choisi, la duplication se réalise de la manière suivante :

- copie de la partie ET du monôme père dans la partie ET des deux monômes fils.
- coupure de la partie OU du monôme père. Le choix du point de coupure est très important pour la diagonalisation de la matrice OU.
- copie dans les monômes fils des deux parties OU du père résultantes de la coupure.
- destruction du monôme père.
- placement des monômes fils dans les matrices en fonction de leurs caractéristiques.



On peut fixer plusieurs critères pour déterminer le point de coupure des monômes à dupliquer :

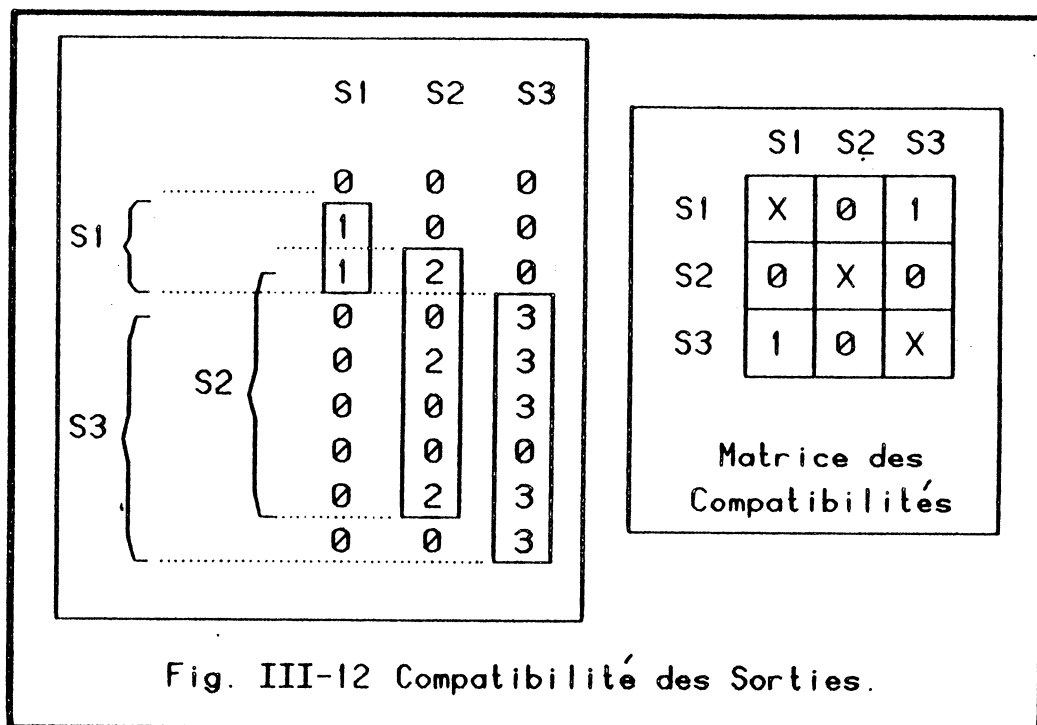
- le barycentre $BARY[m]$
- le point milieu $(CPT[m] + CDT[m])/2$
- le «trou horizontal» le plus grand.
- le «trou horizontal» qui couvre le barycentre
- le «trou horizontal» qui couvre le point milieu

Chaque critère donnant des résultats en fonction des caractéristiques tant des monômes comme des sorties qui leur sont associées.

III.E COMPACTAGE DES MATRICES

La phase de compactage des matrices réalise l'optimisation proprement dite des matrices. Son but est de regrouper le plus de segments possible dans un minimum de colonnes.

Le regroupement de sorties sur une même colonne se fait en fonction de la COMPATIBILITE des sorties.



Pour regrouper plusieurs sorties dans une même colonne il est nécessaire qu'elles soient compatibles deux à deux.

Le nombre de possibilités pour regrouper N objets en P groupes de 1 à N éléments avec P compris entre 1 et N croît de manière exponentielle avec N, il faut donc utiliser des heuristiques pour résoudre même des exemples de taille moyenne. Et il est alors impossible de garantir un résultat optimal.

Nous avons donc proposé l'heuristique suivante :

- trier les sorties par nombre de compatibilités décroissant.
- à partir de la sortie qui a le moins de compatibilités choisir un ensemble de sorties compatibles deux à deux pour les regrouper dans une même colonne. Ces sorties seront éliminées du groupe de départ.
- répéter le pas précédent jusqu'à ne plus avoir de sorties.

Le tri des sorties permet de repérer rapidement les sorties qui ont le moins de compatibilités, de les traiter et donc de réduire rapidement la complexité du problème.

III.E.1 Regroupement Des Sorties.

L'ensemble de départ est la sortie ayant le nombre le plus faible

de compatibilités plus toutes les sorties qui lui sont compatibles. L'algorithme de regroupement est le suivant :

1. si toutes les sorties de l'ensemble sont compatibles deux à deux le regroupement est fini : toutes les sorties peuvent être placées dans la même colonne. Sinon on passe à l'étape deux.
2. On détermine pour chacune des sorties de l'ensemble le nombre de compatibilités qu'elle a dans cet ensemble.
3. On rejette la sortie qui a le moins de compatibilités dans le groupe et on boucle à l'étape numero un.

Cet algorithme garantit qu'à chaque passage on détermine un ensemble de sorties compatibles deux à deux, le fait de rejeter la sortie la moins compatible de l'ensemble accélère le traitement, mais l'absence de recherche exhaustive ne garantit pas une solution optimale.

III.F TRAITEMENTS COMPLEMENTAIRES

L'optimisation faite jusqu'ici est réalisée sur une représentation symbolique qui ne montre qu'une partie des informations nécessaires à la définition du PLA. Pour obtenir le dessin des masques complet du PLA optimisé il faut encore interpréter cette description symbolique pour tracer les éléments graphiques qui décrivent :

- les matrices de transistors
- et ce que nous appellerons les accès aux segments, c'est à dire le tracé interne des connexions qui relieront chaque segment avec son connecteur.

Il existe des problèmes liés à ces deux traitements supplémentaires qui, du fait du symbolisme utilisé, ne sont pas directement visibles lors de l'optimisation. Ces problèmes peuvent être ignorés pendant celle-ci, et ils devront alors être résolus au niveau où ils apparaissent, mais il est aussi possible de les rendre visibles au niveau symbolique par le truchement des informations utilisées lors de l'optimisation.

On utilisera alors certaines techniques pour faire « remonter » ces contraintes, propres au dessin et au tracé des connexions, de telle sorte qu'elles puissent être résolues pendant le compactage.

III.F.1 Conflits Topologiques Dans La Matrice OU.

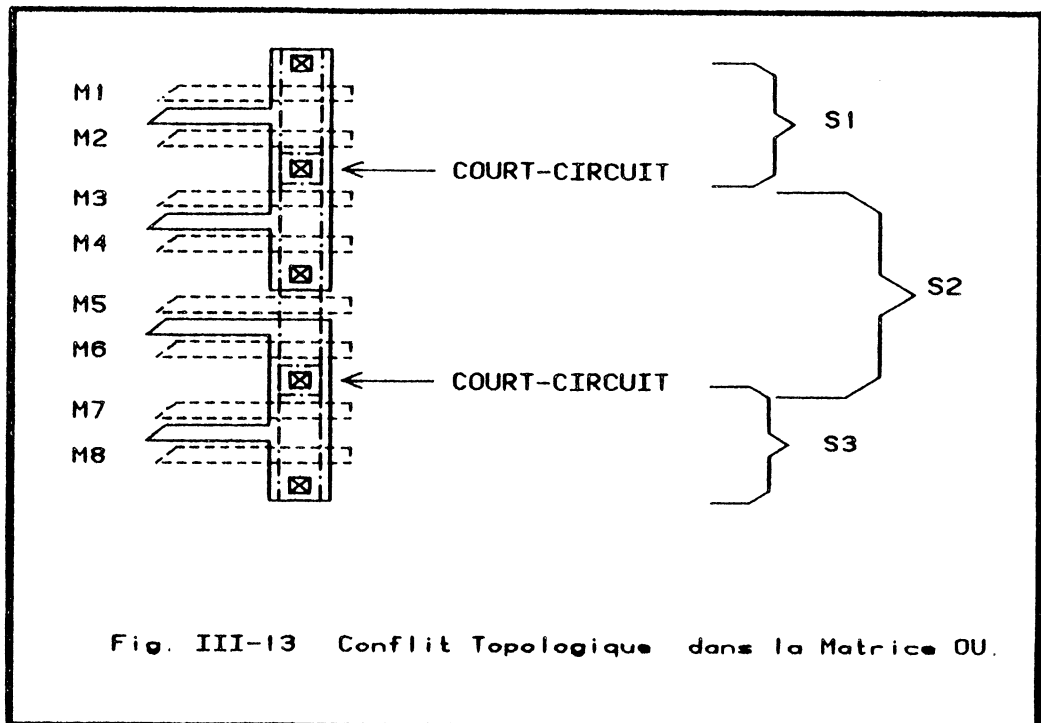


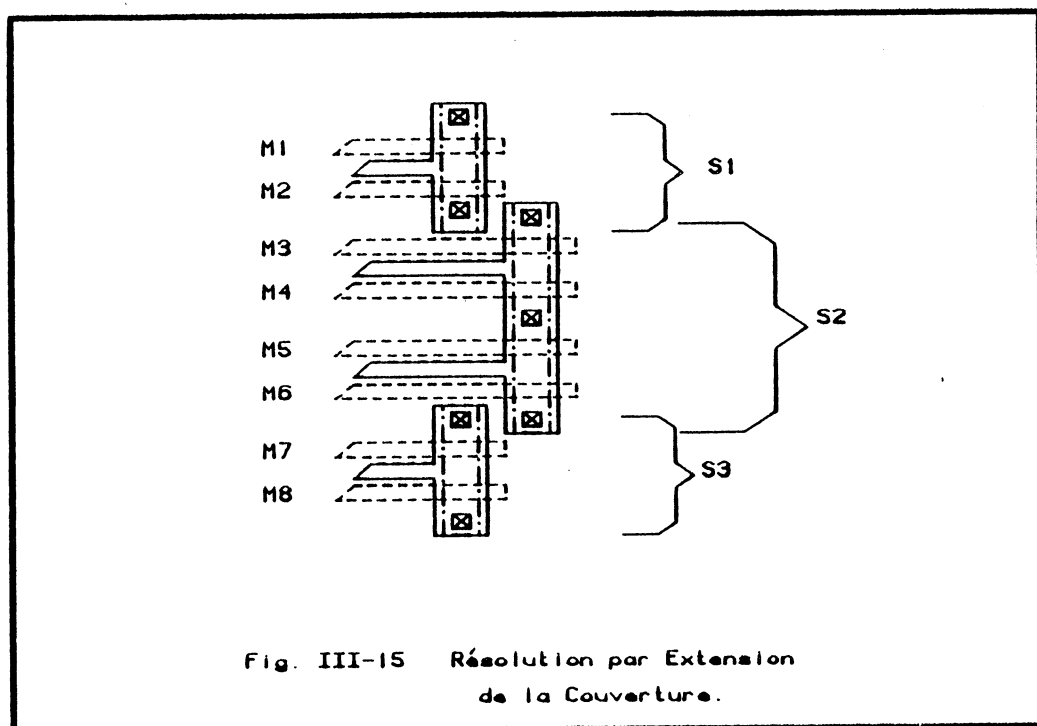
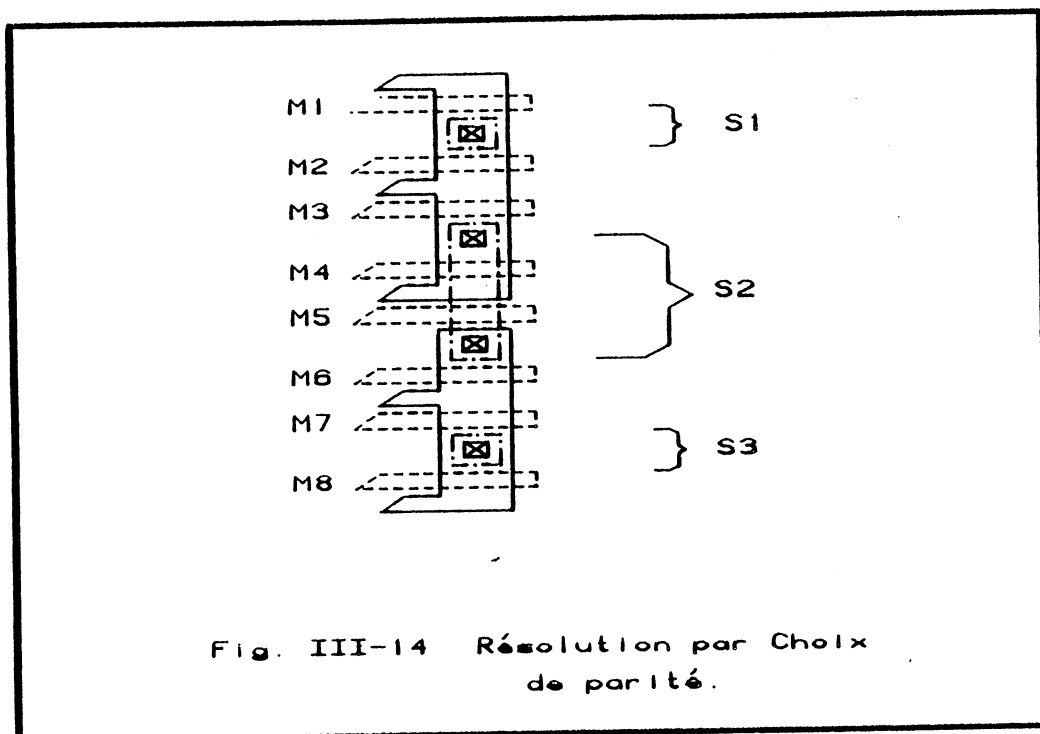
Fig. III-13 Conflit Topologique dans la Matrice OU.

Dans les PLAs NOR-NOR en technologie N-MOS classique, les points sont tels que chaque transistor partage avec un de ses voisins une ligne de masse et, avec le voisin opposé, un contact. Lorsque deux sorties dans la même colonne sont contigües elles vont partager soit une ligne de masse, soit un contact.

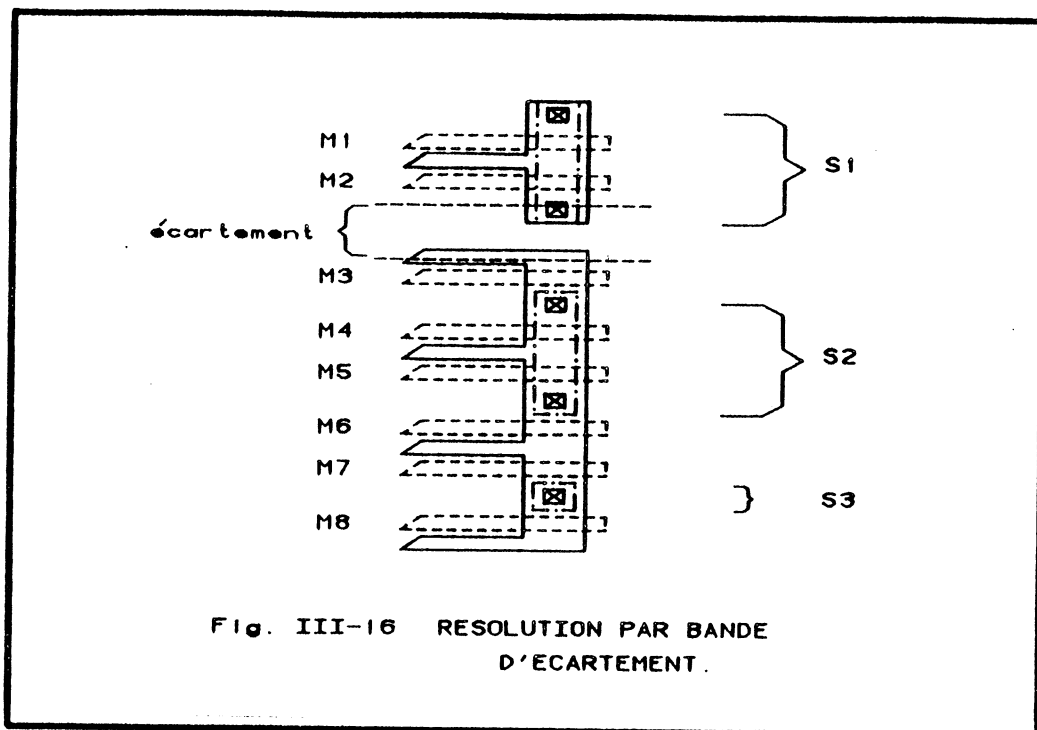
Ce dernier cas est appelé un **CONFLIT TOPOLOGIQUE** qui, s'il n'est pas résolu, entraînera un court-circuit entre les deux sorties.

La résolution de ces conflits topologiques peut être faite de différentes manières :

- dans le cas où tous les conflits apparaissent dans les lignes paires ou dans les lignes impaires, il est possible d'éviter les conflits par un choix de l'ordre des lignes de masse et des contacts.
- éviter que deux sorties dont les couvertures sont juxtaposées soient placées dans la même colonne. Ceci peut se réaliser par une extension d'un pas de la couverture de chaque sortie pour les rendre incompatibles lors du compactage. Cette extension peut se faire vers le haut ou vers le bas, les résultats étant identiques. Malheureusement ceci rend le compactage beaucoup moins performant.
- utiliser ce conflit pour introduire une bande d'écartement. Cette bande devra être répercutée dans la matrice ET et elle facilitera le tracé des connexions internes. Par contre, si



cette technique ne modifie en rien l'optimisation symbolique,
elle rend le PLA plus haut (long) que prévu.



Il est possible d'utiliser ces techniques individuellement ou en couplant la première avec l'une des deux autres. Il faut toutefois savoir que les PLAs ayant plus de lignes que de colonnes sont moins «sensibles» à un allongement (augmentation des lignes).

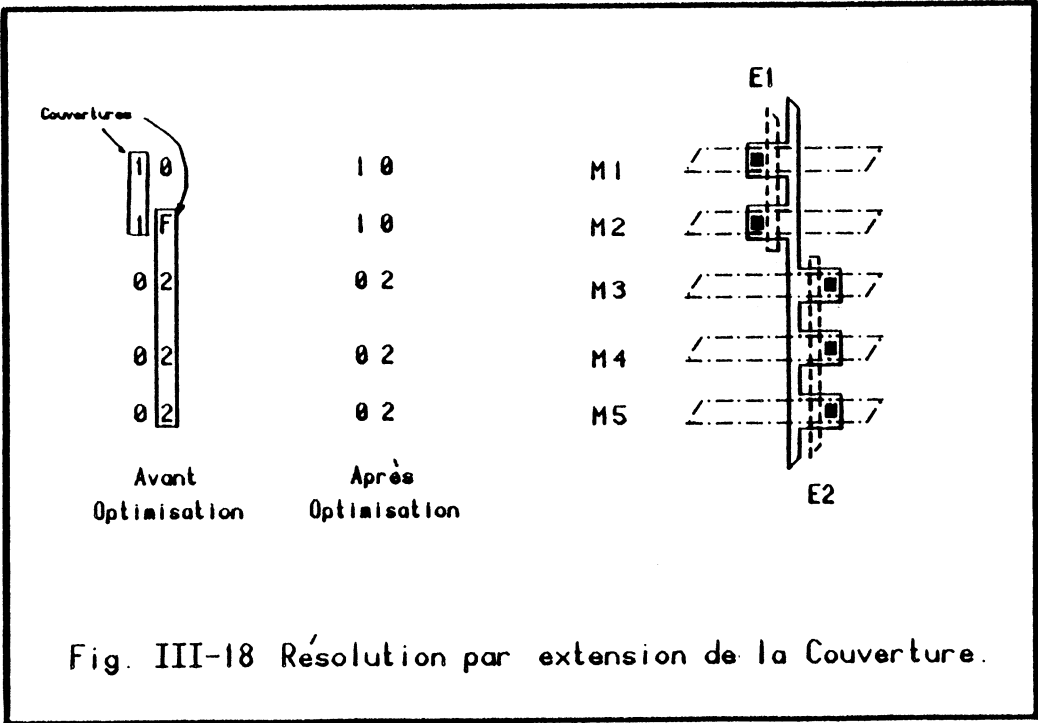
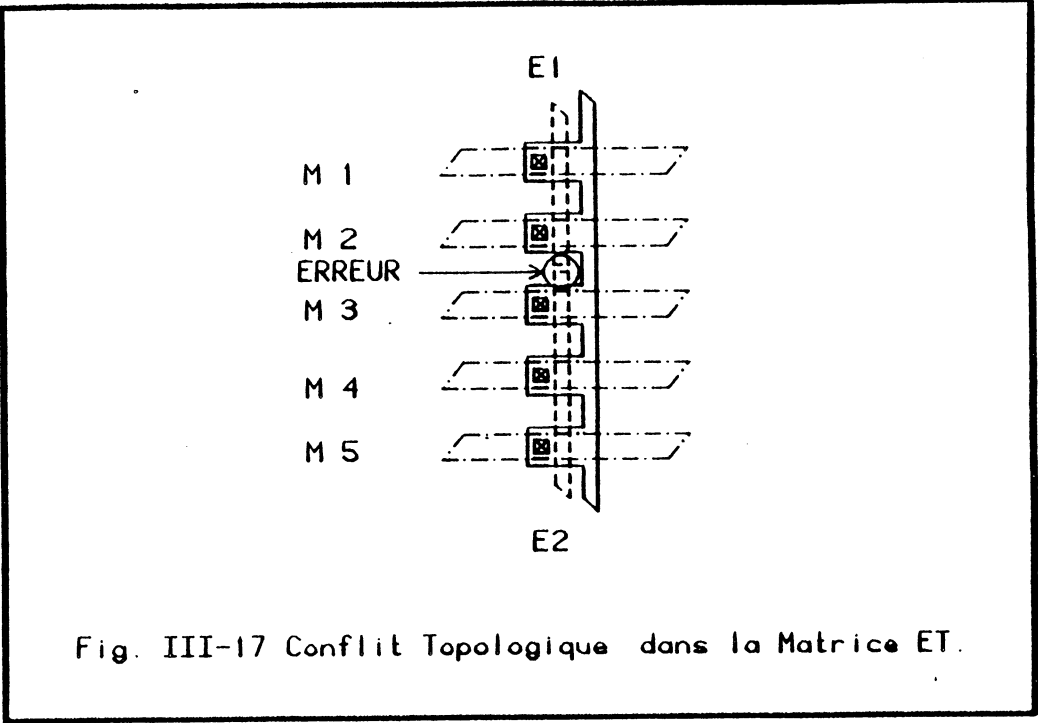
Pratiquement, les PLAs complexes ayant un très grand nombre de monômes, la deuxième technique ne sera jamais utilisée, et on préférera la dernière couplée avec la première, lorsque ceci est possible, pour faciliter le tracé interne des connexions (cf. sous-chapitre suivant).

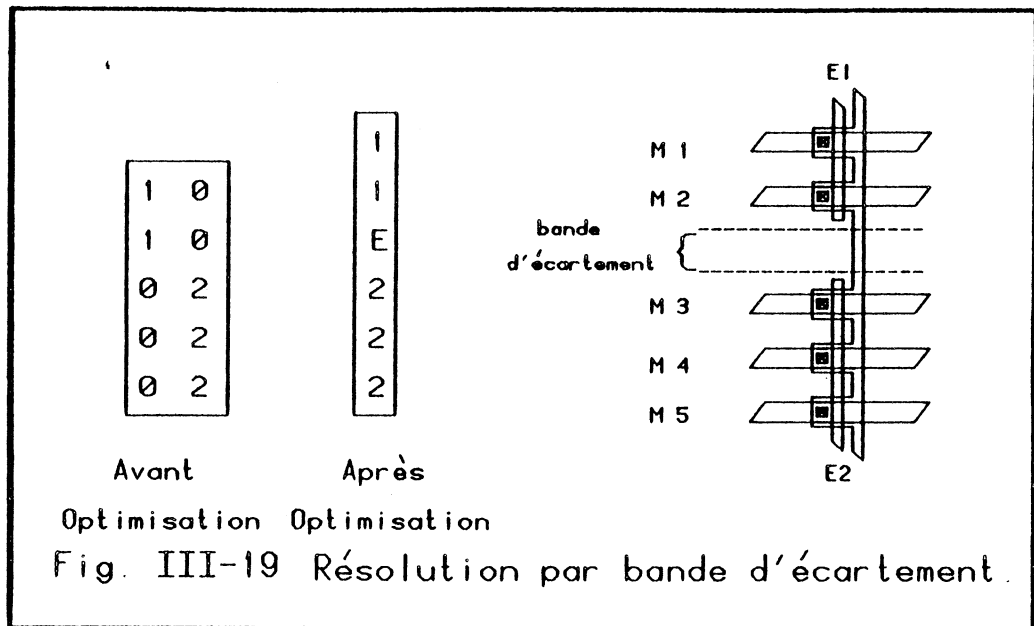
III.F.2 Conflits Topologiques Dans La Matrice ET.

Les conflits topologiques de la matrice ET arrivent lorsque deux entrées appartenant à la même colonne commandent deux transistors voisins.

Les solutions possibles sont :

- l'extension de la couverture avant le compactage.
- l'interposition d'une ligne de rappel de masse qui sera répercutée dans la matrice OU par une bande d'écartement.
- l'interposition d'une bande d'écartement dans les deux matrices.





Les répercussions sur la qualité du compactage sont les mêmes que pour les conflits de la matrice OU et on préférera l'interposition tant de rappels de masse comme de bandes d'écartement qui aident au tracé des connexions internes.

III.F.3 Connecteurs En Haut Et En Bas.

L'optimisation décrite jusqu'ici ne tient pas compte du côté sur lequel sont placés les connecteurs, et il est évident que si un connecteur est placé en haut ou en bas de la matrice, le segment correspondant doit s'étendre jusqu'au bord de celle-ci.

Pour tenir compte de ceci au niveau du compactage, les couvertures des segments en question doivent être étendues jusqu'au bord de la matrice.

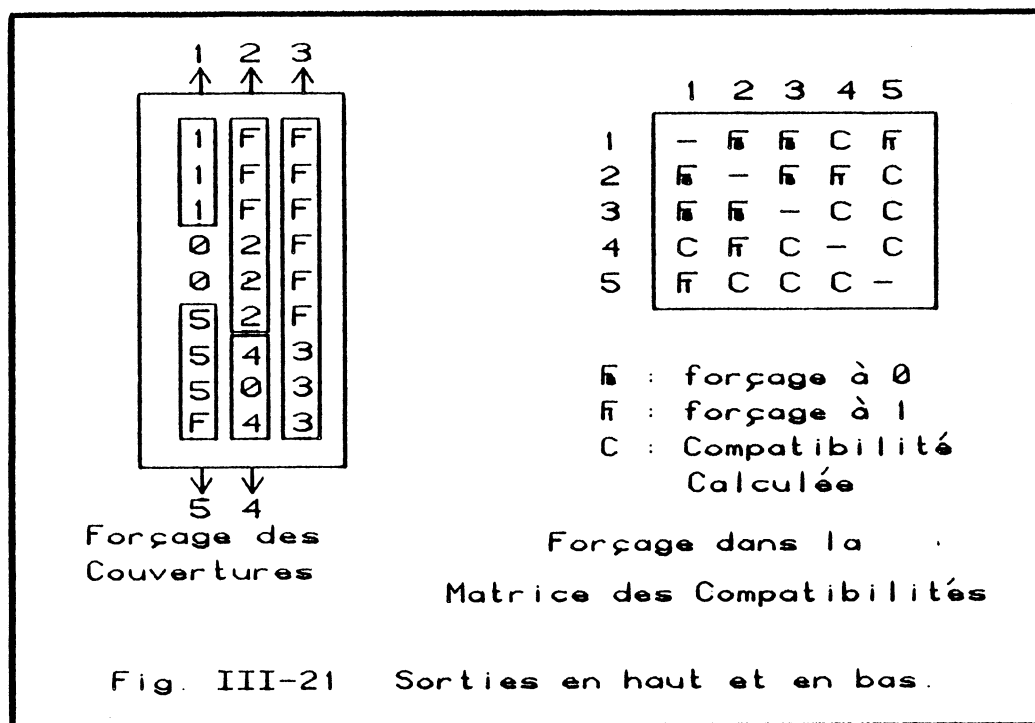
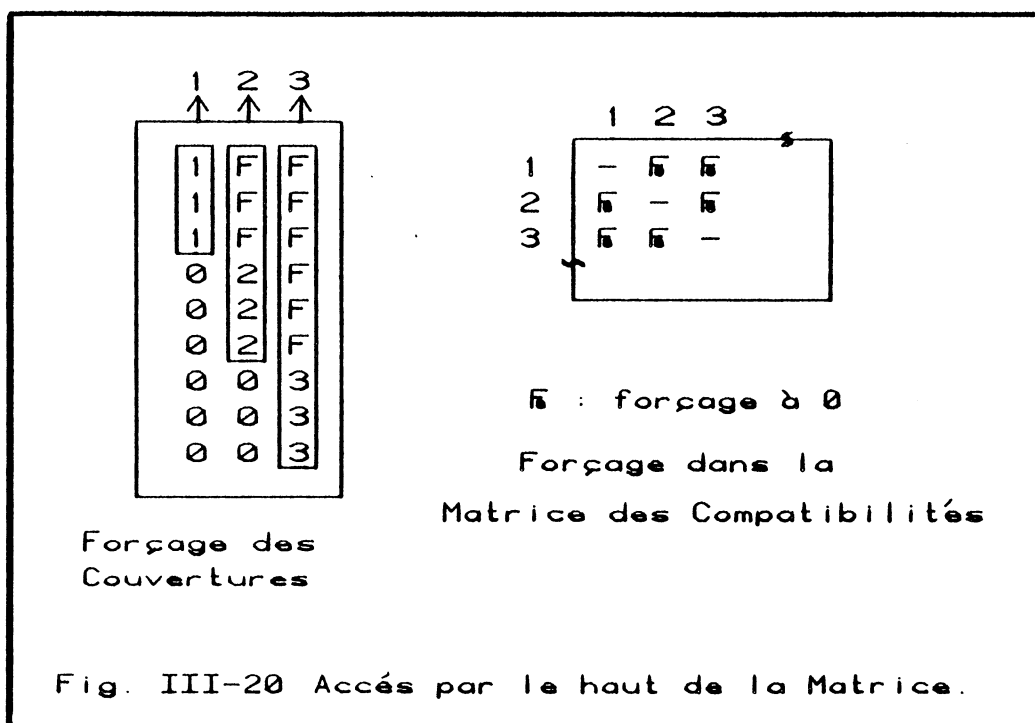
Lorsque la position exacte des connecteurs supérieurs et inférieurs est fixée, on doit éviter le placement dans la même colonne de segments définis dans des colonnes différentes. Ceci est réalisé en forçant l'incompatibilité entre ces sorties avant le compactage.

De même lorsque deux sorties ont été définies comme étant dans la même colonne (une avec son connecteur en haut et l'autre en bas), il faut les rendre indissociables pendant la phase de regroupement.

Après le regroupement des segments il faudra désigner une colonne particulière aux groupes comportant des sorties dont l'accès se trouve aux extrémités des matrices.

III.F.4 Problèmes Liés Au Tracé Des Connexions Internes.

Une fois les lignes de diffusion, silicium polycristalin et d'aluminium ainsi que les transistors des matrices étant déterminées,



le tracé interne des connexions doit être réalisé. Ce tracé doit relier électriquement chaque segment avec son connecteur placé sur un des bords de la matrice. Il faudra donc trouver un chemin à travers les emplacements vides à l'intérieur des matrices du PLA, en contournant les transistors et en évitant de trop se rapprocher des conducteurs déjà placés (respect des règles de dessin). Cette

recherche est souvent difficile et quelquefois impossible [CHU-84]. Or si dans l'optimisation symbolique il est pratiquement impossible de prévoir tous ces chemins, un traitement supplémentaire peut, par contre, aider beaucoup au tracé.

Ce traitement tire profit de la liberté de choix de la colonne où chaque ensemble de segments sera placé. Les ensembles ayant des segments associés aux connecteurs supérieurs et inférieurs ont leur colonne désignée. Pour les autres, un critère devra être déterminé pour que leur assignation aux colonnes rende plus aisé le tracé interne. Quelques critères ont été proposés :

- nombre de transistors par colonne (NTC).
- nombre de segments par colonne (NSC).
- somme des couvertures par colonne (SCV).
- facteur d'utilisation de la colonne (NM/NTC).
- nombre moyen de transistors par segment (NTC/NSC).

Les colonnes les plus transparentes sont ainsi placées vers le côté externe de la matrice, vers lequel convergent toutes les connexions internes.

Ces différents critères ont été testés sans qu'aucun d'eux se relève significativement plus performant que les autres.

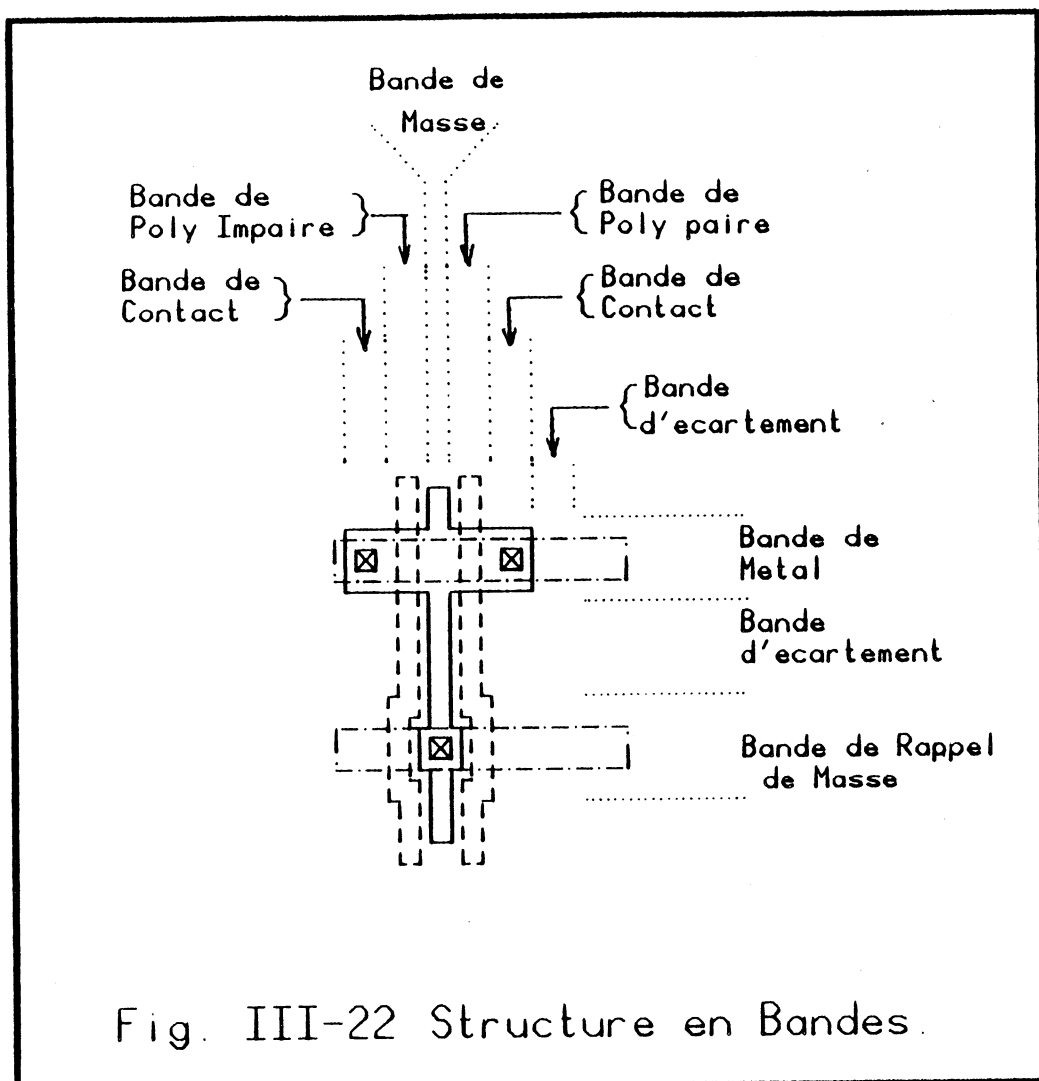
Une autre technique a été proposée, elle consiste à étendre la couverture des segments avant le compactage jusqu'à la position de son connecteur. Mais les performances du compactage ont été très dégradées.

III.G GENERATION DU DESSIN DES MASQUES ET TRACE INTERNE DES CONNEXIONS

Cette phase du traitement des PLAs correspond au travail réalisé par M. Samuel CHUQUILLANQUI [CHU-84].

La génération des dessins des masques peut se faire directement à partir de la description symbolique utilisée lors de l'optimisation topologique. Mais pour réaliser le tracé interne des connexions il est préférable de passer par un niveau supplémentaire appelé «structure en bandes».

Dans ce niveau de description sont visualisés non seulement les transistors avec leur emplacement dans la matrice, mais aussi les contacts, les lignes de masse, les lignes rappels de masse, ainsi que les connexions dans les matériaux utilisés (diffusion, Si-poly, aluminium, etc..). Par contre les caractéristiques dimensionnelles ne



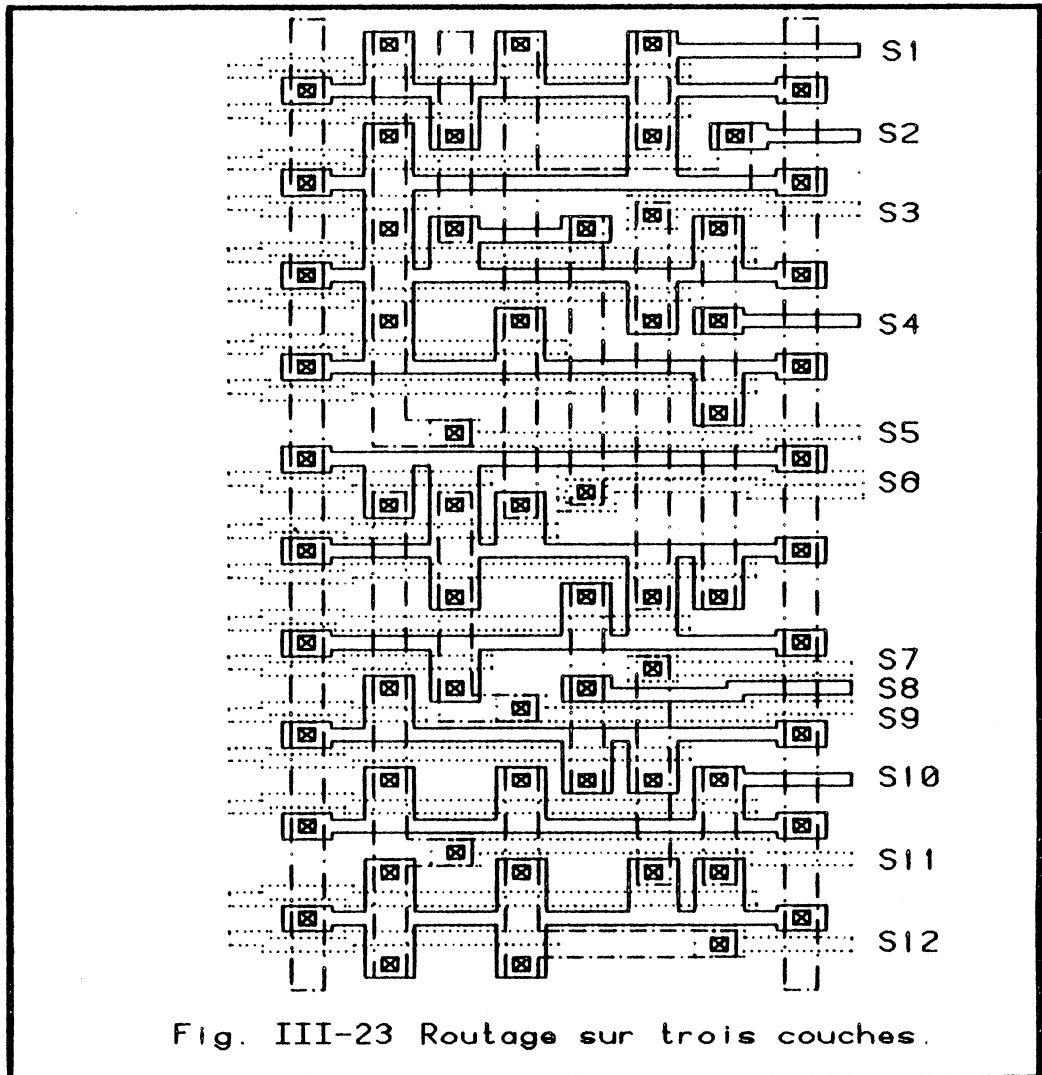
sont représentées que par des voies indirectes.

Le traitement s'effectue en quatre phases :

- transformation du symbolique en structure en bandes.
- prise en compte de la transparence désirée du PLA.
- tracé des connexions internes.
- génération des dessins dans un des multiples langages possibles.

Ce découpage permet une grande indépendance vis à vis des langages de description de masques.

La transparence du PLA est assurée, tant verticalement qu'horizontalement par étirement du PLA aux emplacements désignés. ceci est réalisé par les programmes de dessin des masques et de tracé des connexions internes [CHU-84].



Le tracé des connexions internes peut être réalisé soit sur les trois couches des technologies N-MOS classiques, soit sur une quatrième couche (deuxième métal ou siliciure) dans le cas où une technologie plus avancée est utilisée.

Dans le premier cas, les connexions doivent éviter les obstacles formés par la structure physique des matrices. Lors d'une impossibilité de tracé on doit recourir à l'insertion d'une bande d'écartement pour créer un passage.

L'utilisation de la quatrième couche n'accueillant que les connexions internes simplifie grandement le tracé de celles-ci car les seuls obstacles à éviter sont les connexions elles mêmes.

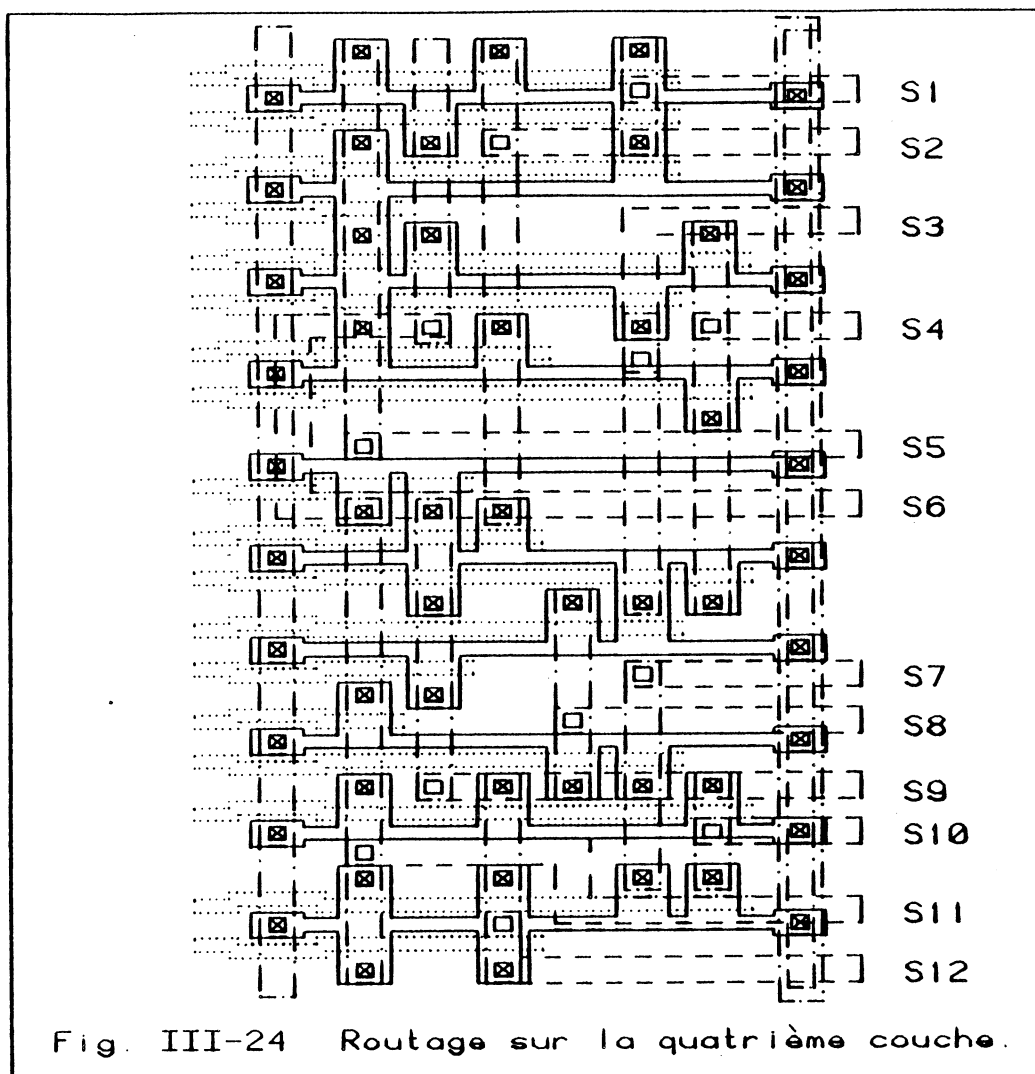


Fig. III-24 Routage sur la quatrième couche.

III.H RESULTATS - CONCLUSION

Plusieurs PLAs provenant de microprocesseurs huit bits existants ont été utilisés comme exemples durant ce travail. Dès les premiers résultats [PER-80], les performances ont été surprenantes : réduction d'au moins 50% de la surface des matrices OU. Les matrices ET étant de leur côté plus difficilement optimisées.

L'introduction de la duplication des monômes a amélioré substantiellement l'optimisation (fig. III-25).

Sur un ordinateur HB-68 avec le système MULTICS, le temps d'unité centrale nécessaire à l'optimisation de PLAs de taille moyenne (20 entrées, 80 monômes et 50 sorties) ne dépasse pas la minute. Pour des PLAs plus grands (22 entrées, 150 sorties et 300 monômes) quatre minutes de CPU sont nécessaires. Ces temps sont très acceptables si on tient compte de la taille des matrices optimisées.

	PLA 1	PLA 2	PLA 3	PLA 4	
CLASSIQUE	2990 (100%)	4320 (100%)	3375 (100%)	7467 (100%)	Surface en Nb. de Points PLA.
OPTIMISATION SANS DUPLICATIONS	1820 (60,8%)	1512 (35%)	1950 (57%)	4454 (59%)	
OPTIMISATION AVEC DUPLICATIONS	1794 (60%)	393 (9%)	1584 (46,9%)	3840 (51,4%)	

Fig. III-25 Quelques Résultats D'optimisation de Matrices OU.

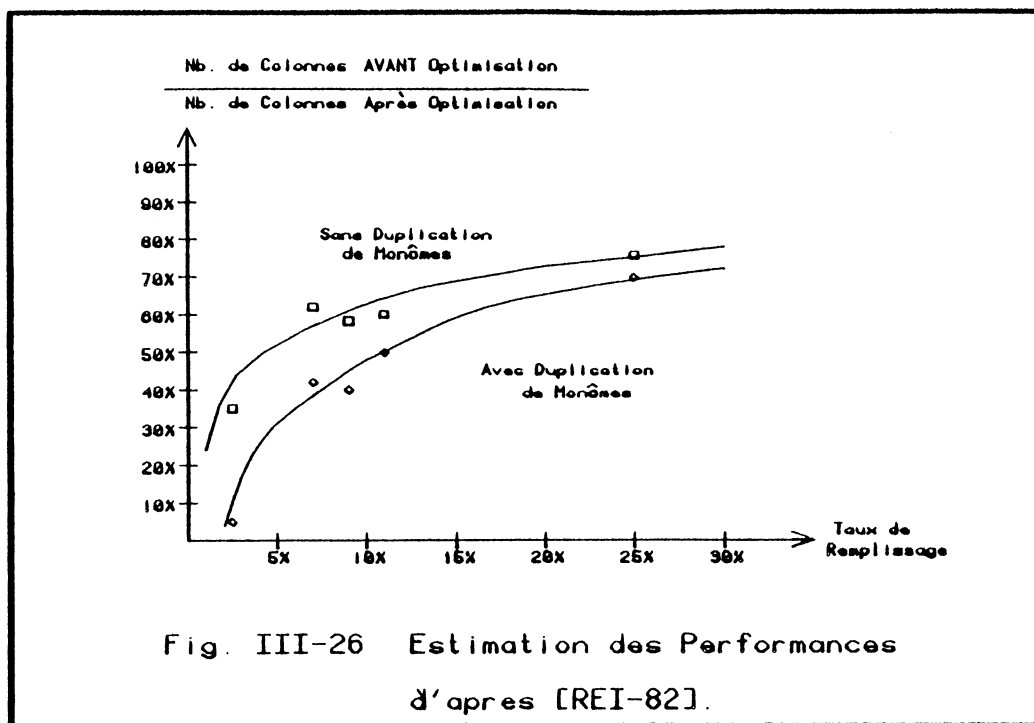
Il est important de constater qu'il existe une forte corrélation entre le taux de remplissage de la matrice OU et le gain en surface obtenu par cette méthode d'optimisation. Statistiquement une relation des performances en fonction de ce taux a été établie par M. Reis [REI-82]. La figure III-26 montre cette relation dans les deux cas suivants :

1. pas de duplication de monômes
2. duplication de 20% des monômes.

Nous avons pu distinguer deux types de PLA qui donnent des résultats différents lors de l'optimisation.

Le premier type correspond aux PLAs de décodage qui contiennent un faible nombre d'entrées, un grand nombre de monômes et un nombre important de sorties. Dans ces PLAs chaque sortie est générée par un petit nombre de monômes (de 1 à 10). Les performances de la méthode des lignes brisées sont alors très bonnes, atteignant même un gain en surface de plus de 90%. Ce type de PLA correspond à la génération de microcommandes à partir de l'état d'un automate, et est utilisé pratiquement dans tous les microprocesseurs.

Le deuxième type de PLA correspond aux PLAs d'encodage, il contient un faible nombre d'entrées et de sorties, le nombre de monômes étant sans importance. Dans ce cas un nombre important de monômes est utilisé dans la génération de chaque sortie, et l'utilisation de notre méthode d'optimisation ne réduit pratiquement pas la surface. Ce type de PLA est utilisé parfois dans



l'implémentation d'automates, le PLA servant comme organe de calcul de l'état suivant. Toutefois, lorsque ces PLAs nécessitent des conditions externes pour le choix de l'état suivant, l'optimisation par lignes brisées permet de réduire la surface de la matrice ET, les entrées de condition n'étant utilisées que très localement dans la génération des monômes.

Finalement on peut dire qu'une méthode d'optimisation topologique telle que la méthode des lignes brisées, en cassant la régularité des PLAs, permet une utilisation plus universelle de ceux-ci dans la conception des circuits à très haute intégration. Les inconvénients dûs précisément à cette régularité, qui rend la génération des PLAs si facile, étant fortement réduits.

CHAPITRE IV
PRESENTATION
DU SYSTEME PAOLA

IV.A DESCRIPTION GENERALE

Dans le cadre du projet CAPRI (Conception Architecturale des PRocesseurs Intégrés) qui se déroule dans l'équipe de conception des ordinateurs de l'IMAG, un certain nombre d'outils d'aide à la conception et d'automatisation de certains processus de cette conception, ont été réalisés ou sont en voies de réalisation. La methodology utilisée est la conception descendante avec l'automatisation des niveaux inférieurs.

Le traitement des PLAs, à partir d'une description structurelle et jusqu'au dessin des masques est réalisé par le système PAOLA. Ce traitement comprend :

1. la minimisation logique éventuellement adaptée à l'étape suivante.
2. l'optimisation topologique par la méthode des lignes brisées.
3. une évaluation du comportement électrique.
4. le dimensionnement des éléments du PLA (transistors des matrices, transistors de charge et amplificateurs d'entrée et de sortie).
5. le dessin des masques dans un langage de description graphique avec tracé des connexions internes.
6. la génération des informations nécessaires à un assembleur de blocs (LUBRICK).

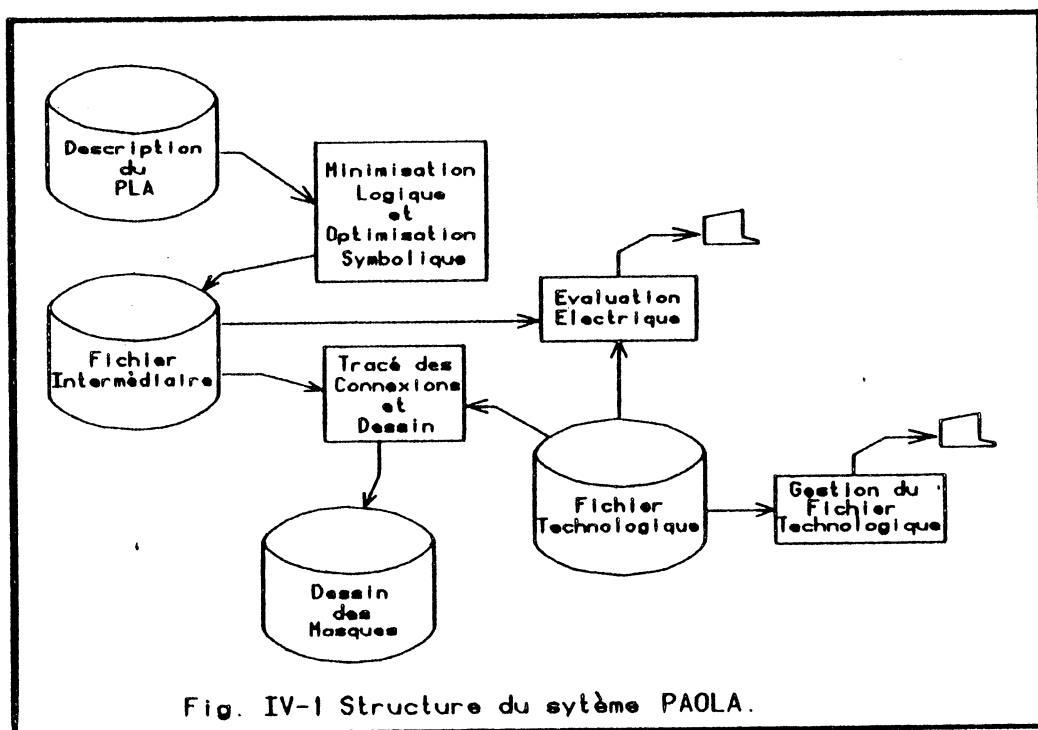
avec les caractéristiques suivantes :

- possibilité de réaliser indépendamment chacun de ces traitements.
- indépendance des règles de dessin.
- indépendance vis à vis des langages de description graphique.
- indépendance des caractéristiques de la technologie utilisée.
- facilité d'extension par ajout de nouveaux modules.

Dans un premier temps les PLAs étaient générés dans une technologie N-MOS à trois ou quatre couches et, dernièrement, la possibilité de le faire dans des technologies bipolaires a été ajoutée [CHU-84/a].

L'ensemble du système a été écrit dans le langage PASCAL pour assurer une grande portabilité.

Le système PAOLA est actuellement composé de quatre modules (fig.IV-1) :



1. module d'optimisation topologique.
2. module de tracé des connexions internes et de dessin.
3. module d'évaluation électrique.
4. module de gestion du fichier technologique.

La communication entre les modules se réalise par le biais de fichiers textuels, ayant la même syntaxe, ce qui permet à l'utilisateur de vérifier, de créer ou éventuellement de modifier ces informations.

IV.B MODULE D'OPTIMISATION TOPOLOGIQUE

Il est composé de six programmes (fig. IV-2) effectuant les traitements suivants :

1. ANALSYN, analyse syntaxique des fichiers contenant la description du PLA à traiter et génération d'un fichier de commandes communes pour les programmes d'optimisation.

2. POSET, réordonnancement des monômes d'après la matrice ET, avec répercussion sur la matrice OU.
3. POSOU, réordonnancement des monômes suivant la matrice OU et duplication des monômes avec repercussion dans la matrice ET.
4. COMPET, compactage de la matrice ET.
5. COMPOU, compactage de la matrice OU.
6. INTERFACE, préparation des matrices pour les phases suivantes.

Les heuristiques choisies sont les suivantes :

- Les critères de choix du monôme à dupliquer sont appliqués suivant l'ordre de performances. Lorsqu'un critère n'est pas capable d'isoler un seul monôme, le critère suivant est appliqué.
- Le découpage des monômes se fait au point désigné par le barycentre.
- La résolution des conflits topologiques peut être faite soit par détermination du pas des bandes, soit par extension des couvertures, soit par écartement au niveau du dessin, ceci en fonction d'une option de compilation.

Le concepteur choisit quelle est la matrice prioritaire à optimiser, et ce choix détermine l'ordre dans lequel les deux programmes POSET et POSOU seront exécutés.

Le nombre de duplications à réaliser est donné interactivement.

Le résultat de l'optimisation symbolique est donné dans deux fichiers textuels, lisibles par l'utilisateur.

IV.C MODULE D'EVALUATION ELECTRIQUE

Ce module réalise l'évaluation du temps de réponse du PLA optimisé en fonction des caractéristiques de la technologie utilisée ainsi que du point PLA. Il permet aussi le dimensionnement tant des transistors de charge des matrices, comme des amplificateurs d'entrée du PLA [DAN-83].

IV.D MODULE DE TRACE DES CONNEXIONS ET DE DESSIN DES MASQUES

Il comporte sept programmes pour réaliser le tracé des connexions

MODULE DE TRACE DES CONNEXIONS ET DE DESSIN DES MASQUES

internes suivant un des deux schémas déjà décrits, et le dessin des masques dans un langage graphique spécifique.

- TESTCONF détecte les conflits tant dans la matrice ET comme dans la matrice OU. Il génère un fichier décrivant ces conflits.
- ANDPLACE et ORPLACE traduisent le symbolique décrivant le PLA en structures en bandes, insèrent les rappels de masse, les bandes de transparence et résolvent les conflits à l'aide de bandes d'écartement.
- IROUTING et OROUTING recherchent les chemins du tracé interne en trois couches (diffusion, silicium polycristallin et aluminium). Cette recherche tient compte des règles de dessin codées dans le fichier technologique.
- ROUTING3 fait de même mais dans une quatrième couche et seulement pour la matrice OU.
- LAYOUT transforme la structure en bandes résultante en une description graphique dans un langage donné.

Il existe une version du programme LAYOUT par langage graphique. De toutes facons PAOLA n'utilise que des rectangles, primitive existante dans tous les langages graphiques, dans la description des masques.

IV.E MODULE DE GESTION DU FICHIER TECHNOLOGIQUE

Pour réaliser le tracé interne, le dessin des masques et l'évaluation électrique, les programmes ont besoin d'un certain nombre d'informations dérivées tant de la technologie que du point PLA utilisés. Pour permettre une grande souplesse dans la définition de certains paramètres, tout en gardant la cohérence de toutes les informations, un module de gestion du fichier technologique a été écrit.

IV.F EXTENSIONS POUR D'AUTRES TECHNOLOGIES

Les bonnes performances de PAOLA dans l'optimisation des PLAs ont entraîné des demandes de modification pour des technologies autres que le N-MOS. Ces technologies sont :

- une technologie de type Arséniure de Gallium
- deux technologies bipolaires (STL et ECL).

Ces extensions ont entraîné des modifications dans la définition et l'interprétation de la structure en bandes.

IV.G CONCLUSION

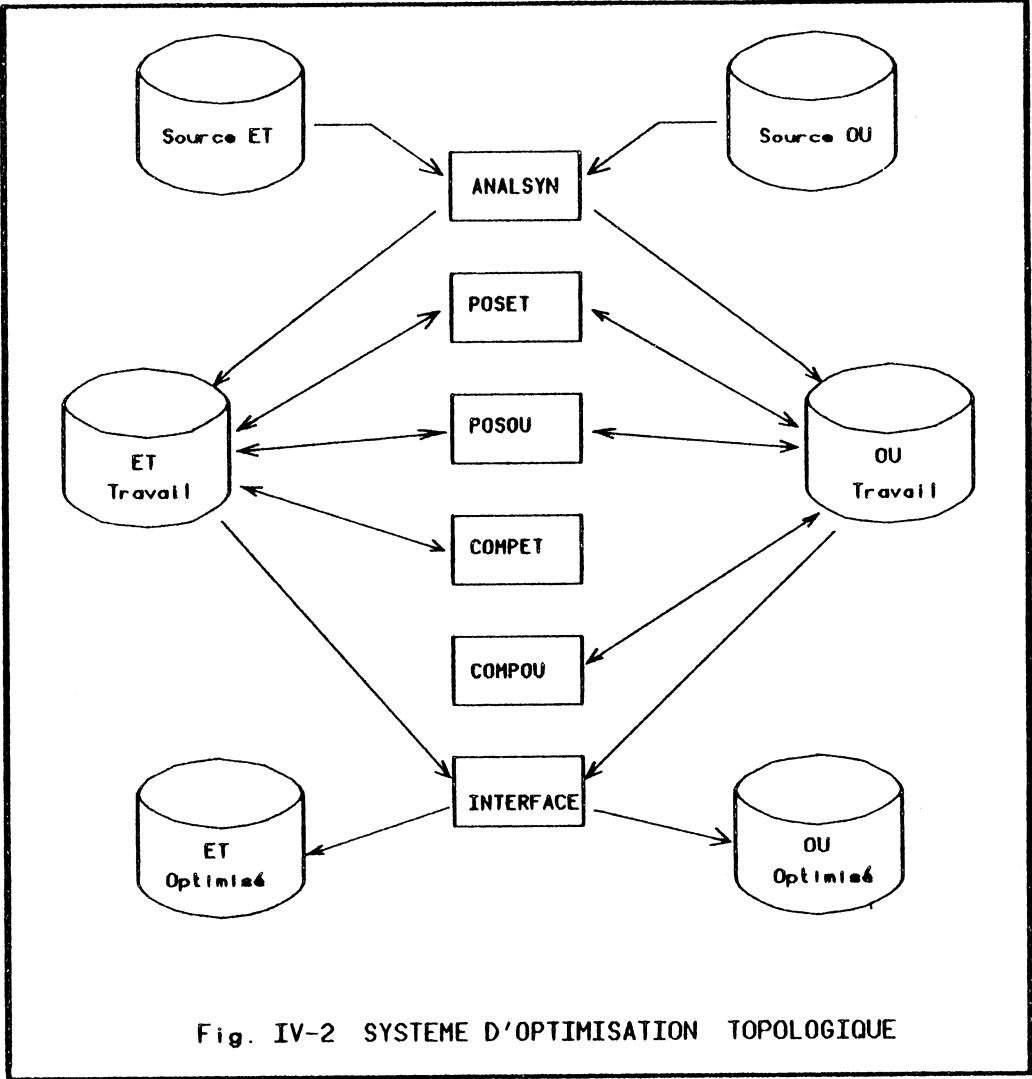
Le système PAOLA est capable de traiter les PLAs des nouvelles générations de circuits intégrés, ayant de bonnes performances dans l'optimisation topologique et, surtout, un temps d'exécution très réduit. Il a été expérimenté par plusieurs entreprises, notamment chez :

- Digital Equipment Corporation, Hudson, Ma. USA
- Silvar Lisco, Louvain, Belgique
- Thomson-CSF dans les services DCI (Saint Egrève) et DIS (Paris)
- Matra Harris Semiconductors, Nantes
- Bull Système, Les Clayes
- EFCIS, Grenoble

ainsi que dans plusieurs universités :

- Universidad Autonoma de Barcelona, Espagne
- Cork College, Irlande
- Instituto Tecnico Superior, Lisbonne, Portugal
- Institut National Polytechnique de Grenoble

L'ensemble du système étant écrit en langage PASCAL, PAOLA est facilement transportable et, actuellement, il existe des versions pour différents ordinateurs tels que HB-68 sous Multics, VAX sous VMS, HP-1000 sous RTE/A et METEUS lambda 7500.



BIBLIOGRAPHIE

REFERENCES BIBLIOGRAPHIQUES

- [AGA-79] Multiple fault Detection in Programmable logic Arrays.
V. Agarwal.
9th Symp. on Fault Tolerant computing, Madison, Wisconsin,
June 1979, pp. 227-223.
- [ALL-81] VLSI Design Automation Activities at M.I.T.
J. Allen and P. Penfield.
IEEE. Tr. Circ. Syst., Vol. CAS-28, No. 7,
July 1981, pp. 645-653.
- [ANC-80] Architecture and Design of Von Newmann Microprocessors.
F. Anceau
Nato Advanced Summer Institute on Design Methodologies
for VLSI Circuits.
July 8-18, 1980, Louvain, Belgique.
- [ANC-82/a] Complex Integrated Circuit Design Strategy.
F. Anceau and R. Reis.
IEEE J. of Solid-State Circuits, Vol. SC-17, No. 3,
June 1982, pp. 459-464.
- [ANC-82/b] CAPRI: A Silicon compiler.
F. Anceau and J. Schoellkopf.
Workshop on Silicon Compilation, Edimburgh, July 1982.
- [ARE-78] A Method to Simplify a Boolean Function into a Near Minimal
Som-of-Products for Programmable Logic Arrays.
Z. Arevalo and J. Bredeson.
IEEE. Trans. on Computers, Vol. C-27, No. 11,
Nov. 1978, pp. 1028-1039.
- [AUG-78] An Alogorithm for Designing Multiple Boolean Functions:
Application to PLAs.
M. Augin, F. Boeri and C. André.
Digital Processes, 4, 1978, pp. 215-230.
- [AUG-79] A New Algorithm for Designing Multiple Boolean Functions:
Application to PLAs.
M. Augin, F. Boeri and C. André.
Digital Processes 4, 1979.
- [AUG-80] Systematic Method of Realization of Interpreted Petri Nets.
M. Augin, F. Boeri and C. André.
DIGITAL PROCESSES, 6, 1980, pp. 55-68.
- [AYA-77] Synthesis of Logic Systems with PLAs,
Logique Cablée ou Logique Programée?
J. Ayache and P. Le Danois.
EPFL, Lausanne, Suisse, 1977.

- [AYR-79]** Silicon Compilation - A Hierarchical use of PLAs.
R. Ayres.
16 DAC, June 1979, pp. 314-326.
- [BEC-84]** A 32b Microprocessor With On-Chip Virtual Memory Management.
J. Beck et al.
ISSCC 84, Feb. 1984, pp. 178-179.
- [BON-81]** Les PLAs NMOS Statiques.
M. Bonnet et J. Tant.
Rapport de projet d'Ingénieur, INPG-ENSERG, Juni 1981.
- [BOS-82]** Test Generation for Programmable Logic Arrays.
P. Bose and J. Abraham.
19 DAC, Las Vegas, Nevada, June 1982, pp. 574-580.
- [BOW-70]** A Method for FAST Approximate Solution of Large Prime Implicant Charts.
R. Bowman and E. McVey.
IEEE Trans. on Comp. February 1970, pp. 169-173.
- [BRA-84]** ESPRESSO-II: A New Logic Minimizer for Programmable Logic Arrays.
R. Brayton et al.
IEEE Custom Integrated Circuits Conference, Rochester, New York, May 1984, pp 370-376.
- [BRI-78]** Multiple output PLA Minimization: EMIN.
p. Bricaud and J. Campbell.
Proc. 1978 WESCON professional Program, Los Angeles, Ca., Sept. 1978.
- [CAN-83]** VLSI in Focus: Designing a 32-bit CPU Chip.
M. Canepa, E. Weber and H. Talley.
VLSI Design, Jan. 1983, pp. 20-24.
- [CAR-72]** Programmable Logic Arrays.
W. Carr and J. Mize.
MOS/LSI Design and Application, pp. 229-258,
Texas Instruments electronic Series,
McGraw-Hill Book Company, New York, N.Y., 1972.
- [CAV-79/a]** Field-Programmable Arrays Part 1: Powerful Alternatives to Random Logic.
N. Cavlan and S. Durham.
Electronics, july 15, 1979, pp.109-114.
- [CAV-79/b]** Field Programmable Arrays, part 2: Sequencers and Arrays Transforms Truth Tables into working Systems.
N. Cavlan and S. Durham.
Electronics, July 19, 1979, pp. 132-139.

- [CHA-75] Test Pattern Generation for Shorts in PLA.
C. Cha.
IBM Technical Disclosure bulletin, vol. 18, No. 5, Oct. 1975.
- [CHA-76] Testing of PLA Cross-Point Defects.
C. Cha.
IBM Technical Disclosure Bulletin, Vol. 18,
No. 8, Jan. 1976.
- [CHA-78] A Testing Strategy for PLAs.
C. Cha.
15th DAC, June 1978, Las Vegas, Nevada, pp. 326-334.
- [CHU-82/a] PAOLA: A Tool For Topological Optimization of Large PLAs.
S. Chuquillanqui and T. Pérez Segovia.
19 DAC, Las Vegas, Nevada, June 1982, pp. 300-306.
- [CHU-82/b] PAOLA, un Systeme CAO Pour L'Optimisation Topologique
et le Dessin Automatique des Masques de PLA Complexes.
S. Chuquillanqui et T. Pérez Segovia.
AFCET, Lille, 17 Novembre 1982.
- [CHU-83/a] Internal Connection Problem in Large Optimized PLAs.
S. Chuquillanqui.
20 DAC, Miami, Florida, June 27-29, 1983.
- [CHU-83/b] A VLSI Topological Optimization Strategy Applied
to PLA Design.
S. Chuquillanqui, T. Pérez Segovia, R. Krasicki.
IEEE. Int. Conf. on CAD, Santa Clara, CA,
Sept. 1983, pp. 184-189.
- [CHU-84/a] Adaptation du Système PAOLA à la technologie
Bipolaire HBIP3A-1.
S. Chuquillanqui.
Thompson CSF/DCI, Rapport No. SC/MC/564-84, Saint Egrève,
Août 1984.
- [CHU-84/b] Une Nouvelle Approche Pour l'Optimisation et
l'Automatisation du Dessin des Masques de PLA Complexes.
S. Chuquillanqui.
Thèse de Docteur Ingénieur INPG, Grenoble, 15 octobre 1984.
- [COO-67] Automatic Artwork Generation for Large Scale Integration.
p. Cook, W. Donath, G. Lemke and A. Brennemann.
IEEE Journal of Solid-State Circuits, dec 1967, p. 190.
- [COO-79/a] One Micron Mosfet PLAs.
*P. Cook, S. Schuster, D. Freedman, J. Parrish
and V. Dilonardo.*
ISSCC 79, Feb. 1979, pp. 62-63.

- [COO-79/b]** A Study in the Use of PLA-Based Macros.
P. Cook C. Ho and S. Schuster.
IEEE J. of Solid-State Circuits, VOL SC-14, No. 5,
October 1979, pp. 833-840.
- [COO-81]** Array Logic.
P. Cook
Digital Technology Status And Trends,
edited by H. Painke. R. Odenburg, VERLAG MUNCHEN WIEN 1981,
pp. 117-144.
- [DAE-81]** A Hardware Approach to Self-Testing of Large PLA's.
W. Daehn and J. Mucha.
IEEE. Trans. on Computers, November 1981, pp. 829-833.
- [DAN-83]** Evaluations Electriques et Temporelles des PLA Complexes.
A. Dandache.
Thèse de 3ème Cycle, INPG, 21 Nov. 1983.
- [DE-83]** PLASCO : A Silicon Compiler for NMOS and CMOS PLA's.
*H. De Man, L. Reynders, M. Bartholomeus
and J. Cornelissen.*
VLSI'83, Trondheim, Norway, Aug. 1983, pp. 171-181.
- [DEM-83/a]** SMILE: a Computer Program for Partitionning of
Programmed Logic Arrays.
G. DeMicheli and M. Santomauro.
Computer-Aided-Design, Vol. 15, No. 2, March 1983, pp. 89-97.
- [DEM-83/b]** Multiple Folding of PLAs.
G. De Micheli and A. Sangiovanni Vicentelli.
Proc. Int. Symp. on Circ. and Sys., Newport Beach,
Ca. , May 1983.
- [DEM-83/c]** PLEASURE: A Computer Program for Simple/Multiple
Constrained/Unconstrained Folding of Programmable
Logic Arrays.
G. DeMichelli and A. Sangiovanni Vicentelli.
20 DAC, Miami, Florida, June 1983, pp. 530-537.
- [DEM-83/d]** Topological Partitionning of Programmable Logic Arrays
G. De Micheli and M. Santomauro.
IEEE. Int. Conf. on Computer Aided Design
Sep. 1983, pp. 182-183.
- [DEM-84]** PLEASURE: a Computeuter Program for Simple/Multiple
Constrained/Unconstrained Folding of Programmable
logic Arrays.
G. DeMicheli and A. Sangiovanni Vicentelli.
Computer Aided Design, Vol. 16, No. 1, January 1984, pp. 2-12.

- [DIE-80] Automated PLA Synthesis of the Combinational Logic of a DDL Description.
D. Dietmeyer and M. Doshi.
Design Automation and Fault Tolerant Computing,
Vol. 3, Issue 3/4, pp. 241-257.
- [DIR-81] A Design Methodology and Computer Aids for Digital VLSI Systems.
S. Director, A. Parker, D. Siewiorek and D. Thomas.
IEEE. Tr. on Cir. Syst., Vol. CAS-28, No. 7,
July 1981, pp. 634-645.
- [DON-82] PHILO - A VLSI Design System.
R. Donze, M. Jenkins, G. Sprorzynski.
19 DAC, Las Vegas, Nevada, June 1982, pp. 163-169.
- [DUT-81] Standford Overview in VLSI Research.
R. Dutton.
IEEE. Tr. Circ. Syst., Vol. CAS-28, No. 7,
July 1981, pp. 654-665.
- [EIC-80] A Heuristic test-Pattern Generator for Programmable Logic Arrays.
E. Eichelberger and E. Lindbloom.
IBM J. Res. Develop., Vol. 24, No. 1, Jan. 1980, pp. 15-22.
- [EGA-82] Optimal Bipartite Folding of PLA.
J. Egan and C. Liu.
19 DAC, Las Vegas, Nevada, June 1982, pp. 141-146.
- [EGA-84] Bipartite Folding and Partitionning of a PLA.
J. Egan and C. Liu.
IEEE trans. on Computer-Aided-Design, Vol. CAD-3, No. 3,
July 1984, pp. 191-199.
- [ELE-80] New PALS Can do Arithmetic, Too
Electronics, July 31, 1980 P 128-129.
- [ELZ-82] ALDS Research Report.
Y. El-Ziq and P. Rao
Honeywell ALDS Research Report #1
- [FLE-75] An Introduction to Array Logic.
H. Fleisher and L. Maissel.
IBM J. Res. Develop., Vol. 19, No. 2, March 1975, pp.98-109.
- [FLA-84] A Complete and Automatic System for Sequencer Design.
E. Flammand.
ICCD 84, Oct. 1984, Port Chester, N.Y., pp. 324-330.
- [FLE-79] Introduction to Special Section on Programmable Logic Arrays
H. Fleisher
IEEE tr. on computers, Vol C-28, No 9, september 1979, p 593.

- [FOR-83]** The Automatic Generation of PLAs from Algorithmic State Machine Description.
J. Forrest and M. Edwards.
VLSI 83, Trondheim, Norvège, Août 1983, pp. 183-193.
- [FRA-80]** An Approach to Debugging Custom Integrated Circuits.
E. Frank and R. Sproull.
Carnegie Mellon University Annual Report 1979-1980.
- [FUJ-81]** A Design of Programmable Logic Arrays With Universals Tests.
H. Fujiwara and K. Kinoshita.
IEEE. trans. on Computers, November 1981, pp. 823-828.
- [FUN-82]** A High-Density PLA Macro and Its Layout Generator.
B. Fung, T. Macnee and G. Rugila.
IEEE. ISSCC Digest of Technical Papers, Feb. 1982, pp. 58-59.
- [GLA-80/a]** An Interactive PLA Generator as an Archetype for a New VLSI Design Methodology.
L. Glasser and P. Penfield.
VLSI MEMO No. 80-25 July 1980, M.I.T.
- [GLA-80/b]** An Interactive PLA Generator as an Archetype for a New VLSI Design Methodology.
L. Glasser and P. Penfield.
ICCC 80, Port Chester, N.Y., Sept. 1980.
- [GLA-80/c]** Implementation of Regular Logic With Feedback Reduced PLAs.
L. Glasser.
Integrated Circuit Memo No. 80-8, MIT, Cambridge, MASS., Feb 1980.
- [GOL-80]** Design Automation and the Programmable Logic Array Macro.
R. Golden, P. Latus and P. Lowy.
IBM J. Res. Develop., Vol. 24, Jan. No. 1, 1980, pp. 23-31.
- [GOR-73]** The Programmable Logic Array: A New Approach to Microprogramming.
K. Gorman.
EDN, Nov. 20 1973, pp. 98-109.
- [GRA-80]** Implementing a Set of Switching Functions in Terms of Programmable Logic Arrays (PLA).
W. Grass.
Digital Processes, Vol.6, No. 1, 1980, pp. 75-96.
- [GRA-82/a]** A Depth-First Branch-and-Bound Algorithm for Optimal PLA Folding.
W. Grass.
19 DAC, Las Vegas, Nevada, June 1982, pp. 133-140.

- [GRA-82/b] A Self-Testing PLA.
G. Grassl and H. Pleiderer
ISSCC 82, pp. 60-61.
- [GRE-76] An Associative Logic Matrix.
D. Greer.
IEEE Journal of Solid-State Circuits, Vol SC-11, No. 5,
October 1976, pp. 679-691.
- [GUY-79] The MOM 400 Single Chip Microcomputer.
A. Guyot, M. Henry and M. Vergniault.
ESSIRC 1979.
- [GUY-82] FLOPE: un Editeur de Plan de Masse.
A. Guyot et I. Supriana.
IMAG, Rapprot Interne, Jan. 1982.
- [GUY-83] FLOPE : Un Editeur Graphique de Plan de Masse de
Circuits Intégrés.
A. Guyot, R. Reis et I. Supriana.
Rapport de Recherche No. 333, IMAG, Janvier 1983.
- [HAC-80] Some Results in Optimal PLA Folding.
G. Hachtel, A. Sangiovanni-Vicentelli and A. Newton
ICCC 80, October 1-3, 1980, Port Chester, N.Y., pp. 1023-1027.
- [HAC-81] An Algorithm for Optimal PLA Folding.
IBM Research Report RC8668, Jan. 1981.
- [HAC-82/a] An Algorithm for Optimal PLA Folding.
G. Hachtel, G. Newton and A. Sangiovanni Vicentelli.
IEEE. Trans. CAD Int. Circ. Sys. Vol. CAD-1, No. 2,
April 1982, pp. 63-77.
- [HAC-82/b] Techniques for Programmable Logic Array Folding.
G. Hachtel.
19 DAC, Las Vegas, Nevada, June 1982, pp. 147-155.
- [HAY-81] Strip Architecture Fits Microcomputer Into Less Silicon.
J. Hayn, K. McDonough and J. Bellay.
Electronics, Jan., 27, 1981, pp. 107-111.
- [HEB-76] High-Speed Programmable Logic Arrays in ESFJ-SOS technology.
E. Hebenstreit and K. Horninger.
IEEE. J. Solid-State Circ., Vol. SC-11, No. 3,
June 1976, pp. 370-374.
- [HEM-76] The PLA: A Different Kind of ROM.
A. Hemei.
Electronic Design 1, Jan. 5, 1976, pp. 78-84.

- [HEN-83] Partitionning Programmable Logic Arrays.
J. Hennessy.
ICCAD, Apr. 1983, pp. 63-76.
- [HOF-81] A Method for Topological Compaction of Programmed Logic Arrays.
M. Hoffman.
Master Report ERL University of California,
Berkley, CA, USA, 1981.
- [HON-72] On Complementation of Boolean Functions.
J. Hong and D. Ostapko
IEEE. Trans. on Computers September 1972, p. 1022.
- [HON-74] MINI: A Heuristic Approach for Logic Minimization.
S. Hong, R. Cain and A. Newton.
IBM Res. Develop., Vol. 18, No. 5, Sept. 1974, pp. 443-458.
- [HON-79] Fault Analysis and Test Generation for Programmable Logic Arrays (PLA's).
S. Hong and D. Ostapko.
IEEE. trans. on Computers, Vol. C-28, Sept. 1979, pp. 617-626.
- [HU-83] Optimum Reduction of Programmable Logic Array.
T. Hu and Y. Kuo.
20 DAC, Miami, Florida, June 1983, pp. 553-558.
- [HUL-75] A Prime Implicant Algorithm with Factoring.
B. Hulme and R. Worrel.
IEEE. Trans. on Computers, November 1975, pp. 1129-1131.
- [HWA-74] A Method for Generating Prime Implicants of a Boolean Expression.
H. Hwa.
IEEE trans. on Computers June 1974, pp. 636-641.
- [JON-75] Array Logic Macros.
J. Jones.
IBM J. Res. Develop., Vol. 19, March 1975, pp. 120-126.
- [JOH-84/a] A VLSI Superminicomputer CPU.
W. Johnson.
ISSCC 84, Feb. 1984, pp. 174-175.
- [JOH-84/b] VLSI Circuit design reaches the level of architectural description.
S. Johnson.
Electronics, may 3, 1984, pp 121-128.
- [KAM-79] Logic Design of Programmable Logic Arrays.
Y. Kambayashi.
IEEE. Tr. On Computers, Vol. C-28, No. 9,
Sep. 1979, pp. 609-617.

- [KAN-81/a] Automatic PLA Synthesis from a DDL-P Description.
S. Kang and W. Van Cleemput.
18 DAC, pp.391-397.
- [KAN-81/b] Automated Synthesis of PLA Based Systems.
S. Kang.
Ph. D. Thesis, Stanford University, March 1981.
- [KAN-84] Gate Matrix layout of Random Control Logic in a 32-bit CMOS CPU Chip Adaptable to Evolving Logic Design.
S. Kang, R. Krambeck, H. Law and A. Lopez.
IEEE. Tr. on CAD, Vol. CAD-2, No. 1, Jan 1983, pp.18-29.
- [KHA-82] Concurrent Error Detection and Testing for Large PLAs.
D. Khakbaz and E. McCluskey.
IEEE J. of Solid-State Circuits, Vol. SC-17, No. 2, April 1982, pp. 386-394.
- [KRA-83] Etude d'un Optimiseur Logique de PLA.
R. Krasicki.
Thèse d'Ingénieur CNAM, CUEFA, Grenoble, 28 Octobre 1983.
- [KWA-77/a] Logical Systems Design Using PLAs and Petri Net: Programmable Hardware Systems.
C. Kwan, C. Michel and P. Le Beux.
Information Processing 77,
IFIP (North-Holland Publ. Co., 1977).
- [KWA-77/b] The Design of Structured Digital Systems Controlled by PLA.
C. Kwan, P. Le Beux and C. Michel.
EUROMICRO 1977, North-Holland Publ. Co., pp. 44-50.
- [KYU-82] Testing for Delay Faults in a PLA.
K. Son and D. Pradhan.
ICCC 82, New York, N.Y., Oct. 1982, pp. 346-349.
- [LAW-82] PLA Design for the Bellmac-32a Microprocessor.
H. Law and M. Sohi.
ICCC 82, New York, N.Y., Oct. 1982, pp. 161-164.
- [LIP-76] ARRAY LOGIC
H. Lipp.
Second Symposium on Micro Architecture, Euromicro 1976,
North-Holland Publishing Company, pp. 57-63.
- [LIU-83] Bounds on the Saved Area Ratio Due to PLA Folding.
W. Liu and D. Atkins.
20 DAC, Miami, Florida, June 1983, pp. 538-544.
- [LOG-75] Hardware Implementation of a Small System in Programmable Logic Arrays.
*J. Logue, N. Brickman, F. Howley,
J. Jones and W. Wu.*
IBM Res. Develop, Vol 19, No. 2, March 1975, pp. 110-119.

- [LUB-82] Some Theoretical Results on the Optimal PLA Folding Problem.
*M. Luby, U. Vazirani, V. Vazirani
and A. Sangiovanni Vicentelli.*
ICCC 82, New York, N.Y., Oct. 1982, pp. 165-170.
- [MAR-83] PRONTO: Quick PLA Product Reduction.
J. Martinez Carballido and V. Powers.
20 DAC, Miami, Florida, June 1983, pp. 545-552.
- [MAY-76] High-Speed Static Programmable Logic Array in LOCMOS.
P. May and F. Schierek.
IEEE. J. of Solid-State Circ., Vol. SC-11, No. 3,
June 1976, pp. 365-369.
- [MCC-57] Minimization of Boolean Functions.
E.J. Mc Cluskey.
Bell Systems Technical Journal, Vol. 35,
April 1957, pp. 1417-1444.
- [MCC-79] Designing with PLA's
E. McCluskey
Technical Note No. 168. Center for Reliable Computing.
Departments of Electrical Engineering and Computer Science,
Standford University, Standford, Ca.
- [MEA-80] Introduction to VLSI Systems.
C. Mead and L. Conway.
Addison-Wesley, 1980.
- [MIC-78] Hardwired Logic and Microprocessors.
C. Michel and C. Kwan.
EUROMICRO Journal 4, 1978, pp. 167-180.
- [MOR-67] Partioned List Algorithms for Prime Implicant Determination
from Canonical Forms.
E. Morreale.
IEEE Transactions on Electronic Computers, Vol. EC-16, No. 5,
oct 1967, pp. 611-620.
- [MOR-70] Recursive Operators for Prime Implicant and Irredundant
Normal Form Determination.
E. Morreale.
IEEE. Trans. on Comp. vol. C-19, No. 6,
June 1970, pp. 504-509.
- [NEW-81] Design Aids for VLSI: The Berkeley perspective.
*A. Newton, D. Pederson, A. Sangiovanni Vicentelli
and C. Sequin.*
IEEE. Tr. Circ. Syst., Vol. CAS-28, No. 7,
July 1981, pp. 666-680.

- [OES-80]** PLASYS: Final Report
D. Oestreicher.
Silicon Structures Project Report 3655
Computer Science Department, California Institute of
Technology, May 1980.
- [PAI-81]** Optimization of the PLA Area.
J. Paillotin.
18 DAC, pp 406-410.
- [PAP-84]** VLSI Design of a PLA Based Microcontrol Scheme.
C. Papachristou, R. Rashid and S. Gambhir.
ICCD 84, Oct. 1984, Port Chester, N.Y., pp. 771-777.
- [PAT-79]** A Programmable Logic Approach for VLSI.
S. Patil and T. Welch.
IEEE Tr. on Computers, Vol. C-28, No. 9,
September 1979, pp. 594-601.
- [PER-79]** Etude de la Conception et la Minimisation des PLAs-
Dessin des PLAs du MOM-400.
T. Pérez Segovia.
Rapport Interne, Equipe d'Architecture des Ordinateurs,
ENSIMAG, Septembre 1979.
- [PER-80/a]** Optimisation en Surface des PLAs.
T. Pérez Segovia.
Rapport DEA., ENSIMAG, 26 Juin 1980.
- [PER-80/b]** Optimisation Topologique des PLAs.
T. Pérez Segovia.
IMAG. Rapport de Recherche No. 216, Octobre 1980.
- [POM-82]** A 32-bit Execution Unit in Advanced NMOS Technology.
M. Pomper, W. Beifus, K. Horninger and W. Kaschte.
IEEE. J. of Solid-State Circuits, Vol. SC-17,
No. 3, June 1982.
- [POT-75]** Choix d'une Structure de sequenceur.
J. Potet et J. Laurent.
Rapport Interne, Unité de projets, E.F.C.I.S., Octobre 1975.
- [POW-84]** An Automated, Low Power, High Speed Complementary PLA Design
System for VLSI Applications.
S. Powell, E. Iodice and E. Friedman.
ICCD 84, Oct. 1984, Port Chester, N.Y., pp. 314-319.
- [PRI-82]** A CMOS PLA Generator.
W. Pritchard.
ESSIRC 1982, Bruxelles, Belgique, pp. 94-97.

- [PUL-84]** A Recursive Bipartitioning Approach for PLA Partitioning and Layout.
V. Pulges and M. Lighter.
ICCD 84, Oct. 1984, Port Chester, N.Y., pp. 320-323.
- [RAO-79]** Etude de la Structure Interne et du Fonctionnement du Microprocesseur INTEL 8085.
Rao Malladi
Rapport DEA. ENSIMAG, Juin 1979.
- [REI-82]** TESS: A Topological Evaluator Tool.
R. Reis.
ICCC 82, New York, N.Y., Oct. 1982, pp. 539-542.
- [REU-75]** Generation of Prime Implicants from Subfunctions and an Unifying Approach to the Covering Problem.
B. Reush.
IEEE. Tr. Comp., Vol C-24, No. 9, Sept. 1975, pp. 924-930.
- [ROT-74/a]** Methods in Design Optimization Array Logic.
J. Roth.
IBM Research Report RC-4943, 1974.
- [ROT-74/b]** Decreasing the Size of Associative Logic Arrays.
J. Roth.
IBM Technical Disclosure Bulletin, Vol. 17, No. 4, Sept. 1974.
- [ROT-78]** Programmed Logic Array Optimization
J. Roth.
IEEE. Trans. on Computers Vol. C-27, Feb. 1978, pp. 174-176.
- [SAS-81]** Multiple Value Decomposition of Generalized Functions and the Complexity of Programmable Logic Arrays.
T. Sasao.
IEEE. Trans. on Computers Vol. C-30, Sept. 1981, pp. 635-643.
- [SCH-80]** Design of Large ALUs Using Multiple PLA Macros.
M. Schmookler.
IBM J. Res. Develop. Vol. 24, No. 1, January 1980, pp. 2-14.
- [SCH-83]** LUBRICK: A Silicon Assembler and Its Application to Data-Path Design for FISC.
J. Schoellkopf.
VLSI'83, pp. 435-445.
- [SCH-84]** A 32b Bus Interface Chip.
R. Schumann and W. Parker.
ISSCC 84, Feb. 1984, p. 176.
- [SHI-79]** Demystifying Microprocessor Design.
M. Shima.
IEEE. Spectrum, July 1979.

- [SLA-70] A New Algorithm for Generating Prime Implicants.
J. Slagle, C. Chang and R. Lee.
IEEE. tr. Computers, Vol. C-19, No. 4,
April 1970, pp. 304-310.
- [SMI-79] Detection of Faults in Programmable Logic Arrays.
J. Smith.
IEEE. Trans. on Computers, Vol. C-28, Nov. 1979, pp.845-853.
- [SMI-82] Structured Logic Design of Integrated Circuits Using
Storage/logic Array (SLA).
k. Smith, T. Carter and C. Hunt.
IEEE J. of Solid-State Circuits, Vol. SC-17, No.2,
April 1982, pp. 395-406.
- [SOB-83] The Universal Synchronous Machine.
R. Sobol.
VLSI Design, November 1983, pp. 60-64.
- [SOU-82] PLA Versus Bit Slice: comparison for a 32 bit Alu.
E. Soutschek, M. Pomper and K. Horninger.
IEEE J. of Solid-State Circuits, Vol. SC-17, No. 3,
June 1982, pp. 584-586.
- [STE-82] A Fully Automatic, Technology-Independent PLA
Macrocell Generator.
*M. Stebnisky, M. McGinnis, J. Werbickas,
R. Putatunda and A. Feller.*
ICCC 82, New York, N.Y., Oct. 1982, pp. 156-160.
- [STE-83] APSS: An Automatic PLA Synthesis System.
M. Stebnisky, M. McGinnis and J. Werbickas.
20 DAC, Miami, Florida, June 1983, pp. 430-435.
- [SU-69] Computer Reduction of Two-Level, Multiple-Output
Switching Circuits.
Y. SU and D. Dietmeyer.
IEEE. Trans. on Computers, Vol C-18, No. 1,
Jan. 1969, pp. 58-63.
- [SUM-83] A New Integrated System For PLA Testing and Verification.
F. Sumenzi, S. Gai, M. Mezzalama and P. Prinetto.
20 DAC, Miami, Florida, June 1983, pp. 57-63.
- [SUR-75] Minimization of Switching Functions- A Fast technique.
Sureshchander.
IEEE Trans. on Computers, july 1975, pp. 753-756.
- [SUW-81] A Computer-Aided-Design System for Segmented-Folded
PLA Macro-Cells.
I. Suwa and W. Kubitz.
18 DAC, Nashville, Tennessee, June 1981, pp. 398-405.

- [TIS-67] Generalization of Consensus Theory and Application to the Minimization of Boolean Functions.
P. Tison.
IEEE. tr. on Electronic Computers, Vol. EC-16, No. 4, August 1967, pp. 446-456.
- [TIS-71] An Algebra for Logic Systems-Switching Circuits Application.
P. Tison.
IEEE.Tr. Comp., March 1971, pp. 339-351.
- [TRI-81] A Structured Design Methodology and Associated Software Tools.
S. Trimberger, J. Rowson, C. Lang and J. Gray.
IEEE. Tr. on Circuits and Systems, Vol. CAS-28, No. 7, July 1981, pp. 618-633.
- [TEE-82] A Logic Minimizer for VLSI PLA.
B. Teel and D. Wilde.
19 DAC, Las Vegas, Nevada, June 1982, pp. 156-162.
- [VAN-79] Hierarchical Design for VLSI: Problems and Advantages.
W. Van Cleemput.
Proceedings of Caltech Conference on VLSI, January 1979, pp. 259-274.
- [VAR-84] Optimisation et Dessin des Masques de PLA Monomatrice.
P. Varinot.
Rapport de D.E.A. en Electronique/Circuits Intégrés.
USMG, Juin 1984.
- [VAR-85] Method of PLA's Implementation : The Monomplane PLA.
P. Varinot et S. Chuquillanqui.
ESSIRC 18-22 Septembre 1985, Toulouse.
- [VER-80] Macro Generation Algorithms for LSI Custom Chip Design.
B. Vergnieres.
IBM Res. Develop., Vol. 24, No. 5, May 1980, pp. 612-621.
- [WEB-79] High level Design for Programmed Logic Arrays.
H. Weber.
4th Int. symp. on Computer Hardware Description Languages, Palo Alto, Ca, Oct. 1979, pp. 96-101.
- [WEI-67] Large Scale Integration of MOS Complex Logic: A Layout Method.
A. Weinberger.
IEEE Journal of Solid-State Circuits, Dec 1967, pp. 182-190.
- [WEI-78] Parallel Adders Using Standard PLAs.
A. Weinberger.
4th Symp. on Computer Arithmetic, Santa Monica, CA, Oct. 1978.

- [WEI-79] High-Speed Programmable Logic Array Adders.
A. Weinberger.
IBM J. Res. Develop. Vol. 23, No. 2, March 1979, pp. 163-178.
- [WIM-83] HOPLA: PLA Optimization and Synthesis.
S. Wimer and N. Sharfman.
20 DAC, Miami, Florida, June 1983,, pp. 790-794.
- [WOO-75] High-Speed dynamic Programmable Logic Array Chip.
R. Wood.
IBM J. Res. Develop., Vol. 19, 1975, p. 379.
- [WOO-79] A High Density Programmable Logic Array Chip.
R. Wood.
IEEE Tr. on Computers, Vol. C-28, No. 9,
September 1979, pp. 602-608.

AUTORISATION de SOUTENANCE

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

VU le rapport de présentation de Monsieur ANCEAU, Professeur INPG en disponibilité

Monsieur PEREZ SEGOVIA Tomàs

est autorisé à présenter une thèse en soutenance en vue de l'obtention du titre de
DOCTEUR DE TROISIEME CYCLE, spécialité "Informatique".

Fait à Grenoble, le 10 octobre 1985

D. BLOCH
Président
de l'Institut National Polytechnique
de Grenoble

P.O. le Vice-Président,

A handwritten signature in black ink, appearing to be 'D. Bloch', with a stylized, flowing script.

