



**HAL**  
open science

## Liaison d'un outil de description des circuits intégrés à un microscope électronique à balayage

Isabelle Guiguet

► **To cite this version:**

Isabelle Guiguet. Liaison d'un outil de description des circuits intégrés à un microscope électronique à balayage. Electronique. 1985. dumas-00319081

**HAL Id: dumas-00319081**

**<https://dumas.ccsd.cnrs.fr/dumas-00319081>**

Submitted on 5 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.

**CONSERVATOIRE NATIONAL DES ARTS ET METIERS**

**CENTRE AGREE**

**DE GRENOBLE (C.U.E.F.A.)**

**MEMOIRE**

**présenté en vue d'obtenir**

**le DIPLOME D'INGENIEUR C.N.A.M.**

**en**

**informatique**

**par**

**Isabelle GUIGUET**

**LIAISON D'UN OUTIL DE DESCRIPTION DES CIRCUITS INTEGRES**

**A UN MICROSCOPE ELECTRONIQUE A BALAYAGE**

**Soutenu le 28 octobre 1985**

**JURY**

**Président : MM L. BOLLIET et J.Y. RANCHIN**

**Membres : M B. COURTOIS**

**M J.P. MOREAU**

**M Y. VERNAY**



## **RESUME**

Ce document présente l'établissement d'un lien entre la description CALMA d'un circuit et un microscope électronique à balayage, en vue d'une mise au point des circuits intégrés assistée par ordinateur. En phase de validation, le concepteur identifie certaines connexions par un nom dans le fichier des masques et il peut connaître leur niveau logique dans différents états du circuit. Ces niveaux sont issus des images obtenues à l'aide du microscope. Une correspondance est ainsi établie entre le dessin des masques et l'image microscope du circuit.

### **Mots clés :**

CAO.

Microscope électronique à balayage,

Contraste de potentiel,

Validation de circuits prototypes.





Ce travail a été réalisé au sein de l'équipe d'Architecture des Ordinateurs du Laboratoire TIM3, dans le cadre d'un congé formation.

Je remercie très chaleureusement,

MM L.BOLLIET et J.Y.RANCHIN, Professeurs au Conservatoire National des Arts et Métiers, qui m'ont fait l'honneur de présider ce jury.

M Bernard COURTOIS, Chargé de Recherches au CNRS, qui m'a accueilli dans son équipe et a dirigé mes travaux.

M Jean-Pierre Moreau, Responsable Stratégie CAO à Thomson Semi Conducteurs, et M Yves VERNAY, Ingénieur au CNET, pour avoir accepté de participer au jury.

M Jacques LAURENT, Docteur de l'INPG, qui m'a initié à la microscopie électronique et M Jean-Paul EYNARD, Ingénieur au CNRS, pour les conseils et le soutien qu'ils m'ont apportés tout au long de ce travail.

M Pierre JUTIER, Maître de Recherches à l'INSERM, pour le temps consacré à me former au traitement d'images.

Tous les membres de l'équipe et du CIME, et plus particulièrement Claire RUBAT DU MERAC, Laurent BERGHER et Dominique MICOLLET pour leur gentillesse et leur aide constantes.

Me Isabelle AMIELH et Mlle Isabelle DI BILIO qui ont assuré la frappe de ce mémoire.

Le service de reprographie de l'IMAG qui a assuré le tirage de ce document.



## TABLE DES MATIERES

<b>INTRODUCTION.....</b>	<b>9</b>
<b>1 - <u>LE MICROSCOPE ELECTRONIQUE A BALAYAGE (MEB)</u>.....</b>	<b>11</b>
1.1 - Interaction entre le faisceau incident et l'échantillon .....	11
1.2 - Structure schématique d'un MEB .....	13
1.3 - Le contraste de potentiel .....	16
1.4 - Facteurs perturbant une image en contraste de potentiel .....	18
1.4.1 - Présence d'électrons rétrodiffusés .....	18
1.4.2 - L'émission secondaire ne dépend pas uniquement du contraste de potentiel .....	19
1.5 - Les applications du MEB en microélectronique .....	21
1.5.1 - Caractérisation .....	21
1.5.2 - Restructuration .....	21
1.5.3 - Recherche de fonctionnement limite .....	22
1.5.4 - "Reverse engineering" .....	22
1.5.5 - Inspection - contrôle de qualité .....	22
1.5.6 - Analyse de défaillances .....	23
1.5.7 - Mise au point de circuits prototypes .....	23
<b>2 - <u>L'OUTIL</u> .....</b>	<b>25</b>
2.1 - Le MEB .....	25
2.2 - Le conditionneur .....	27
2.3 - Le système de traitement d'images .....	30
2.3.1 - Organisation de la mémoire et des processeurs .....	30
2.3.2 - Architecture fonctionnelle .....	32
2.3.3 - Architecture matérielle .....	32

2.3.4 - Système d'adressage .....	36
2.3.5 - Fonctionnement .....	37
2.3.6 - Saisie d'une image .....	37
2.4 - Le système informatique de traitement et de contrôle .....	40
<b>3 - <u>LE "STREAM FORMAT" CALMA</u></b> .....	<b>43</b>
3.1 - Description d'un enregistrement .....	43
3.2 - Représentation machine des différents types de données .....	44
3.2.1 - Description .....	44
3.2.2 - Représentation schématique des données numériques .....	45
3.2.3 - Exemples .....	46
3.3 - Les différents types d'enregistrements .....	47
3.4 - La syntaxe CALMA .....	51
<b>4 - <u>MISE AU POINT DE CIRCUITS PROTOTYPES</u></b> .....	<b>55</b>
4.1 - Présentation du problème .....	55
4.2 - Test des défauts de fabrication .....	56
4.3 - Test des défauts de conception .....	58
4.3.1 - Localisation d'un bloc défectueux .....	59
4.3.2 - Localisation à l'intérieur d'un bloc de défauts modélisables .....	60
4.3.3 - Localisation à l'intérieur d'un bloc de défauts non modélisables .....	60
<b>5 - <u>REALISATIONS</u></b> .....	<b>63</b>
5.1 - Transfert des programmes translatables sur ROMUALD .....	63
5.1.1 - Format des programmes à transférer .....	63
5.1.2 - Programmes développés .....	65

5.2 - Détermination des coordonnées théoriques des points à observer .....	67
5.2.1 - Identification d'un polygone .....	67
5.2.2 - Programme de traduction du fichier de base .....	69
5.2.3 - Programme d'affichage .....	79
5.2.4 - Programme de calcul des points d'observation .....	94
5.2.5 - Récapitulatif.....	95
5.3 - Détermination de l'état logique des points à observer .....	96
5.3.1 - Présentation .....	96
5.3.2 - Correspondance entre l'image théorique et l'image pratique .....	97
5.3.3 - Calcul de l'adresse des points dans la mémoire image .....	100
5.3.4 - Correspondance entre un niveau de gris et un niveau logique .....	105
5.3.5 - Programmes développés .....	107
<b>6 - TESTS</b> .....	<b>113</b>
6.1 - Tests relatifs au calcul des coordonnées théoriques des points à observer .....	113
6.2 - Tests relatifs à la détermination du niveau logique des points à observer .....	114
6.2.1 - Influence d'une variation de l'adresse des points de recalage .....	114
6.2.2 - Influence de la position du point de recalage horizontal par rapport au point de recalage origine .....	116
<b><u>CONCLUSION</u></b> .....	<b>117</b>

<b><u>ANNEXES</u></b> .....	121
Annexe 1 : Images microscope .....	121
Annexe 2 : Algorithmes .....	126
A2.1 - Algorithme du programme de traduction .....	126
A2.2 - Algorithme du programme de calcul des points d'observation .....	130
Annexe 3 : Résultats du programme d'affichage .....	132
Annexe 4 : Erreurs de positionnement .....	135
Annexe 5 : Superposition d'une image pratique et d'une image théorique .....	138
<b><u>BIBLIOGRAPHIE</u></b> .....	143

## **INTRODUCTION**

Lors de la mise au point de Circuits Intégrés (CI) prototypes, le concepteur peut conduire deux types d'analyse :

### **\* Une analyse externe**

Elle consiste à étudier le comportement du circuit à travers ses plots d'entrée/sortie. Des machines de test spécialisées sont utilisées dans ce cas. Cependant l'analyse externe se heurte à la faible accessibilité des circuits complexes. En effet, un circuit ne comportant que quelques dizaines de broches pour un nombre toujours croissant de transistors, les informations recueillies sont insuffisantes pour une étude complète de sa logique interne.

### **\* Une analyse interne**

Elle consiste à contrôler le fonctionnement des CI par prélèvement de valeurs logiques et électriques en divers points internes du circuit, au niveau des composants (transistors, connexions...). Elle facilite beaucoup l'établissement d'un diagnostic précis. Les outils actuellement disponibles pour réaliser ces mesures sont les machines à pointes. Elles sont peu adaptées aux technologies actuelles caractérisées par une décroissance des dimensions des éléments des CI, allant de pair avec une intégration croissante.

En effet :

- les mesures sont difficiles : le positionnement des pointes est long et délicat en raison du fort risque d'endommagement de la surface du CI.
- les pointes introduisent une capacité qui perturbe le fonctionnement électrique du circuit et fausse les mesures.
- les points de mesure doivent être explicitement implantés lors de la conception du circuit. Ceci entraîne une perte de place sur la surface du CI et laisse peu de souplesse à la stratégie de test.



Une des alternatives à cette sonde mécanique est la sonde électronique. L'analyse de points internes du CI peut être effectuée à l'aide d'un Microscope Electronique à Balayage (MEB). L'utilisation récente de cet appareil, pour mettre au point des prototypes, nécessite de le relier aux outils CAO permettant de décrire les circuits. Le langage de description choisi est le "stream format" de CALMA. Il s'agit donc de relier les descriptions CALMA au MEB. Les différentes étapes sont les suivantes :

- \* Déterminer les points à observer à partir de la description CALMA des masques du circuit
- \* Retrouver ces points sur l'image issue du microscope
- \* Déterminer le niveau logique de chacun en fonction de son niveau de gris.

La première partie de ce rapport décrira un microscope électronique à balayage. Le chapitre suivant présentera l'outil de test organisé autour de cet appareil.

Dans la troisième partie, le langage de description de circuits CALMA sera décrit. La quatrième partie présentera les méthodes utilisables pour mettre au point des circuits prototypes.

Enfin, les deux derniers chapitres seront consacrés au travail effectué, en décrivant respectivement les programmes implantés et les différents tests réalisés.

## 1 - LE MEB

### 1.1 - Interaction entre le faisceau incident et l'échantillon

Lorsqu'un faisceau d'électrons accélérés (appelé faisceau incident et noté FI) entre en contact avec une cible, les chocs entre ces électrons et les atomes du solide provoquent l'émission de plusieurs types d'électrons et de rayonnements électromagnétiques (figure 1.1) [MEB 78], notamment :

\* des électrons rétrodiffusés (ER)

Il s'agit des électrons incidents renvoyés par l'échantillon. Le taux de rétrodiffusion dépend de la nature de l'échantillon : il augmente avec le nombre atomique de ce dernier. Ces électrons ayant une énergie élevée peuvent émerger de couches assez profondes (de 150 à 1300 nm), et leur propagation est rectiligne.

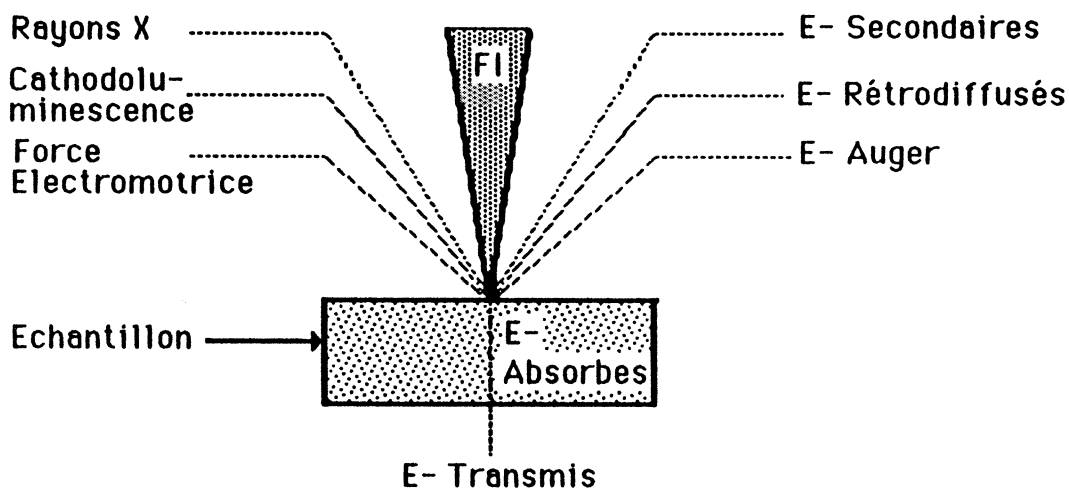


Figure 1.1 - Signaux émis par les interactions entre le F.I. et l'échantillon

\* des électrons secondaires (ES)

Il s'agit d'électrons d'énergie inférieure à 50 eV provenant de l'excitation des atomes de l'échantillon par les électrons incidents ou rétrodiffusés. En raison de leur faible énergie, ils ne peuvent émerger de l'échantillon que s'ils sont émis à partir d'une couche superficielle (de 1 à 10 nm), c'est-à-dire avant diffusion du faisceau dans la matière. (La figure 1.2. permet de situer cette zone d'émission). En conséquence, ce signal fournit la résolution la plus fine, approximativement égale au diamètre de la sonde. C'est l'image qu'il génère qui est exploitée pour étudier l'état du potentiel de surface des CI. L'annexe 1 présente des images générées par ce signal. Le taux d'émission secondaire dépend :

- de la nature de l'échantillon
- de l'angle formé par la surface de l'échantillon et le FI
- du potentiel de surface de l'échantillon (voir paragraphe 1.3)

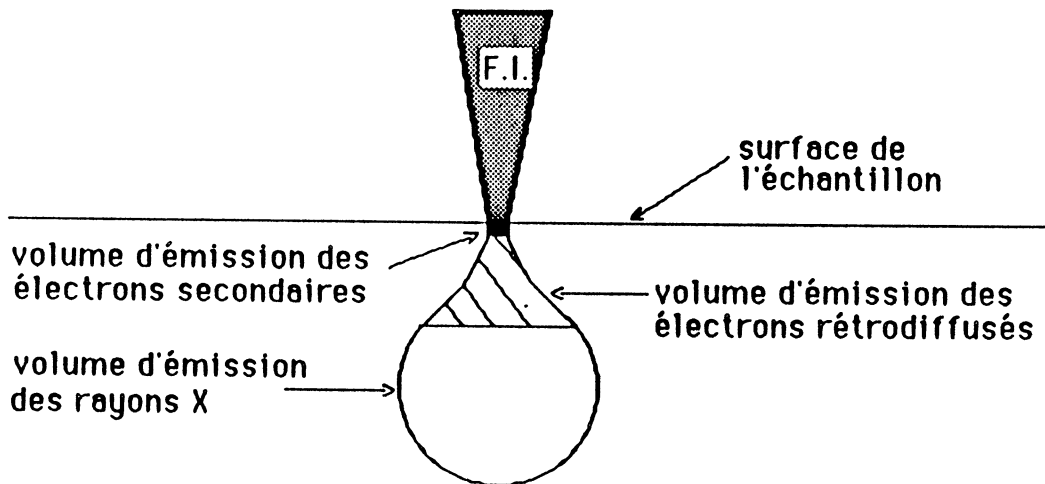


Figure 1.2 - Répartition des zones d'émission

\* des électrons absorbés (CE)

Il s'agit des électrons incidents non réémis par l'échantillon en raison de l'importante perte d'énergie qu'ils subissent lors des chocs successifs.

## **1.2. - Structure schématique d'un MEB**

Un MEB, représenté sur les figures 1.3 et 1.4 [MOL 80] comprend :

- Une platine goniométrique qui supporte l'objet à examiner et permet de le déplacer dans la chambre à vide du MEB.
- Un canon à électrons qui génère et accélère le faisceau incident. Il est formé d'une cathode thermoémissive (souvent constituée d'un filament de tungstène), d'une anode servant à accélérer les électrons et d'un wehelt qui concentre les électrons en une image du filament appelée le "cross-over". La tension d'accélération et le courant du filament sont réglables.
- Un ensemble de focalisation : il s'agit de deux ou trois lentilles électromagnétiques ayant pour but de faire converger le faisceau incident sur la surface du CI, c'est à dire de former une image ponctuelle du cross-over au niveau de l'échantillon.
- Un diaphragme permettant de faire varier le diamètre de la sonde.
- Un dispositif de déviation du faisceau qui permet de déplacer le spot dans deux directions perpendiculaires afin d'obtenir un balayage ligne et un balayage trame, de fréquence et d'amplitude réglables.
- Un détecteur d'ES : le plus répandu est un dispositif mis au point par Everhart et Thornley. Les ES libérés sont attirés par le collecteur qui est porté à un fort potentiel positif et le signal résultant est amplifié. Des détecteurs permettant de recueillir les autres signaux peuvent également être présents.
- Un écran de contrôle sur lequel est visualisé le signal amplifié. Le nombre d'ES recueillis module l'intensité du signal analogique qui arrive sur le canon à électrons de l'écran du MEB, et détermine ainsi la luminosité de l'image:

Si ce canon est soumis à un signal fort, de nombreux électrons sont générés et viennent "blanchir" l'écran; sinon, peu d'électrons sont émis et l'écran reste noir. Le balayage du MEB est synchrone de celui de l'écran de contrôle assurant ainsi une correspondance entre tout point balayé par le FI et le point image le représentant.

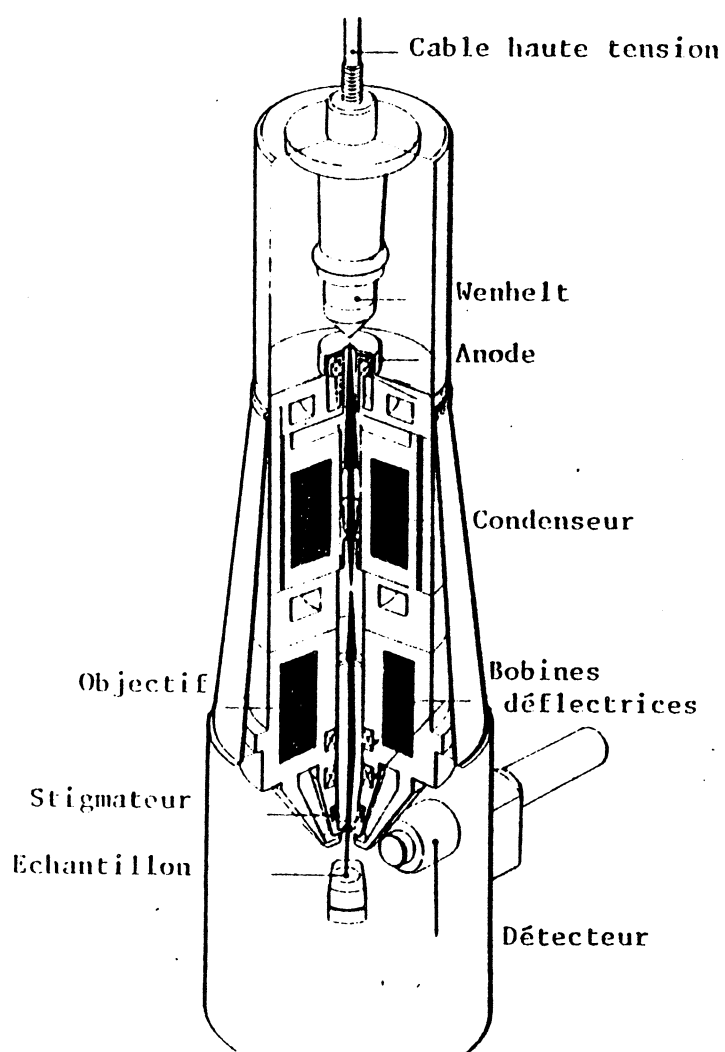
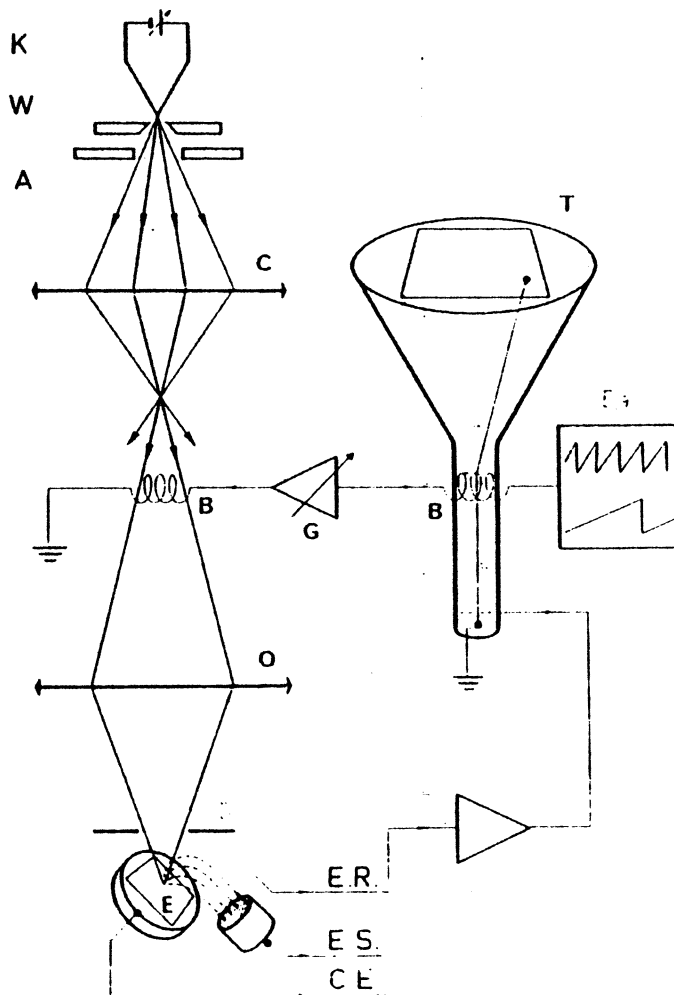


Figure 1.3 - Coupe schématique d'un MEB

Légende :

- K : cathode thermoémissive
- W : Wenhelt
- A : anode
- C, O : condenseur, objectif (lentilles électromagnétiques)
- B : bobines de déflexion
- Ba : dispositif de balayage (balayages ligne et trame)
- D : diaphragme
- T : tube cathodique
- G : contrôle du grandissement
- E : échantillon
- ER, ES, CE : détecteurs de différents signaux

Figure 1.4 - Schéma d'un MEB

### **1.3 - Le contraste de potentiel**

Le nombre d'ES émis dépend entre autres paramètres de la distribution de potentiel sur la surface du CI. Ce phénomène, manifesté lorsque le circuit est alimenté, est appelé contraste de potentiel et est schématisé en figure 1.5. [FAZ 81].

Les électrons secondaires ont une faible énergie, et sont donc très sensibles aux champs électriques. A cause du fort potentiel du collecteur (300v), presque tous les électrons secondaires émis par une surface à la masse sont captés. Par contre, si l'interconnexion est à +5v, des champs électriques sont créés à sa surface ; ils se superposent à celui du collecteur et peuvent complètement annuler son effet.

Les électrons secondaires issus d'interconnexions à la masse sont soumis à un champ d'extraction qui les accélère vers le collecteur. Par contre, ceux issus d'interconnexions à +5v doivent passer à travers un champ retardateur qui renvoie la plupart d'entre eux sur les interconnexions. Le collecteur recevant moins d'électrons secondaires des interconnexions à +5v que de celles à la masse, les états logiques du CI peuvent être visualisés : les zones du circuit à un potentiel haut ("1" logique) apparaissent sombres, tandis que les zones à un potentiel bas ("0" logique) apparaissent claires.

Le contraste de potentiel permet donc d'effectuer des mesures qualitatives de niveaux logiques. Par contre il n'est pas adapté aux mesures quantitatives du potentiel car il ne varie pas linéairement avec le potentiel de surface. Cette non linéarité est principalement due aux deux facteurs suivants [BER 82] [BAR 84] :

- le signal mesuré est un courant et n'est pas une fonction linéaire du potentiel de surface.
- le signal obtenu en un point peut être perturbé par le potentiel des connexions voisines

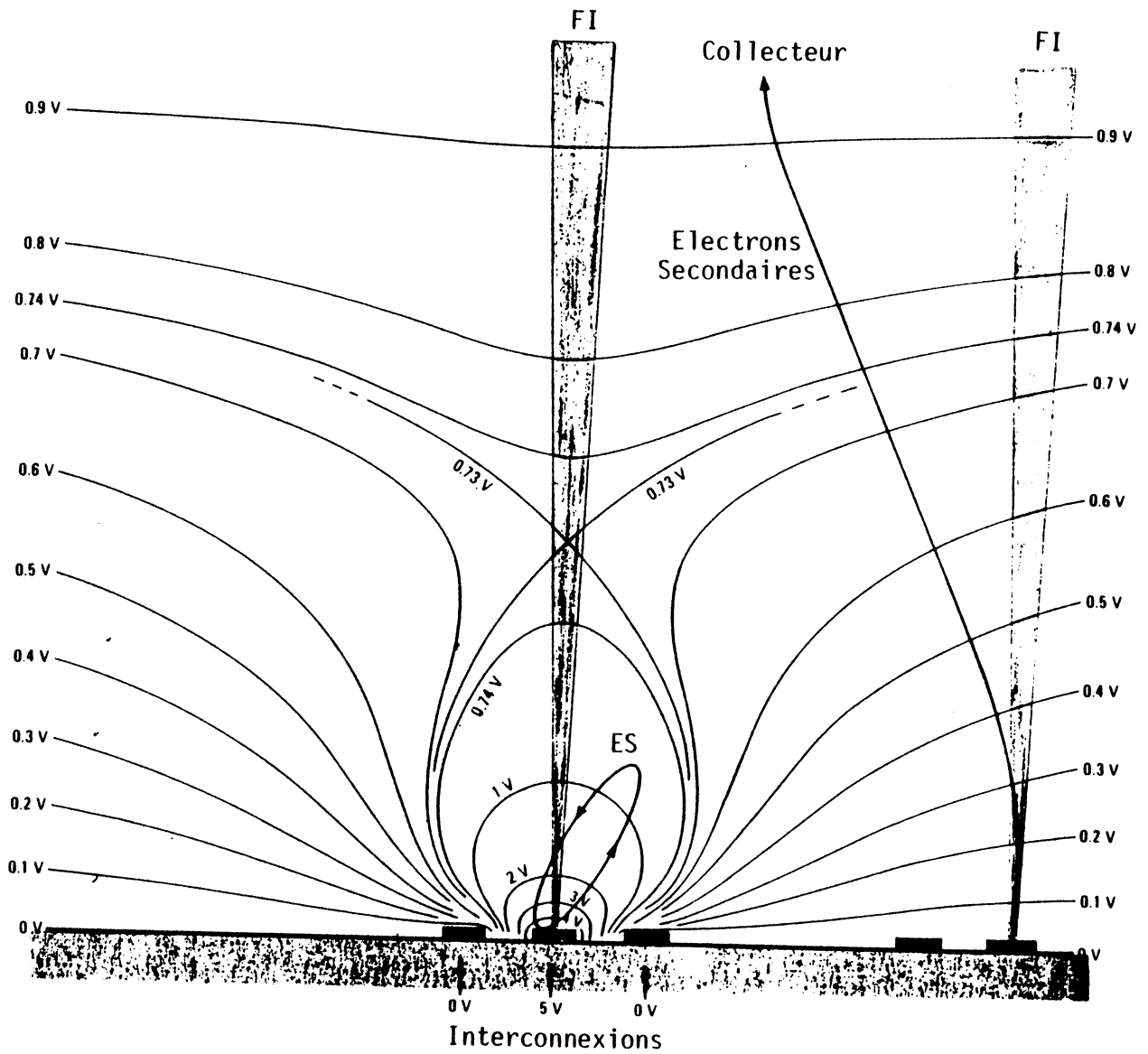


Figure 1.5 - Contraste de potentiel. Les électrons secondaires émis par les connexions à +5v sont repoussés par le champ électrique local, alors que ceux issus de connexions à la masse sont accélérés vers le collecteur.



Cependant, l'adjonction d'un spectromètre électronique [FLE 74] [FEU 82] permet d'effectuer des mesures quantitatives. Il comprend :

- Une grille d'extraction portée à un potentiel important et située au dessus de l'échantillon.

- Un champ retardateur qui sert à maintenir constant le courant de sortie par application d'une contre-réaction. On ne mesure plus le signal de sortie (qui est constant), mais la contre-réaction appliquée qui évolue linéairement avec la variation du potentiel de surface.

#### **1.4 - Facteurs perturbant une image en contraste de potentiel**

Une image en contraste de potentiel ne dépend pas uniquement de la distribution de potentiel à la surface du CI. Ceci est principalement dû aux phénomènes suivants :

##### **1.4.1 - Le signal détecté comprend des électrons rétrodiffusés**

Contrairement aux électrons secondaires, les électrons rétrodiffusés suivent une trajectoire rectiligne. La figure 1.6 montre que certains sont captés par le collecteur.

Ils seront ou non collectés en fonction de l'orientation de la surface du circuit au point d'émission par rapport au collecteur. Les électrons rétrodiffusés représentent approximativement 5 à 10% des électrons collectés, ce qui fausse la mesure des électrons secondaires.

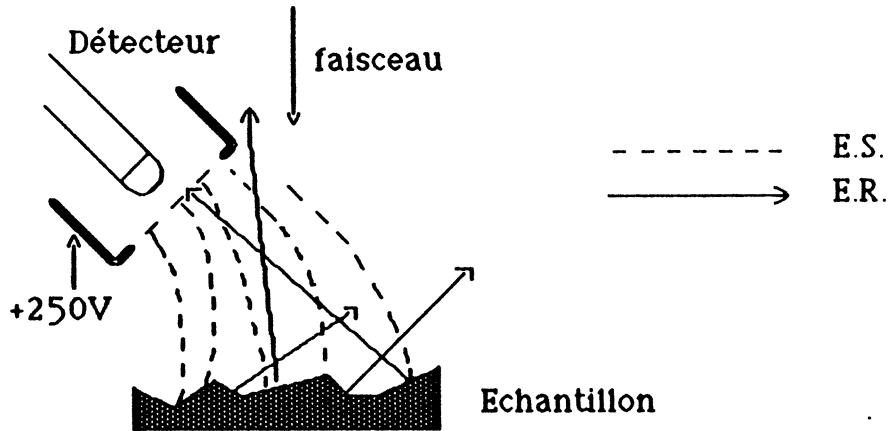


Figure 1.6 - Détection des E.S.

#### 1.4.2 - L'émission secondaire ne dépend pas uniquement du contraste de potentiel

En effet, les facteurs suivants interviennent :

##### \* Le contraste topographique

Le relief de la surface de l'échantillon influence le taux d'émission secondaire. En effet, le nombre d'ES émis varie avec l'angle d'incidence  $\theta$  formé par le faisceau incident et la droite perpendiculaire à la surface (voir figure 1.7). Si le faisceau incident frappe une surface oblique (respectivement plane), beaucoup (respectivement peu) d'ES sont libérés et l'image apparaît claire (respectivement sombre). Ce phénomène qui s'explique par le fait que les électrons secondaires s'extraient de l'échantillon d'autant plus facilement que la surface est plus inclinée, constitue le contraste topographique.

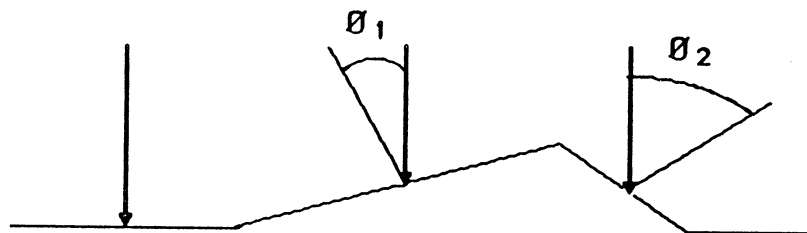


Figure 1.7 - Représentation de l'angle d'incidence du faisceau

Lorsque le circuit n'est pas alimenté, seul ce contraste est présent. Ce mode de fonctionnement est notamment utilisé pour inspecter la couche métallique supérieure des CI (détection de coupures de lignes, de court-circuits,...).

Lorsque le circuit est alimenté, il s'ajoute au contraste de potentiel et intervient ainsi dans l'image obtenue.

Une technique de comparaison permet de s'affranchir du contraste topographique ainsi que de la présence des ER. Elle consiste à faire une observation alors que le circuit n'est pas alimenté. La même opération est ensuite effectuée lorsqu'il est alimenté. Enfin, la première observation est soustraite à la seconde, ce qui permet d'obtenir un résultat ne dépendant que du contraste de potentiel.

#### \* La nature de l'échantillon

Lorsque le faisceau frappe le circuit certains électrons incidents sont absorbés. Dans un isolant, leur accumulation crée une charge négative qui entraîne une diminution du taux d'émission secondaire et par suite, la perte progressive du contraste. Cet effet est notamment observé pour la couche d'oxyde de passivation (couche d'isolant déposée sur les CI). Il est donc préférable d'observer des circuits non passivés ou dépassivés. Il faut cependant noter que de tels circuits finissent également par se charger si le temps d'observation est important, en raison de la présence d'isolants internes.

\* L'effet de proximité

Le signal obtenu au point d'impact peut être perturbé par le potentiel des points voisins. Cet effet est d'autant plus important que les connexions sont plus rapprochées, c'est à dire que la densité d'intégration est plus élevée.

Il peut être atténué en plaçant une grille d'extraction portée à un fort potentiel au dessus de l'échantillon, ce qui est la cas du spectromètre électronique pour les mesures quantitatives.

## **1.5 - Les applications du MEB en microélectronique**

Pour certaines applications le circuit n'est pas alimenté. Pour d'autres, il est observé en fonctionnement dans la chambre à vide du MEB .

### **1.5.1 - Caractérisation**

Elle consiste à étudier les semi-conducteurs afin de tester une technologie (pour contrôle ou pour mise au point). Le MEB permet de mesurer différents paramètres physiques tels que la profondeur de la jonction, le dopage, etc...[BRE 77].

### **1.5.2 - Restructuration**

Deux types de restructuration peuvent être utilisés [SHA 81] :

\* Une restructuration temporaire

Elle sert à aider un test.

\* Une restructuration définitive

Elle sert à isoler une zone défectueuse afin d'améliorer le rendement de production ou poursuivre le test sur le reste du circuit.

### 1.5.3 - Recherche de fonctionnement limite

Elle s'effectue en deux phases [HUE 84] :

1) Détermination des frontières de fonctionnement du circuit en fonction de certains paramètres. Pour cela, il est simplement nécessaire de disposer d'un critère de fonctionnement/non fonctionnement du CI.

2) Localisation du point géographique cause du passage fonctionnement/ non fonctionnement à l'aide du MEB.

### 1.5.4 - "Reverse engineering"

Cette application consiste à analyser un circuit de structure non connue afin de retrouver ses schémas électriques (et éventuellement logiques). Deux étapes sont nécessaires :

#### 1) Relevé de la description des masques

Cette opération est actuellement réalisée manuellement par un opérateur. Il effectue une observation du circuit et peut utiliser le MEB afin de comprendre son fonctionnement. L'automatisation de cette étape qui touche le domaine de l'analyse d'images est en cours de développement.

#### 2) Détermination des schémas électriques à partir de la description des masques

Cette phase est réalisée à l'aide d'un extracteur [JER 83].

### 1.5.5 - Inspection - contrôle de qualité

Le CI n'est pas alimenté. Sa surface est observée afin de détecter certains défauts de fabrication tels que des coupures de lignes, des court-circuits, des rétrécissements de métallisation, etc... [RUB 84]

Deux types d'inspection peuvent être envisagés :

- Une inspection visuelle où l'opérateur observe le circuit et décide de l'accepter ou de le rejeter selon la gravité des défauts qu'il présente.

- Une inspection automatique où l'opérateur est assisté par un système de traitement d'images qui inspecte avec référence :

- \* aux règles de dessin,
- \* à un circuit bon,
- \* à la description du CI inspecté.

#### 1.5.6 - Analyse de défaillance

Elle peut avoir lieu à deux moments distincts de la vie d'un CI :

- Après fabrication afin de localiser puis examiner des défauts de fabrication.
- Après défaillance afin de localiser puis examiner des défauts se produisant pendant la vie du circuit.

Elle est réalisée en deux phases [BER 85] :

##### 1) Localisation du défaut

Le MEB est utilisé en contraste de potentiel (circuit alimenté)

##### 2) Diagnostic

- . Recherche du type de défaut (court-circuit, coupure, etc...)
- . Recherche du procédé physico-chimique entraînant la défaillance (corrosion, électromigration, etc...).

#### 1.5.7 - Mise au point de circuits prototypes

Il s'agit de l'une des applications les plus importantes du MEB. Elle est considérablement facilitée si les outils CAO ( descriptions, extracteurs, simulateurs, etc...) sont accessibles.

Elle peut couvrir la détection des défauts :

- \* de fabrication,
- \* et de conception (mise au point proprement dite).

Cette application fait l'objet de ce rapport et sera détaillé dans les chapitres suivants.



## **2 - L'OUTIL**

Le développement des diverses applications nécessite de relier le MEB à d'autres appareils. L'ensemble constitue l'outil de test par faisceau d'électrons dont le schéma est représenté en figure 2.1 [LAU 84]. Les principaux composants reliés au MEB sont :

- un conditionneur permettant de faire fonctionner le circuit à analyser,
- un système de traitement d'images,
- un système informatique assurant le contrôle de l'ensemble de ces éléments ainsi que le traitement des informations.

L'outil mis en œuvre est constitué par :

### **2.1 - Le MEB**

Le microscope utilisé est un JEOL 35 C. Il fournit un signal visualisé sur un écran de contrôle et éventuellement transmis au système de traitement d'images. Il présente les caractéristiques suivantes :

- le facteur de grossissement est compris entre 10 et 90000.
- la tension d'accélération des électrons peut varier entre 1 et 39 KV.
- la platine porte objet permet de déplacer le circuit en translation (suivant les axes x et y), en hauteur (suivant l'axe z), ou en rotation et de modifier son inclinaison par rapport au faisceau incident. Ce dernier réglage sert à atténuer les déformations d'images observées lorsque le CI n'est pas absolument perpendiculaire au faisceau (en raison du collage de la puce sur le boîtier et du montage du boîtier sur son support). Le déplacement de la platine est réalisé manuellement par l'opérateur.



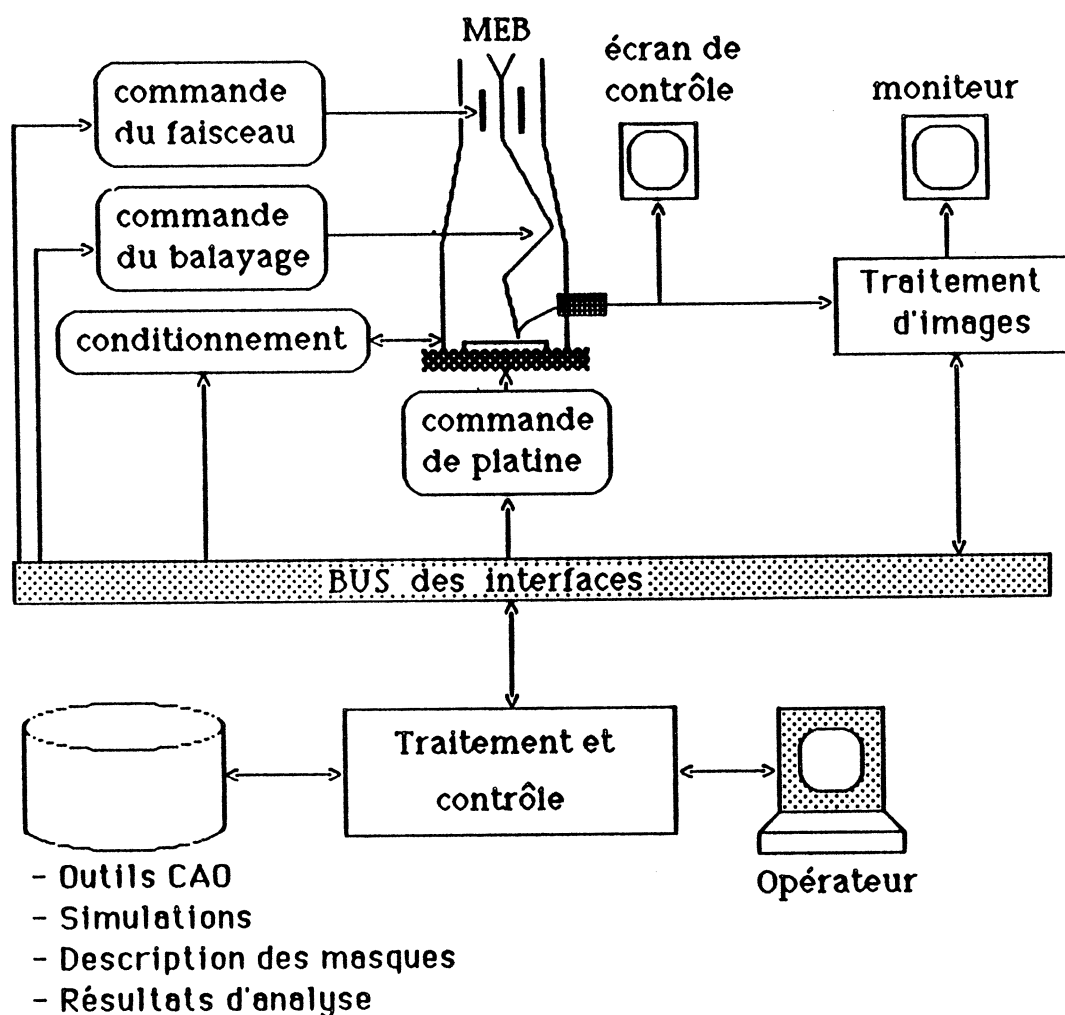


Figure 2.1 - Organisation générale de l'outil

- il ne possède pas de dispositif permettant de positionner le faisceau en tout point du circuit, et ainsi de ne balayer qu'une partie ou qu'un point de l'image. En conséquence la totalité de l'image contenant le(s) point(s) à observer est systématiquement balayée.

- la stroboscopie ne peut être effectuée que sur échantillonnage. En effet l'absence d'un modulateur de faisceau [FOU 81] en impulsions de très courte durée ne permet pas actuellement d'utiliser la stroboscopie de faisceau.

## **2.2 - Le conditionneur**

Il s'agit d'un outil programmable d'aide à la mise au point des CI développé dans l'équipe [BAU 85], son schéma est représenté en figure 2.2. Il permet de conditionner les circuits ayant jusqu'à 64 broches et d'analyser leur réponse à ce conditionnement. Pour cela, il a accès à toutes les broches du CI placé dans la chambre à vide du MEB, par l'intermédiaire d'un câble comprenant un fil par broche et traversant la paroi du microscope par des passages étanches. Il peut soit être utilisé de façon autonome, soit être relié à un système informatique.

Ses principales fonctions, contrôlées par microprocesseur 8 bits (6809), sont les suivantes :

### **a) Relier ses entrées/sorties aux broches du CI**

Cette fonction est réalisée à l'aide d'une matrice de connexions comportant 64 colonnes correspondant aux pattes du circuit et 9 lignes correspondant à leur nature. Elle permet de particulariser chaque patte, chacune pouvant être une alimentation, une masse, une horloge ou une donnée (en entrée, en sortie, ou bidirectionnelle)

### **b) Stocker un microprogramme de test**

Avant exécution, des vecteurs d'entrée sont rangés dans une mémoire de test. Il s'agit d'une RAM comportant 4K mots de 72 bits. Un mot mémoire décrit :

- le vecteur de test destiné à exciter le circuit (bits 0 à 63). L'ensemble de ces microinstructions constitue le microprogramme de test.

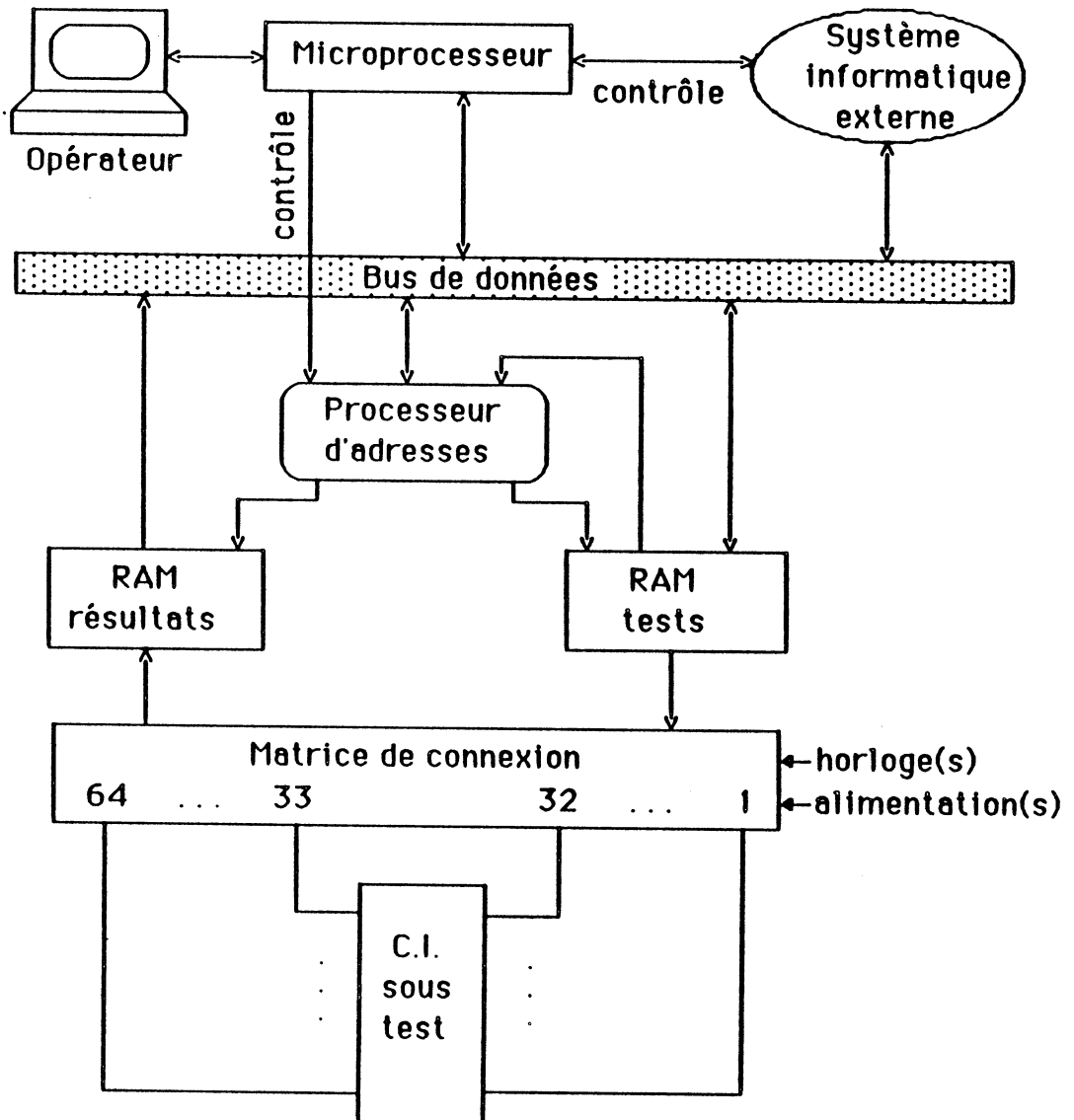


Figure 2.2 - Schéma du conditionneur

- des informations servant au contrôle de ce microprogramme :
  - . le bit 64 indique si un point d'arrêt est ou non présent au niveau de la micro-instruction,
  - . les bits 65 à 71 représentent une séquence à laquelle obéit le processeur d'adresse chargé du déroulement (séquencement) du microprogramme. Ces informations de séquencement permettent :
    - \* le passage en séquence,
    - \* le saut à une autre adresse de test,
    - \* des boucles sur une même micro-instruction.

A l'issue de ce remplissage mémoire, effectué par l'opérateur ou par un système informatique externe, le conditionneur est prêt pour l'analyse du comportement du circuit.

### c) Contrôler le déroulement du microprogramme de test

Les broches d'entrée sont excitées à la fréquence désirée avec les vecteurs de test stockés en mémoire. Il est possible de choisir l'un des types d'exécution suivants :

- fonctionnement pour n cycles :

Un compteur de cycles est chargé via le microprocesseur avec le nombre de cycles voulu, puis la main est donnée au processeur d'adresse. A chaque cycle, le compteur est décrémenté, et son passage à zéro redonne la main au microprocesseur. Ce mode de fonctionnement permet une exécution en pas à pas (n=1).

- fonctionnement jusqu'à un point d'arrêt :

Une commande permet de placer des points d'arrêt aux adresses voulues. Elle a pour effet de positionner le bit 64 de la mémoire de test à "1". Si l'exécution est lancée avec l'option point d'arrêt, la détection de ce bit stoppe l'exécution et redonne la main au microprocesseur.

- fonctionnement continu.

Une commande permet à l'utilisateur d'arrêter à tout moment l'exécution du microprogramme de test. Le système imprime alors l'adresse de la micro-instruction suspendue, ainsi que la valeur du compteur de cycles.

#### d) Recueillir les réponses externes du circuit

Au même rythme que le déroulement des vecteurs d'entrée, les réponses du circuit sont prélevées sur les broches de sortie et rangées dans une mémoire de résultats comportant 4K mots de 64 bits. Il s'agit d'une RAM en écriture seule pour le circuit (elle est remplie séquentiellement en synchronisme avec le circuit), et en lecture seule pour le conditionneur (les 4096 derniers événements sur les sorties du circuit peuvent être affichés à l'écran ou transmis au système informatique).

### **2.3 - Le système de traitement d'images**

Il s'agit d'un multiprocesseur interactif SIMD (single instruction, multiple data) où le même flot d'instructions est exécuté sur plusieurs ensembles de données simultanément. Il a été réalisé au sein du Centre Interuniversitaire de Calcul de Grenoble, et a été appelé ROMUALD [BRE 80] [BRE 84] [RUB 83].

#### **2.3.1 - Organisation parallèle de la mémoire et des processeurs**

Une image TV peut être divisée en tranches, chacune comportant un certain nombre de lignes adjacentes. La mémoire dans laquelle est rangée l'image peut être partitionnée de la même manière (figure 2.3) et un processeur, dit "processeur local" (PL), est associé à chaque tranche d'image en mémoire. Un processeur général (PG) contrôle l'ensemble des processeurs locaux.

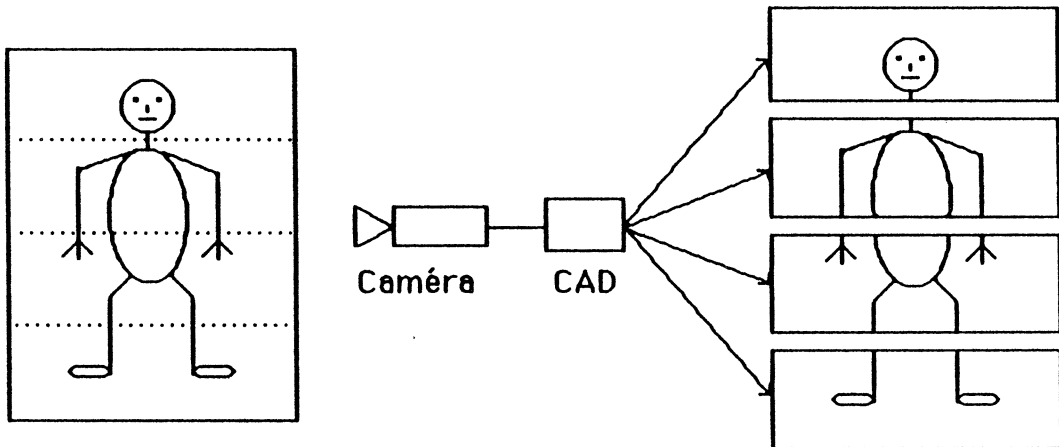


Figure 2. 3 - Partition de l'image et projection en mémoire

Les problèmes d'accès des processeurs aux programmes sont résolus en copiant ces programmes dans des mémoires privées associées à chaque processeur. Chaque processeur local disposant d'une partie cohérente de l'image et d'une copie du code nécessaire à son traitement, l'exécution peut être effectuée en total parallélisme. Cependant, un problème d'accès à l'image se pose pour tout algorithme affectant à chaque pixel une valeur fonction de celle de ses voisins. En effet, lorsqu'il opère près de la frontière d'une tranche, le voisinage déborde sur la tranche voisine. Dans ce cas, le processeur général recopie au dessus de l'image de chaque processeur local un certain nombre de lignes de l'image du processeur précédent. Le transfert inverse est effectué après le traitement (figure 2.4).

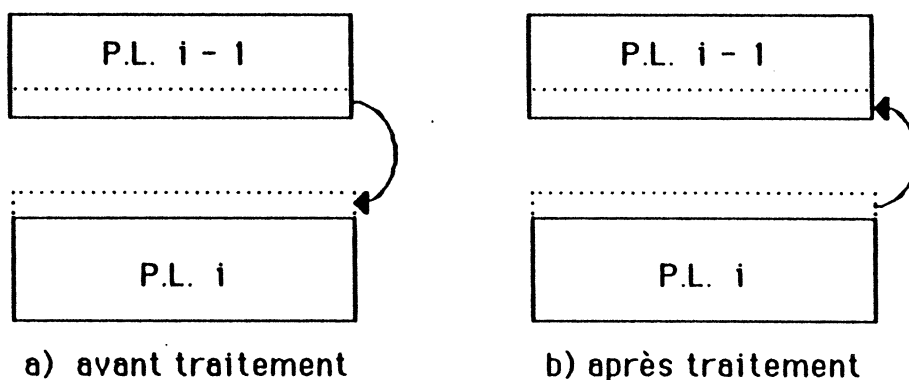


Figure 2. 4 - Recopie de lignes image entre processeurs locaux

### 2.3.2 - Architecture fonctionnelle (Figure 2.5)

Quatre unités logiques composent ce système de traitement d'images :

- \* Une unité de saisie convertit le signal reçu de la caméra, ou du MEB, et range l'image numérisée en mémoire. Une image est digitalisée en une matrice de 512 lignes, chaque ligne comprenant 512 pixels. La valeur de chaque pixel est codée sur huit bits ce qui permet de différencier 256 niveaux de gris. La saisie d'une image est détaillée dans le paragraphe 2.3.6.

- \* Une unité d'affichage convertit l'image numérisée en un signal vidéo standard transmis à l'écran d'affichage.

- \* Une unité de traitement effectue sur l'image les opérations désirées (lissage, seuillage, ... etc).

- \* Une unité de commande assure d'une part le contrôle et l'ordonnancement des tâches confiées aux trois autres unités, et permet d'autre part, via un terminal, le dialogue interactif avec l'utilisateur.

### 2.3.3 - Architecture matérielle (figure 2.5)

Ce processeur d'images comprend :

- \* Un mégabus autour duquel s'organise son architecture. Toutes les unités opératives lui sont connectées. Toutes les informations circulant dans le système transitent par lui.

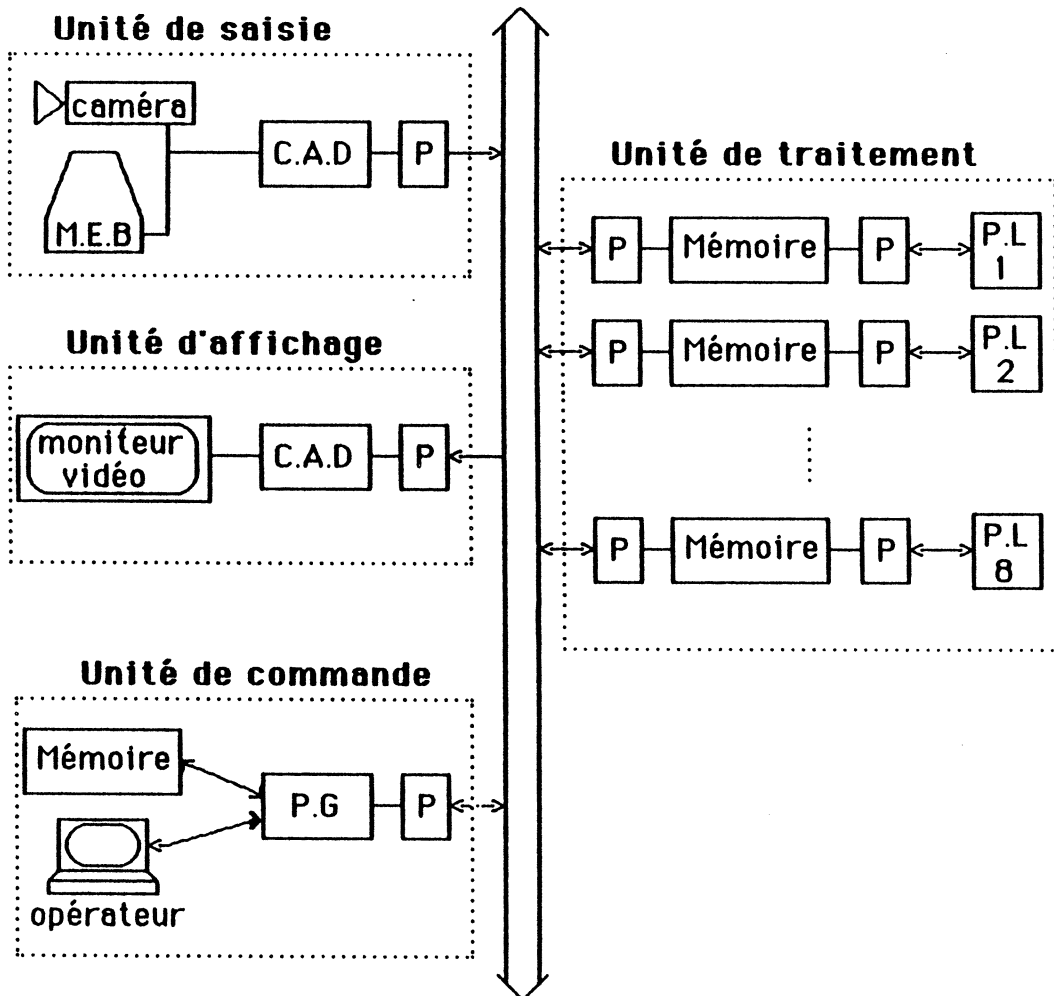


Figure 2.5 - Architecture du système de traitement d'images

P : porte

CAD : convertisseur analogique/digital,

CDA : convertisseur digital/analogique.



\* Une unité de traitement constituée de huit parties appelées micro-unités de traitement. Elles sont physiquement séparées et chacune comprend :

- un processeur de traitement (Z8002);
- une mémoire composée de 32 koct de mémoire image (contenant 64 lignes consécutives de l'image), et de 32 koct de mémoire de traitement (contenant les programmes, les zones de travail, et une zone de communication avec l'unité de commande). Sa structure est représentée en figure 2.6.

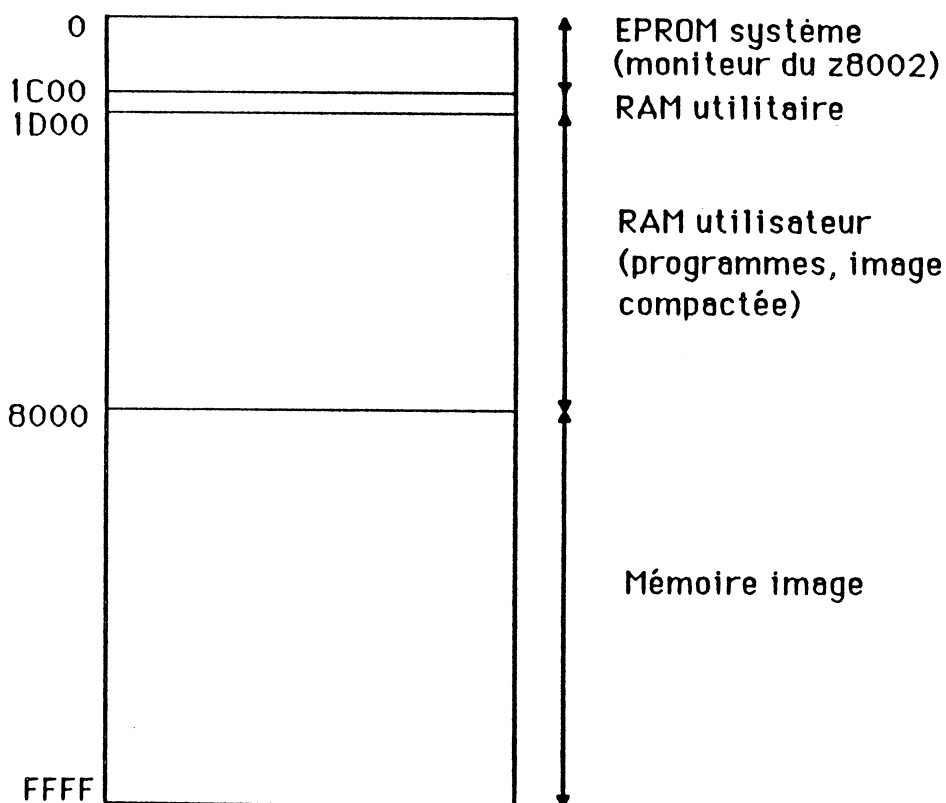


Figure 2. 6 - Organisation de la mémoire locale d'un processeur local

\* Un système de portes permettant d'isoler les micro-unités de traitement du reste de la machine et de reconnaître parmi les informations circulant sur le mégabus celles qui s'adressent à chaque micro-unité. Un exemple d'une telle organisation est présenté en figure 2.7. La mémoire  $i$  peut être accédée par le processeur local  $i$  (porte P1 ouverte, porte P2 fermée) ou par le "monde extérieur" (porte P1 fermée, porte P2 ouverte). Il est ainsi possible de reconfigurer dynamiquement la machine. (Le chemin de données peut être modifié en quelques nanosecondes).

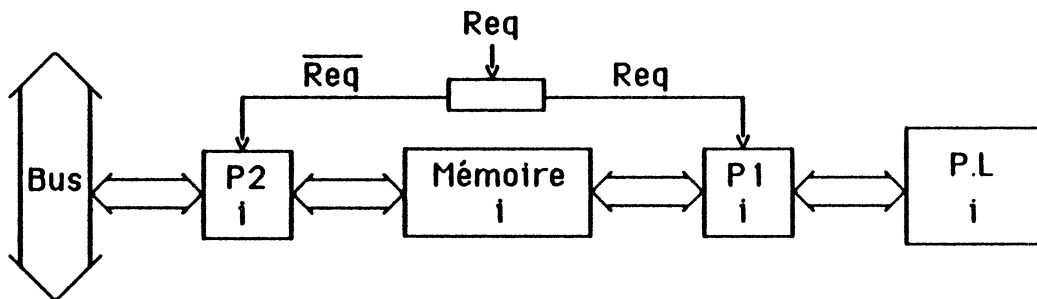


Figure 2. 7 - Mécanisme de portes

\* Un Z8001 assume le rôle du processeur général. Il dispose d'une mémoire de travail comprenant cinq segments de 64 koct chacun. La figure 2.8 donne l'organisation du segment numéro 0. Les quatre autres segments sont entièrement disponibles à l'utilisateur pour, par exemple, stocker une image (stockage de 128 lignes par segment). Ce processeur d'images permet ainsi de ranger deux images : l'une dans la mémoire image des processeurs locaux, l'autre dans la mémoire de travail du processeur général. Une image compactée, image dans laquelle seuls les six bits poids forts de chaque pixel sont conservés, peut de plus être rangée dans la mémoire de traitement des processeurs locaux.

\* Un Z8 pilotant les unités de saisie et d'affichage. Elles sont connectées en mode DMA à toute la mémoire image.

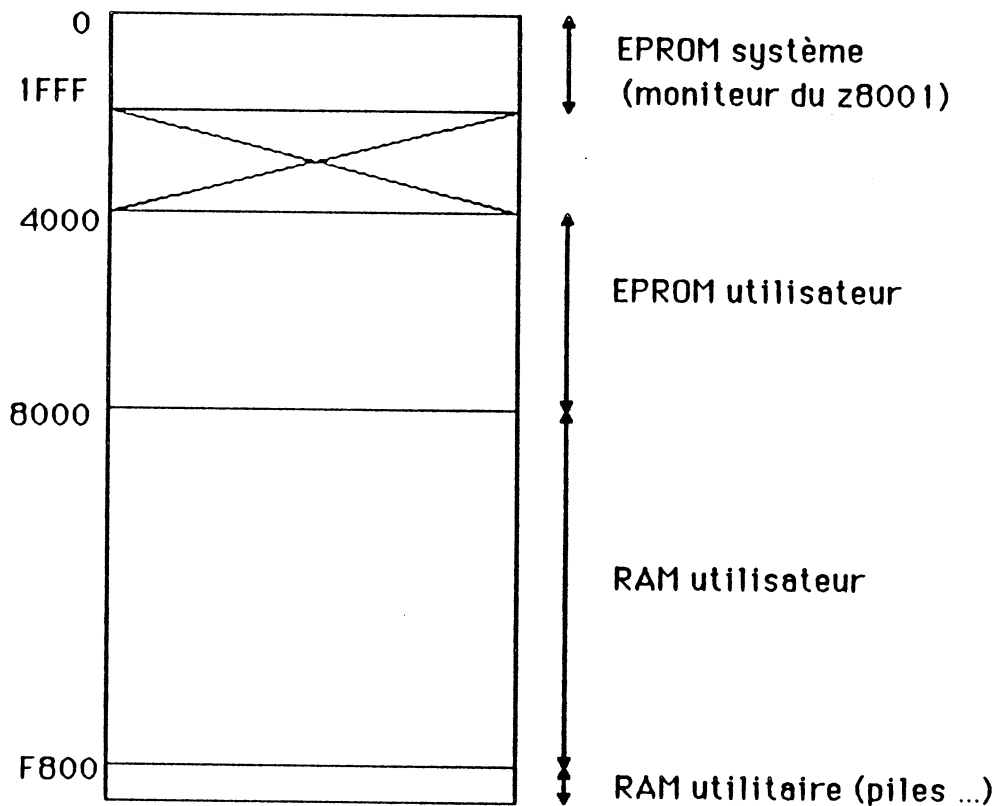


Figure 2. 8 - Segment zéro de la mémoire du processeur général

#### 2.3.4 - Systeme d'adressage

Les adressages suivants peuvent être effectués :

- \* chaque processeur local peut adresser la partie de la mémoire dont il est responsable,
- \* le processeur général peut adresser sa mémoire de travail,
- \* le processeur général et les convertisseurs peuvent adresser l'ensemble des mémoires des processeurs locaux. Pour cela, les adresses circulant sur le mégabus comportent d'une part le nom de la micro-unité accédée, d'autre part le déplacement à l'intérieur de sa mémoire.

### 2.3.5 - Principe du fonctionnement

L'ordonnement du fonctionnement des quatre unités et leur interaction sont réalisés sur le principe d'une soumission totale des PL au PG. Cette soumission est obtenue à partir d'une politique précise sur les interruptions et l'utilisation du mégabus : les PL et les convertisseurs sont interruptibles par le PG uniquement. Ce dernier n'est interruptible que par le Convertisseur Analogique-Digital (CAD) en fin de saisie; le reste du temps, il est le maître absolu de la situation.

### 2.3.6 - Saisie d'une image

#### 2.3.6.1 - Signal Vidéo

Une image TV classique est composée de 625 lignes réparties en une ou deux trames. Le signal vidéo obtenu, schématisé en figure 2.9, comporte :

- L'information analogique relative au niveau de gris. L'intensité lumineuse rencontrée le long d'une ligne est convertie en un signal électrique analogique.
- Des informations de synchronisation :
  - \* 1 ou 2 top trames (impulsion indiquant le début d'une trame),
  - \* N tops lignes (impulsion indiquant le début d'une ligne), N étant le nombre de lignes balayées par le FI.

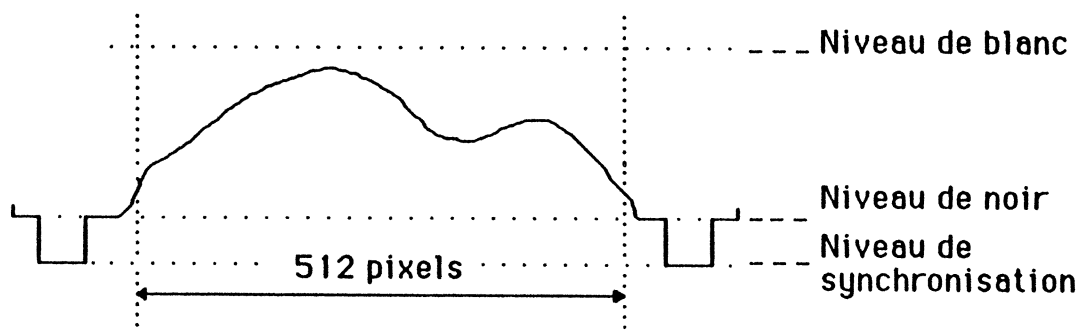


Figure 2.9 - Signal vidéo

### 2.3.6.2 - Le système d'acquisition

Le MEB fournit au processeur d'images un signal de type vidéo (ses tops de synchro sont des tops de synchro vidéo standard) comportant une seule trame. La digitalisation de l'image s'obtient en échantillonnant ce signal analogique. Le CI étant balayé en N lignes, N pixels sont relevés pour chaque ligne. Les instants d'échantillonnage sont déterminés par les tops lignes et tops trame.

L'image de la totalité d'un CI n'étant pas suffisamment détaillée pour être exploitable, le circuit doit être analysé par zones. La situation et les dimensions d'une zone sont libres et son image détaillée peut être obtenue par un zoom du MEB. Il consiste à centrer le point à observer par rapport au faisceau incident en déplaçant la platine porte-objet, puis à augmenter le grossissement jusqu'à ce que la topologie de la zone analysée soit très nette. L'ensemble du circuit étant visualisé en plusieurs fois, pour un même vecteur d'entrée, il faudra saisir une série d'images en déplaçant à chaque fois la platine porte objet. Si le déplacement de cette platine est d'une précision insuffisante, il entraîne la perte des références de position du circuit dans la chambre. Un recentrage est alors nécessaire pour chaque saisie d'image, ce qui provoque d'importantes pertes de temps. Pour pallier à cet inconvénient, ce déplacement doit être automatisé de façon extrêmement précise, ce qui nécessite un équipement de mise en œuvre délicate et onéreuse.

Déplacer la platine entraînant soit d'importantes pertes de temps, soit l'utilisation d'un équipement coûteux, une méthode d'acquisition a été développée afin de limiter ce nombre de déplacements [BAI 82] : le MEB est réglé sur un grossissement fixe permettant de visualiser si possible tout le circuit; lors de la saisie d'une image, l'opérateur indique le numéro de la zone à analyser. Seuls les points échantillonnés appartenant à cette zone sont visualisés sur l'écran du processeur d'images (figures 2.10 et 2.11).

Le balayage est réglé de façon à ce que l'on dispose de 2048 lignes et que sur chacune d'elle, 2048 points soient échantillonnés.

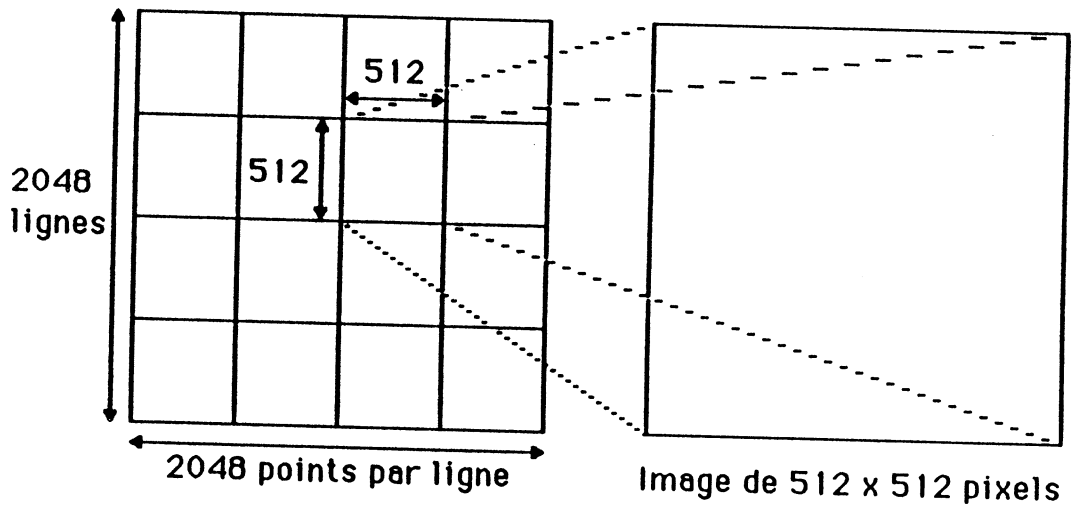


Figure 2. 10 - Système d'acquisition d'une image

Sur les lignes  $m$  à  $m+511$ , les échantillons  $n$  à  $n+511$  sont prélevés. L'acquisition des données ne commence que lors du  $n^{\text{ième}}$  top de la  $m^{\text{ième}}$  ligne.

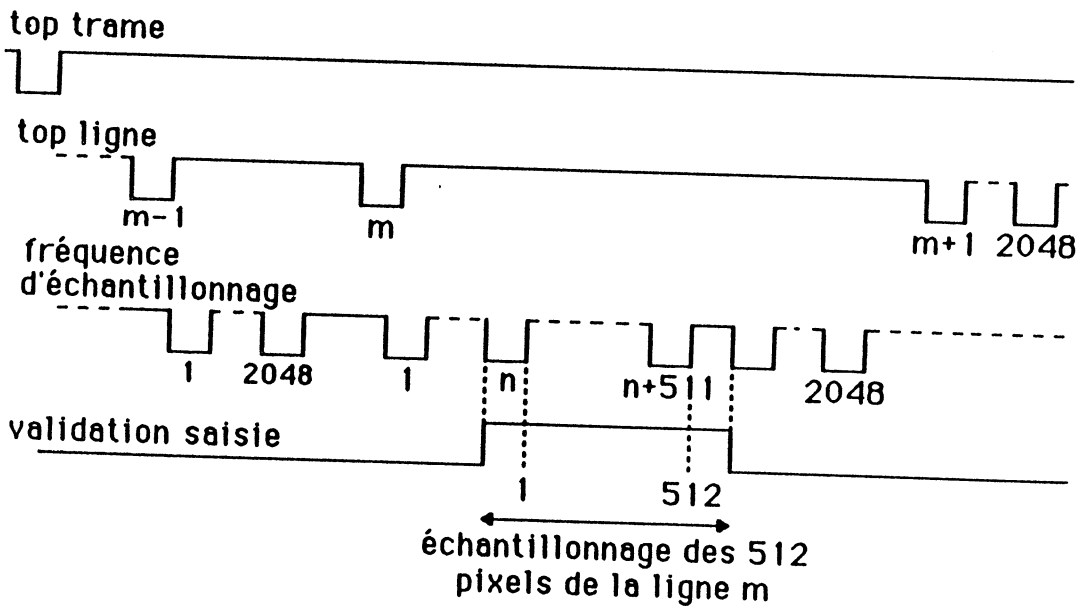


Figure 2. 11 - Echantillonnage d'une zone

## **2.4 - Le système informatique de contrôle et de traitement**

Il s'agit d'un VAX 780-VMS disposant d'une unité centrale (32 bits) d'une capacité mémoire de 4 Moctets , et de disques durs permettant d'enregistrer 500 Moctets. Son importante capacité mémoire et sa puissance de traitement lui permettent d'assurer :

### **\* Le stockage des outils CAO**

Il s'agit des outils permettant par exemple d'éditer un circuit et de générer les fichiers de description de masques, ou de simuler son fonctionnement.

### **\* Le traitement des observations**

Ce système peut récupérer le résultat des tests, à savoir :

- La valeur des broches de sortie du circuit; ces données se trouvent dans la mémoire de résultats du conditionneur (analyse externe).
- Le niveau logique de chaque point observé (analyse interne). Un document comportant notamment les informations suivantes peut être généré :
  - . les coordonnées des points,
  - . la valeur des vecteurs de test,
  - . la valeur logique observée,
  - . la valeur logique de référence (valeur obtenue avec un circuit bon ou calculée par simulation du fonctionnement du circuit).
- Les images digitalisées rangées dans la mémoire du système de traitement d'images. Il est alors possible de développer des applications nécessitant le stockage d'un nombre d'images supérieur aux possibilités offertes par le processeur d'images.

\* Le contrôle de l'ensemble de l'outil

Il peut aider à l'élaboration d'une stratégie de test : en fonction des résultats expérimentaux et des résultats de simulation, il peut déterminer de nouveaux points à tester, et générer des vecteurs d'entrée.

Il pilote donc :

- Le conditionneur en lui transmettant les vecteurs de test à appliquer et en contrôlant leur séquençement (durée de boucle sur un vecteur, choix du vecteur suivant, etc...).
- Le MEB en lui transmettant les coordonnées des points à observer. Celles-ci peuvent notamment être déterminées en manipulant le fichier de description des masques (affichage d'une zone du circuit, sélection des points à observer). La structure d'un tel fichier est décrite dans le chapitre suivant.





### **3 - LE STREAM FORMAT CALMA**

Compte tenu de la grande variété des outils utilisés en CAO de VLSI, il est apparu intéressant de choisir la description utilisée pour la fabrication des masques (point de passage obligatoire) pour retrouver les informations utiles aux tests. A ce niveau, le "stream format" de CALMA a été choisi en raison de son caractère standard. D'autres formats de description pourraient être considérés, dès l'instant qu'un transformateur adéquat permet de passer de l'un à l'autre.

#### **3.1 - Description d'un enregistrement**

Le fichier de sortie décrivant le circuit est composé d'enregistrements de longueurs variables. La structure d'un enregistrement, représentée en figure 3.1 est une suite de données précédée d'un champ de description dont :

- \* deux octets pour donner la longueur totale de l'enregistrement (ce nombre tient compte de ces deux octets),
- \* un troisième pour indiquer le type d'enregistrement.  
(Le paragraphe 3.3 liste les différents types existants),
- \* le quatrième et dernier octet de ce champ donne un numéro indiquant le type des données contenues dans l'enregistrement. Les types disponibles sont les suivants :
  - . 0 : aucune donnée
  - . 1 : chaîne de bits
  - . 2 : entier sur deux octets
  - . 3 : entier sur quatre octets
  - . 4 : réel sur quatre octets
  - . 5 : réel sur huit octets
  - . 6 : chaîne de caractères ASCII

Les octets suivants contiennent les données.

Un enregistrement comporte toujours un nombre pair d'octets, une chaîne de caractères de longueur impaire étant complétée par un caractère NULL.

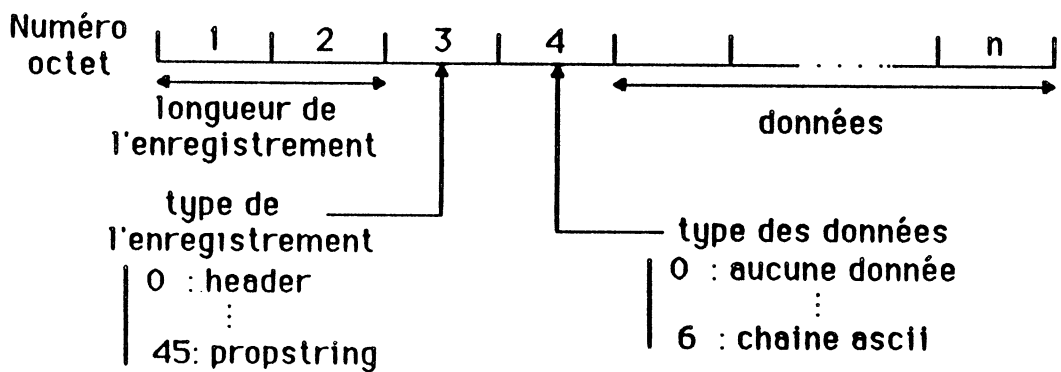


Figure 3. 1 - Schéma d'un enregistrement

## 3. 2 - Représentation machine des différents type de données

### 3. 2. 1 - Description

- Une chaîne de bits est représentée par deux octets.
- Une chaîne de caractères comprend le code ASCII de chaque caractère.
- Un entier est représenté en complément à deux.
- Un réel est représenté en virgule flottante :

une telle représentation se compose de trois parties : le signe, l'exposant et la mantisse. La valeur du nombre est calculée en appliquant la formule suivante :  
 $\text{mantisse} \times 16^{\text{exposant} - 64}$ .

La mantisse est normalisée, c'est à dire toujours comprise entre 1/16 et 1. Il est ainsi possible de considérer que le point se trouve juste à gauche du bit 8 (les bits étant numérotés b0, b1, ...). Le bit 8 représente donc la valeur 1/2, le 9 la valeur 1/4, etc ...

Normaliser consiste à décaler la mantisse par blocs de quatre bits jusqu'à ce que les quatre bits de poids le plus forts soient non nuls. Lors de chaque décalage l'exposant est décrémenté de un.

La représentation d'un nombre négatif diffère de celle du même nombre positif uniquement par le bit le plus fort qui correspond au signe .

Dans la représentation sur huit octets, les quatre premiers octets sont les mêmes que dans la représentation sur quatre octets. Les quatre octets suivants permettent d'avoir une plus grande précision.

### 3. 2. 2 - Représentation schématique des données numériques

- Entier sur 2 octets noté i2 : smmmmmmm M
- Entier sur 4 octets noté i4 : smmmmmmm M M M
- Réel sur 4 octets noté R4 : seeeeeee M M M
- Réel sur 8 octets noté R8 : seeeeeee M M M M M M M

où :

- s représente un bit de signe,
- e représente un bit d'exposant,
- m représente un bit de mantisse,
- M représente un octet de mantisse.

3. 2. 3 - Exemples

* i2 :	137	00000000	10001001		
	- 137	11111111	01110111		
* R4 :	0,7	01000000	10110011	00110011	00110011
			(2 <sup>-1</sup> +2 <sup>-3</sup> +2 <sup>-4</sup> +2 <sup>-7</sup> +2 <sup>-8</sup> +...)		
	1,7	01000001	00011011	00110011	00110011
		( 16x	(2 <sup>-4</sup> +2 <sup>-5</sup> +2 <sup>-7</sup> +2 <sup>-8</sup> +...))		
	1	01000001	00010000	00000000	00000000
		( 16x	(2 <sup>-4</sup> ))		
	- 1	11000001	00010000	00000000	00000000
	10	01000001	10100000	00000000	00000000
		( 16x	(2 <sup>-1</sup> +2 <sup>-3</sup> ))		
	100	01000010	01100100	00000000	00000000
		( 16 <sup>2</sup> x	(2 <sup>-2</sup> +2 <sup>-3</sup> +2 <sup>-6</sup> ))		
	1000	01000011	00111110	10000000	00000000
		( 16 <sup>3</sup> x	(2 <sup>-3</sup> +2 <sup>-4</sup> +2 <sup>-5</sup> +2 <sup>-6</sup> +2 <sup>-7</sup> +2 <sup>-9</sup> ))		

### 3.3 - Les différents types d'enregistrement existants

Format de description :

Enregistrement		Données	
TYPE	LIBELLE	NOMBRE	TYPE
COMMENTAIRES			

**0      HEADER            1                    12**

Numéro de version du stream format

**1      BGNLIB                12                    12**

Contient la date et l'heure de la dernière modification de la bibliothèque ainsi que la date et l'heure du dernier accès à cette bibliothèque. Trois entiers représentent d'une part une date (jour, mois, an), et d'autre part une heure (heure, minute, seconde)

**2      LIBNAME                1                    chaine Ascii**

Nom de la bibliothèque

**3      UNITS                    2                    R8**

Unité utilisée

**4      ENDLIB                0**

Marque la fin de la bibliothèque

**5      BGNSTR                12                    12**

Contient la date et l'heure de création d'une structure ainsi que la date et l'heure de sa dernière modification, et marque le début de cette structure.

- |    |  |   |                     |
|----|--|---|---------------------|
| 6  | <b>STRNAME</b>   | 1 | <b>Chaine Ascii</b> |
|    | Nom de la structure en cours de description. Il comporte au plus 32 caractères appartenant à l'ensemble suivant : (lettres majuscules, lettres minuscules, chiffres, signes "-" et "\$").  |   |                     |
| 7  | <b>ENDSTR</b>  | 0 |                     |
|    | Marque la fin d'une structure.   |   |                     |
| 8  | <b>BOUNDARY</b>  | 0 |                     |
|    | Marque le début d'un polygone.   |   |                     |
| 9  | <b>PATH</b>  | 0 |                     |
|    | Marque le début d'un chemin.   |   |                     |
| 10 | <b>SREF</b>  | 0 |                     |
|    | Marque le début d'un appel simple d'une seule structure.   |   |                     |
| 11 | <b>AREF</b>  | 0 |                     |
|    | Marque le début d'un appel d'un tableau de structures.   |   |                     |
| 12 | <b>TEXT</b>  | 0 |                     |
|    | Marque le début d'un texte.  |   |                     |
| 13 | <b>LAYER</b>   | 1 | <b>I2</b>           |
|    | Numéro compris entre 0 et 63 représentant le niveau dans le circuit (métal, diffusion, etc...).  |   |                     |
| 14 | <b>DATATYPE</b>  | 1 | <b>I2</b>           |
|    | Inutilisé (égal à 0).  |   |                     |
| 15 | <b>WIDTH</b>   | 1 | <b>I4</b>           |
|    | Largeur d'un chemin ou d'une ligne de texte.   |   |                     |
| 16 | <b>XY</b>  | n | <b>I4</b>           |
|    | Ensemble de couples de coordonnées (x,y), chaque valeur étant représentée par un i4. Leurs nombre et signification dépendent de l'élément auquel cet enregistrement se rapporte :  |   |                     |
|    | - Un chemin ou un polygone comporte au plus 200 couples; un chemin en comporte au moins deux et un polygone au moins quatre. Chaque couple représente les coordonnées d'un sommet du polygone ou du chemin. Le premier et le dernier point coïncident. |   |                     |

- Un texte ou un appel simple comprend un seul couple, correspondant au point où est respectivement placée la chaîne de caractères, ou l'origine de la structure appelée.

- Un appel de tableau comporte trois couples qui représentent :

- . Le point origine du tableau.
- . Un point sur la même ligne que le point origine. Ces deux points sont distants de la largeur du tableau.
- . Un point sur la même colonne que le point origine. Ces deux derniers points sont distants de la hauteur du tableau.

17	<b>ENDEL</b>	<b>0</b>	
	Marque la fin d'un élément.		
18	<b>SNAME</b>	<b>1</b>	<b>chaîne Ascii</b>
	Nom de la structure appelée. (Il obéit à la syntaxe décrite lors de la présentation de l'enregistrement STRNAME).		
19	<b>COLROW</b>	<b>2</b>	<b>12</b>
	Nombre de colonnes et nombre de lignes d'un tableau de structures		
22	<b>TEXTYPE</b>	<b>1</b>	<b>12</b>
	Nombre compris entre 0 et 63 représentant le type du texte.		
23	<b>PRESENTATION</b>	<b>1</b>	<b>chaîne de bits</b>
	Indicateurs spécifiant la mise en forme d'un texte. Ainsi :		
	- Les bits 1 et 2 indiquent la justification horizontale (0 = gauche, 1 = centre, 2 = droite).		
	- Les bits 3 et 4 contiennent le numéro de la police de caractères utilisée.		
	- les bits 5 et 6 indiquent la justification verticale (0 = haut, 1 = milieu, 2 = bas)		
26	<b>STRANS</b>	<b>1</b>	<b>chaîne de bits</b>
	Indicateur de réflexion : si le bit de poids fort est égal à 1, une réflexion par rapport à l'axe des x est appliquée (avant une éventuelle rotation angulaire).		
27	<b>MAG</b>	<b>1</b>	<b>R8</b>
	Facteur de grossissement		



- 28 ANGLE 1 R8**  
Facteur de rotation angulaire en degrés. Lors d'un appel de tableau, ce facteur est appliqué globalement au tableau (par rapport à son point origine), puis à chaque élément individuel du tableau (par rapport à son point origine local).
- 31 REFLIBS n chaînes Ascii**  
Nom des bibliothèques externes utilisées. Ce type d'enregistrement n'est pas employé car toutes les structures référencées sont supposées décrites dans le même fichier.
- 32 FONTS n chaînes Ascii**  
Nom des fichiers définissant les différentes polices de caractères. Chaque nom comporte 44 caractères.
- 33 PATHTYPE 1 I2**  
Indique la façon dont se termine un chemin. S'il est égal à 1, une demie largeur est ajoutée en fin de chemin. S'il est égal à 0, le chemin se termine au niveau du dernier sommet. Ceci est représenté par le schéma suivant :
- The diagram shows two horizontal lines. The first line, labeled '0', ends with a vertical bar. The second line, labeled '1', ends with a vertical bar that has a small horizontal extension to the right, representing a half-width addition.
- 34 GENERATIONS 1 I2**  
Nombre de copies
- 43 PROPATTR 1 I2**  
Numéro d'attribut compris entre 1 et 127. Seul l'attribut 64 annonçant le nom d'un polygone ou d'un chemin est pris en compte.
- 44 PROPVALUE 1 chaîne Ascii**  
Chaîne (comportant au plus 126 caractères) associée à l'attribut spécifié dans le précédent enregistrement PROPATTR. Dans le cas de l'attribut 64, cet enregistrement contient le nom d'un élément.

### 3.4 - La syntaxe CALMA

Le fichier est organisé en une bibliothèque composée d'un ensemble de structures, chacune d'elles étant nommée et comportant une série d'éléments de différents types. Un élément peut être :

\* Un polygone : il est défini par le niveau auquel il appartient et par les coordonnées de chacun de ses sommets. Le premier et le dernier sommet sont confondus.

Par exemple, le polygone représenté en figure 3.2 est décrit par l'ensemble de sommets suivant : { P1, P2, P3, P4, P5, P6, P1 }

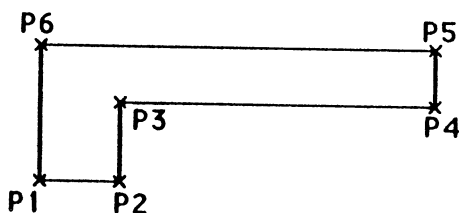


Figure 3.2 - Représentation d'un polygone

\* Un chemin : il est défini par

- l'ensemble de ses sommets,
- une largeur (répartie de part et d'autre du chemin),
- le niveau auquel il appartient,
- un type spécifiant la façon dont il se termine.

Un chemin est schématisé en figure 3.3

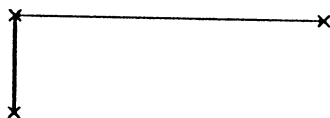


Figure 3.3 - Représentation d'un chemin

\* Un appel simple d'une seule structure . Il comporte :

- le nom de la structure appelée,
- les opérations à appliquer à la structure (rotation, grossissement, réflexion par rapport à l'axe des x),
- les coordonnées du point où placer l'origine de la structure.

\* Un appel d'un tableau de structures . Un tel appel comprend :

- le nom de la structure appelée,
- les opérations à appliquer au tableau et à ses différents éléments,
- le nombre de colonnes et de lignes du tableau,
- les trois points définissant le tableau.

Un tableau est représenté en figure 3.4

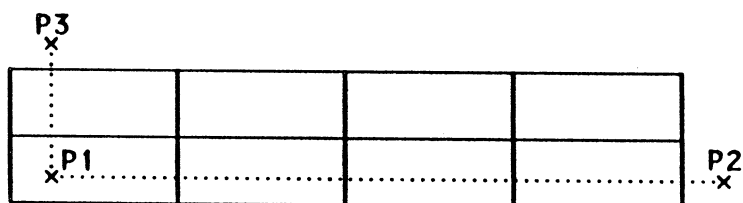
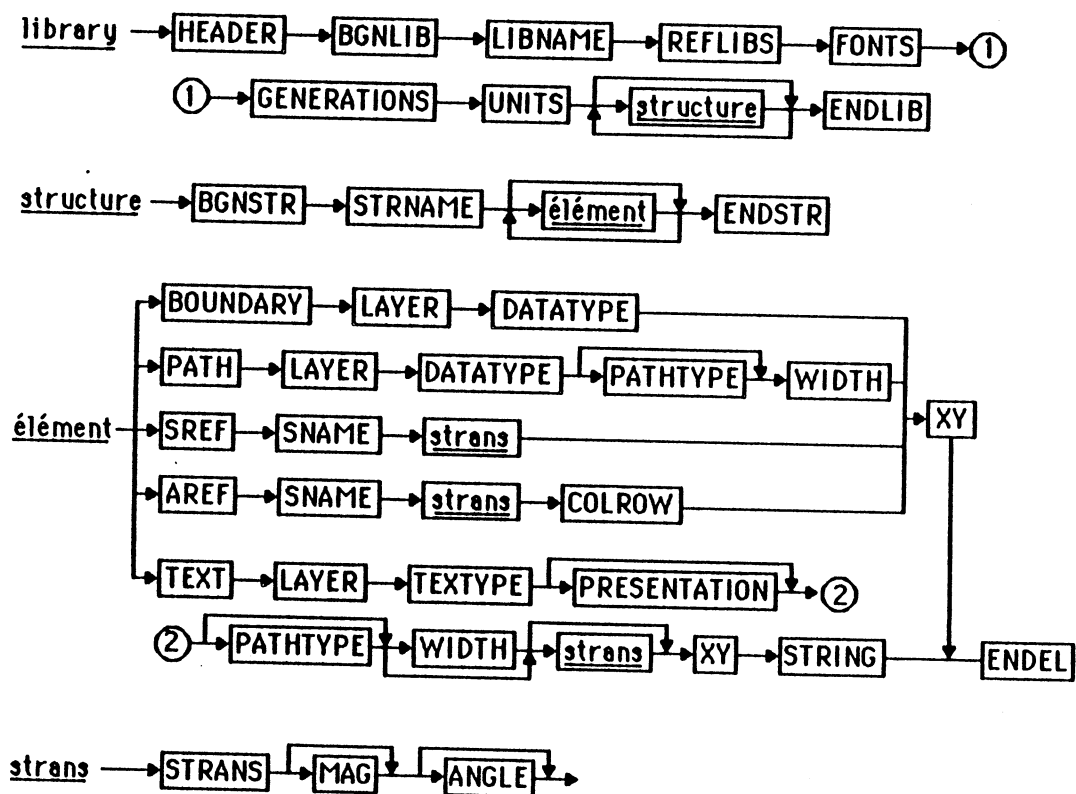


Figure 3.4 - Représentation d'un tableau

\* Un texte. Il s'agit d'un commentaire ajouté au dessin, il comprend :

- le libellé du commentaire,
- le point à partir duquel il doit être placé,
- le niveau auquel il appartient,
- des informations de mise en forme,
- la police de caractères utilisée.

Cette syntaxe est schématisée en figure 3.5

Figure 3.5 - Syntaxe CALMA



## 4 - MISE AU POINT DE CIRCUIT PROTOTYPES

### 4.1 - Présentation du problème

Il s'agit d'obtenir le premier bon circuit. Le processus de conception d'un circuit est représenté figure 4. 1.

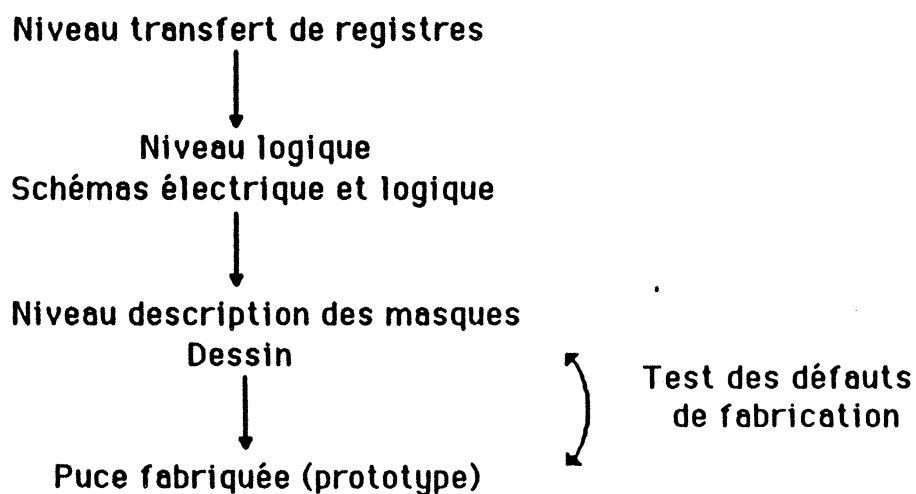


Figure 4. 1 - Etapes de la conception d'un prototype

Deux problèmes distincts existent :

#### 1) Les défauts de fabrication (Passage du dessin au prototype)

Exemples : - impuretés dans les matériaux,

- inhomogénéité du substrat,
- mauvais alignement des masques,
- coupures, court-circuits.

## 2) Les défauts de conception

Par abus de langage cette rubrique regroupe l'ensemble des erreurs imputables au concepteur, à savoir :

### \* Les défauts fonctionnels

(Passage du cahier des charges aux schémas électrique et logique)

Ils proviennent d'une mauvaise transcription du cahier des charges. Dans ce cas, le circuit fonctionne, mais ses réponses ne correspondent pas aux résultats escomptés. Dans la suite de ce chapitre, de telles erreurs sont supposées résolues. (Les schémas électrique et logique sont considérés corrects).

### \* les défauts de conversion

(Passage des schémas électrique et logique au dessin)

Exemples : - non respect des règles de dessin,

- non respect de règles d'implémentation :

- . omission de connexions,
- . une porte ET dessinée comme une porte OU,
- . une connexion de X à Y au lieu de X à Z.

## 4.2 - Test des défauts de fabrication

Une première étape consiste à vérifier si la puce est bien la réalisation physique de la description des masques à partir de laquelle elle a été produite, c'est à dire si le dessin a été correctement fabriqué. Trois niveaux d'intervention sont possibles :

- Les tests "technologiques", application du MEB dite caractérisation permettent au fondeur de vérifier la qualité des matériaux (homogénéité, épaisseur des dépôts...), et de relever les paramètres physiques caractéristiques de la technologie utilisée. Ces tests sont le plus souvent effectués sur des motifs prévus à cet usage.

- Le "contrôle dimensionnel" est une opération généralement réalisée sur les masques et/ou sur les résines, à chaque phase de la fabrication. Les dimensions prévues sont vérifiées.

- L'inspection, manuelle ou automatique, a pour but de contrôler la conformité des réalisations (masques et circuit) à leurs descriptions originales. A ce niveau, peuvent être détectés des défauts tels que mésalignement, non respect des règles de dessin ou encore amorces de rupture, court-circuits, etc...

- Les tests fonctionnels sont réalisés dans une seconde étape à l'aide des données fournies par le concepteur :

- \* Tests "statiques" pour contrôler les caractéristiques électriques et physiques (tenue en tension, en température, etc ...).

- \* Tests "dynamiques" conduits par des vecteurs de test spécifiques, ils attestent du bon fonctionnement du circuit.

Si des défauts sont présents la méthode suivante, dont le schéma est représenté en figure 4. 2, peut être appliquée :

- 1) Extraire le schéma électrique de la description des masques, puis tirer le schéma logique du schéma électrique. Comparer les résultats aux schémas originaux et simuler le fonctionnement sur des points considérés comme "sensibles".

- 2) Effectuer une comparaison en certains points entre la valeur générée par le circuit (obtenue avec le MEB) et celle issue des simulations.

Le nombre de points de comparaison et le nombre de simulations différentes à effectuer afin d'avoir une forte probabilité de prendre la bonne décision devraient être déterminés à l'aide d'expérimentations.

Les points de comparaison sont tels que leur valeur simulée puisse être obtenue facilement, et tels qu'ils soient facilement observables au MEB (points d'interconnexions métalliques).



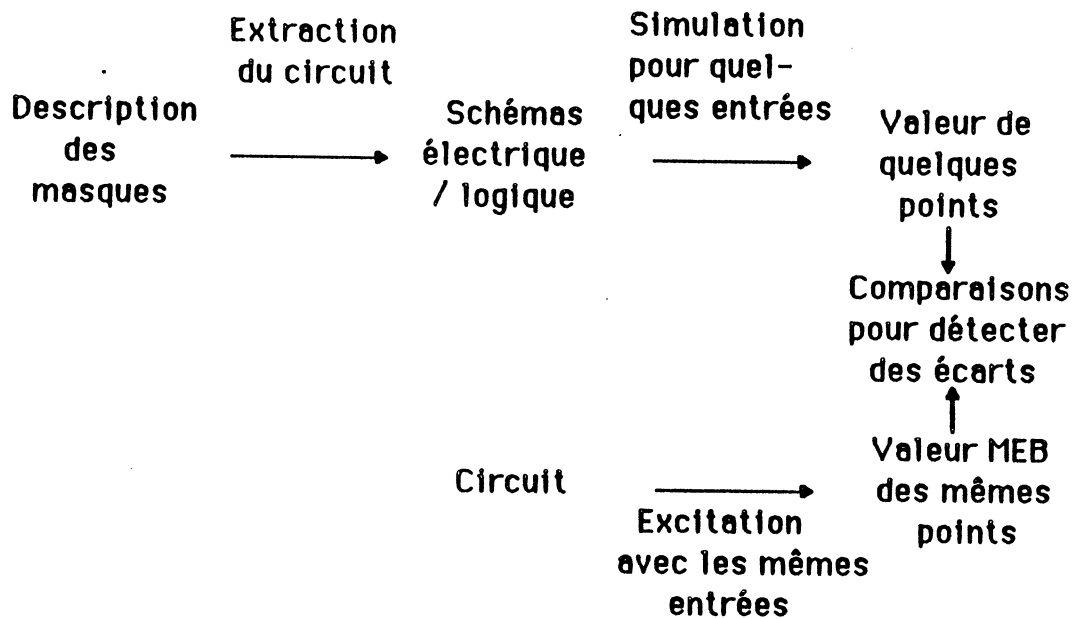


Figure 4. 2 - Test des défauts de fabrication

Un prototype ayant réussi ce test est donc bien la réalisation physique de la description des masques et est prêt pour l'étape suivante.

### 4. 3 - Test des défauts de conception

Il s'agit de vérifier le passage des schémas électrique et logique au dessin. Cette opération est nécessairement effectuée par le concepteur, et ne concerne que les circuits pour lesquels ce passage n'est pas réalisé de façon automatique par un outil CAO.

Cette vérification est rendue difficile par le manque de modèles représentant les défauts de conception. En effet, certains défauts ne peuvent être modélisés par les modèles classiques (modèles représentant les fautes de collage, de court-circuit, etc...) [COU 82].

La suite de ce chapitre donnera :

- une méthode pour localiser les blocs défectueux,
- des méthodes pour localiser à l'intérieur d'un bloc des défauts modélisables,
- une aide pour localiser à l'intérieur d'un bloc des défauts non modélisables.

#### 4. 3. 1 - Localisation d'un bloc défectueux

Un circuit est conçu comme un assemblage de blocs, chaque bloc étant constitué de cellules de base à contrôler. Un bloc est défectueux (contient un défaut de conception) s'il existe des erreurs sur ses interfaces de sortie et non sur celles d'entrée. Il s'agit donc de vérifier les interfaces entre les blocs. Pour cela, les valeurs aux interfaces (obtenues avec le MEB) sont comparées avec les valeurs déterminées à partir du schéma logique établi par le concepteur.

A l'issue de cette étape, un bloc défectueux a donc pu être identifié. Le problème est alors de localiser le défaut à l'intérieur du bloc représenté en figure 4. 3. La méthode employée pour le résoudre diffère si le défaut peut être ou non représenté par l'un des modèles disponibles.

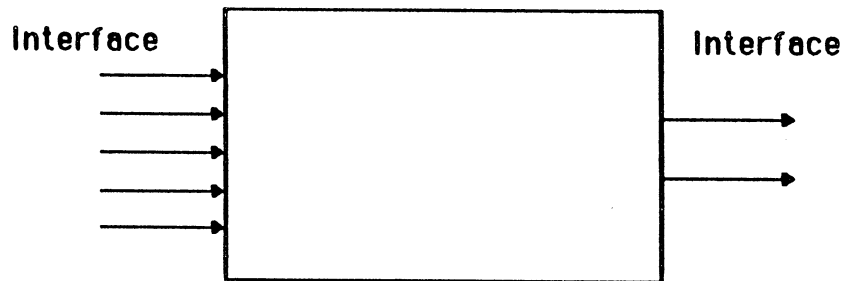


Figure 4. 3 - Le défaut de conception est à l'intérieur de ce bloc.  
Les entrées et les sorties sont les interfaces accédées par le MEB pour identifier le bloc défectueux.

#### 4. 3. 2 - Localisation à l'intérieur d'un bloc de défauts modélisables

Les deux méthodes suivantes peuvent être appliquées [BER 83] :

- sensibilisation / désensibilisation de chemins
- utilisation d'un résolveur automatique de problèmes

#### 4. 3. 3 - Localisation à l'intérieur d'un bloc de défauts non modélisables

Les modèles de panne disponibles, mal adaptés aux tests avec investigation du fonctionnement interne, ne peuvent représenter tous les défauts de conception [COU 82].

Des recherches sont nécessaires dans les domaines suivants :

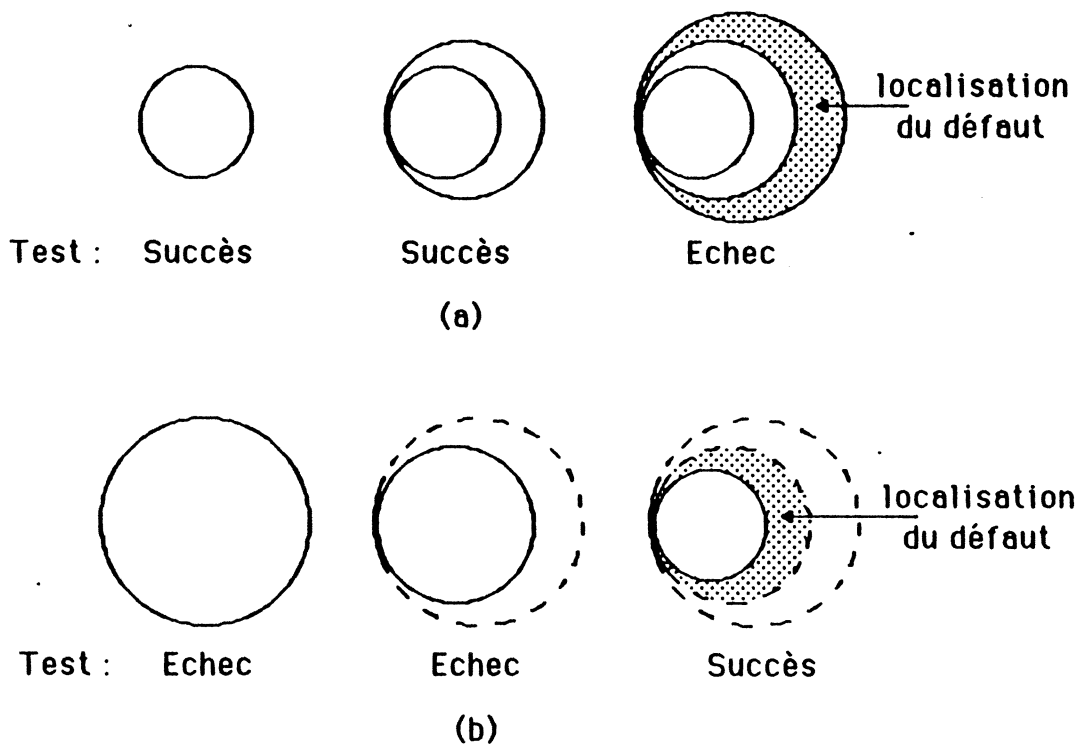
\* la modélisation des défauts de conception :

- comment les modèles de faute existants peuvent-ils représenter ces défauts ?
- quels modèles spécifiques doivent être créés ?

\* les méthodes de diagnostic basées sur ces nouveaux modèles.

Une possibilité consiste à restreindre progressivement la zone potentielle du défaut en testant des portions de bloc. Tester une zone consiste à comparer pour sa limite (constituée par des points similaires aux interfaces d'un bloc) les valeurs relevées avec le MEB, aux valeurs obtenues à partir du schéma logique du concepteur. Deux approches possibles conduisent aux méthodes représentées en figure 4. 4, nommées "start-big" et "start-small".

Les premiers résultats concernant la plus grande zone (le bloc), la méthode "start-big" est préférée à la méthode "start-small". Cependant si le bloc est constitué de cellules, au préalable testées par ailleurs, la méthode "start-small" peut se révéler plus rapide.



**Remarques :**

\* On suppose que les points interfaces existent à la fois sur le schéma logique et sur la puce. Ceci peut ne pas être vrai car de gros défauts peuvent exister.

\* Des tests exhaustifs sont théoriquement nécessaires pour les tests d'interfaces car on ne dispose pas de modèles de fautes.

## 5 - REALISATIONS

L'objet de ce chapitre est de décrire la réalisation de l'outil de base, c'est à dire le lien entre une description (CALMA) et le microscope; outil sur lequel des nouvelles méthodes de mise au point de circuits prototypes pourront être étudiées.

### 5. 1 - Transfert des programmes translatables sur ROMUALD

Le VAX supporte le code assembleur des programmes de ROMUALD ainsi que les utilitaires nécessaires à leur assemblage et édition de liens. Les programmes mis au point sont enregistrés en EPROM, mais durant leur phase de développement, leur code translatable doit être chargé en RAM depuis le VAX. Ce traitement est décrit ci-après.

#### 5. 1. 1 - Format des programmes à transférer

Les programmes sont codés suivant le format INTEL présenté ci-dessous.

##### \* format général

caractère d'en tête	longueur de l'enregis- trement	adresse de départ	type de l'enregis- trement	données	checksum
------------------------	--------------------------------------	----------------------	----------------------------------	---------	----------

\* Description :

Nom	Nombre de caractères ASCII	Description
Caractère d'entête	1	caractère ":"
Longueur de l'enregistrement	2	Deux chiffres hexa représentant le nombre d'octets de données. Ils sont égaux à zéro dans le dernier enregistrement
Adresse de départ	4	Quatre chiffres hexa représentant l'adresse du premier octet de données de l'enregistrement
Type de l'enregistrement	2	00 = enregistrement de données 01 = dernier enregistrement du fichier 02 = premier enregistrement du fichier
Données	2*N	N octets de données, chacun représenté par deux chiffres hexa
Checksum	2	Deux chiffres hexa représentant le "complément à 2" de la somme, modulo 256, des octets précédents de l'enregistrement

\* exemple : la figure 5.1 présente un programme en format Intel

```

:020000020000FC
:200000007F04340400C27F0F340400F87F0C340200F57F1091E4340400C47F0FD7E7EF03A
:200020007F087F0AE8F814028000F8AC0C28340400C87F1D7F1E0A083F3FEEFC340400C6A
:200040007F1095E4DFE4340200BC0C25FFFF9E08340400B37F0C21000010C90DBA440096F
:20006000E6028D81E80B0B00000FE1028D81E8068D02010000103208008C8D839E087F1EE
:200080000A082A2AEFC3402007CC6047F1E0A080D0DE6FC0A080A0AE6F90A082424E60A9
:2000A0002E28A930F60D3402005C7F0EE70C2F40A951E8E90C253030A930F60434020046E
:2000C0007F0EE7012F409E081561647265737365206368617267656D656E74203F20186E5
:2000E0006F6C206475206669636869657220612063686172676572203F200872756E20744
:0A01000072616E73690D8000F8ACA7
:00000001FF

```

Figure 5. 1 - Programme en format Intel

Explications relatives à la deuxième ligne:

- Longueur = 20 : l'enregistrement comporte 32 octets de données (soit 64 chiffres hexadécimaux)
- adresse de départ = 0000 : la première donnée (7F) se trouve à l'adresse 0000 de la mémoire du programme. Les octets suivants (04, 34, 04, ...) seront chargés aux adresses 0001, 0002, ...
- type = 00 : il s'agit d'un enregistrement de données
- données = 7F...EF03 : paires de chiffres hexadécimaux représentant les octets de données de l'enregistrement.
- checksum = A6 : "complément à 2" de 5A qui correspond à la somme hexadécimale suivante :  $20 + 00 + 00 + 00 + 7F + 04 + \dots + EF + 03 = 5A$

5. 1. 2 - Programmes développés

Deux programmes ont été mis en place l'un sur ROMUALD, l'autre sur le VAX.

\* Programme ROMUALD recevant le code et le rangeant en mémoire

Il effectue :

- la saisie de l'adresse à partir de laquelle doit être chargé le programme,
- la saisie du nom du programme à transférer,
- le passage en mode transparent (dans ce mode tout ce que ROMUALD reçoit du VAX est envoyé au terminal),
- le lancement du programme VAX de transfert,
- l'attente d'un '?' que le VAX transmet lorsqu'il est prêt à recevoir le nom du programme à transférer,
- l'envoi de ce nom au VAX,
- l'attente d'un astérisque indiquant le début du transfert,



- le rangement du programme en mémoire à partir de l'adresse saisie. Lors de la réception d'un caractère, le programme teste s'il s'agit d'un '\$'. Si c'est le cas, il sait que la totalité du programme a été transférée. En effet, le VAX envoyant un '\$' à la fin de l'exécution d'un programme, ce caractère indique la fin de transmission. ROMUALD reçoit le code ASCII hexadécimal des caractères . Ces codes sont transformés en la valeur des caractères, regroupés deux à deux en octets, puis rangés en mémoire,
- le retour au mode normal.

\* Programme VAX transférant le fichier

Il est lancé par le programme ROMUALD précédent, et effectue :

- l'envoi d'un '?' indiquant qu'il attend le nom du programme à transférer,
- la lecture de ce nom,
- l'envoi d'un astérisque indiquant qu'il va transférer le programme,
- le transfert du programme. Lors du parcours séquentiel du fichier, le type de chaque enregistrement est testé :

\* S'il ne s'agit pas du dernier enregistrement, ses données sont transférées. Les autres informations (entête, longueur, adresse, type et checksum) ne sont pas transmises.

\* Sinon, le transfert est terminé. Le VAX envoie un '\$' et se met en attente d'une commande.

## **5.2 - Détermination des coordonnées théoriques des points à observer**

### **5.2.1 - Identification d'un polygone**

Pour observer un polygone au microscope, il faut indiquer son nom au système. Ceci n'est pas suffisant pour identifier de façon unique une occurrence d'un polygone dans les deux cas suivants :

\* La structure à laquelle il appartient est appelée en différents endroits du circuit. Le polygone apparaît alors plusieurs fois et il faut également indiquer l'appel à prendre en compte. Afin de distinguer ces différents appels, le système génère un numéro d'ordre. Ultérieurement, un nom d'appel affecté par l'utilisateur remplacera ce numéro.

\* La structure à laquelle il appartient est appelée sous forme d'un tableau de "l" lignes et "c" colonnes. Afin d'observer l'une de ces l x c occurrences, les numéros de ligne et de colonne doivent être indiqués.

L'identification complète d'un polygone comporte donc :

- . le nom du polygone,
- . le numéro de l'appel,
- . le numéro de la ligne (si appel d'un tableau),
- . le numéro de la colonne (si appel d'un tableau).

#### **Exemple :**

Soit un circuit appelant trois fois la structure représentée en figure 5.2, soit : deux appels simples et un appel sous forme d'un tableau de quatre lignes et deux colonnes (figure 5. 3).

Le polygone hachuré à été caractérisé par un nom. Afin d'observer l'une de ses dix occurrences, il faut indiquer son nom et le numéro d'appel. Pour un appel simple (appels 1 ou 3), ces deux informations sont suffisantes pour déterminer les coordonnées du point à observer. Mais pour une occurrence appartenant au tableau, les numéros de ligne et de colonne doivent également être indiqués.

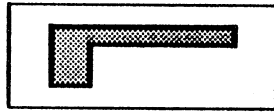


Figure 5. 2 - Structure appelée

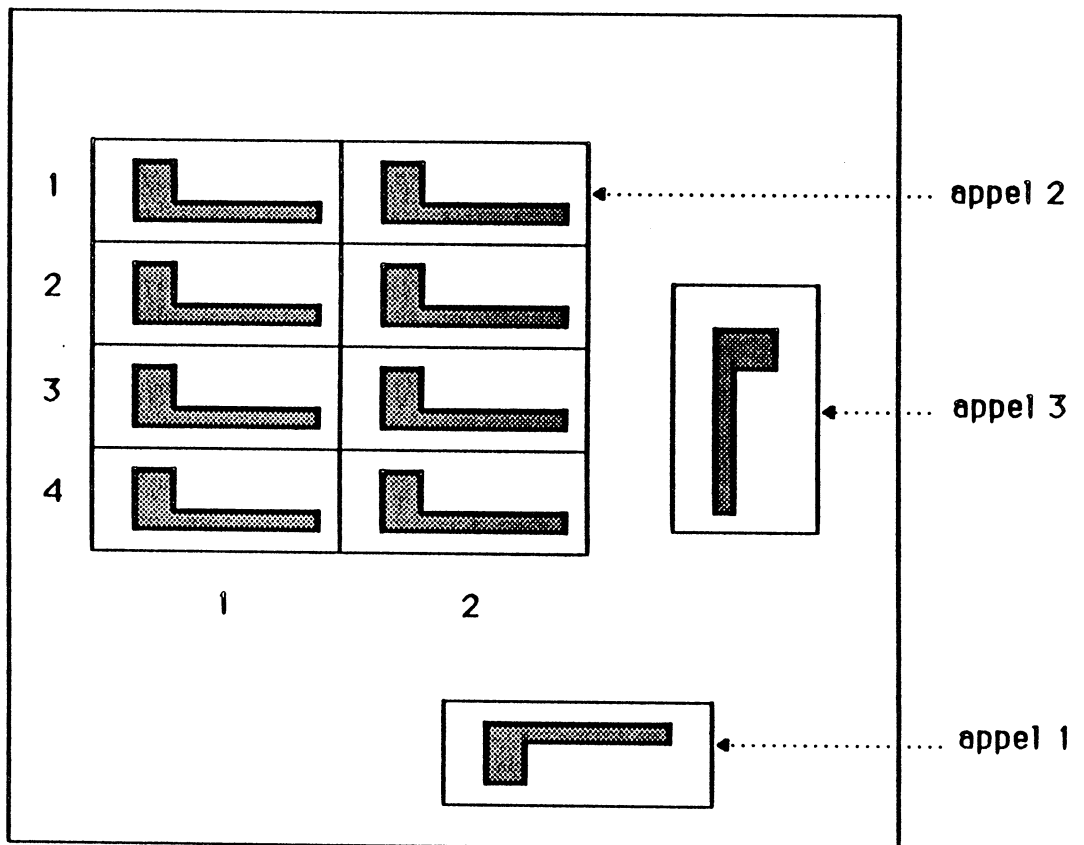


Figure 5. 3 - Exemple de multiples occurrences d'un polygone

## 5. 2. 2 - Programme de traduction du fichier de base

### 5. 2. 2. 1 - Présentation générale

Il s'agit d'un traitement batch, exécuté une seule fois, effectuant la mise en forme des données c'est à dire la traduction du fichier de base en un ensemble de fichiers. Pour cela, il travaille sur les fichiers suivants :

\* En entrée : le fichier dump contenant la description CALMA du circuit.

\* En sortie : les cinq fichiers suivants :

- un fichier des instructions CALMA,
- un fichier des sommets des polygones,
- un fichier des structures,
- un fichier des noms des polygones,
- un fichier des coordonnées du point à observer à l'intérieur de chaque

occurrence de tout polygone nommé lors de la conception du circuit.

L'algorithme de ce programme est décrit en annexe 2.1.

### 5. 2. 2. 2 - Description des fichiers

#### **a) Fichier de base**

Il s'agit du fichier dump hexadécimal de format CALMA contenant la description du circuit. Un exemple est représenté en figure 5.4.. Il est composé d'enregistrements de 2048 octets et comprend :

\* des lignes d'entête au début de chaque page,

\* des lignes d'entête au début de chaque enregistrement,

\* des lignes de données comportant chacune :

- l'adresse de son premier octet dans l'enregistrement auquel il appartient,
- la représentation ASCII de 32 caractères,
- la représentation hexadécimale de ces 32 caractères qui sont formatés en

huit mots de quatre octets et constituent l'information utile du fichier.

Ces données sont lues de droite à gauche, mais les deux chiffres hexadécimaux formant un caractère sont lus de gauche à droite.

Par exemple, la séquence "1C2B0300 020A0601" est lue "01060A02 00032B1C".

DUMP OF FILE SYS:SYSDEVICE:CLBG.IGUIGUETJCIRCLIT1.SOU:1 ON 23-JUL-1985 13:37:14.99  
FILE ID (2064,4,0) END OF FILE BLOCK 66 / ALLGATED 66

0C000000 C11A0600 5248535F E3554C50 06120C00 000A0400 5248535F 515F5355 US\_Q\_SHR.....PLUS\_SHR..... 00  
0C00000A C4C00011 04002CC9 000048E8 01000310 0CC00000 00000000 E442051C ..B4.....MH..I..... 00  
03100C0C 00C00000 00008442 051C0C00 0090011A 06005248 535F4353 41420E12 ..BASC\_SHR.....B4..... 00  
03104C0C C0C0020E 06000800 C20DC000 00080400 00110400 DC050000 E4DC0C00 ..\4.....\4.....\4.....Zã...H..Zã 00  
E4D4C000 0C110100 E4D40000 0C8EC100 5CC10000 0C8E0100 5CC10000 E8930100 ..H..A\.....A\.....TC.....TC 00  
C0DA000C E8930100 C0CA0000 CC18C100 84DC0000 DC180100 84CC0000 0C110100 .....4.....\4.....\4.....Zã...H..Zã 00  
03104C00 00C0020E 06000800 C20D0600 00080400 00110400 5CC10000 E8930100 ..H..A\.....A\.....TC.....TC 00  
20670000 54F2FFFF F06E0000 54F2FFFF F06E0000 48F4FFFF 78E60000 48F4FFFF ..TH..FX..TH..NF..RT..NP..RT..G 00  
78E60000 24FAFFFF 00000000 24FAFFFF 00000000 48F4FFFF 20670000 48F4FFFF ..TH..G ..TH.....Zã.....Zã..FX 00  
03103C0C G0C0020E 06000800 C20DC000 00080400 0C110400 78E60000 48F4FFFF ..TH..FX.....Zã.....Zã< 00  
C0DA0000 E0C30100 C0DA0000 8C34C000 E4D40000 8C340000 E4D40000 30080100 ..O..TC..4<..TC..4<..Zã...ã..Zã 00  
0008040C C0110400 E4C40000 30C8C100 84DC0000 30080100 84CC0000 E0030100 .....ã\4.....0.....\4.....0.....TC..... 00  
08D6000C 0G130100 C0CA0000 0013C100 02102C00 0G0G020E 06000600 C20D0600 .....ã\4.....0.....\4.....0.....TC..... 00  
0008040C C0110400 C0CA0000 0013C100 C0DA0000 E2160100 09D60000 E6160100 ..H..VX...ã..Zã.....Zã.....VX 00  
08D60000 54C50100 C0DA0000 54050100 03102C00 0000020E 06000600 C20D0600 .....ã\4.....0.....\4.....0.....TC..... 00  
00080400 00110400 C0DA0000 2805C10C C0DA0000 3C090100 08C60000 3C090100 .....ã\4.....0.....\4.....0.....TC..... 00  
ACC5C00C 28040100 ECDB0000 2804C100 03104C00 00C0020E 06000500 C20C0600 .....ã\4.....0.....\4.....0.....TC..... 00  
54BA0000 28040100 94C00000 2804C100 94C00000 48C70100 ACD50000 48070100 ..H..U,..F..ã.....(..ã.....(..: 00  
00080400 C0110400 ECDB0000 2804C100 ECDB0000 68CA0100 54BA0000 680A0100 ..H...:..F..EL...C..EL..... 00

RECORD NUMBER 3 (00000003), 2048 (0800) BYTES

ACD50000 14180100 ECC80000 1418C100 03104C00 0GC020E 06000500 C20D0600 .....L.....CL.....U, 00  
78B40000 74C80100 58910000 74080100 58B10000 F4140100 ACC50000 F4140100 ..T..U,..T..1X..T..1X..T..4X 00  
0008040C G0110400 ECC80000 14180100 ECDB0000 D4110100 78B40000 C4110100 .....T..4X...T..CL.....CL.....# 00  
ACA30000 ECC80000 CCA60000 ECC60000 03104C00 0CC0020E 06000500 C20D0600 .....L.....CL.....CL.....CL.....# 00  
78B40000 74C80100 58B10000 7408C100 58B10000 9CF90000 ACA30000 F0EBC000 ..KP..#..Y...1X...T..1X...T..4X 00  
0008040C C0110400 CCA60000 ECC60000 CCA60000 60EA0000 78B40000 0CF60C00 ..X...4X..Jã..EL..CL..EL..... 00  
6C6E0000 3CF00000 48710000 3CF00000 03104C00 0G0G020E 06000800 C20D0600 .....ã\4.....0.....\4.....0.....TC..... 00  
9C630000 C8E10000 CC580000 C8E10C00 CC580000 98E90000 6C680000 98E90000 ..I...KL...I...CL..AH..CL..AH..C. 00  
0008040C C0110400 48710000 3CF00000 48710000 BCE30000 9C630000 BCE30000 ..C<..C<..C<..QF..P<..QF..... 00  
A8610000 ECE30000 C0500000 ECE30000 03102C00 0000020E 06000600 C20D0600 .....ã\4.....0.....\4.....0.....TC..... 00  
0008040C G0110400 C0500000 ECE30000 C0500000 A4E70000 A8610000 A4E70000 ..G\$..A<..G\$..Jã..C<..C<..A< 00  
182E0000 90E20000 D4620000 90E2C000 03104C00 0G0G020E 06000500 C20D0600 .....ã\4.....0.....\4.....0.....TC..... 00  
945C0000 E0E50000 D8270000 E0E5C000 D8270000 70DF0000 182E0000 70DF0000 ..P.....F...X..E0...X..E0... 00  
0008040C C0110400 D4620000 90E20000 D4620000 DGE80000 945C0000 C0E80000 ..HP...ã\4.....ã\4.....ã\4.....ã\4..... 00  
EC2C0000 SCE00000 04290000 9CE0C000 03102C00 0G0G020E 06000600 C20D0600 .....ã\4.....ã\4.....ã\4.....ã\4..... 00  
0008040C C0110400 04290000 9CE0C000 04290000 94E40000 EC2C0000 E4E40000 ..D...L...D...ã\4.....ã\4.....ã\4..... 00  
10270000 0CCF0000 E02E0000 0CCF0000 03102C00 0G0G020E 06000800 C20C0600 .....ã\4.....ã\4.....ã\4.....ã\4..... 00  
0008040C 00110400 E02E0000 0CDF0000 E02E0000 78E60000 10270000 78E60000 ..FX...'..FX...ã\4.....ã\4.....ã\4..... 00  
14690000 48F4FFFF FC6C0000 48F4FFFF 03102C00 0GC020E 06000600 C20C0600 .....ã\4.....ã\4.....ã\4.....ã\4..... 00  
0008040C C0110400 FC6C0000 48F4FFFF FC6C0000 30F8FFFF 14690000 30F8FFFF ..XO..I...XC..L\..TH..L\..TH..I. 00  
E8670000 1CF3FFFF 286E0000 1CF3FFFF 03103C00 0000020E 06000500 02C00600 .....ã\4.....ã\4.....ã\4.....ã\4..... 00  
286E0000 5CF9FFFF 08690000 5CF9FFFF 08680000 FE2A0000 E9670000 F82A0000 ..X..GH..X..K...Y\..K...Y\..N< 00  
03104C00 00C0020E 06000800 C20DC000 00080400 00110400 286E0000 1CF3FFFF ..S...N<.....ã\4.....ã\4.....ã\4..... 00  
8C890000 10270000 8C890000 F82A0000 08CF0000 F82A0000 08CF0000 10270000 ..'...D...X...D...X...9...9... 00  
78E60000 10270000 79E60000 00000000 00000000 0CC00000 00000000 10270000 ..'.....ã\4.....ã\4.....ã\4.....ã\4..... 00  
3A0C340C 0ECC1800 08005400 02051C00 00070400 00110400 08CF0000 1G270000 ..'...D.....ã\4.....ã\4.....ã\4..... 00  
020C0600 C0C80400 5248535F 43534142 060E0C00 05001E00 11000300 09005400 ..T.....ã\4.....ã\4.....ã\4.....ã\4..... 00  
A00F0000 ACC00000 88080000 C409C000 88080000 03102C00 0000020E C6000600 .....ã\4.....ã\4.....ã\4.....ã\4..... 00  
020D0600 C0C80400 00110400 C4090000 880E0000 C4090000 A00F0000 AC0C0000 .....ã\4.....ã\4.....ã\4.....ã\4..... 00  
801D0000 ACC00000 C8190000 C409C000 C8190000 03102C00 0000020E 060C0600 .....ã\4.....ã\4.....ã\4.....ã\4..... 00  
020D0600 00080400 00110400 C4090000 C8190000 C4090000 801D0000 AC0D0000 .....ã\4.....ã\4.....ã\4.....ã\4..... 00  
C02E0000 ACC00000 D8270000 C409C000 D8270000 03102C00 0G0G020E C6000600 .....ã\4.....ã\4.....ã\4.....ã\4..... 00  
020C0600 00C80400 00110400 C409C000 D8270000 C4090000 C02B0000 ACC0C000 .....ã\4.....ã\4.....ã\4.....ã\4..... 00  
0039G00C ACC00000 E8350000 C409C000 E8350000 03102C00 0000020E C6000600 .....ã\4.....ã\4.....ã\4.....ã\4..... 00  
020C0600 C0C80400 00110400 C409C000 E8350000 C4090000 00390000 AC0D0000 .....ã\4.....ã\4.....ã\4.....ã\4..... 00

Figure 5. 4 - Extrait d'un fichier dump CALMA

Ce fichier est uniquement utilisé par le programme de traduction qui l'ouvre en lecture. La procédure le lisant ne doit prendre en compte que les 32 caractères codés en hexadécimal de chaque ligne. Ne sont pas traités les lignes entêtes, la représentation ASCII, l'adresse dans l'enregistrement et les espaces entre mots de chaque ligne de données.

### **b) Fichier des Instructions CALMA**

\* Il contient la description de la composition de chaque structure.

\* Il est en accès direct.

\* Un enregistrement occupe quatre mots de 32 bits (numérotés b31...b0) de type entier. La signification des différents champs dépend du type de l'enregistrement. Les différents types d'enregistrement sont les suivants :

#### **1) Enregistrement indiquant le début d'une structure**

<b>N° mot</b>	<b>Description</b>
1 : b20 - b 31	Type de l'enregistrement (égal à 6).
1 : b10 - b19	Numéro de la structure dont la description débute. Il s'agit du numéro de l'enregistrement auquel elle correspond dans le fichier des structures.
autres bits	Inutilisés et mis à zéro

#### **2) Enregistrement décrivant un polygone**

<b>N° mot</b>	<b>Description</b>
1 : b20 - b31	Type de l'enregistrement (égal à 8).
1 : b10 - b19	Numéro du niveau auquel appartient le polygone (métal,...).
1 : b0 - b9	Inutilisés et mis à zéro.

- 2 Numéro de l'enregistrement du fichier des sommets qui contient les coordonnées de son premier point.
- 3 Nombre de sommets qu'il comporte
- 4 Numéro de l'enregistrement du fichier des noms qui contient son nom. (Il est égal à zéro si le polygone n'est pas nommé).

### 3) Enregistrements décrivant un appel

Un seul enregistrement permet de décrire un appel simple. Un deuxième enregistrement est ajouté pour un appel de tableau.

#### - Premier enregistrement

<b>N° mot</b>	<b>Description</b>
1 : b20 - b31	Type de l'enregistrement (10 pour un appel simple, 11 pour un appel de tableau).
1 : b10 - b19	Numéro de la structure appelée.
1 : b1 - b9	Angle représentant la rotation à appliquer à la structure.
1 : b0	Indicateur de réflexion par rapport à l'axe des x. Il est égal à 1 si la structure doit être réfléchi, à 0 sinon.
2	Abscisse du point où placer l'origine de la structure.
3	Ordonnée de ce point. Pour un tableau, il s'agit de l'origine d'une occurrence située dans l'un des quatre coins.
4 : b10 - b31	Facteur de grossissement à appliquer à la structure. Ce nombre pouvant être décimal, il est multiplié par 1000 avant d'être rangé dans l'enregistrement.
4 : b0 - b9	Numéro d'appel de la structure.

- Deuxième enregistrement

Il contient les dimensions d'un tableau :

N° mot	Description
1	Nombre de colonnes du tableau.
2	Nombre de lignes du tableau.
3	Largeur d'une colonne.
4	Largeur d'une ligne.

4) Enregistrement indiquant la fin d'une structure

N° mot	Description
1 : b20 - b31	Type de l'enregistrement (égal à 7).
autres bits	Inutilisés et mis à zéro.

\* Organisation des différents enregistrements dans le fichier

Le fichier contient une suite de descriptions de structure, chacune étant organisée comme suit :

Type enregistrement	Description
6	Début de structure
8	Polygones de la structure
⋮	
8	
10 ou 11	Appels de la structure
⋮	
10 ou 11	
7	Fin de structure



Ce classement par type d'éléments (polygones puis appels) permet un important gain de temps lors du dessin des contours des structures.

Remarque : tout chemin est transformé en un polygone.

\* Utilisation du fichier

- Il est créé par le programme de traduction,
- il est lu par les programmes de traduction, d'affichage et de calcul des points d'observation,
- il est modifié par le programme d'affichage.

c) Fichier des sommets des polygones

- \* ce fichier est en accès direct,
- \* il contient un enregistrement pour chaque sommet de tout polygone,
- \* Un enregistrement occupe deux mots de type entier ; sa structure est la suivante :

<b>N° mot</b>	<b>Description</b>
1	Abscisse d'un sommet.
2	Ordonnée de ce point.

\* Utilisation du fichier :

- il est créé par le programme de traduction,
- il est lu par les programmes d'affichage et de calcul des points d'observation.

**d) fichier des structures**

- \* il est en accès direct sur un numéro de structure,
- \* il contient un enregistrement par structure,
- \* organisation d'un enregistrement :

<b>N° mot</b>	<b>Type</b>	<b>Description</b>
1	entier	coordonnée x maximum - coordonnée x minimum
2	entier	coordonnée y maximum - coordonnée y minimum
3	entier	coordonnée x minimum
4	entier	coordonnée y minimum
		Ces quatre valeurs indiquent l'encombrement de la structure. Elles tiennent compte des appels qu'elle effectue.
5	entier	Adresse du premier élément (polygone ou appel) de la structure dans le fichier des instructions CALMA.
6 : b1-b31	entier	Adresse du premier appel de la structure dans le fichier des instructions. Si la structure n'effectue aucun appel, il s'agit de l'adresse de l'enregistrement marquant la fin de la structure.
6 : b0	entier	Indicateur spécifiant si la structure appelle ou non d'autres structures. Il est égal à 1 s'il s'agit d'une structure englobante, à 0 sinon.
7-14	caractère	Chaîne de 32 caractères contenant le nom de la structure

- \* Utilisation du fichier
  - il est créé par le programme de traduction,
  - il est lu par les programmes d'affichage et de calcul des points d'observation.

e) Fichier des noms de polygones

- \* il est en accès direct sur un numéro de nom,
- \* il contient un enregistrement par nom d'élément,

\* structure du premier enregistrement :

N° mot	Type	Description
1	entier	Numéro du dernier enregistrement rangé dans le fichier. Cette information est utilisée lors de l'ajout de nouveaux noms.
2	entier	Numéro du dernier enregistrement du fichier pour lequel le calcul des points d'observation a été effectué.
3-8		Inutilisés.

\* structure des enregistrements suivants :

N° mot	Type	Description
1-8	caractère	Chaine de 32 caractères contenant un nom de polygone

\* Utilisation du fichier :

- il est créé par le programme de traduction,
- il est lu et modifié par les programmes d'affichage et de calcul des points d'observation.

### **f) Fichier des points d'observation**

\* il est en accès direct,

\* un enregistrement est créé pour chaque occurrence de tout polygone nommé. Il contient les coordonnées du point à observer et occupe sept mots de type entier. Ce fichier permet de ne pas afficher les masques des connexions à analyser.

\* Structure du premier enregistrement :

<b>N° mot</b>	<b>Description</b>
1	Numéro du dernier enregistrement rangé dans le fichier.
2-7	Inutilisés.

\* Structure des enregistrements suivants :

<b>N° mot</b>	<b>Description</b>
1 : b12 - b31	Numéro du polygone (il s'agit du numéro de l'enregistrement du fichier des noms qui contient le nom de cet élément).
1 : b0 - b11	Numéro de l'appel du polygone.
2	Abscisse du point à observer.
3	Ordonnée de ce point. Pour un appel de tableau, il s'agit du point se trouvant dans l'occurrence inférieure gauche.
4	Largeur d'une colonne du tableau.
5	Largeur d'une ligne du tableau.
6 : b16 - b31	Nombre de lignes du tableau.
6 : b0 - b15	Nombre de colonnes du tableau.
	Ces quatre dernières informations, égales à zéro pour un appel simple, permettent de calculer le point à observer à l'intérieur de chaque occurrence d'un tableau.
7	Numéro de la structure englobante contenant ce point.

\* Utilisation du fichier :

- il est créé par le programme de traduction,
- il est lu par les programmes d'affichage et de calcul des points d'observation
- il est modifié par le programme de calcul des points d'observation.

### 5. 2. 2. 3 - Types d'enregistrements CALMA pris en compte

Afin de réaliser la génération de fichiers désirée, tous les types d'enregistrements existants ne doivent pas nécessairement être traités. Les types pris en compte sont listés ci-dessous après classement par type de données.

\* Aucune donnée :

- 7 endstr
- 8 boundary
- 9 path
- 10 sref
- 11 aref
- 12 text : ce type est pris en compte bien que les commentaires figurant dans la description du circuit ne soient pas traités.
- 17 endel

\* Entier :

- 13 layer
- 15 width
- 16 xy
- 19 colrow
- 33 pathtype
- 43 propattr : un enregistrement de ce type n'est retenu que si l'entier qu'il contient est égal à 64.

\* Réel :

- 27 mag
- 28 angle : seules les valeurs 90, 180 et 270 sont traitées.

\* Chaines de bits : strans

\* Chaîne de caractères :

- 6 strname
- 18 sname
- 44 propvalue : un tel enregistrement n'est pris en compte que s'il contient le nom d'un polygone.

### 5. 2. 3 - Programme d'affichage

#### 5. 2. 3. 1 - Les commandes disponibles

Ce programme interactif permet :

- \* d'afficher le dessin des masques du circuit,
- \* d'affecter un nom à des polygones,
- \* de sélectionner les polygones à observer au microscope :
  - soit en indiquant leur identification,
  - soit en les désignant à l'aide du réticule après les avoir dessinés sur l'écran.

Afin de réaliser ces différentes opérations, les commandes suivantes sont disponibles :

- **LS** : liste le numéro et le nom de chaque structure présente dans la description du circuit.
- **ST s** : sélectionne la structure de travail, et affiche son nom.  
Le paramètre "s" correspond au numéro de cette structure. Cette commande a principalement pour effet de définir la zone du circuit à afficher comme étant égale à l'encombrement de la structure "s".
- **NI n1 [n2][n3]** : sélectionne le(s) niveau(x) de masques à prendre en compte. Chaque niveau est désigné par son numéro. Le nombre maximal de niveaux choisis simultanément est égal à trois, mais une valeur du premier paramètre permet de dessiner tous les niveaux des masques.
- **EN p** : dessine uniquement le contour des structures et définit la fenêtre à afficher comme étant égale à l'encombrement de la structure précédemment sélectionnée. Le paramètre "p" indique la profondeur à appliquer :
  - \* p=1 : dessin du contour de la structure de travail uniquement,
  - \* p=2 : dessin du contour de la structure de travail ainsi que du contour des structures qu'elle appelle. Incréments "p" de un permet de descendre d'un niveau dans la description de la structure de travail.
- **Pr xmin, xmax, ymin, ymax** : définit la fenêtre à afficher.  
Les paramètres xmin,...ymax sont des coordonnées absolues de la structure de travail.
- **GR** : affiche un réticule servant à désigner un ou deux points afin d'exécuter l'une des sous-commandes suivantes :
  - \* **zz** : désigne la partie du dessin à prendre en compte lors du prochain ordre d'affichage. La fenêtre à afficher est spécifiée en frappant un 'z' en chacun des deux points définissant son encombrement. Le résultat de cette opération est identique à celui de la commande PR précédemment décrite.
  - \* **?** : liste l'identification du polygone rempli pointé par le réticule. Si un polygone non rempli est désigné, aucune opération est effectuée.
  - \* **s** : la frappe d'un 's' en un point d'un polygone rempli affiche son identification et le sélectionne pour être observé au microscope.

Un astérisque est affiché à coté de l'identification pour indiquer que le polygone a été sélectionné. Il permet de distinguer cet affichage de celui obtenu avec la sous-commande précédente.

- **AF [nom-polygone]** : affiche le détail des polygones de la fenêtre sélectionnée. Cette sélection est effectuée par les commandes ST, EN, PR ou GR. Seuls les polygones appartenant au(x) niveau(x) de masques sélectionné(s) sont tracés. Si le nom d'un polygone est spécifié, chaque occurrence est remplie lors de l'affichage.
- **RE nom-polygone** : remplit chaque occurrence du polygone spécifié dans la fenêtre sélectionnée, et calcule pour chacune d'elles les coordonnées du point qui sera analysé si elle est sélectionnée pour être observée à l'aide du microscope.
- **LI** : liste les polygones choisis pour être observés au microscope. Pour chaque sélection, l'identification du polygone et les coordonnées du point à observer sont affichées.
- **SU Identification polygone** : supprime l'occurrence indiquée de la liste des sélections. Si les numéros d'appel, de ligne et de colonne ne sont pas précisés, ils sont considérés égaux à zéro.
- **RZ** : supprime toutes les entrées de la liste des sélections.  
 Cette commande s'est avérée utile car lors du lancement du programme, les sélections effectuées au cours de l'exécution précédente sont actives.
- **ID [nom-polygone]** : affiche un réticule utilisé pour désigner l'un des polygones dessinés. (Un 'm' est frappé en chacun des deux points définissant son encombrement). Si ce polygone est nommé, son nom est affiché. De plus, si un nom est spécifié dans la commande, il est affecté au polygone désigné. Préalablement, une confirmation est demandée à l'opérateur si le polygone est déjà nommé. Il est ainsi possible de :
  - \* retrouver le nom d'un polygone affiché,
  - \* renommer un polygone,
  - \* nommer un polygone qui n'a pas été identifié lors de la conception du circuit.



- **OB Identification polygone** : sélectionne l'occurrence indiquée pour être observée au microscope. Les coordonnées du point à observer sont lues dans le fichier des points d'observation (décrit p 75). Si aucun enregistrement ne correspond à l'identification spécifiée, un message d'erreur est affiché. Cette commande dont le résultat est identique à celui de la troisième fonction de la commande GR, permet de sélectionner un polygone sans l'afficher au préalable.
- **RR** : affiche le réticule pour désigner un, deux, ou trois points de recalage. Il s'agit des points permettant d'établir la correspondance entre le dessin des masques du circuit et l'image microscope. Après positionnement sur l'un de ces trois points, taper :
  - \* "o" pour désigner le point origine,
  - \* "h" pour désigner le point définissant le déplacement horizontal théorique,
  - \* "v" pour désigner le point définissant le déplacement vertical théorique.
- **RM  $x_0$ ,  $y_0$ ,  $\Delta x_t$ ,  $\Delta y_t$**  : définit les informations nécessaires au recalage, à savoir :
  - \* les coordonnées ( $x_0$ ,  $y_0$ ) du point origine,
  - \* le déplacement horizontal théorique ( $\Delta x_t$ ),
  - \* le déplacement vertical théorique ( $\Delta y_t$ ).

Cette commande est équivalente à la commande précédente. Les informations de recalage peuvent donc être données soit manuellement, soit à l'aide du réticule.
- **AU** : fin d'exécution du programme.

### 5. 2. 3. 2 - Exemples :

Les commandes permettant de sélectionner un polygone pour l'observer à l'aide du MEB sont présentées ci-après. Deux cas sont distingués :

\* polygone d'identification connue

Nom	numéro d'appel	ligne	colonne	Commentaire
Polygone 1	2	0	0	appel simple
Polygone 2	4	3	1	appel d'un tableau

Afin d'observer les deux polygones figurant dans le tableau précédent, les commandes suivantes doivent être frappées :

- OB polygone 1, 2
- OB polygone 2, 4, 3, 1

\* polygone d'identification non connue

Il est alors impossible d'employer la commande précédente. Les étapes suivantes doivent être suivies :

- affichage de l'occurrence du polygone à observer :
  - . Sélection de la structure et des niveaux de masques :
    - ST n<sup>o</sup>-structure
    - NI n<sup>o</sup>-niveau
  - . Tracé de contour et choix de la fenêtre à afficher :
    - EN profondeur
    - GR zz
  - . Affichage des polygones de la fenêtre :
    - AF
- Remplissage et sélection
  - . Remplissage de toutes les occurrences du polygone dans la fenêtre :
    - RE nom-polygone
  - . Désignation à l'aide du réticule de chaque occurrence à observer :
    - GR s

### 5. 2. 3. 3 - Description du programme

#### a) Fichiers manipulés :

Ce programme utilise les cinq fichiers générés au cours de la phase de traduction et gère un fichier contenant les coordonnées des points sélectionnés devant être observés au microscope. Ce fichier sera transmis au système de traitement d'images. Ses caractéristiques sont les suivantes:

- \* il est en accès direct,
- \* il contient un enregistrement par point sélectionné,
- \* un enregistrement occupe sept mots de type entier.

\* Structure du premier enregistrement :

N° mots	Description
1	Numéro du dernier enregistrement du fichier. Il est utilisé lors de l'ajout de points dans le fichier. Les informations suivantes concernent la correspondance entre un dessin des masques et une image issue du MEB.
2	Abscisse théorique <sup>(1)</sup> du point de recalage origine.
3	Ordonnée théorique <sup>(1)</sup> de ce point.
4	Déplacement horizontal théorique <sup>(1)</sup>
5	Déplacement vertical théorique <sup>(1)</sup>
6-7	Inutilisés.

\* Structure des enregistrements suivants :

<b>N° mots</b>	<b>Description</b>
	Les quatre premiers mots contiennent l'identification du polygone à analyser.
1	numéro du polygone
2	numéro d'appel
3	numéro de ligne
4	numéro de colonne
5	abscisse théorique <sup>(1)</sup> du point d'observation
6	ordonnée théorique <sup>(1)</sup> de ce point
7	indicateur spécifiant si la valeur du point doit ou non être relevée :
	1 : point à observer
	0 : demande d'observation annulée

(1) "théorique" qualifie une valeur issue de la description des masques

\* Utilisation du fichier

Il est créé et mis à jour par le programme d'affichage. Il est lu par un programme transférant sur ROMUALD les coordonnées des points sélectionnés.

### b) Remplissage de polygones :

Un polygone étant défini par les coordonnées de ses sommets, l'algorithme suivant a été choisi [SAR 82] :

Le polygone est parcouru de bas en haut. A partir d'une ligne comportant au moins un sommet, une ou plusieurs régions sont remplies jusqu'à ce qu'un autre sommet soit rencontré. Ce procédé est représenté en figure 5.5.

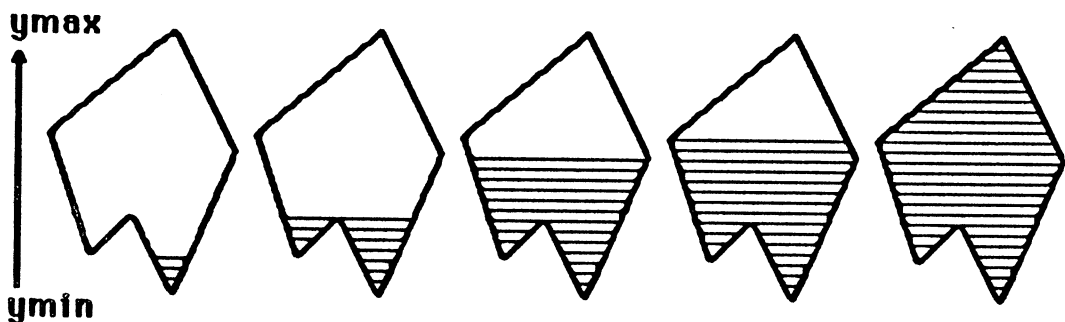
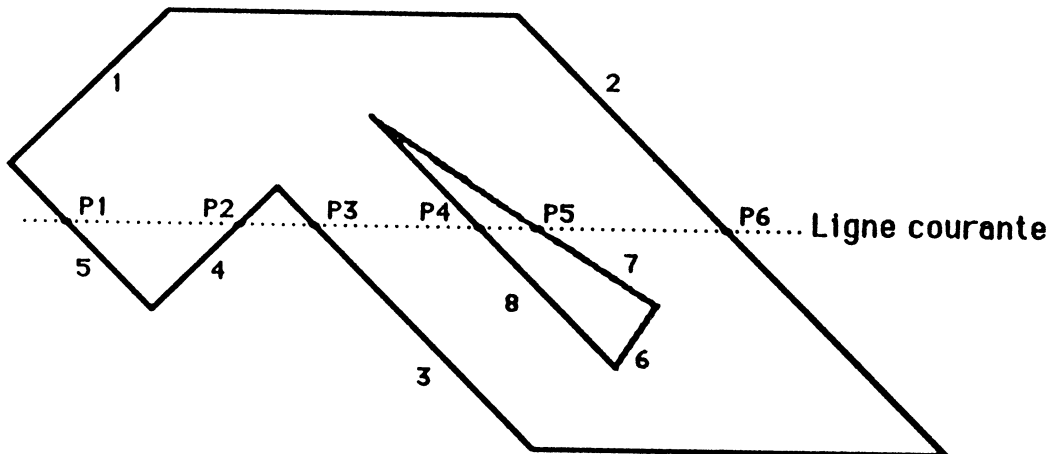


Figure 5. 5 - Remplissage d'un polygone

Le remplissage d'une ligne s'effectue en joignant deux à deux les points de cette ligne qui croisent une arête du polygone. Pour cela, une liste des arêtes intersectant la ligne à remplir, appelée liste des arêtes courantes est gérée. Les arêtes qu'elle contient sont triées de gauche à droite. Cette liste n'est mise à jour que lors du remplissage de lignes comportant des sommets, des régions pouvant alors être créées, modifiées ou supprimées.

Il faut noter que les arêtes horizontales sont traitées d'une façon particulière. En effet, elles sont immédiatement dessinées et ne figurent pas dans l'ensemble des arêtes du polygone.

Un exemple est représenté en figure 5. 6.



Ensemble des arêtes du polygone : 1, 2, .....7, 8

Liste des arêtes courantes : 5, 4, 3, 8, 7, 2

Points d'intersection de la ligne courante et des arêtes du polygone :

P1, P2, P3, P4, P5, P6

Vecteurs tracés : P1-P2, P3-P4, P5-P6

Figure 5. 6 - Remplissage d'une ligne

Cet algorithme permettant de remplir tout polygone a été adapté aux caractéristiques des polygones à traiter, à savoir :

- la direction d'une arête est horizontale, verticale ou diagonale,
- le premier et le dernier sommet décrivant un polygone sont confondus,
- le seul sommet figurant plusieurs fois dans la description d'un polygone est le sommet initial.

Description de l'algorithme :

### Procédure REMPLISSAGE

#### \* Identification des arêtes

[Parcours séquentiel des sommets du polygone]

**Pour**  $J = 1$  à nombre de sommets -1 **faire**

**Si** l'arête joignant les sommets  $J$  et  $J + 1$  est horizontale **alors**

Tracé d'un vecteur reliant ces deux sommets.

**Sinon**

- numérotation de l'arête,
- détermination de ses sommets de départ et d'arrivée (après orientation de l'arête du bas vers le haut),
- calcul de l'incrément à ajouter à l'abscisse d'un point de l'arête lors du passage à la ligne supérieure (cette valeur est égale à 0 1 ou -1 selon la direction de l'arête (verticale, diagonale avant, diagonale arrière),
- enregistrement de cette arête dans la liste des arêtes entrantes du sommet de départ, et dans celle des arêtes sortantes du sommet d'arrivée. Ces deux tableaux permettent de connaître les arêtes entrantes et sortantes de chaque sommet. Les différentes configurations possibles en un sommet sont représentées en figure 5. 7.

**Finsi**

**Finpour**

\* Classement des sommets par ordonnées croissantes et calcul des extréma  $y_{min}$  et  $y_{max}$ .

#### \* Remplissage du polygone

Lecture du premier sommet dans la liste triée par ordonnées.

[Traitement ligne à ligne et de bas en haut]

**Pour**  $y = y_{min}$  à  $y_{max}$  **faire**

- mise à jour de la liste des arêtes courantes :

**Tant que** le sommet lu est sur la ligne courante  $y$  **faire**

- . retirer de la liste des arêtes courantes les arêtes sortantes du sommet,

- . ajouter dans cette liste les arêtes entrantes du sommet (en respectant un tri de gauche à droite),
- . lire les coordonnées du sommet suivant.

**Finfaire****- Remplissage de la ligne y :**

[La liste des arêtes courantes est parcourue séquentiellement et les arêtes sont prises deux à deux.]

J  $\leftarrow$  1

**Tant que J < nombre d'arêtes courantes faire**

- . calcul et sauvegarde des abscisses  $x_1$  et  $x_2$  des points d'intersection de la ligne y respectivement avec les arêtes J et J+1 (un tel calcul s'effectue en ajoutant l'incrément en x de l'arête traitée à l'abscisse de son point d'intersection avec la ligne précédente),
- . tracé du segment joignant les points  $(x_1, y)$  et  $(x_2, y)$ ,
- . J  $\leftarrow$  J + 2.

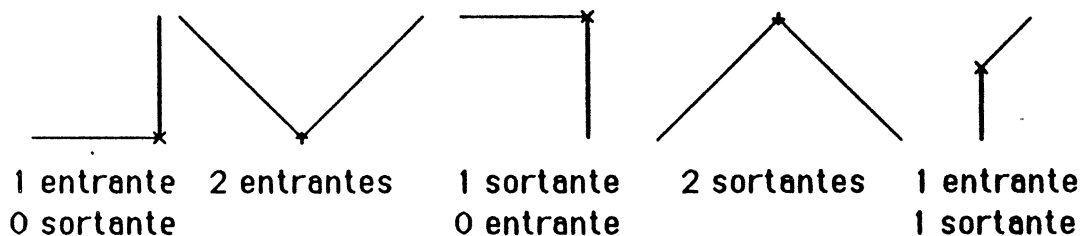
**Finfaire****Finpour****Fin**

Figure 5. 7 - Représentation des différentes configuration d'arêtes entrantes et sortantes en un sommet



### c) Décomposition d'un polygone en trapèzes

Cette procédure suit la même démarche que la procédure de remplissage. Les régions mises en évidence lors du remplissage constituent les différents trapèzes du polygone. Par exemple, le polygone représenté en figure 5. 8. a est décomposé en sept trapèzes (figure 5. 8. b).

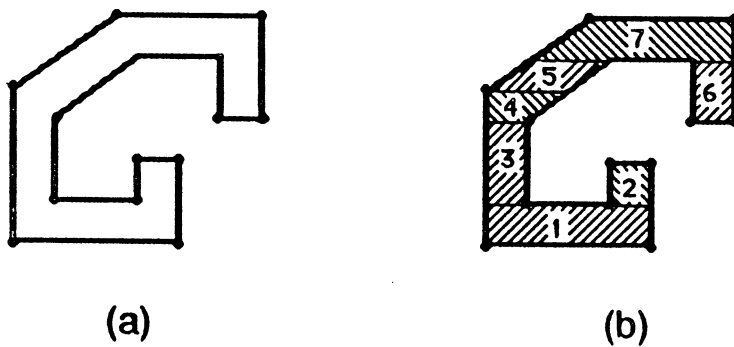


Figure 5. 8 - Décomposition d'un polygone en trapèzes

Afin de réaliser cette décomposition, le programme détermine si toute arête courante forme un bord gauche ou un bord droit du polygone. Un exemple d'une telle classification est représenté en figure 5.9.

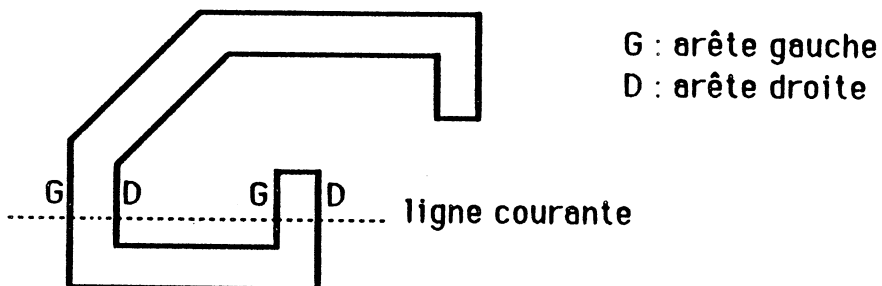


Figure 5. 9 - Classification des arêtes courantes

Un trapèze se termine au niveau d'une ligne chaque fois qu'elle comporte un sommet remplissant l'une des quatre conditions suivantes :

\* Deux arêtes entrantes :

Un trapèze est formé si elles sont insérées entre deux arêtes qui sont dans l'ordre de type gauche puis droite. Il est alors borné par ces deux dernières arêtes. Ceci est illustré en figure 5.10.a.

\* Une arête entrante et aucune sortante :

S'il s'agit d'une arête droite insérée entre une gauche puis une droite, un trapèze est délimité par les deux arêtes entre lesquelles a lieu l'insertion. Ceci est représenté en figure 5.10.b.

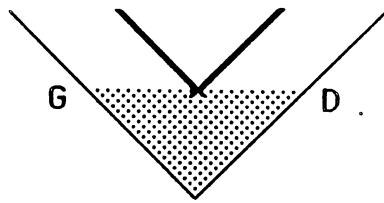
\* Une arête sortante (0 ou 1 entrante) :

Un trapèze est formé entre cette arête et l'arête courante suivante ou précédente suivant le type gauche ou droite de l'arête sortante. La figure 5.10.c illustre ce cas.

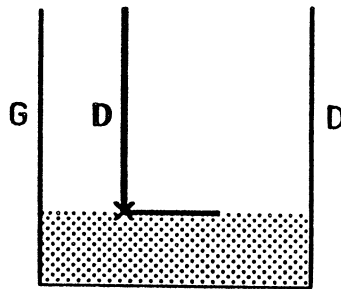
\* Deux arêtes sortantes :

- la première génère un trapèze défini de la même façon que dans la configuration précédente.

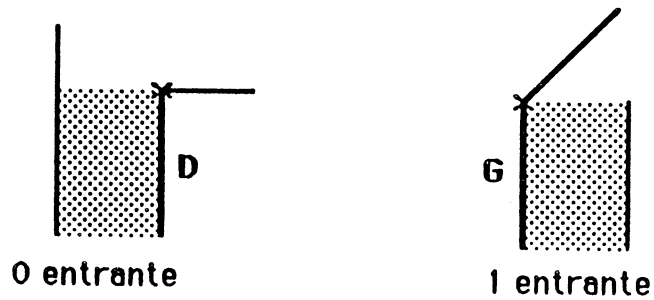
- si la deuxième est une arête gauche, un trapèze est formé entre cette arête et l'arête courante suivante comme représenté en figure 5.10.d.



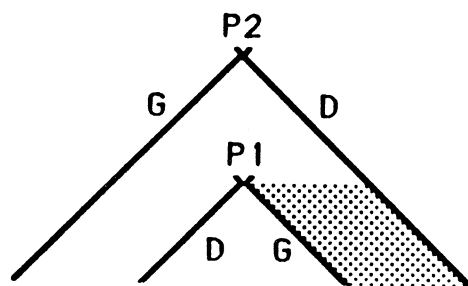
a) Deux arêtes entrantes



b) une arête entrante, aucune sortante



c) Une arête sortante



d) Deux arêtes sortantes (la deuxième arête sortante de P1 génère un trapèze, mais non celle de P2)

Figure 5.10 - Formation d'un trapèze

**d) Calcul des coordonnées du point à observer à l'intérieur d'un polygone**

La procédure de décomposition en trapèzes précédemment décrite permet d'observer tout point du polygone. Dans un premier temps, il a été choisi d'observer un point appartenant au premier trapèze localisé. Il s'agit du barycentre d'un triangle ou d'un parallélogramme, et du barycentre du plus grand rectangle compris dans tout autre trapèze.

En conséquence, la procédure de décomposition servant à identifier un seul trapèze, les suivants n'étant pas recherchés, n'est pour l'instant que partiellement utilisée. Cependant, il est possible que dans certain cas, le fait d'observer ainsi un point du premier trapèze ne soit pas satisfaisant. Une procédure de choix du point à observer décomposant totalement le polygone sera alors nécessaire.

Cette procédure de calcul a été intégrée à la procédure de remplissage et est appelée par les deux traitements batch, le calcul du point d'observation étant réalisé dans les trois cas suivants :

- \* remplissage d'un polygone,
- \* création du fichier des points à observer à l'intérieur de chaque occurrence de tout polygone nommé,
- \* ajout d'enregistrements à ce fichier. Ils correspondent aux polygones nommés à l'aide du programme d'affichage.

#### 5. 2. 4 - Programme de calcul des points d'observation

Il s'agit d'un programme batch calculant les coordonnées du point à observer à l'intérieur de chaque occurrence de tout polygone nommé interactivement à l'aide du programme d'affichage et pour qui ce calcul n'a pas été effectué.

Ce programme travaille sur les fichiers suivants :

\* en entrée : les fichiers des structures, des instructions et des sommets

\* en entrée et sortie :

- le fichier des noms de polygones. Le numéro du dernier polygone traité contenu dans le premier enregistrement est mis à jour,

- le fichier des points d'observation. Les enregistrements relatifs aux polygones traités au cours de l'exécution du programme sont ajoutés.

Son algorithme, décrit en annexe 2.2, est similaire à la partie du programme de traduction effectuant ce calcul pour les polygones nommés lors de la conception du circuit.

5. 2. 5 - Récapitulatif

La figure 5.11 représente un schéma des fichiers et traitements décrits précédemment. Le programme transférant les coordonnées des points à observer du VAX sur ROMUALD sera décrit ultérieurement (voir § 5. 3. 2).

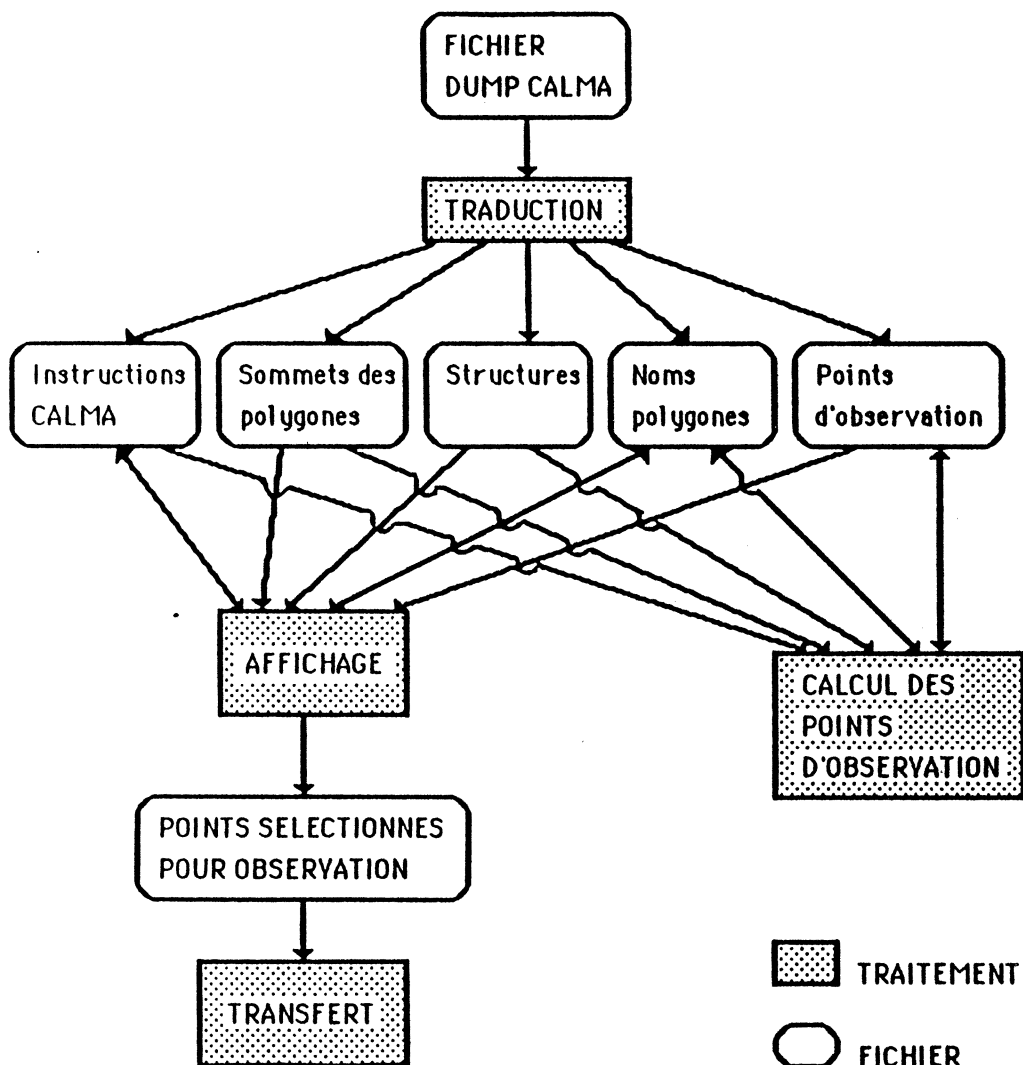


Figure 5. 11 - Schéma récapitulatif des traitements et des fichiers

### 5. 3 - Détermination de l'état logique des points à observer

#### 5. 3. 1 - Présentation

Les programmes précédemment décrits permettent, par manipulation du fichier de description des masques, de désigner les points à observer. Il faut alors localiser chacun d'eux sur l'image issue du microscope et déterminer son niveau logique à partir de son niveau de gris.

Dans la suite de ce rapport, nous appellerons :

- \* "image théorique", le dessin des masques obtenu à l'aide du programme d'affichage décrit en 5. 2.3,
- \* "image pratique", l'image microscope digitalisée par ROMUALD,
- \* "coordonnées théoriques", les coordonnées (x, y) exprimées en lambda, d'un point dans l'image théorique, le lambda étant l'unité de longueur utilisée pour la description des masques,
- \* "adresse image", l'adresse d'un point dans la mémoire image de ROMUALD. Elle se compose :
  - du numéro du processeur local dont la mémoire image contient le point considéré,
  - d'un déplacement représenté sur deux octets permettant d'adresser les 64 K de mémoire locale,
- \* "fenêtre", la partie du circuit couverte par le MEB. Une fenêtre comporte 16 zones numérotées (0, 0) (0, 1)...(3, 2) (3, 3), chacune correspondant à une image de 512 x 512 pixels (figure 5. 12).

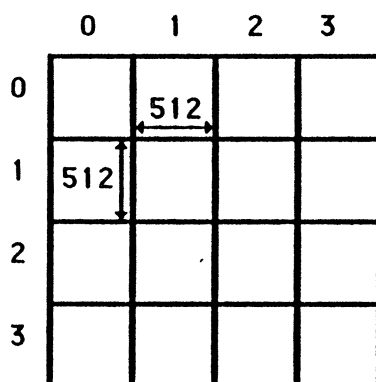


Figure 5. 12 - Fenêtre du microscope

### 5. 3. 2 - Correspondance entre l'image théorique et l'image pratique

Les points à observer étant définis par leurs coordonnées théoriques, il s'agit de retrouver le point de l'image pratique auquel chacun correspond. Pour cela, l'opérateur indique au système les coordonnées théoriques et l'adresse image de trois points du circuit appelés "points de recalage". La correspondance théorique/pratique étant connue pour ces trois points, il est alors possible de déterminer l'adresse image de tout point défini par ses coordonnées théoriques. Cette correspondance reste vraie tant que les paramètres du MEB (zoom, balayage, position platine, tension d'accélération, etc...) sont constants.

Les points de recalage, représentés en figure 5.13, se composent :

- \* d'un point origine O,
- \* d'un point H déterminant un déplacement horizontal  $\Delta x$  par rapport au point O,
- \* d'un point V déterminant un déplacement vertical  $\Delta y$  par rapport au point O.



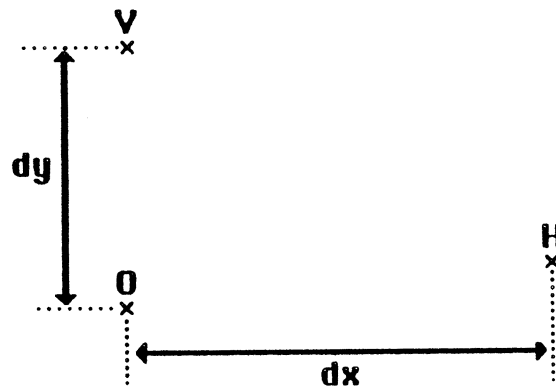


Figure 6. 13 - Points de recalage

Les trois opérations décrites ci-après doivent être effectuées :

\* Choix des points de recalage :

Il tient compte des considérations suivantes :

- Ces points doivent être facilement observables et identifiables sur l'image microscope (coins de plot, ...).
- Ils doivent être tels que les images microscope soient exploitables, c'est à dire que le nombre de pixels représentant une connexion soit suffisant. Ce nombre dépend du zoom MEB, ce dernier étant choisi de façon à couvrir une fenêtre qui contienne les trois points. Le tableau de la figure 5.14 indique le nombre de pixels correspondant à  $4 \lambda$  dans une technologie telle que un  $\lambda$  est égal à  $2,25 \mu\text{m}$ . Il permet de connaître le grossissement minimum à appliquer afin d'obtenir une image suffisamment détaillée.
- Il est préférable de choisir des points se trouvant dans trois zones différentes. Ceci permet de couvrir une partie importante de la fenêtre, et ainsi d'améliorer les résultats.

Zoom	Nombre de pixels représentant une connexion de 4 lambdas
27	4
30	5
36	6
43	7
50	8
60	9
65	10
75	11
80	12

Figure 5. 14 - Nombre de pixels par connexion en fonction du zoom du MEB

\* Repérage des points de recalage sur les images microscope

L'image d'une zone contenant au moins un point de recalage est saisie. Le numéro de cette zone est noté, ainsi que l'adresse image de ce(s) point(s). Ces opérations sont répétées jusqu'à ce que l'adresse de chacun des points de recalage soit connue.

Ces adresses permettent de calculer :

- les coordonnées (xfo, yfo) du point origine dans la fenêtre,
- le déplacement horizontal pratique noté  $\Delta xp$ ,
- le déplacement vertical pratique noté  $\Delta yp$ .

Ces valeurs correspondant à un nombre de pixels, sont comprises entre 1 et 2048.

Remarque : L'adresse image d'un point est déterminée à l'aide d'un reticule. En effet un programme permet de faire apparaître un réticule (sur le moniteur de ROMUALD visualisant l'image saisie), de le positionner en tout point de l'écran, et d'afficher l'adresse image et le niveau de gris du point sur lequel il est placé.

\* Repérage des points de recalage sur le dessin des masques

Une commande du programme d'affichage décrit en 5.2.3 permet de désigner les points de recalage à l'aide d'un réticule. A l'issue de cette opération les informations suivantes sont connues :

- les coordonnées théoriques du point origine,
- le déplacement horizontal théorique noté  $\Delta x_t$ ,
- le déplacement vertical théorique noté  $\Delta y_t$ .

(Une commande permettant de saisir directement la valeur de ces informations est également disponible).

5. 3. 3 - Calcul de l'adresse des points dans la mémoire image

Un point étant défini par ses coordonnées théoriques ( $x_t$ ,  $y_t$ ), coordonnées données sous forme de déplacements par rapport au point de recalage origine, il s'agit de déterminer son adresse image.

Les valeurs suivantes sont supposées connues :

- les déplacements horizontaux théorique et pratique ( $\Delta x_t$ ,  $\Delta x_p$ ),
- les déplacements verticaux théorique et pratique ( $\Delta y_t$ ,  $\Delta y_p$ ),
- les coordonnées ( $x_{fo}$ ,  $y_{fo}$ ) du point origine dans la fenêtre.

Les différentes étapes du calcul sont les suivantes :

5. 3. 3. 1 - Calcul des coordonnées ( $x_f$ ,  $y_f$ ) du point dans la fenêtre

Elles sont données par les formules suivantes :

$x_f$  = abscisse de l'origine dans la fenêtre + nombre de pixels séparant le point et l'origine dans le sens horizontal

$$x_f = x_{fo} + \frac{\Delta x_p * x_t}{\Delta x_t}$$

$y_f$  = ordonnée de l'origine dans la fenêtre - nombre de pixels séparant le point et l'origine dans le sens vertical

$$y_f = y_{fo} - \frac{\Delta y_p * y_t}{\Delta y_t}$$

Les règles de trois précédentes permettent de transformer des déplacements en lambda ( $x_t, y_t$ ), en déplacement en pixels.

Horizontalement, le déplacement est ajouté à l'origine car les numéros de colonnes sont incrémentés de gauche à droite.

Verticalement, il lui est soustrait car les numéros de lignes sont incrémentés de haut en bas.

Ces coordonnées sont comprises entre les bornes suivantes :

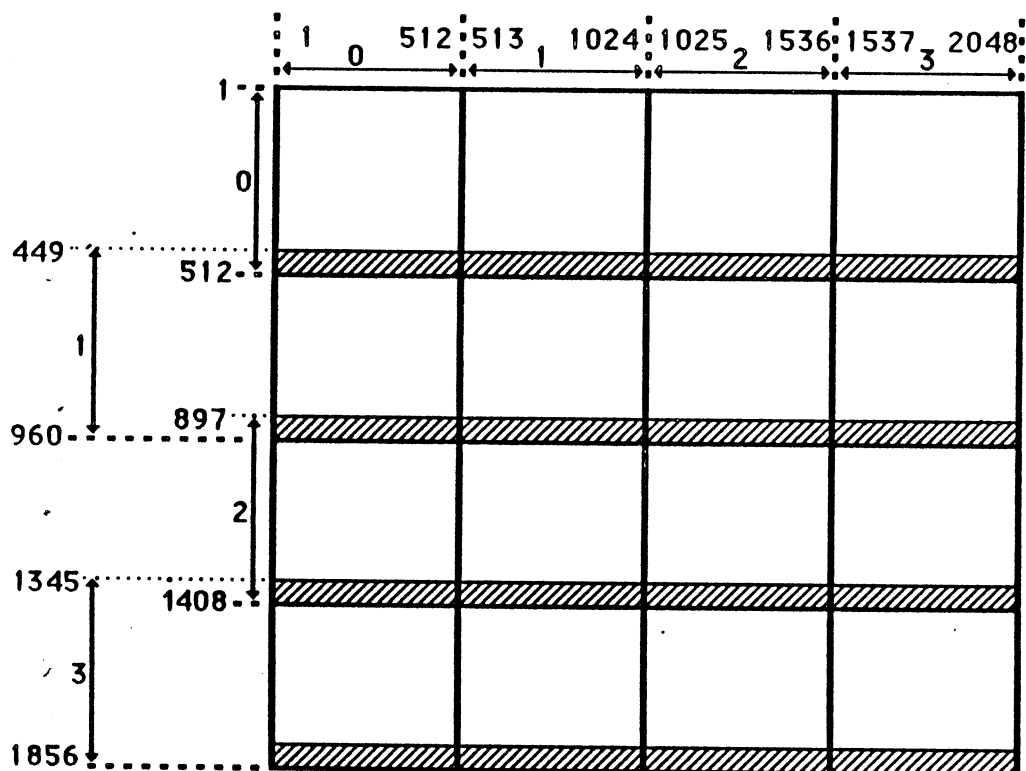
$$1 \leq x_f \leq 2048$$

$$1 \leq y_f \leq 1856$$

Un point ayant des coordonnées hors de ces limites, se trouve hors de la fenêtre couverte par le microscope et ne peut donc pas être analysé au cours de l'observation basée sur cette correspondance théorique / pratique. Afin de l'observer, le circuit devra être déplacé dans la chambre du MEB, et un nouveau recalage devra être effectué.

Remarque :

La limite supérieure de  $y_f$  n'est pas égale à 2048 (qui correspond à quatre images de 512 lignes) car lors de l'acquisition d'une image, il existe actuellement, un recouvrement de 64 lignes entre deux zones consécutives verticalement (figure 5.15). Ainsi, les 64 dernières lignes des zones situées en  $(0,x)$  sont identiques aux 64 premières de celles situées en  $(1,x)$ . Il en est de même entre les zones  $(1,x)$  et  $(2,x)$ , ainsi qu'entre  $(2,x)$  et  $(3,x)$ .



 64 lignes de recouvrement

Description d'une zone :

- $\begin{array}{c} a \\ \leftarrow c \rightarrow b \end{array}$ 
 $c$  : Indice colonne de la zone  
 $a, b$  : Abscisses minimale et maximale de la zone
- $\begin{array}{c} a \\ c \\ \downarrow b \end{array}$ 
 $c$  : Indice ligne de la zone  
 $a, b$  : Ordonnées minimale et maximale de la zone

Figure 5.15 - Découpage d'une fenêtre en 16 zones

5.3.3.2 - Calcul du numéro (nolig, nocol) de la zone à laquelle appartient le point, et des coordonnées (xz, yz) du point dans la zone.

Ces valeurs, qui ne dépendent que des coordonnées du point dans la fenêtre, sont données par les tableaux suivants :

<b>xf</b>	<b>nocol</b>	<b>xz</b>
$1 \leq xf \leq 512$	0 xF	
$513 \leq xf \leq 1024$	1	xF - 512
$1025 \leq xf \leq 1536$	2	xF - 1024
$1537 \leq xf \leq 2048$	3	xF - 1536

<b>yf</b>	<b>nolig</b>	<b>yz</b>
$1 \leq xz \leq 448$	0	yF
$449 \leq xz \leq 896$	1	yF - 448
$897 \leq yf \leq 1344$	2	yF - 896
$1345 \leq yf \leq 1856$	3	yF - 1344

5. 3. 3. 3 - Calcul de l'adresse du point dans la mémoire image.

\* Détermination du numéro "nopl" du processeur local contenant le point

L'image étant découpée en huit parties de 64 lignes consécutives, ce numéro varie de 1 à 8. Il est fonction du numéro de la ligne de l'image sur laquelle se trouve le point et est déterminé comme suit :

yz	nopl
$1 \leq yz \leq 64$	1
$65 \leq yz \leq 128$	2
$129 \leq yz \leq 192$	3
$193 \leq yz \leq 256$	4
$257 \leq yz \leq 320$	5
$321 \leq yz \leq 384$	6
$385 \leq yz \leq 448$	7
$449 \leq yz \leq 512$	8

\* Modification de l'ordonnée yz du point dans la zone

Cette coordonnée représente un numéro de ligne dans la zone, et varie de 1 à 512. Elle est transformée en un numéro de ligne dans la mémoire image locale du processeur contenant le point. Ce numéro varie de 1 à 64, et est déterminé en retranchant à yz, 64 lignes par processeur situé avant celui dont la mémoire comprend le point. Ce calcul est représenté par la formule suivante :

$$yz = yz - ((nopl-1)*64).$$

\* Calcul de l'adresse du point dans la mémoire image locale

Les 64 lignes image de chaque processeur étant rangées dans la partie de la mémoire locale allant des adresses 8000H à FFFFH, cette valeur est déterminée en appliquant la formule donnée ci-après.

$$\begin{aligned}
 \text{déplacement} & & \text{(adresse du} & & \text{Nombre de pixels} & & \text{Abscisse du} \\
 \text{en mémoire} & = & \text{premier point} & + & \text{précédant la ligne} & + & \text{point dans la} \\
 \text{locale} & & \text{image)} - 1 & & \text{qui contient le point} & & \text{ligne image} \\
 & = & 7FFF_{\text{Hexa}} & + & (xz - 1) * 512 & + & xz
 \end{aligned}$$

#### 5. 3. 4 - Correspondance entre un niveau de gris et un niveau logique.

Une fois calculée l'adresse image du point à observer, il est possible de lire son niveau de gris. Le problème est alors de calculer son état logique.

La méthode utilisée est basée sur la détermination de seuils délimitants la plage de niveaux de gris correspondant à chaque état logique.

##### 5. 3. 4. 1 - Seuillage de l'image

Il est possible de binariser ou de trinariser une image.

###### \* Binarisation :

Un seuil doit être connu. Les points de niveau de gris inférieur à ce seuil sont noirs, les autres blancs.

###### \* Trinarisation :

Deux seuils sont alors nécessaires. Les points de niveau de gris, inférieurs au seuil de noir sont noirs, ceux de niveau de gris supérieur au seuil de blanc sont blancs, les autres sont gris.

Ces seuils sont demandés interactivement à l'opérateur. L'image seuillée est affichée afin de vérifier s'ils sont corrects. Si ce n'est pas le cas, le seuillage est recommencé avec d'autres valeurs.

Remarque : Ces seuils sont saisis par l'opérateur car leur calcul automatisé n'est pas parfaitement satisfaisant. En effet, les problèmes suivants sont rencontrés :

- \* la répartition des niveaux de gris d'une image dépend beaucoup des réglages du MEB (contraste, intensité...)

- \* l'histogramme des niveaux de gris d'une image en contraste de potentiel comporte généralement plus ou moins de pics que l'histogramme théorique idéal, représenté en figure 5.16.



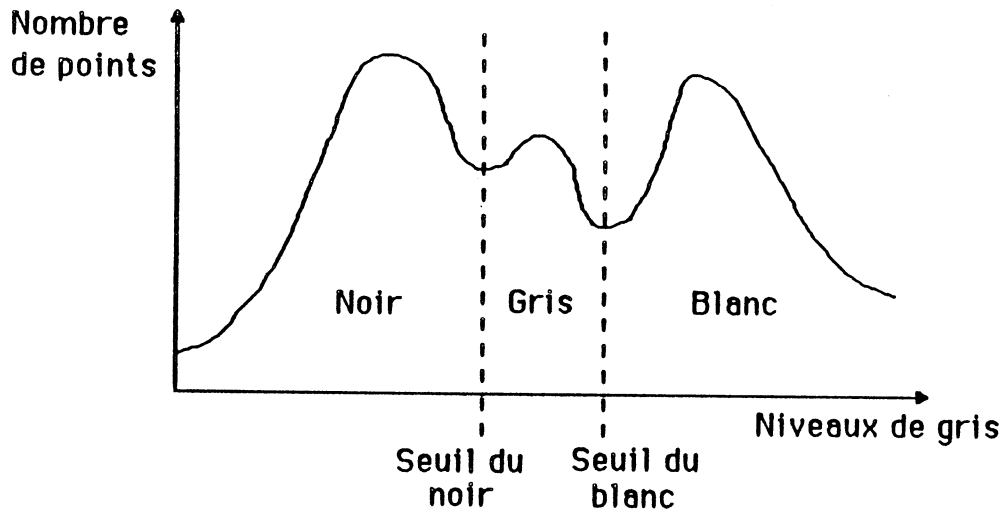


Figure 5.16 - L'histogramme théorique d'une image en contraste de potentiel comporte trois pics

#### 5. 3. 4. 2 - Détermination du niveau logique d'un point

Cette opération est réalisée en comparant le niveau de gris du point à la valeur du (des) seuil(s).

##### \* Binarisation :

Le point est considéré à 1 logique si son niveau de gris est inférieur au seuil, à 0 logique sinon.

##### \* Trinarisation :

Le point est à 1 logique si son niveau de gris, est inférieur au seuil du noir, à 0 logique si son niveau de gris est supérieur au seuil du blanc, et dans un état indéterminé lorsque son niveau de gris est compris entre les deux seuils.

### 5. 3. 5 - Programmes développés

#### 5. 3. 5.1 - Transmission des informations nécessaires à l'observation sur ROMUALD

Les deux programmes suivants ont été réalisés :

##### a) Programme VAX transférant les informations

Les données suivantes sont transférées :

- \* le nombre de points à observer,
- \* des informations de recalage :
  - le déplacement horizontal théorique  $\Delta x_t$ ,
  - le déplacement vertical théorique  $\Delta y_t$ ,
- \* les coordonnées théoriques  $(x_t, y_t)$  des points à observer qui sont exprimées par rapport au point de recalage origine.

Ce programme effectue :

- \* Le transfert d'un astérisque indiquant le début des coordonnées.
- \* La lecture des coordonnées théoriques  $(x_{t0}, y_{t0})$  du point de recalage origine, ainsi que des déplacements théoriques  $\Delta x_t$  et  $\Delta y_t$ , dans le fichier des points à observer généré par le programme d'affichage décrit en 5. 2. 3.
- \* Le transfert des déplacements  $\Delta x_t$  et  $\Delta y_t$ .
- \* Le parcours séquentiel du fichier précédemment cité.

Chaque enregistrement est lu, puis son type est testé afin de déterminer si le point doit être observé. Si c'est le cas, les quatre opérations suivantes sont effectuées :

- transformation des coordonnées  $(x, y)$  relatives à l'origine du circuit et lues dans le fichier, en coordonnées relatives au point de recalage origine. Ceci est obtenu par soustraction de  $x_{t0}, y_{t0}$  respectivement à  $x$  et  $y$ ,

- conversion de chacune des deux coordonnées en huit chiffres hexadécimaux, une valeur négative étant au préalable convertie en son "complément à 2",
- transfert des coordonnées. Il s'effectue par lignes comportant au plus neuf coordonnées,
- incrémentation du nombre de points à observer.

\* Le transfert du nombre de points à observer (après conversion en huit chiffres hexadécimaux).

\* Le transfert d'un astérisque de fin de transmission.

**b) Programme ROMUALD recevant les informations et les rangeant en mémoire**

Il réalise :

- \* La saisie de l'adresse à partir de laquelle seront rangées les informations transférées.
- \* Le lancement du programme VAX de transfert précédemment décrit.
- \* L'attente de l'astérisque de début des données.
- \* La lecture et le stockage des informations en provenance de VAX. Elles sont lues par mots de quatre chiffres hexadécimaux. Lors de la réception du premier de ces chiffres, le programme teste s'il s'agit d'un astérisque. Si ce n'est pas le cas, la totalité des informations n'a pas été transférée, et les trois opérations suivantes sont effectuées :
  - acquisition des trois chiffres suivants,
  - conversion du mot formé par quatre codes ASCII en un nombre codé sur deux octets,
  - rangement en mémoire des deux octets obtenus. Les informations sont rangées consécutivement dans la zone de stockage en laissant libres les quatre premiers octets.

\* La recopie du nombre de points à observer (dernière valeur transférées se trouvant en fin de la zone de rangement) dans les quatre premiers octets prévus à cet effet.

#### 5. 3. 5. 2 - Programme ROMUALD d'analyse des points à observer

Ses différentes étapes sont les suivantes :

a) Saisie de la situation pratique des trois points de recalage. Une telle situation est définie par :

- \* l'adresse image du point (numéro de processeur local et déplacement en mémoire locale),
- \* le numéro de la zone qui le contient (numéro de ligne et numéro de colonne).

b) Calcul des coordonnées de ces points dans la fenêtre. Elles se déduisent de la situation de chaque point et sont notées :

- \* (xfo, yfo) pour le point origine,
- \* (xfh, yfh) " " " horizontal,
- \* (xfv, yfv) " " " vertical.

c) Calcul des déplacements pratiques :

- \* déplacement horizontal,  $\Delta xp = xfh - xfo$ ,
- \* déplacement vertical,  $\Delta yp = yfh - yfo$ .

d) Saisie de l'adresse à partir de laquelle sont rangées les informations transmises depuis le VAX.

e) Calcul de l'adresse image des points observables. Un point est observable s'il appartient à la fenêtre couverte par le microscope.

L'algorithme de ce traitement, basé sur un parcours séquentiel de la liste des points à observer, est donné ci-après.

**Pour** J = 1 à nombre de points à observer **faire**

\* calcul de l'adresse image du point à observer (décrit en 5. 3. 2).

\* **Si** le point est observable **alors**

Création d'une entrée dans une table ordonnée contenant pour chaque point :

- Le numéro de la zone qui le contient. Les points sont rangés dans la table selon l'ordre croissant de leur numéro de zone (tri sur les numéros de colonne puis de ligne). Ceci permet d'analyser tous les points d'une zone en saisissant une seule fois son image.
- Son adresse image.
- Son numéro "J". Les points étant triés à l'intérieur de la table, il est nécessaire de mémoriser leur numéro afin de lier chaque résultat à l'enregistrement correspondant du fichier des points à observer.

**Finsi**

**Finpour**

f) Choix du seuillage à effectuer. L'opérateur tape un 'b' ou un 't' pour respectivement binariser ou trinariser les images.

g) Saisie du nom du fichier VAX dans lequel sont rangés les résultats.

Ce fichier comprend :

- cinq chiffres hexadécimaux par point : les quatre premiers représentent le numéro du point, le dernier son état logique (0, 1 ou 3 correspondant respectivement au 0 logique, au 1 logique et à l'état indéterminé).

- cinq caractères par vecteur de test : les quatre premiers représentent le numéro de vecteur, le dernier est un astérisque permettant de distinguer les numéros de vecteurs de ceux des points. Ces caractères figurent au début de chaque analyse de l'ensemble des points à observer (avant le résultat du premier point).

h) Ouverture de ce fichier résultat en se plaçant sous l'éditeur de texte. Il est créé ou modifié (ajout de lignes à la fin de ce fichier) à l'aide de cet utilitaire.

i) Calcul de l'état logique des points et transfert des résultats sur le VAX. L'algorithme de cette procédure est présenté ci-après :

### **Répéter**

- \* Numérotation du vecteur de test.
- \* Transfert de ce numéro suivi d'un astérisque dans le fichier des résultats
- \* Analyse de tous les points par parcours séquentiel de la table ordonnée :

**Pour** J =1 à nombre de points observables **faire**

- **S'il** s'agit du premier point **ou si** le point n'appartient pas à la zone courante **alors**

\* saisie de l'image de la zone contenant le point, qui devient la zone courante

\* seuillage de l'image obtenue :

### **Répéter**

- saisie du (des) seuil(s),
- affichage de l'image seuillée,
- saisie de la réponse à une question demandant si le seuillage est correct.

**Jusqu'à** seuillage correct

### **Finsi**

- Détermination du niveau logique du point par comparaison de son niveau de gris à la valeur du (des) seuil(s) (voir paragraphe 5.3.3.2).
- Transfert du numéro du point suivi de son état logique dans le fichier des résultats

### **Finpour**

\* Saisie de la réponse à une question demandant si l'état des points doit à nouveau être relevé : à ce stade de l'exécution, tous les points ont été analysés. Il peut alors être intéressant de modifier l'état du circuit (en faisant varier le vecteur de test) et de relever la nouvelle valeur des mêmes points.

Jusqu'à la fin de l'observation des points.

## 6 - TESTS

### 6.1 - Tests relatifs au calcul des coordonnées théoriques des points à observer

6.1.1 - Les premiers tests ont été réalisés sur un petit additionneur dont l'organisation (figure 6.1) est décrite ci-après.

Il comprend :

- deux structures de base, n'appelant aucune structure et nommées BASC\_SHR et PLUS\_SHR,
- une structure PLUS\_Q\_SHR qui se compose d'une occurrence de chacune des deux structures précédentes,
- une structure englobante ADD\_SHR correspondant au circuit et formée :
  - \* d'un tableau de trois lignes et une colonne de la structure PLUS\_Q\_SHR,
  - \* d'une occurrence de la structure BASC\_SHR.

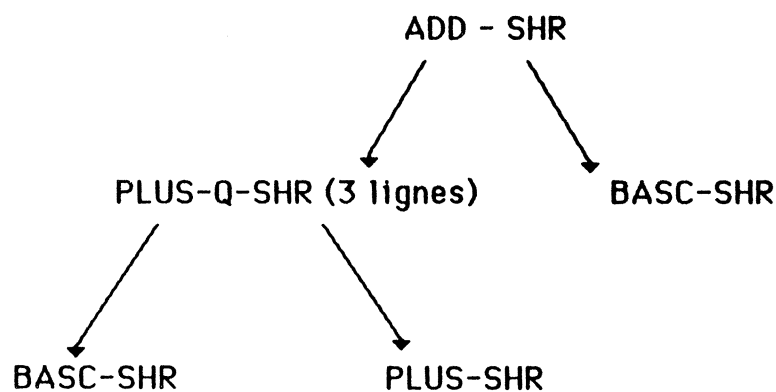


Figure 6.1 - Structure du circuit ADD-SHR



Les résultats de certaines commandes du programme d'affichage figurent en annexe 3.

6. 1. 2 - Dans un deuxième temps, la description d'une mémoire associative a été utilisée. Le temps d'affichage, qui dépend de la zone à dessiner, a alors été considéré. Il augmente avec le nombre de structures concernées par l'affichage.

## **6. 2 - Tests relatifs à la détermination du niveau logique des points**

Ils ont été effectués sur l'analyseur de signature dont le dessin des masques est donné en figure 6. 2. Ne disposant pas de la description CALMA du circuit, les coordonnées théoriques des points à observer ont été déterminées à partir du dessin BENSON des masques.

L'observation a porté sur les 33 points de tests de l'une des 8 lignes. Afin de faciliter le calcul de l'erreur commise lors de la détermination de l'adresse image des points, le coin inférieur gauche de chaque point de test a été observé. Le nombre de pixels de décalage en x, déterminé à l'aide du réticule, est négatif si le point analysé se trouve à la gauche du point réel, positif s'il est à sa droite.

Les résultats de certaines observations sont donnés ci-après.

### **6. 2. 1 - Influence d'une variation de l'adresse des points de recalage**

Lors de la détermination de l'adresse image des points de recalage à l'aide du réticule, une erreur de quelques pixels peut être réalisée. Le tableau de la figure A4.1 (annexe 4) montre l'influence d'un écart d'un pixel sur le résultat. Les valeurs marquées d'un astérisque présentent une différence de un pixel par rapport celles de la première colonne.

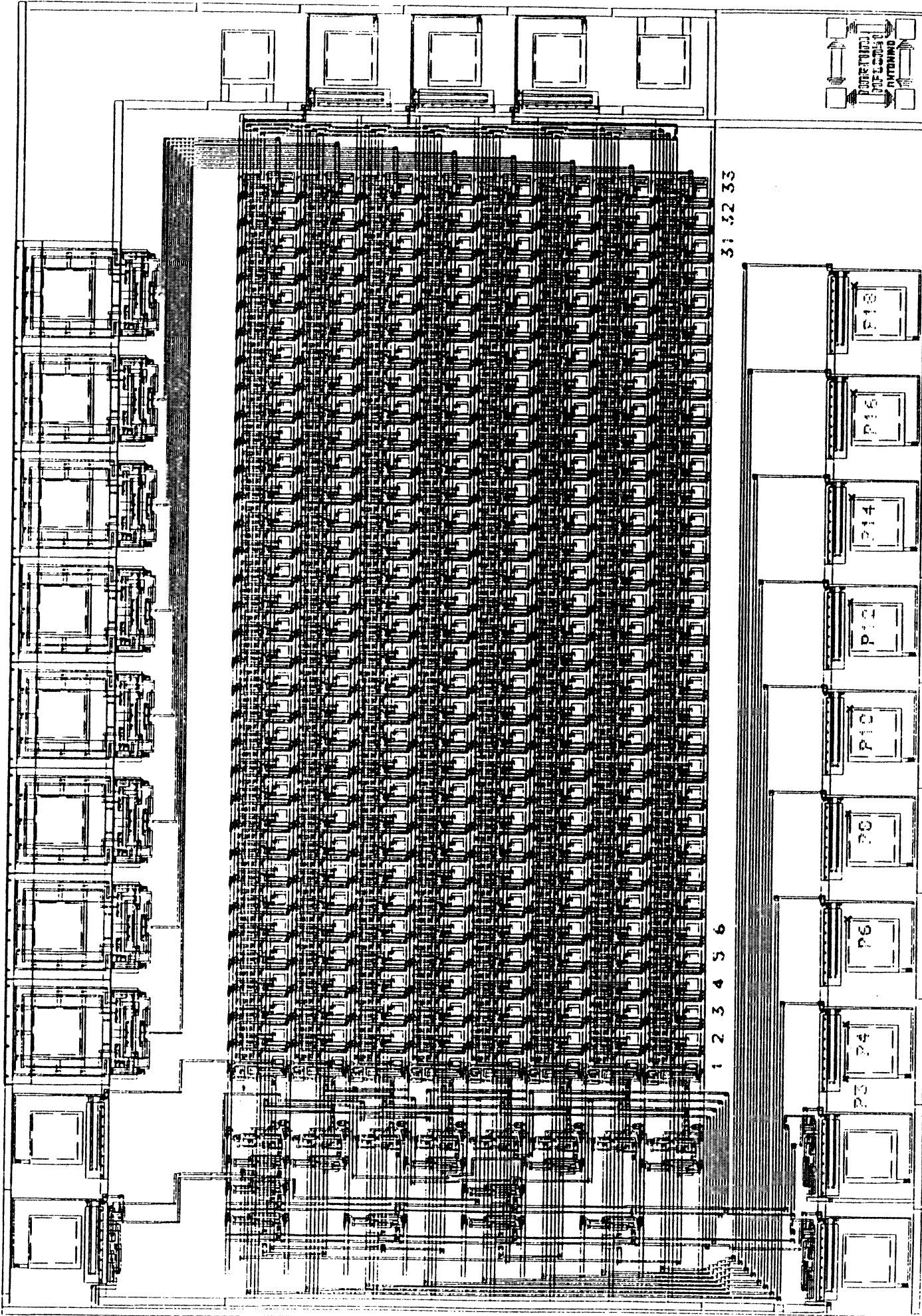


Figure 6-2. Mainframe Data Bus Interconnection

### 6. 2. 2 - Influence de la position du point de recalage horizontal par rapport au point de recalage origine

Les observations ont été réalisées dans les conditions suivantes :

- le grossissement du MEB est égal à 33. Un carré de 11 x 11 pixels représente alors chaque point de test,
- le point de recalage origine est le point P3.

Le tableau de la figure A4.2 présente les résultats obtenus en faisant varier le point de recalage horizontal de P4 à P18. Dans chaque colonne une marque indique la position de ce point de recalage parmi l'ensemble des points à observer.

Les courbes correspondantes, donnant l'écart entre la position réelle et la position observée, sont représentées en figure A5.3. Trois types d'erreurs sont mis en évidence :

- Des erreurs de mesure : il s'agit d'une marge d'erreur de quelques pixels correspondants aux erreurs de recalage.

- Des erreurs dues à la position du point de recalage horizontal.

Elles sont observées pour les points situés à la droite de ce point de recalage.

En effet, pour de tels points, les erreurs de recalage s'additionnent. Les déplacements  $\Delta x$  et  $\Delta y$  doivent donc couvrir toute la fenêtre analysée.

- Des erreurs intrinsèques au microscope :

- \* imprécision de la linéarité des balayages,

- \* aberrations optiques.

Si la courbe d'évolution de ces erreurs est reproductible, il sera sans doute possible de les compenser.

## **CONCLUSION**

Le travail réalisé a permis d'obtenir des premiers résultats. Toutefois, deux types de problèmes subsistent :

### a) Problème de positionnement

Les expériences ont montré que le point analysé peut ne pas appartenir à la connexion à observer. Ceci est principalement dû aux raisons suivantes :

- Les images issues du MEB peuvent présenter des déformations.
- Lors du calcul de l'adresse image des points, le circuit est supposé parfaitement aligné par rapport aux axes x et y. Cet alignement peut être vérifié au niveau d'une image à l'aide du réticule. Cependant sur l'ensemble de la fenêtre, il peut exister quelques pixels de décalage.
- L'adresse image de chaque point de recalage, obtenue à l'aide du réticule, est entachée de quelques pixels d'erreur.

Différentes solutions peuvent être envisagées :

- En considérant la forme du polygone à observer, il est possible de s'affranchir de toutes les causes d'erreurs précédentes. Une méthode consiste à rechercher cette forme autour du point analysé, qui ne sert alors qu'à localiser approximativement la connexion (il peut donc simplement s'agir d'un sommet du polygone).
- La prise en compte des décalages en rotation et des déformations inhérentes au microscope, au niveau du calcul de l'adresse image des points, permet de supprimer les deux premières sources d'erreurs.
- Enfin, la mise en oeuvre d'un dispositif superposant une image théorique et une image pratique sur le moniteur de ROMUALD, permet de s'affranchir des erreurs engendrées par la méthode de recalage utilisée.

Ce dispositif, en cours de réalisation, comprend un mélangeur qui reçoit :

- . le signal vidéo correspondant à l'image microscope,
- . le signal vidéo correspondant au dessin des masques, qui est égal à 0 pour le fond et à 1 pour le contour des connexions.

Il effectue leur somme et la normalise. Le signal obtenu génère une image sur laquelle le dessin théorique du circuit s'ajoute à l'image issue du MEB. Une méthode permettant de superposer ces deux images est exposée en annexe 5. Le recalage pouvant alors être effectué visuellement au pixel près, seuls subsisteront les erreurs dûes aux déformations d'images.

Cette superposition d'images présente donc un grand intérêt en mise au point automatique des circuits. Il est important de noter qu'elle permet également de faciliter leur mise au point visuelle. En effet, le concepteur désirant connaître l'état logique d'un point, aura la possibilité d'afficher simultanément l'image microscope de la zone le contenant ainsi que le contour théorique de la connexion à laquelle il appartient. Le point à analyser peut ainsi être identifié sans ambiguïté (même en présence d'éléments répétitifs), et son niveau logique est "lu" sur l'écran.

#### b) Problème de seuillage de l'image

Le calcul de l'état logique d'un point en fonction de son niveau de gris est obtenue par seuillage de l'image. Le problème consiste à automatiser la détermination des seuils, ces valeurs étant actuellement saisies par l'opérateur. Cette automatisation est rendue difficile par le fait que l'histogramme des niveaux de gris de l'image peut comporter plus de trois pics, ou présenter des pics peu marqués.

Un programme recherchant tous les pics de l'histogramme et gardant les trois principaux est disponible. Le seuil du noir correspond au point minimal situé à droite du pic de gauche, celui du blanc au point minimal situé à gauche du pic de droite.

Afin d'améliorer le résultat de ce programme, il est envisageable de s'aider de niveaux de gris de référence (niveau d'une ligne de masse, d'une ligne d'alimentation et du substrat) relevés automatiquement sur l'image MEB pour rechercher les trois pics de l'histogramme.

Au delà de ces problèmes, d'importants travaux sont nécessaires dans les deux domaines suivants :

a) Etablissement d'une liaison avec le MEB à partir de niveaux d'abstraction plus élevés que CALMA (description symbolique, description de niveau RTL). Les procédures de test ne sont pas prévues dans ces langages de haut niveau.

b) Développement de méthodes de mise au point de CI s'appuyant sur les outils généraux décrits dans le présent document. Le résultat des premières recherches a été brièvement exposé dans le chapitre 4.



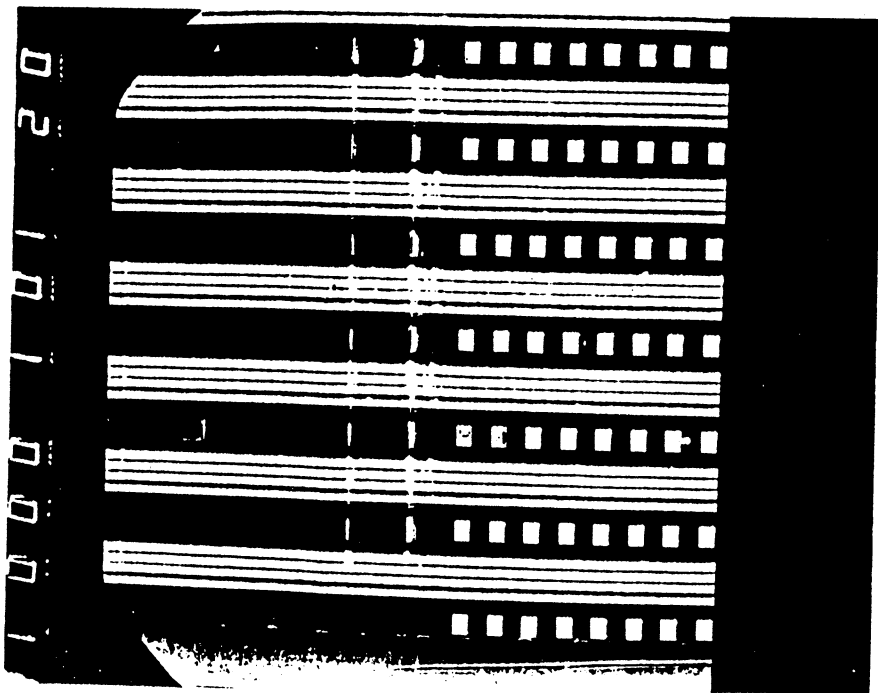
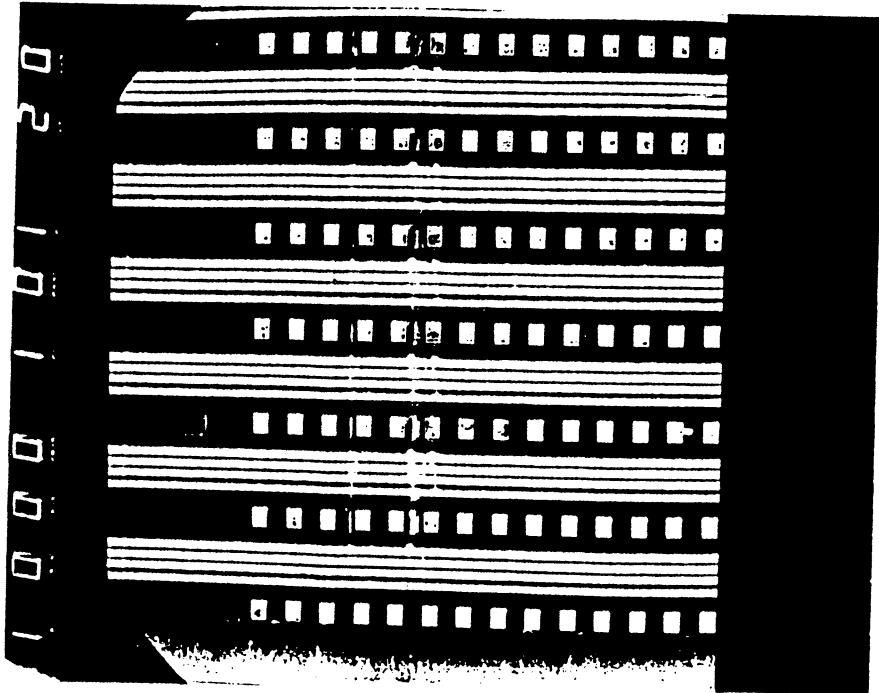
**ANNEXE 1****IMAGES MICROSCOPE**

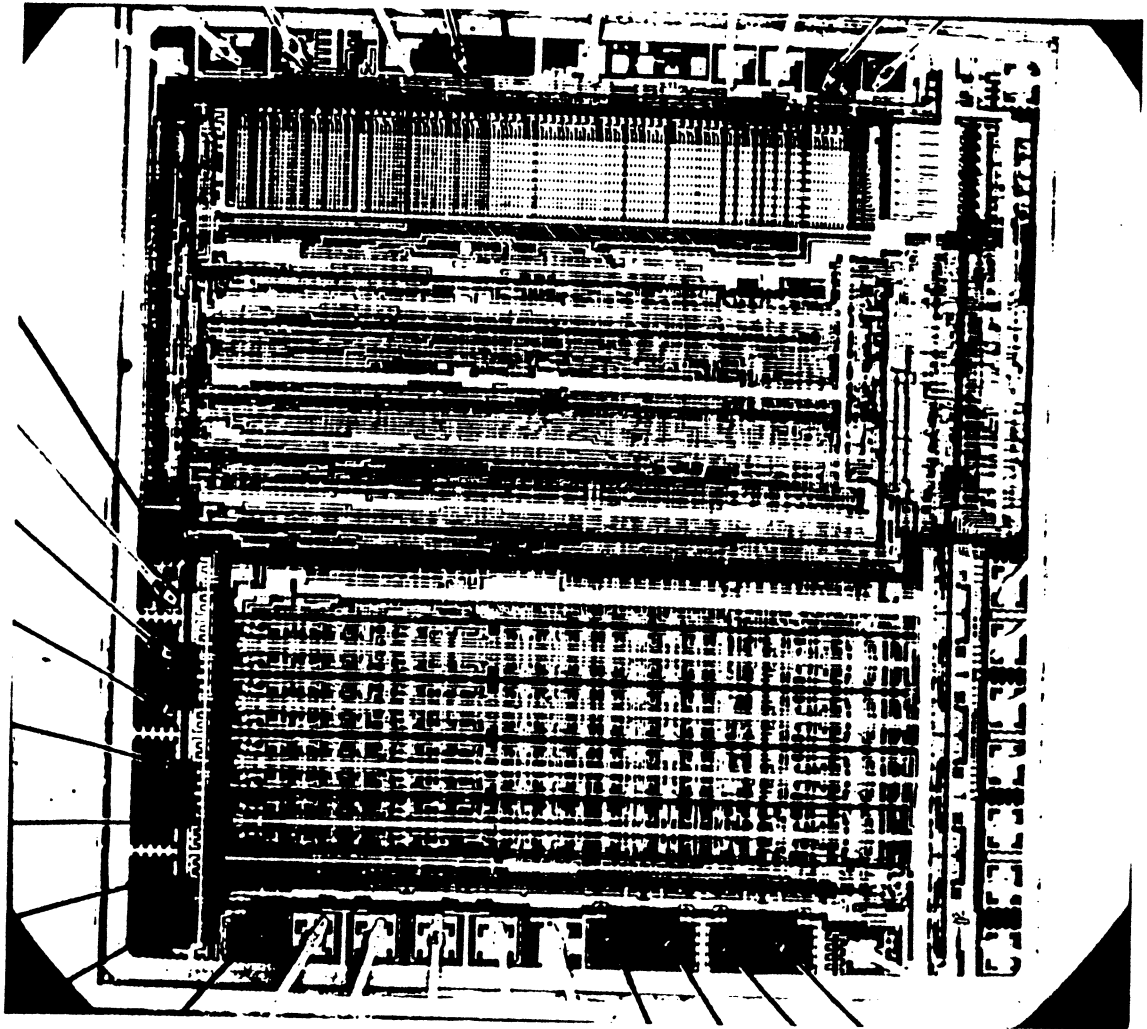
Cet annexe comprend des photographies obtenues lorsque le circuit est en fonctionnement dans la chambre du MEB.

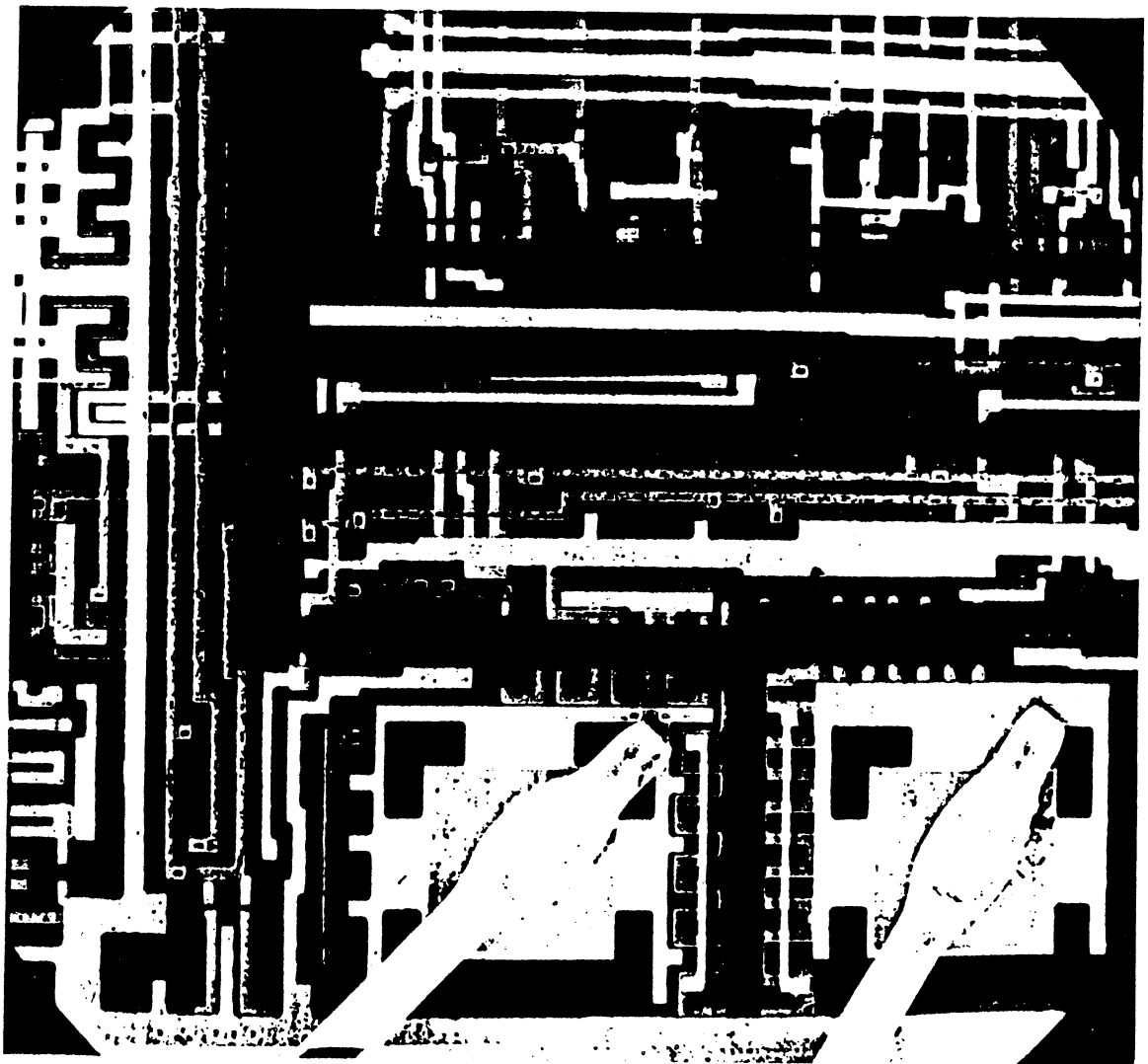
- Les trois premières présentent un décalage de registres.
- Les trois suivantes ont été réalisées en utilisant un grossissement microscope croissant.

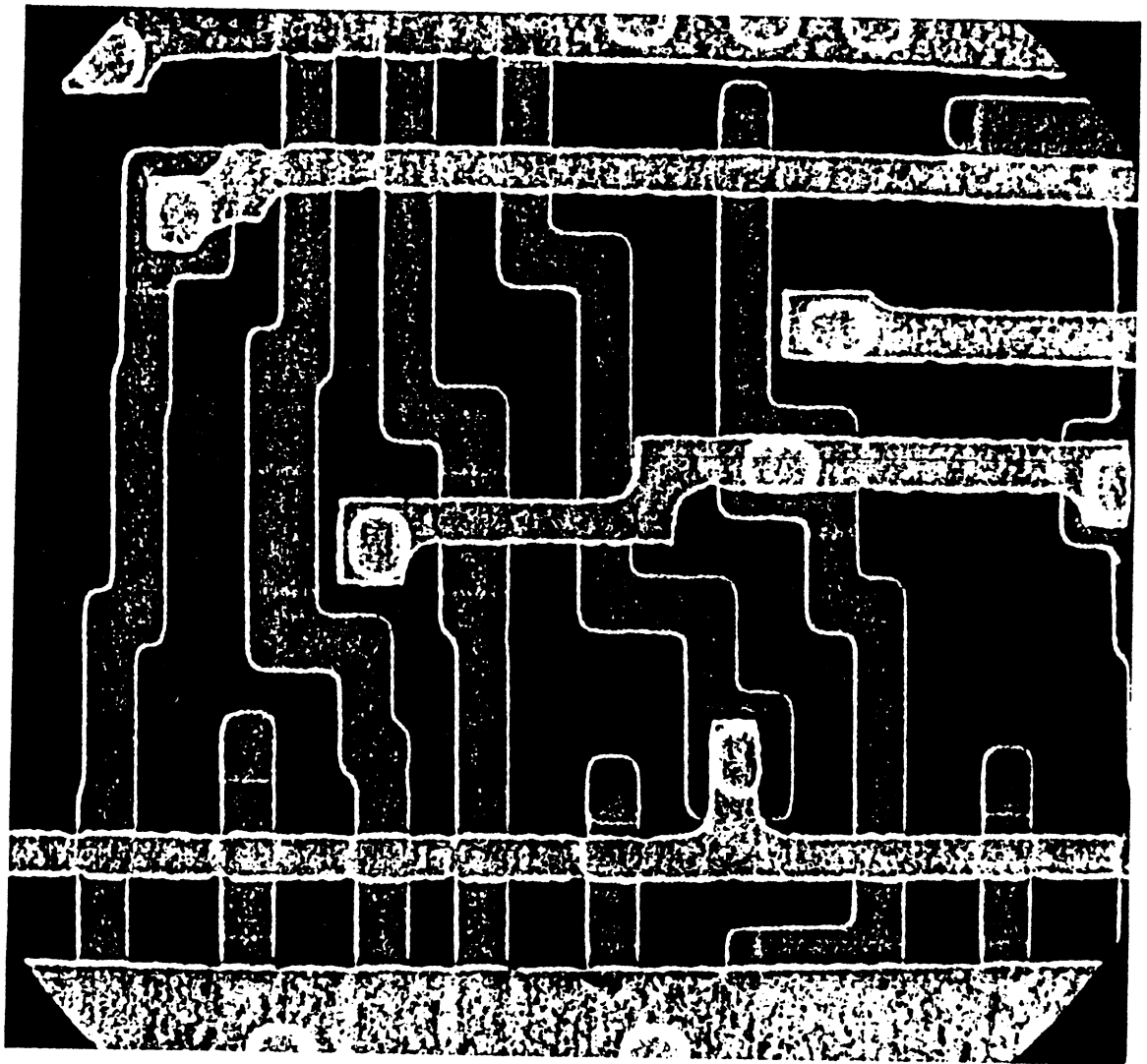












## **ANNEXE 2**

### **ALGORITHMES**

#### **A2.1 - Algorithme du programme de traduction**

##### **A2.1.1 - Création des fichiers des instructions CALMA, des sommets de polygones et des noms de polygones**

[parcours séquentiel du fichier dump CALMA]

Lecture du premier enregistrement.

**Tant que** la longueur de l'enregistrement est non nulle **faire**

- Traitement de l'enregistrement en fonction de son type :

\* **enregistrement non pris en compte** : aucun traitement.

\* **fin d'un chemin** :

- transformation du chemin en polygone,
- exécution de la procédure de fin de polygone décrite ci-après.

\* **fin d'un polygone** :

- création d'un enregistrement dans le fichier des instructions,
- création d'un enregistrement par sommet dans le fichier des sommets,
- mise à jour de l'encombrement de la structure le contenant.

\* **fin d'un appel** :

- s'il s'agit d'un appel de tableau, ses dimensions sont calculées en tenant compte des rotations qui lui sont appliquées,
- création d'une entrée dans une table des appels, contenant pour chaque appel toutes les informations à enregistrer dans le fichier des instructions. Les appels ne sont pas rangés immédiatement dans ce fichier afin qu'ils se trouvent après les polygones de la structure.

- mise à jour d'une table des structures contenant une entrée par structure. L'une de ses zones, qui comprend le nombre d'appels à chaque structure, est incrémentée pour la structure appelée.

\* Fin d'une structure :

- rangement (dans la table des structures) de son encombrement calculé sur ses polygones sans tenir compte de ses appels,
- création dans le fichier des instructions :
  - . d'un enregistrement par appel simple et de deux par appel de tableau effectué par la structure, à partir d'informations lues dans la table des appels,
  - . d'un enregistrement de fin de structure,
- rangement dans la table des structures, de l'adresse de son premier appel dans le fichier des instructions.

\* Autre enregistrement :

- calcul de son nombre de données,
- conversion des données (transformations des chiffres hexadécimaux lus dans le fichier dump en des nombres entiers, des nombres réels ou des caractères),
- les enregistrements ayant une donnée caractère sont traités comme suit :
  - . s'il s'agit du nom d'une structure appelée, elle est numérotée.
  - . s'il s'agit du nom d'une structure dont la description débute :
    - \* la structure est numérotée,
    - \* son encombrement est supposé nul,
    - \* l'adresse de son premier élément, dans le fichier des instructions, est rangée dans la table des structures,
    - \* un enregistrement de début de structure est créé dans le fichier des instructions,
  - . s'il s'agit du nom d'un polygone :
    - \* ce nom est numéroté,
    - \* il est éventuellement ajouté au fichier des noms de polygones,
    - \* les coordonnées du point à observer, dans le repère de la structure, sont calculées.

- Lecture de l'enregistrement suivant

## **Finfaire**

### A2.1.2 - Création des fichiers des structures et des points d'observation

[parcours séquentiel de la table des structures]

**Pour** J = 1 à nombre de structures **faire**

**Si** la structure J n'est appelée par aucune structure **alors**

[lecture de la composition de la structure dans le fichier des instructions CALMA].

Lecture de l'enregistrement indiquant le début de la structure.

#### **Répéter**

traitement de l'enregistrement lu en fonction de son type :

#### \* **début de la structure J**

- création d'une pile contenant le numéro et l'encombrement de la structure,
- lecture du premier élément de la structure J.

#### \* **Appel d'une structure**

- pour un appel de tableau, lecture de l'enregistrement suivant du fichier instruction,
- empilement des informations suivantes :
  - . le numéro et l'encombrement de la structure,
  - . le numéro et le type de l'appel,
  - . l'adresse de retour (dans le fichier des instructions),
  - . les opérations à appliquer à la structure et les coordonnées de son origine,
  - . les dimensions d'un tableau.
- Lecture du premier élément de la structure appelée.

\* Polygone

- **Si** le polygone est nommé **alors**
  - . calcul des coordonnées du point à observer (par rapport à l'origine du circuit) en parcourant la pile afin de prendre en compte les opérations à appliquer aux structures ainsi que leur position dans le circuit,
  - . création d'un enregistrement dans le fichier des points d'observation.

**Finsi**

- Lecture de l'enregistrement suivant.

\* Fin d'une structure

- Enregistrement de cette structure dans le fichier des structures.
- **Si** ce n'est pas la fin de la structure englobante **J** **alors**
  - . modification de l'encombrement de cette structure finie en tenant compte des opérations appliquées lorsqu'elle est appelée,
  - . prise en compte de ce nouvel encombrement dans celui de la structure appelante. L'encombrement d'une structure tient ainsi compte de l'ensemble des encombrements de celles qu'elle appelle.

**Finsi**

- Retirer la structure de la pile
- **Si** la pile n'est pas vide **alors**
  - lecture de l'enregistrement de retour.

**Finsi**

**Jusqu'à** ce que la pile soit vide.

**Finsi****Finpour**



### A2.1.3 - Création du premier enregistrement du fichier des noms de polygone

Ces deux champs sont égaux au nombre d'enregistrements du fichier, ce qui indique que le point à observer à l'intérieur de toute occurrence de chaque polygone nommé, lors de la conception du circuit, a été calculé.

## A2.2 - Algorithme du programme de calcul des points d'observation

- \* lecture du fichier des structures : création d'une table contenant les informations lues et calcul du nombre de structures,
- \* lecture du premier enregistrement du fichier des noms de polygones afin de connaître le numéro du dernier polygone traité,
- \* lecture du premier enregistrement du fichier des points d'observation afin de connaître le numéro du dernier enregistrement créé dans ce fichier,
- \* parcours de la table des structures et traitement de toutes les structures englobantes :

**Pour** J = 1 à nombre de structures **faire**

**Si** la structure J est englobante **alors**

[Lecture de sa composition dans le fichier des instructions].

Lecture de l'enregistrement marquant le début de la structure J.

**Répéter**

Traitement de l'enregistrement lu en fonction de son type :

\* début de la structure J :

- création d'une pile contenant le type de l'enregistrement,
- lecture du premier élément de la structure J.

\* Appel d'une structure :

- lecture des dimensions d'un tableau dans l'enregistrement suivant,
- empilement des informations suivantes :
  - . le numéro et le type de l'appel,
  - . l'adresse de retour,
  - . les opérations à appliquer à la structure et les coordonnées de son origine,
  - . les dimensions d'un tableau,
- lecture du premier élément de la structure appelée.

\* polygone :

**Si** son numéro est supérieur au dernier numéro traité **alors**

[c'est un polygone nommé dont le point à observer à l'intérieur de chacune de ses occurrences n'a pas été déterminé].

- lecture des coordonnées de ses sommets,
- calcul des coordonnées du point à observer à l'intérieur du polygone en tenant compte des opérations effectuées sur cette structure et sur toutes celles se trouvant à niveau supérieur,
- création d'un enregistrement dans le fichier des points d'observation.

**Finsi**

\* Fin d'une structure :

- Retirer cette structure de la pile,
- **Si** la pile n'est pas vide **alors**  
lecture de l'enregistrement de retour.

**Finsi**

**Jusqu'à** ce que la pile soit vide.

**Finsi**

**Finpour**

- \* mise à jour du dernier polygone traité. A l'issue de ce traitement, il est égal au numéro du dernier polygone nommé (car tout polygone nommé a alors été traité).

**ANNEXE 3**

**RESULTATS DU PROGRAMME D'AFFICHAGE**

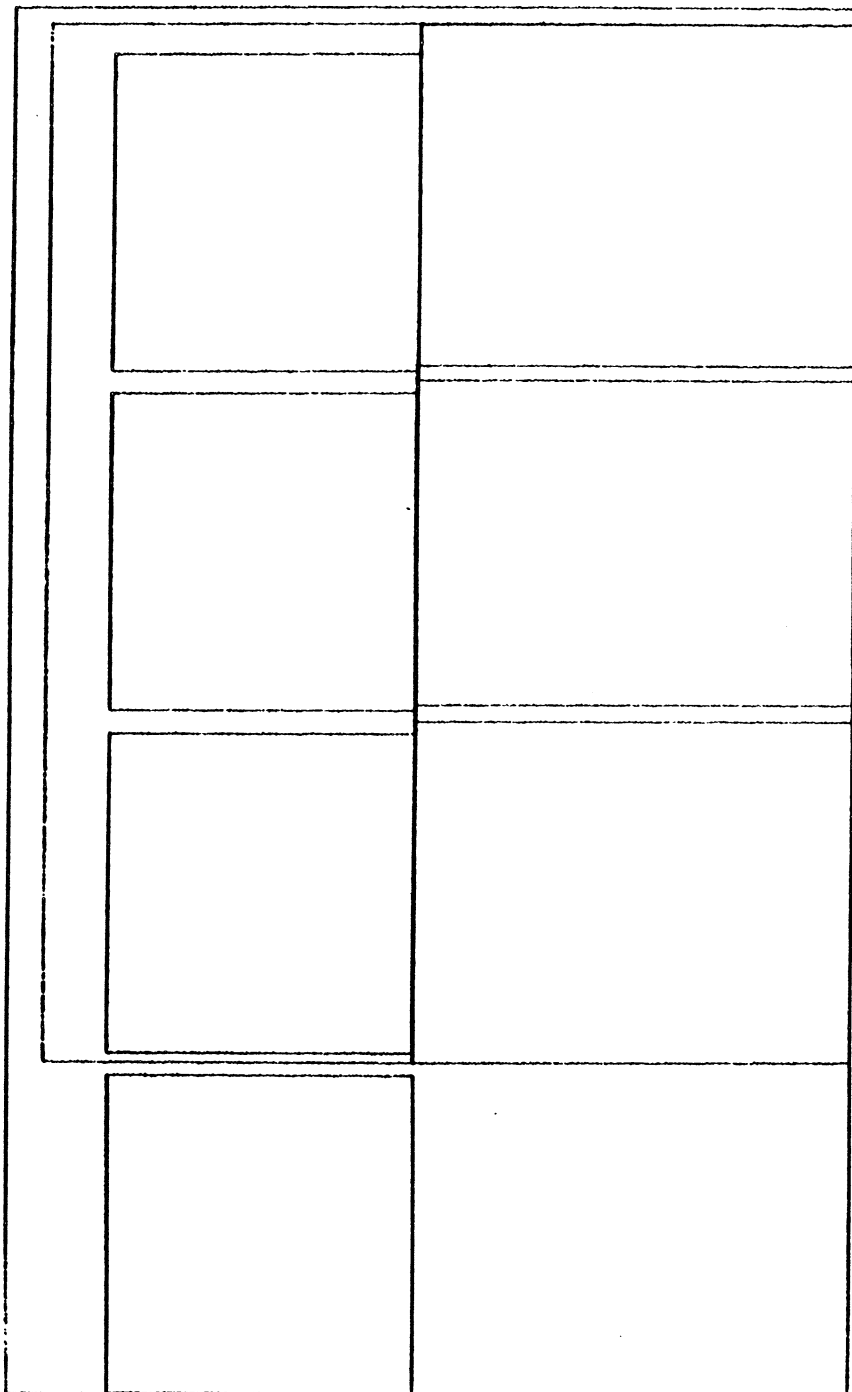


Figure A3.1 - Dessin du contour des structures

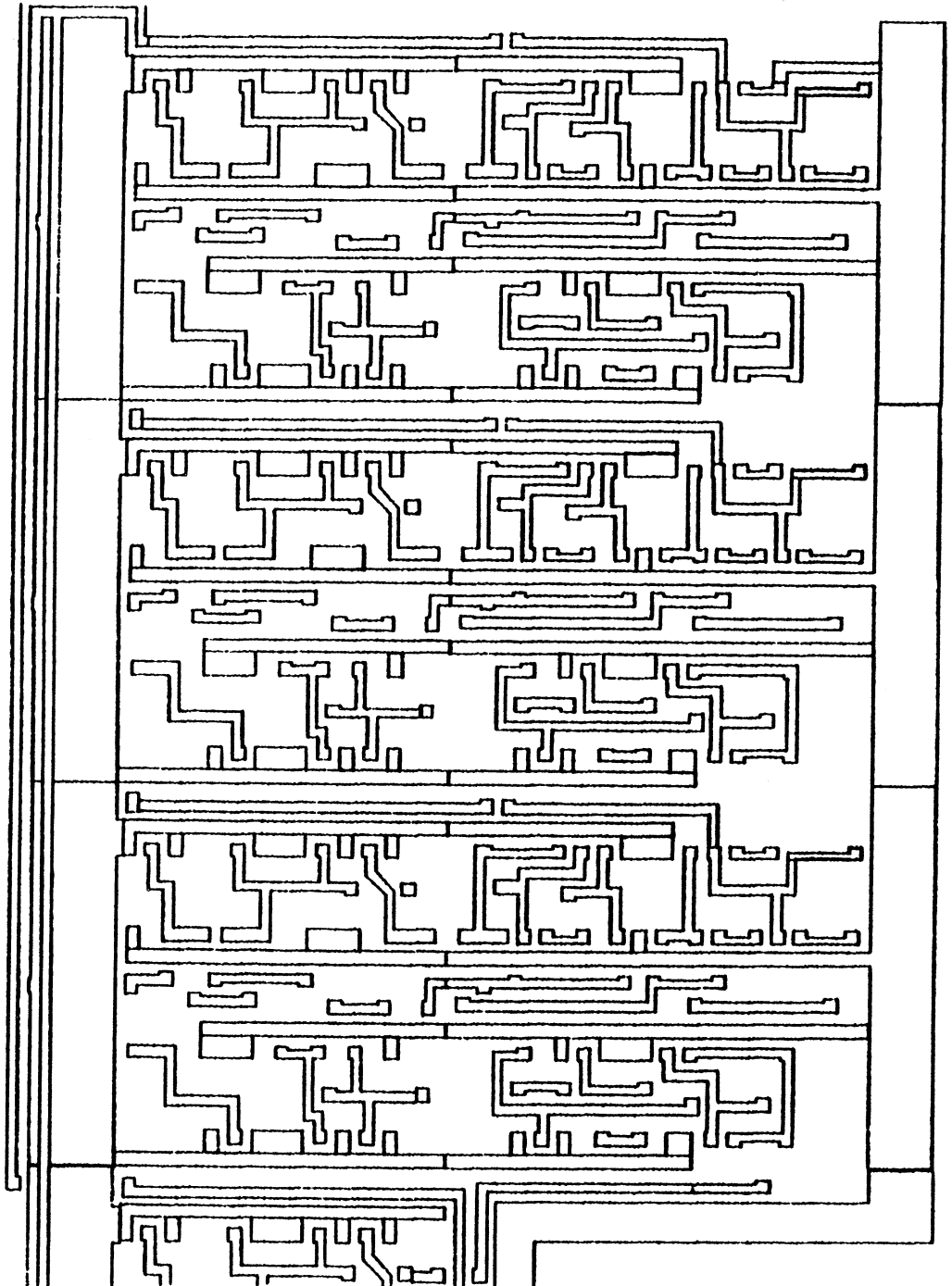


Figure A3.2 - Affichage d'une partie du circuit

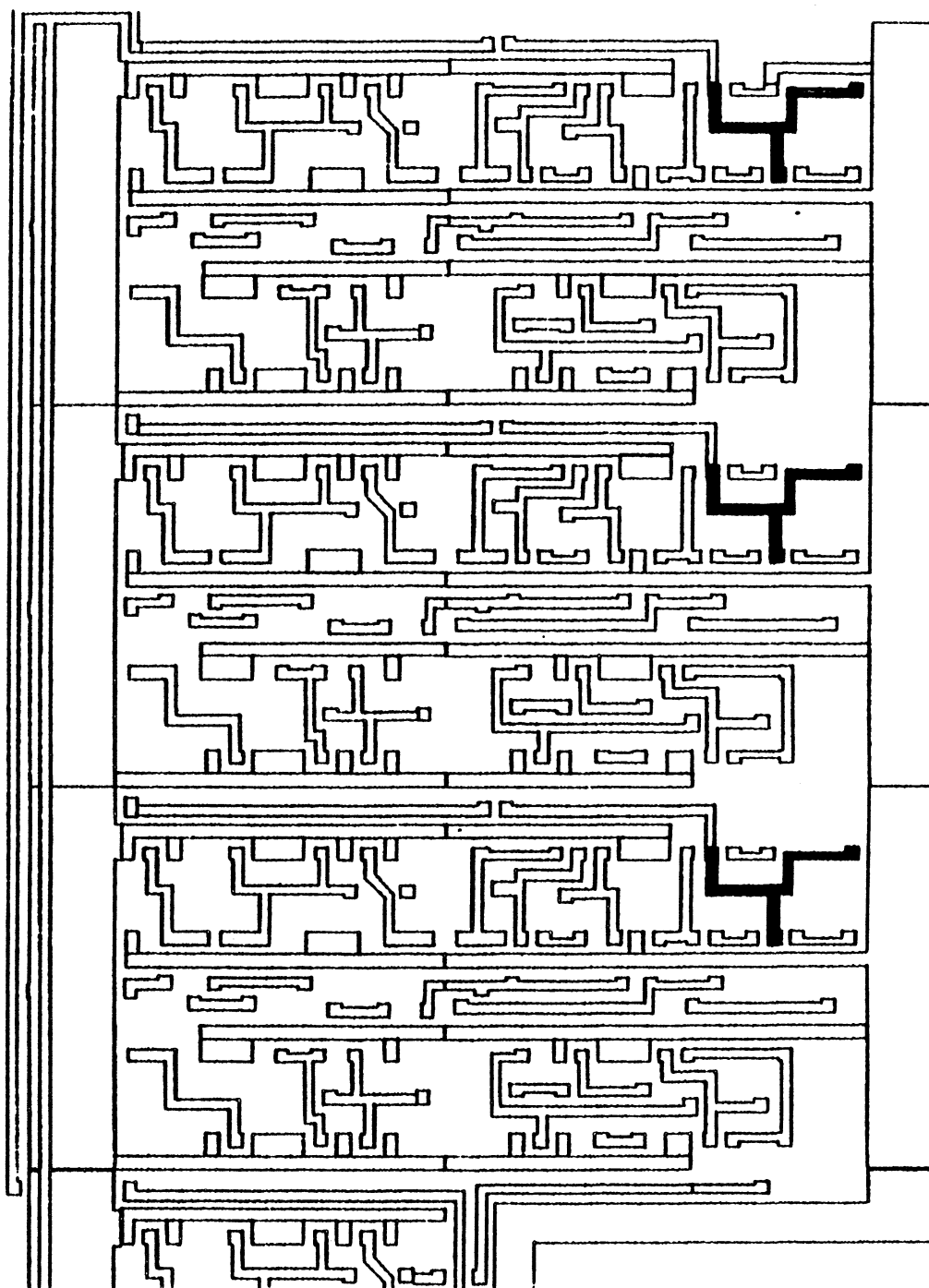


Figure A3.3 - Remplissage d'un polygone

ANNEXE 4**ERREURS DE POSITIONNEMENT**

POINTS DE RECA- LAGE POINT D'OBSER- VATION	P3	P3-1	P3	P3-1	P3+1	P3	P3+1	P3-1	P3+1
	P16	P16	P16-1	P16-1	P16	P16+1	P16+1	P16+1	P16-1
1	0	-1*	0	-1*	+1*	0	+1*	-1*	+1*
2	0	-1*	0	-1*	+1*	0	+1*	-1*	+1*
3	+2	+1*	+2	+1*	+3*	+2	+3*	+1*	+3*
4	+3	+2*	+3	+2*	+4*	+3	+4*	+2*	+4*
5	+3	+2*	+3	+2*	+4*	+3	+5*	+2*	+4*
6	+4	+3*	+4	+3*	+5*	+4	+5*	+4	+5*
7	+4	+4	+4	+3*	+5*	+5*	+5*	+4	+5*
8	+4	+4	+4	+3*	+5*	+5*	+5*	+4	+5*
9	+5	+4*	+4*	+4*	+5	+5	+6*	+4*	+5
10	+4	+3*	+3*	+3*	+4	+4	+5*	+3*	+4
11	+4	+3*	+3*	+3	+4	+4	+5*	+4	+4
12	+4	+3*	+3*	+3*	+4	+4	+5*	+4	+4
13	+5	+4*	+4*	+4*	+5	+5	+6*	+5	+5
14	+6	+5*	+5*	+5*	+6	+6	+7*	+6	+6
15	+6	+6	+6	+5*	+7*	+7*	+7*	+6	+6
16	+5	+5	+4*	+4*	+5	+6*	+6*	+5	+5
17	+5	+5	+5	+4*	+6*	+6*	+6*	+6*	+5
18	+4	+4	+4	+3*	+5*	+5*	+5*	+5*	+4
19	+3	+3	+3	+2*	+4*	+4*	+4*	+4*	+3
20	+3	+3	+3	+2*	+4*	+4*	+4*	+4*	+3
21	+2	+1*	+1*	+1*	+2*	+2	+3*	+2	+1*
22	+3	+3	+2*	+2*	+3*	+4*	+4*	+3	+2*
23	+2	+2	+1*	+1*	+2*	+3*	+3*	+3*	+1*
24	+2	+2	+1*	+1*	+2	+3*	+3*	+3*	+8*
25	+1	+1	+1*	+1*	+1	+2*	+2*	+2*	0\$
26	0	0	-1*	-1*	0	+1*	+1*	+1*	-1*
27	-2	-2	-3*	-3*	-2	-1*	-1*	-1*	-3*
28	-2	-2	-3*	-3*	-2	-1*	-1*	-1*	-3*
29	-2	-1*	-3*	-3*	-2	0*	-1*	0*	-3*
30	-2	-1*	-3*	-3*	-2	0*	-1*	0*	-3*
31	-2	-2	-4*	-3*	-3*	-1*	-1*	-1*	-4*
32	-3	-3	-5*	-4*	-4*	-2*	-2*	-2*	-5*
33	-5	-5	-7*	-6*	-6*	-4*	-4*	-4*	-7*

Figure A4.1 - Résultats sur l'influence d'une variation  
de l'adresse des points de recalage

POINT DE RECALAGE HORIZONTAL POINT D'OBSERVATION	P4	P6	P8	P10	P12	P14	P16	P18
1	0	0	0	0	0	0	0	0
2	-1 →	-1	-1	-1	-1	-1	0	0
3	0	0	+1	+1	+1	+1	+2	+2
4	0	0	+1	+2	+2	+2	+3	+3
5	-1	-1	+1	+1	+2	+2	+3	+3
6	-1	-1 →	+1	+2	+3	+3	+4	+4
7	-2	-2 →	+1	+2	+2	+3	+4	+5
8	-3	-3	0	+1	+2	+3	+4	+5
9	-4	-3	0	+1	+2	+3	+5	+5
10	-5	-5	0 →	0	+1	+1	+4	+4
11	-7	-6	-2 →	-1	+1	+1	+4	+4
12	-8	-7	-2	-1	0	+1	+4	+4
13	-8	-7	-2	0	+1	+2	+5	+6
14	-8	-7	-1	0 →	+2	+3	+6	+7
15	-8	-8	-2	0	+2	+3	+6	+7
16	-11	-10	-2	-2	0	+1	+5	+6
17	-11	-10	-4	-2	0	+1	+5	+6
18	-13	-12	-5	-3	-1 →	0	+4	+5
19	-15	-14	-7	-5	-2 →	-1	+3	+4
20	-16	-15	-7	-5	-3	-1	+3	+4
21	-19	-18	-10	-7	-5	-3 →	+2	+3
22	-19	-18	-9	-7	-4	-3 →	+3	+4
23	-21	-20	-10	-8	-5	-4	+2	+3
24	-22	-21	-11	-8	-5	-4	+2	+3
25	-24	-22	-12	-10	-6	-5	+1 →	+2
26	-26	-24	-14	-11	-8	-6	0 →	+1
27	-29	-27	-16	-13	-10	-8	-2	0
28	-30	-28	-17	-14	-10	-9	-2	0
29	-30	-29	-17	-14	-10	-9	-2	0
30	-31	-30	-18	-15	-11	-9	-2	0 →
31	-31	-32	-19	-16	-12	-10	-2	-1
32	-35	-34	-21	-17	-13	-11	-3	-2
33	-38	-35	-24	-20	-16	-13	-5	-4

Figure A4.2 - Résultats sur l'influence de la position du point de recalage horizontal

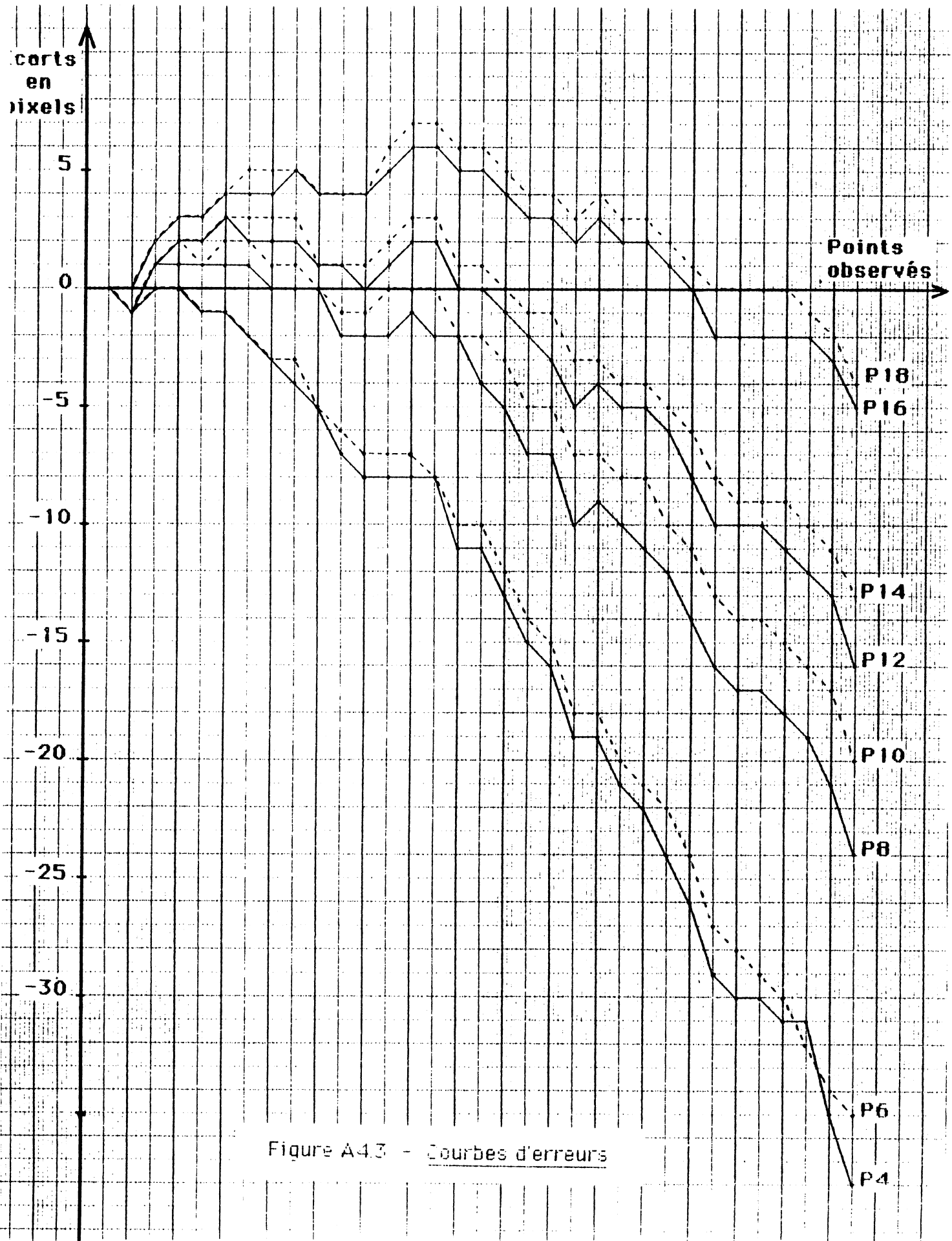


Figure A4.3 - Courbes d'erreurs



## ANNEXE 5

### SUPERPOSITION D'UNE IMAGE PRATIQUE ET D'UNE IMAGE THEORIQUE

La corrélation de ces deux images suppose la réalisation des étapes suivantes :

#### A5.1 - Etablissement d'une correspondance entre le grossissement du MEB et l'échelle d'affichage du dessin des masques

Afin que les dimensions des connexions soient identiques sur les deux images, la longueur en lambda correspondant aux 512 pixels d'une image microscope est déterminée pour chaque grossissement MEB utilisé. Cette valeur est notée nbl et est schématisée en figure A5.1.

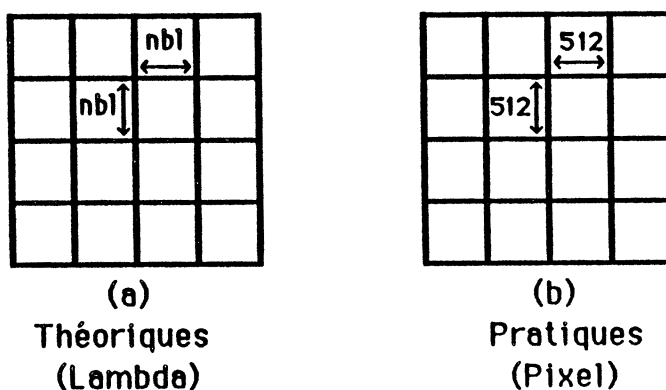


Figure A5.1 - Dimensions d'une zone

Cette correspondance est établie une seule fois.

## **A5.2 - Réalisation de la superposition des deux images**

Elle peut se décomposer en trois phases de la façon suivante :

### **A5.2.1 - Sélection de la longueur en lambda d'une zone (nbl)**

Le grossissement du MEB est indiqué au programme affichant les masques du circuit, afin que la valeur nbl soit déduite de la correspondance précédemment établie.

### **A5.2.2 - Localisation de la fenêtre microscope sur le dessin des masques**

La fenêtre couverte par le microscope est grossièrement repérée sur le dessin des masques, en désignant l'un de ses points à l'aide du réticule. Il peut par exemple s'agir de son coin inférieur gauche. Connaissant la longueur en lambda d'une zone, le dessin des masques peut être divisé en zones de façon analogue à la fenêtre microscope.

### **A5.2.3 - Superposition de deux images sur le moniteur vidéo de ROMUALD**

Pour cela, le dessin des masques de l'une des zones est affiché, et laissé en permanence sur l'écran. L'image microscope correspondante est saisie, et se superpose à son dessin théorique. A la vue des décalages existants, le circuit est déplacé dans la chambre du MEB jusqu'à ce que l'image microscope obtenue coïncide avec son image théorique.

Cette coïncidence peut n'être qu'approximative pour une mise au point visuelle ; mais elle doit être effectuée au pixel près pour une mise au point automatique, car le calcul décrit ci-après est alors réalisé.

### A5.3 - Calcul de l'adresse image de tout point à observer

#### A5.3.1 - Calcul des coordonnées théoriques (xf, yf) du point dans la fenêtre

Le point est défini par ses coordonnées théoriques (xt, yt) relatives à l'origine du circuit. Les coordonnées du point inférieur gauche de la fenêtre dans ce repère sont connues et notées (xof, yof). L'origine est ramenée en ce point en soustrayant xof, yof respectivement à xt et yt, ce qui est représenté en figure A5.2.

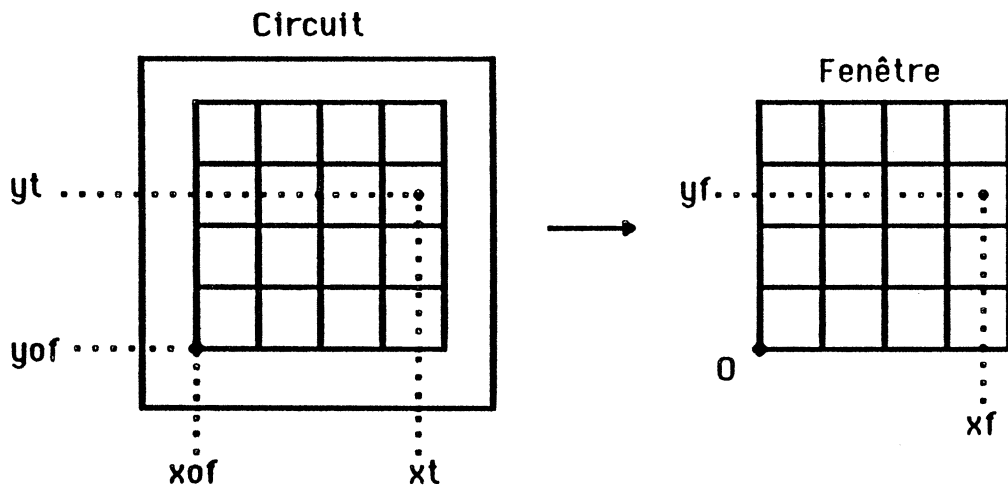


Figure A5.2 - Calcul des coordonnées théorique du point dans sa zone :

$$xzt = xf - xof$$

$$yzt = yf - yof$$

#### A5.3.2 - Détermination du numéro (nolig, nocol) de la zone à laquelle appartient le point

Il est calculé à partir des coordonnées fenêtre (xf, yf) du point et de la longueur nbl d'une zone en lambda.

### A5.3.3 - Calcul des coordonnées théoriques (xzt, yzt) du point dans la zone à laquelle il appartient

Soient (xoz, yoz) les coordonnées fenêtre du point inférieur gauche de la zone considérée. L'origine est ramenée en ce point par soustraction de xoz et yoz respectivement à xf et yf. Ceci est schématisé en figure A5.3.

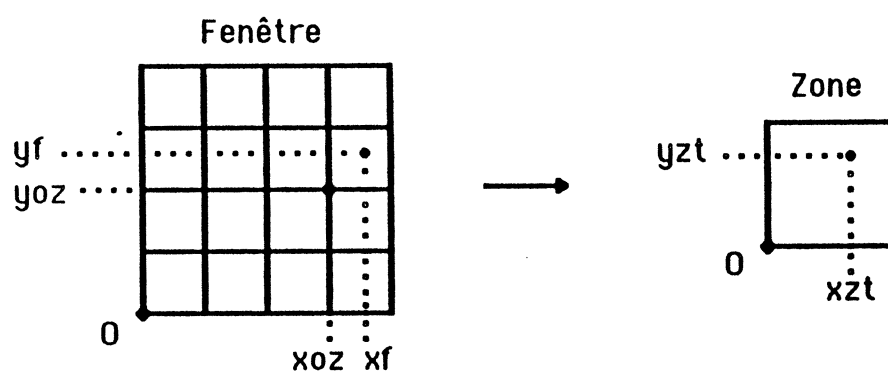


Figure A5.3 - Calcul des coordonnées théorique du point dans sa zone :

$$xzt = xf - xoz$$

$$yzt = yf - yoz$$

### A5.3.4 - Calcul des coordonnées pratiques (xzp, yzp) du point

Elles correspondent au nombre de pixel séparant le point à observer du point inférieur gauche de la zone, et sont donc comprises entre 1 et 512. L'origine étant ramenée au point supérieur gauche de la zone, (figure A5.4) et nbl lambda correspondant à 512 pixels, ces valeurs sont obtenues par les expressions suivantes :

$$xzp = (512 * xzt) / nbl$$

$$yzp = 512 - ((512 * yzt) / nbl)$$

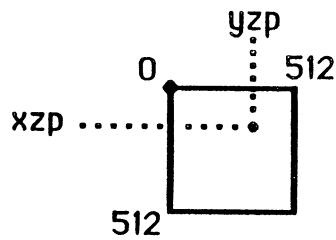


Figure A5.4 - Coordonnées pratiques du point  
dans la zone qui le contient

#### A5.3.5 - Calcul de l'adresse image du point

Le calcul de cette adresse, effectué à partir des coordonnées (xzp, yzp) du point est identique à celui décrit en 5.3.3.3.

**BIBLIOGRAPHIE**

- [BAI 82] BAILLE G., LAURENT J.  
"Etude du comportement interne des circuits VLSI par microscopie électronique"  
 Congrès AFCET : Architecture des machines et systèmes informatiques.  
 Université de Lille, Novembre 1982
- [BAR 84] BARRE M., BONNET J.  
"Système d'analyse fonctionnelle par faisceaux d'électrons"  
 Colloque ADESO : Qualité des composants électronique  
 Université de Bordeaux 1, Mars 1984
- [BAU 85] BAUDRAND F.  
"Présentation de TESSIE : testeur de circuits intégrés"  
 Rapport de contrat DAll. Laboratoire TIM3  
 Institut IMAG, Grenoble, Février 1985
- [BER 83] BERGER-SABBATEL G., COURTOIS B.  
"E-beam testing strategies for VLSI"  
 Rapport de recherche N° 371 Laboratoire TIM3  
 Institut IMAG, Grenoble, Mai 1983
- [BER 82] BERTSHE K.J., CHARLES H.K.  
"The practical implementation of voltage contrast as a diagnostic tool"  
 20th Reliability Physics, San Diégo, Mars 1982
- [BER 85] BERGHER L.  
"Analyse de défaillance de circuits VLSI par microscopie électronique à balayage"  
 Thèse Docteur-Ingénieur de l'INPG, Grenoble, Juin 1985
- [BRE 77] BRESSE J.F.  
"Quantitative use of the electron-beam induced current [EBIC] for the characterisation of semiconductor devices"  
 Scanning Electron Microscopy, Vol 1, 1977, p 683-693
- [BRE 80] BRETAGNOLLE B.Y., RUBAT DU MERAC C., JUTIER P.  
"Looking for parallelism in sequential algorithms :an assistance hardware design"  
 Ecole d'Eté de l'OTAN  
 Editeur G.C. Simon Haralick, 1980

- [BRE 84] BRETAGNOLLE B.Y.  
"Du traitement d'images dans ses rapports avec l'architecture des ordinateurs"  
 Thèse Docteur Ingénieur de l'INPG, Grenoble, Janvier 1984
- [COU 82] COURTOIS B.  
"Debugging VLSI using SEM"  
 Rapport de recherche N° 310 Laboratoire TIM3  
 Institut IMAG, Grenoble, Septembre 1982
- [FAZ 81] FAZEKAS P., FEUERBAUM H.P., WOLFGANG E.  
"Scanning electron beam probes VLSI chips"  
 Electronics, July 14, 1981, p 105-112
- [FEU 82] FEUERBAUM H.P.  
"E.beam testing : methods and applications"  
 Colloque Microcircuit Engineering : microlithography  
 Grenoble, Octobre 1982
- [FLE 74] FLEMING J.P., WARD E.W.  
"Voltage contrast linearization with a hemispherical retarding analyser"  
 Journal of Scientific Instruments, Vol 7, 1974
- [FOU 81] FOURNIER J.M., GEORGES A., RAINARD J.L.  
"Analyse en dynamique des circuits VLSI par stroboscopie au microscope électronique à balayage"  
 Note Technique NP/CNET/GRE/34  
 CNET/CNS Meylan, Grenoble, Mai 1981
- [HUE 84] HUET M., AUVERT G., FOURNIER J.M.  
"Localisation de défauts de tensions marginales de Circuits Intégrés par stimulation électronique ou photonique"  
 4e Colloque International de fiabilité et maintenabilité  
 Perros-Guirec (CNET Lannion), Mai 1984
- [JER 83] JERRAYA A.  
"Une nouvelle approche pour la vérification des masques des circuits intégrés"  
 Thèse Docteur Ingénieur de l'INPG, Grenoble, Novembre 1983
- [LAU 84] LAURENT J.  
"Projet ACIME: Analyse des Circuits Intégrés par Microscopie Electronique"  
 Thèse doctorat de l'INPG, Grenoble, Octobre 1984

- [MEB 78] FONTAINE G., VICARIO E., BRESSE J.F., DUPUY M., PHILIBERT J.  
"Micro analyse et microscopie électronique à balayage"  
Les Editions de Physique, Ouvrage présentant les cours de  
l'école d'été organisée à l'ENS d'Electrochimie et d'Electro  
Métallurgie de Grenoble (ENSEEG) en Septembre 1978.
- [MOL 80] MOLINS R.  
"Le microscope électronique à balayage et la micro analyse X à  
sélection d'énergie"  
Journal du Four Electrique, N° 6 et 7, 1980
- [RUB 83] RUBAT DU MERAC C., JUTIER P., LAURENT J., COURTOIS B.  
"A new domain for image analysis : VLSI circuits testing,  
with ROMUALD, specialized in parallel image processing"  
B.P.R.A. (British Pattern Recognition Association)  
University of Oxford England, September 1983
- [RUB 84] RUBAT DU MERAC C., BAILLE G., COURTOIS B.  
"Inspection automatique de la qualité des circuits intégrés"  
Colloque ADESO : Qualité des composants électroniques  
Université de Bordeaux 1, Mars 1984
- [SAR 82] SARRASIN M.T.  
"Analyse et implémentation du logiciel d'un synthétiseur d'images"  
Thèse CNAM, Grenoble, Octobre 1982
- [SHA 81] SHAVER D.C  
"Electron beam testing and restructuring of IC's"  
VLSI Memo 81-42  
Massachusetts Institute of Technology, February 1981



