



HAL
open science

Conception d'une mémoire reconfigurable intégrée sur tranche

Bassam Nasreddine

► **To cite this version:**

Bassam Nasreddine. Conception d'une mémoire reconfigurable intégrée sur tranche. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1988. Français. NNT: . tel-00327331

HAL Id: tel-00327331

<https://theses.hal.science/tel-00327331>

Submitted on 8 Oct 2008

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

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

THESE

Présentée par

Bassam NASREDDINE

Pour obtenir le titre de **DOCTEUR**

de **L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE**

(Spécialité : Microélectronique)

CONCEPTION D'UNE MEMOIRE RECONFIGURABLE INTEGREE SUR TRANCHE

Date de soutenance : le 6 juillet 1988

Composition du jury :

P. GENTIL	Président
F. DEVOS	Examineurs
F. JUTAND	
C. LANDRAULT	
G. SAUCIER	
J. TRILHE	



sident: Georges LESPINARD

Année 1988

Professeurs des Universités

RIBAUD	Michel	ENSERG	JOUBERT	Jean-Claude	ENSPG
RRAUD	Alain	ENSIEG	JOURDAIN	Geneviève	ENSIEG
JDELET	Bernard	ENSPG	LACOUME	Jean-Louis	ENSIEG
AUFILS	Jean-Pierre	ENSEEG	LESIEUR	Marcel	ENSHMG
MAN	Samuel	ENSERG	LESPINARD	Georges	ENSHMG
DOCH	Daniel	ENSPG	LONGQUEUE	Jean-Pierre	ENSPG
IS	Philippe	ENSHMG	LOUCHET	François	ENSIEG
NETAIN	Lucien	ENSEEG	MASSE	Philippe	ENSIEG
UVARD	Maurice	ENSHMG	MASSELÔT	Christian	ENSIEG
ISSONNEAU	Pierre	ENSIEG	MAZARE	Guy	ENSIMAG
UNET	Yves	IUFA	MOREAU	René	ENSHMG
ILLERIE	Denis	ENSHMG	MORET	Roger	ENSIEG
VAIGNAC	Jean-François	ENSPG	MOSSIERE	Jacques	ENSIMAG
ARTIER	Germain	ENSPG	OBLED	Charles	ENSHMG
ENEVIER	Pierre	ENSERG	OZIL	Patrick	ENSEEG
ERADAME	Herve	UFR PGP	PARIAUD	Jean-Charles	ENSEEG
OVET	Alain	ENSERG	PERRET	René	ENSIEG
HEN	Joseph	ENSERG	PERRET	Robert	ENSIEG
UMES	André	ENSERG	PIAU	Jean-Michel	ENSHMG
RVE	Félix	ENSHMG	POUPOT	Christian	ENSERG
LLA-DORA	Jean-François	ENSIMAG	RAMEAU	Jean-Jacques	ENSEEG
PORTES	Jacques	ENSPG	RENAUD	Maurice	UFR PGP
LMAZON	Jean-Marc	ENSERG	ROBERT	André	UFR PGP
RAND	Francis	ENSEEG	ROBERT	François	ENSIMAG
RAND	Jean-Louis	ENSIEG	SABONNADIÈRE	Jean-Claude	ENSIEG
GGIA	Albert	ENSIEG	SAUCIER	Gabrielle	ENSIMAG
NLUPT	Jean	ENSIMAG	SCHLENKER	Claire	ENSPG
ULARD	Claude	ENSIEG	SCHLENKER	Michel	ENSPG
NDINI	Alessandro	UFR PGP	SILVY	Jacques	UFR PGP
UBERT	Claude	ENSPG	SIRYES	Pierre	ENSIMAG
NTIL	Pierre	ENSERG	SOHM	Jean-Claude	ENSEEG
EVEN	Hélène	IUFA	SOLER	Jean-Louis	ENSIMAG
ERIN	Bernard	ENSERG	SOUQUET	Jean-Louis	ENSEEG
YOT	Pierre	ENSEEG	TROMPETTE	Philippe	ENSHMG
NES	Marcel	ENSIEG	VEILLON	Gérard	ENSIMAG
USSAUD	Pierre	ENSIEG	ZADWORNÝ	François	ENSERG

Professeur Université des Sciences Sociales (Grenoble II)

LLIET Louis

Personnes ayant obtenu le diplômed'HABILITATION A DIRIGER DES RECHERCHES

CKER	Monique	DEROO	Daniel	HAMAR	Roger
DER	Zdenek	DIARD	Jean-Paul	LADET	Pierre
ASSERY	Jean-Marc	DION	Jean-Michel	LATOMBE	Claudine
OLLET	Jean-Pierre	DUGARD	Luc	LE CORREC	Bernard
EY	John	DURAND	Madeleine	MADAR	Roland
LINET	Catherine	DURAND	Robert	MULLER	Jean
MAULT	Christian	GALERIE	Alain	NGUYEN TRONG	Bernadette
RNUEJOLS	Gérard	GAUTHIER	Jean-Paul	PASTUREL	Alain
ULOMB	Jean-Louis	GENTIL	Sylviane	PLA	Fernand
LARD	Francis	GHIBAUDO	Gérard	ROUGER	Jean
NES	Florin	HAMAR	Sylviane	TCHUENTE	Maurice
				VINCENT	Henri

CHERCHEURS DU C.N.R.S

Directeurs de recherche 1ère Classe

CARRE
FRUCHART
HOPFINGER
JORRAND

René
Robert
Emile
Philippe

LANDAU
VACHAUD
VERJUS

Ioan
Georges
Jean-Pierre

Directeurs de recherche 2ème Classe

ALEMANY
ALLIBERT
ALLIBERT
ANSARA
ARMAND
BERNARD
BINDER
BONNET
BORNARD
CAILLET
CALMET
COURTOIS
DAVID
DRIOLE
ESCUDIER
EUSTATHOPOULOS
GUELIN
JOJD

Antoine
Colette
Michel
Ibrahim
Michel
Claude
Gilbert
Roland
Guy
Marcel
Jacques
Bernard
René
Jean
Pierre
Nicolas
Pierre
Jean-Charles

KLEITZ
KOFMAN
KAMARINOS
LEJEUNE
LE PROVOST
MADAR
MERMET
MICHEL
MUNIER
PIAU
SENATEUR
SIFAKIS
SIMON
SUERY
TEODOSIU
VAUCLIN
WACK

Michel
Walter
Georges
Gerard
Christian
Roland
Jean
Jean-Marie
Jacques
Monique
Jean-Pierre
Joseph
Jean-Paul
Michel
Christian
Michel
Bernard

Personnalités agréées à titre permanent à diriger
des travaux de recherche (décision du conseil scientifique)

ENSEEG

CHATILLON
HAMMOU
MARTIN GARIN

Christian
Abdelkader
Régina

SARRAZIN
SIMON

Pierre
Jean-Paul

ENSERG

BOREL

Joseph

ENSIEG

DESCHIZEAUX
GLANGEAUD

Pierre
François

PERARD
REINISCH

Jacques
Raymond

ENSHMG

ROWE

Alain

ENSIMAG

COURTIN

Jacques

EFP

CHARUEL

Robert

C.E.N.G

CADET
COEURE
DELHAYE
DUPUY
JOUVE
NICOLAU

Jean
Philippe
Jean-Marc
Michel
Hubert
Yvan

NIFENECKER
PERROUD
PEUZIN
TAIEB
VINCENDON

Hervé
Paul
Jean-Claude
Maurice
Marc

Laboratoires extérieurs :

C.N.E.T

DEVINE
GERBER

Rodericq
Roland

MERCKEL
PAULEAU

Gérard
Yves



Je tiens à exprimer toute ma reconnaissance à madame Gabrièle SAUCIER, professeur à l'ENSIMAG, pour avoir bien voulu m'accueillir dans son laboratoire de recherche et pour avoir encadré mon travail pendant ces trois dernières années.

Je tiens à remercier:

Monsieur Pierre GENTIL, professeur à l'ENSERG et directeur du CIME pour m'avoir fait l'honneur d'accepter de présider le jury de cette thèse,

Monsieur Francis JUTAND, professeur à l'Ecole Nationale Supérieure de Télécommunication (Paris), d'avoir accepté d'être rapporteur de cette thèse,

Monsieur Christian LANDRAULT, directeur de recherche au LAMM, d'avoir accepté d'être rapporteur de cette thèse,

Monsieur Francis DEVOS, professeur à l'IEF, d'avoir accepté de faire partie de ce jury,

Monsieur Jacques TRILHE, docteur ingénieur à la direction technique de SGS Thomson microélectronics et responsable du projet ESPRIT 824-WSI, pour sa précieuse collaboration depuis le début de ce travail . Je le remercie également d'avoir accepté de faire partie de ce jury,

Je remercie également tous mes collègues de l'Unité de Recherche sur la Conception de Systèmes Intégrés pour l'ambiance de travail agréable qu'ils ont su faire régner au sein de l'équipe, et tout particulièrement Edmond KOUKA, Y-shu WANG et Dominique MARRON pour leur précieuse collaboration .

Ce travail a été réalisé dans le cadre du projet ESPRIT-824 (Wafer Scale Integration) financé par la Communauté Economique Européenne (CEE).



**A mes Parents,
A mon frère Hosni**



*CONCEPTION D'UNE MEMOIRE
RECONFIGURABLE INTEGREE
SUR TRANCHE*



INTRODUCTION



L'augmentation de l'intégration des circuits MOS peut se faire :

- en diminuant la taille de grille des transistors MOS,
- en augmentant la dimension des puces.

Dans le cas du transistor MOS, la diminution de la dimension d'une grille est limitée par la valeur minimale de la charge stockée dans cette grille qui doit rester supérieure au bruit dû à l'agitation thermique des électrons [TRI88]. La taille d'un circuit est en pratique limitée par la surface maximale d'une tranche (sauf les circuit 3D). Les circuits intégrés sur une tranche entière appelés WSI (Wafer Scale Integration), pourraient atteindre 30 à 40 millions de transistors effectifs parmi 50 à 60 millions de transistors intégrés sur tranche de 5 pouces. Le but de cette étude est d'étudier leur faisabilité.

Les avantages de ce type d'intégration poussé à l'extrême sont, tout d'abord la diminution du coût du système-WSI, dû à l'économie du nombre de boîtiers utilisés. En effet, un système intégré sur tranche se monte sur un seul boîtier (un système équivalent en circuit imprimé nécessite quelques dizaines de boîtiers classiques montés sur plusieurs cartes de circuits imprimés). Ceci réduit le coût global du système dans la mesure où le prix d'un boîtier-WSI reste raisonnable. D'autre part, la diminution du nombre de boîtiers permet une très bonne compaction. En effet un système réalisé avec des circuits imprimés occupe une surface 6 à 7 fois plus grande que celle occupée par un système intégré sur tranche.

La réduction du nombre de boîtiers offre deux autres avantages:

- la diminution du nombre d'amplificateurs d'entrée/sortie qui implique une baisse de consommation de courant et une augmentation de la vitesse de fonctionnement du système.

- Le dernier avantage et peut être le plus important est l'augmentation de la fiabilité due à la diminution du risque d'oxydation et des mauvaises soudures des contacts puces-boîtiers et/ou boîtiers-carte imprimée.

Il existe d'importantes difficultés à surmonter pour réussir un produit WSI; Tout d'abord le problème de la baisse de rendement due d'une part, à l'augmentation de la taille de circuit, et d'autre part à l'utilisation de longues lignes augmentant les risques de coupure et de court-circuit. Il faut que le rapport nombre de transistors effectifs /surface occupée par le circuit soit le plus grand possible. Cet objectif peut être atteint par l'utilisation de la redondance à différents niveaux du circuit permettant de tolérer les défauts de fin de fabrication.

Le deuxième problème est lié à la propagation des signaux sur des distances qui quelques fois dépassent plusieurs cm (9 cm) sur des lignes R, L, C dont l'impédance est plus importante que dans le cas des circuits imprimés.

Enfin, le troisième problème est celui de la fabrication des boîtiers à l'échelle de la tranche. En effet, un boîtier WSI doit avoir un nombre de broches qui peut dépasser parfois les 800 broches. Il doit dissiper uniformément et le plus rapidement la chaleur produite par le circuit en fonctionnement. Cette chaleur sera d'autant plus importante que le circuit sera plus rapide [VAL86]. La mauvaise dissipation de la chaleur a d'autres conséquences non négligeables telles que la dilatation de la tranche dans le boîtier qui risque d'entraîner sa cassure ou son décollage.

La prise en compte des problèmes cités ci dessus va dans le sens de l'obtention de systèmes fiables et performants.

Dans le chapitre 1, nous étudierons les différentes architectures des circuits WSI et en particulier les architectures des mémoires.

Le chapitre 2 sera consacré à l'étude de faisabilité ainsi qu'aux contraintes liées aux différentes étapes de réalisation de la mémoire.

Le chapitre 3 traite de l'architecture et de la réalisation de la mémoire-WSI.

Enfin on donnera dans le chapitre 4 la stratégie de test et de configuration à différents niveaux de la mémoire intégrée sur tranche.



CHAPITRE I

ARCHITECTURE WSI



I. INTRODUCTION

La réalisation d'un circuit WSI repose sur l'intégration de puces élémentaires sur une tranche de silicium. La principale difficulté est due au rendement de fabrication de ce circuit dont la taille implique l'existence de défauts, d'où la nécessité d'une tolérance aux défauts de fin de fabrication qui dépend de la structure du circuit considéré.

Dans ce chapitre, après un rappel du principe de tolérance aux défauts, seront étudiées les architectures intégrables sur tranche.

II. LA TOLERANCE AUX DEFAUTS

La tolérance aux défauts de fabrication repose sur la prévision d'un certain nombre d'éléments de réserve (de redondance) destinés à remplacer les éléments défectueux d'un circuit. Ce principe de remplacement nécessite donc la détection et la localisation des défauts sur un élément défectueux et réparable.

La stratégie de redondance est fortement liée à l'architecture du circuit et à l'apparition (où à la distribution) des défauts sur un circuit. En effet le principe de redondance consiste à chercher dans un circuit les différentes structures régulières formées d'éléments corrigibles. Ces structures sont fonction de l'architecture du circuit (par exemple, pour une mémoire SRAM, le plan mémoire est une structure régulière). Il s'agit ensuite d'ajouter pour chaque structure régulière des éléments de réserve dont le nombre est fonction de la distribution des défauts [MOO86].

La technique de remplacement des éléments défectueux par des éléments sains, nécessite l'utilisation de dispositifs de connexion ou de déconnexions, tels que les fusibles, les anti-fusibles et les transistors à grilles flottantes. Ces dispositifs seront étudiés plus en détail dans le chapitre II.

III. ARCHITECTURES INTEGRABLES SUR TRANCHE

La recherche menée dans le domaine de l'intégration depuis une dizaine d'années a conduit à l'apparition de produits divers.

III.1. L'APPROCHE "HYBRIDE"

La solution hybride proposée par des sociétés américaines telles que "MOSAIC SYSTEM" permet d'interconnecter des puces sur une tranche de silicium en utilisant une matrice d'interconnexion standard intégrée sur cette tranche, la configuration électrique de la matrice est réalisée par la programmation des anti-fusibles. Dans cette technique, on utilise des puces testées et découpées et ensuite collées sur la tranche avec la possibilité de pouvoir coller une deuxième, pour chacune, en remplacement dans le cas d'un endommagement des premières. Cette solution offre des avantages de gain en surface par rapport aux circuits imprimés, mais son coût est élevé.

III.2. L'APPROCHE "WSI"

Le principe général de réalisation d'un produit WSI consiste à intégrer sur tranche les puces formant le circuit WSI et les réseaux d'interconnexion sans utiliser de masques supplémentaires personnalisés. Il s'agit ensuite d'immerger les puces intégrées sur tranche dans le réseau d'interconnexion. Les puces peuvent être identiques pour les systèmes dits répétitifs (mémoires, réseaux systoliques,...) ou différentes pour les systèmes non répétitifs (systèmes à base de microprocesseurs...)

III.2.1. Les systèmes WSI non répétitifs

Exemple : Les systèmes à base de microprocesseurs

Ces systèmes pourraient permettre d'intégrer une carte complète de circuits sur une surface nettement moindre, avec des avantages évidents en fiabilité, consommation et rapidité.

Cette structure comporte un certain nombre de puces prédéfinies telles que, microprocesseurs, ROM, RAM, périphériques personnalisés...immergées dans une structure d'interconnexion flexible qui doit permettre la configuration d'un système complet en fin de fabrication (voir figure 1.1).

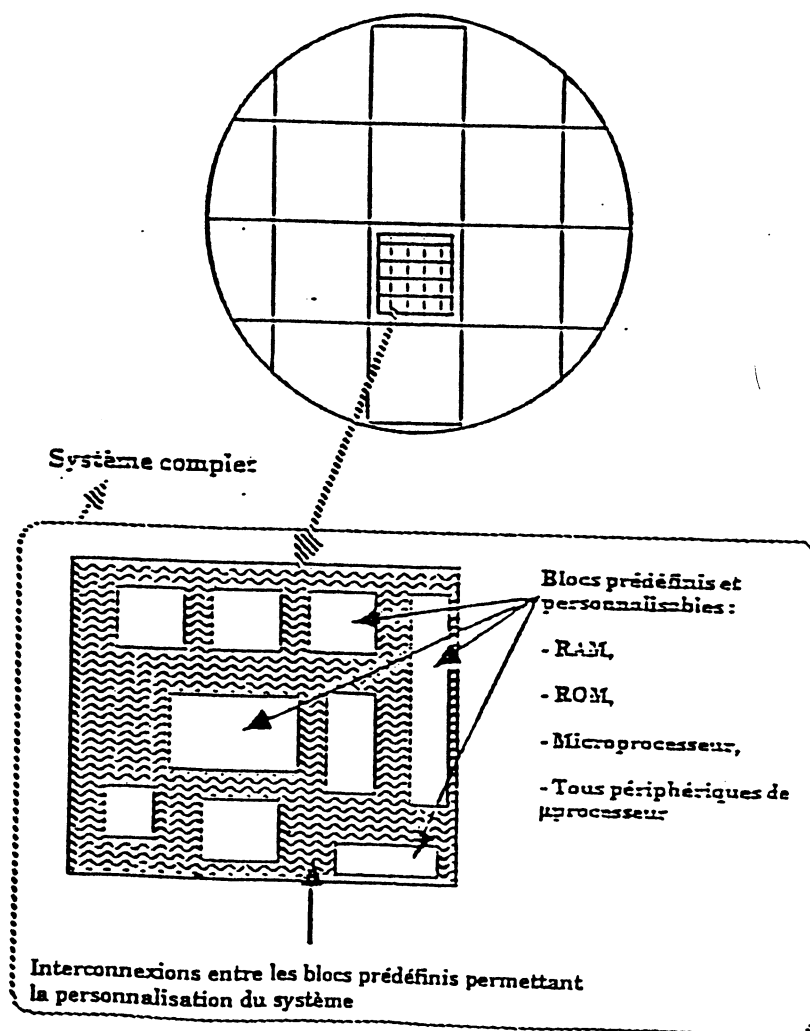


figure 1.1 : systèmes à base de microprocesseurs

Le problème essentiel à résoudre dans cette approche concerne le faible rendement du système complet du au faible rendement des puces composant le système. Pour remédier à ce problème, il faut adopter une stratégie de reconfiguration à deux niveaux, qui consiste à :

- dupliquer massivement des puces de petites tailles.
- utiliser des puces avec redondance, ceci concerne les puces de grande taille.

La duplication des puces consiste à intégrer des puces de réserve qui ont pour rôle de remplacer les puces défectueuses. Il faut donc qu'elles soient le plus près des puces défectueuses et dotées de dispositifs de reconfiguration et de remplacement.

L'utilisation de la redondance dans les puces a pour but de reconfigurer une puce jugée défectueuse après le premier test. La reconfiguration d'une puce de type microprocesseur consiste à reconfigurer la partie opérative [GEN87] et la partie contrôle [DAN86] par l'ajout de bloc de réserve dans les deux parties.

III.2.2. Les systèmes WSI répétitifs : réseaux systoliques mémoires

Contrairement aux systèmes répétitifs, les puces utilisées sont identiques, chacune réalisant une tâche précise. Par exemple, dans le cas d'un réseau systolique intégré sur tranche, les puces utilisées, dites Processeur Élémentaire (PE) comportent une unité de calcul et éventuellement des petites mémoires pour stocker les coefficients qui servent pour le calcul. Dans le cas de la mémoire on utilise des puces mémoire.

* *Les réseaux systoliques [LEI85], [MOR86]*

L'idée est d'intégrer sur tranche des puces dites processeurs élémentaires (PEs) dans une structure souple d'interconnexion. Cette application est bien adaptée à l'intégration sur tranche car les communications se font entre les PEs voisins, et ainsi, le temps de propagation des signaux sur les grandes lignes d'interconnexion est réduit. Plusieurs études sont en cours sur ce sujet, nous citerons en exemple les circuits pour le calcul parallèle tels que le "CHIP" (Configurable Highly Parallel) développé à la North Carolina University par K.Hedlund [HED82], les circuits spécialisés pour le traitement d'image tels que le WASP de l'université de Brunel, au Royaume Uni [LEA85] et le circuit développé au sein du projet esprit 824-WSI pour faire de la compaction d'image ou de la transformation géométrique d'image en temps réel [SYS87] (voir figure 1.2). Ce dernier circuit intègre quelques 18000 processeurs élémentaires.

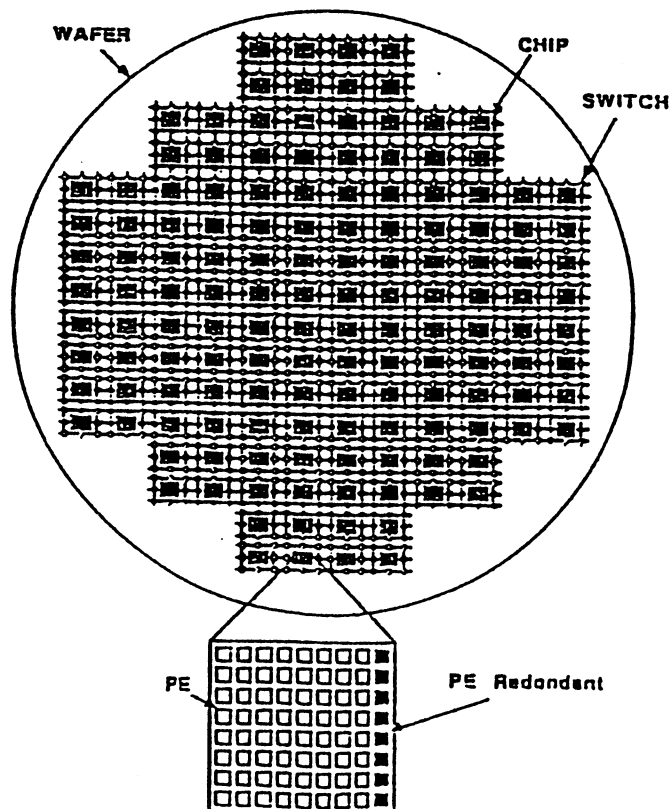


figure 1.2 : réseaux systolique à base de "ASAP"

Chaque processeur élémentaire (PE), appelé ASAP, de type GAPP NCR se compose d'une UAL de 1bit, d'une mémoire RAM séparée en deux blocs de 64 bits indépendamment adressables, de 5 registres et de 6 multiplexeurs.

La tolérance aux défauts en fin de fabrication est assurée par une reconfiguration à deux niveaux :

- la reconfiguration au niveau du "chip", les PEs sont groupés en matrice (chip) avec une colonne de redondance qui permet de récupérer un PE défectueux par ligne,
- la reconfiguration au niveau de la tranche ; un réseau de commutateur permet de contourner les chips défectueux.

* *Les mémoires*

Les mémoires ont été les circuits le plus étudié en WSI, ceci pour diverses raisons ; d'abord pour la demande sans cesse croissante des dispositifs de stockage dans la technologie de pointe. D'autre part pour la facilité d'implantation des circuits mémoires sur une tranche entière.

La plupart des circuits mémoires déjà réalisés ou en cours de réalisation sur tranche sont des mémoires vives statiques (SRAM). En effet une mémoire SRAM nécessite des cellules mémoires élémentaires composées de six transistors ce qui implique qu'une SRAM supérieure à quelque Mbits dépasserait le dizaine de millions de transistors qui ne peuvent pas être implantés sur une surface normale de puce (de l'ordre de 100 mm^2). Par contre sur une pareille surface, on arrive à implanter des mémoires dynamiques (DRAM) dont la cellule mémoire élémentaire est composée d'un seul transistor [LEW85] (une firme japonaise a réalisé une mémoire DRAM de 4 Mbits en technologie CMOS, sur une surface de 95 mm^2 de silicium [YAM88]).

Nous allons présenter quelque mémoires-WSI, et les techniques utilisées pour leurs réalisations.

Mémoire NTT

NTT a réalisé en 1980 une mémoire ROM de 4 Mbits sur une tranche de 4 pouces, et une autre SRAM en 1984 de 1.5 Mbits sur une tranche de 4 pouces [UEO84]. La même stratégie a été appliquée dans les deux cas. Nous allons résumer l'architecture de la 1.5 Mbits SRAM ; elle est organisée en blocs de 256k de 6 bits, avec duplication massive de chaque bloc et possibilité de contrôle de parité de deux blocs similaires. En écriture, les informations sont écrites dans les deux blocs semblables, et inversement en lecture, avec choix de l'information ayant la bonne parité. Une telle organisation assure non seulement la tolérance aux défauts de fabrication, mais aussi aux pannes survenant durant la vie du circuit. Cette mémoire présente quelques inconvénients ; d'abord elle n'est pas de grande capacité (1.5 Mbits), ensuite, elle n'est pas très rapide (avec un temps d'accès de 520 ns).

Mémoire Anarmatic

La société Sinclair a réalisé une mémoire série de 0.5 Mbits DRAM (en technologie NMOS sur tranche de 4 pouces) pour remplacer le disque mémoire "Winchester" utilisé dans les micro-ordinateurs. Le temps d'accès annoncé est de 10 μ s, ce qui est largement suffisant pour une telle application.

Anarmatic, a repris l'activité mémoire de Sinclair et elle a annoncé une DRAM de 7 Mbits sur une tranche de 5 pouces. La stratégie de configuration utilisée est inspirée de Aubusson et Catt, elle consiste à construire une spirale en utilisant seulement les éléments de connexion bons [AUB78], ensuite les cellules mémoires saines sont connectées à la spirale. La formation de cette spirale est effectuée à chaque mise sous tension de la mémoire. Cette stratégie de configuration comme dans le cas de la mémoire NTT tolère les défauts de fabrication ainsi que les pannes survenant durant la vie de la mémoire.

Mémoire INOVA

INOVA est une société américaine qui fabrique surtout des mémoires intégrées sur tranche en technologie CMOS. Elle a réalisé en 1985 une mémoire SRAM de 1 Mbits (avec un temps d'accès de 150 ns) à partir de 144 cellules mémoires de 16 kbits sur une tranche de 4 pouces [BEN85], [HAW85]. Une autre mémoire de 24 Mbits a été annoncée (avec un temps d'accès de 65ns) à partir des blocs de 256k x 1bits sur une tranche de 4 pouces organisée en 3 Mbits x 8bits. Pour les deux mémoires la stratégie de configuration est la même, par exemple, la mémoire 24 Mbits se configure à partir des blocs de 256k x 1bit, le système de décodage est implanté dans la colonne centrale de la tranche. Les blocs 256k sont redondants car un bloc peut être utilisé à moitié avec un autre demi bloc. La configuration matérielle de la mémoire (blocs) est réalisée par laser. Avec cette stratégie INOVA obtient un rendement global de 40 à 60%.

Ceci nous permet d'introduire le but de notre étude : Il s'agit de concevoir une mémoire-WSI vive statique de 4.5 Mbits intégrée sur tranche [NAS86]. L'intégration est faite en intégrant des puces de base de 64Kbits. La principale difficulté à résoudre sera d'obtenir un rendement suffisamment élevé pour la mémoire-WSI. Pour cela on adoptera une stratégie de reconfiguration à deux niveaux :

- reconfiguration au niveau des puces par l'ajout de colonnes redondantes dans le plan mémoire .
- reconfiguration au niveau de la tranche grâce à la prévision de puces de réserves.

CHAPITRE II

*ETUDE DE FAISABILITE ET
CONTRAINTE*



I. INTRODUCTION

Ce chapitre est consacré à l'étude de faisabilité de la mémoire 4.5 Mbits intégrée sur une tranche de 4". Cette mémoire doit répondre aux spécifications suivantes :

- Son format de données doit être de 18bits où 2x(8 bits+ 1bit de parité)
- Elle doit être rapide ; son temps d'accès ne doit pas dépasser 100ns.
- Au niveau électrique, elle doit être compatible avec les circuits TTL, alimentée par une tension de 5 volts. Sa consommation en courant doit être inférieure à 7 μ A au repos et inférieure à 3 Ampères en mode opération.
- Enfin, elle doit être facilement testable. Le temps de test ne doit pas excéder 10 minutes sur tranche, et 3 minutes en boîtier.

La première partie de ce chapitre est dédiée aux différentes étapes de réalisation de la mémoire. La seconde concerne les contraintes liées aux étapes d'implantation et de test.

II. LES DIFFERENTES ETAPES DE REALISATION

II.1. ORGANISATION VIRTUELLE

La mémoire 4.5 Mbits est réalisée à partir des puces mémoires SRAM 64kx1bit reconfigurables. Cette réalisation nécessite 72 bonnes mémoires 64k (72 x 64k x 1bit = 4.5 Mbits) organisées en 18 blocs de 256kx1bit [INT86]. Chaque bloc est donc constitué de 4 mémoires 64kx1bit connectées à la même ligne de bit (voir figure 2.1).

Bloc de 256kx1bit

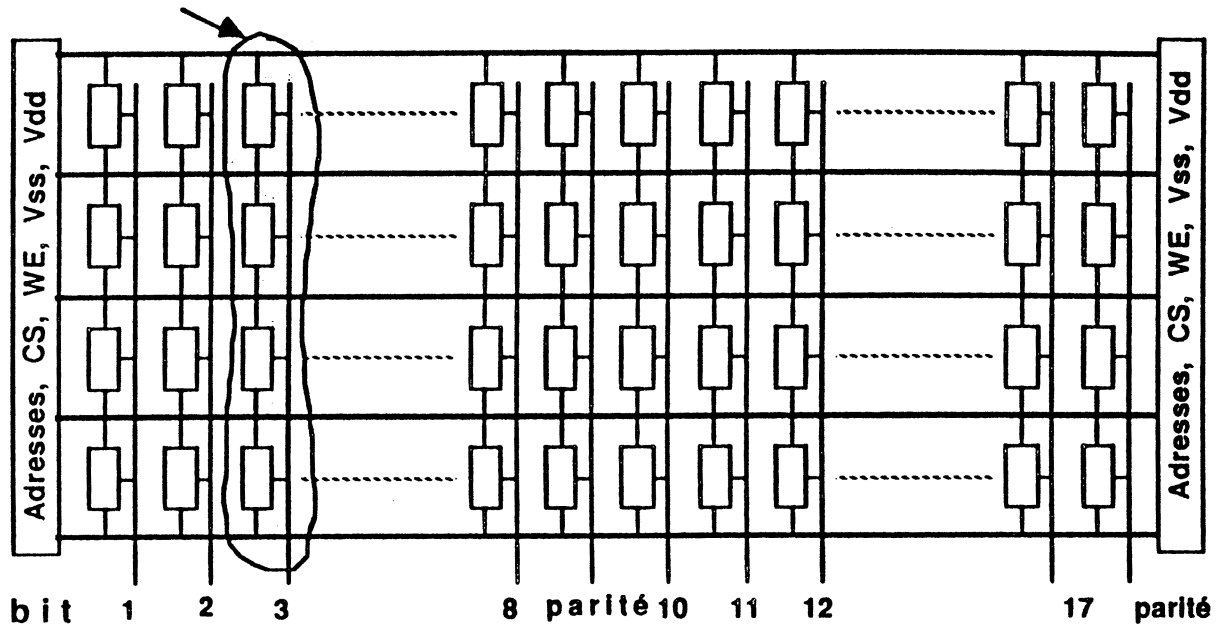


figure 2.1 : Organisation virtuelle de la mémoire 4.5 Mbits

Cette organisation nécessite un dispositif qui sélectionne à chaque fois 64k x18bit. Ce qui revient à sélectionner dans chaque bloc une mémoire 64k parmi 4. Nous avons pour ce faire proposé une solution dont l'originalité consiste en l'intégration d'un décodeur (1/4) programmable à proximité (ou à l'intérieur de chaque mémoire 64k), cette solution est illustrée dans la figure suivante :

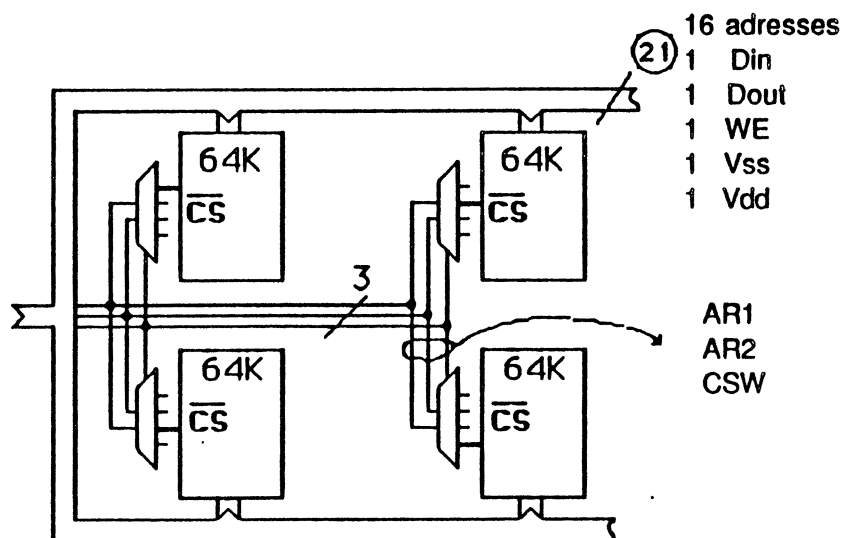


figure 2.2 : bloc de quatre mémoires 64k

Ce décodeur permet l'économie de consommation de la mémoire-WSI du

fait qu'il ne sélectionne dans tous les cas de fonctionnement que le quart de la tranche.

D'autre part, toutes les mémoires 64K sont adressées et contrôlées en même temps et seules les mémoires sélectionnées (18 mémoires) par le décodeur peuvent communiquer avec l'extérieur (écriture/lecture) à travers les lignes de bits (18 lignes). Cette structure nécessite donc des réseaux d'interconnexion pour accéder et alimenter les mémoires 64k, et des dispositifs de connexion/déconnexion pour connecter ou non les mémoires 64k aux réseaux d'interconnexion.

Les réseaux d'interconnexion nécessaires pour cette structure sont :

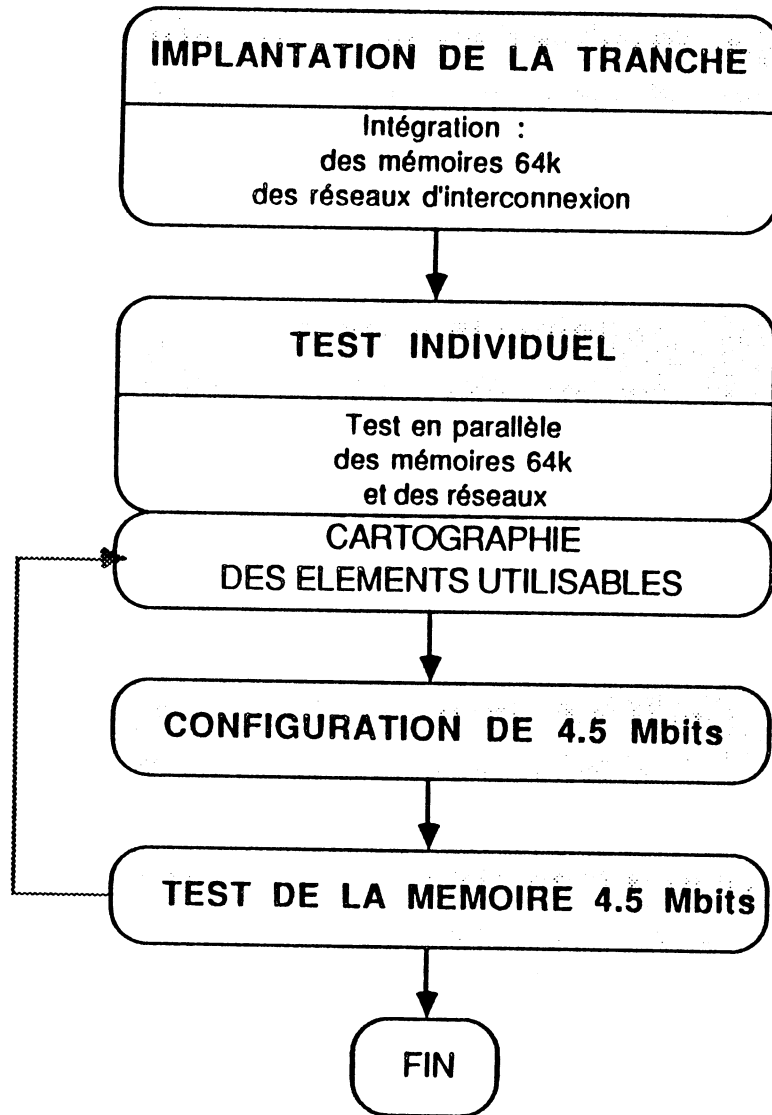
- Deux réseaux d'alimentations : Vdd, Vss*.
- 16 réseaux d'adresses : A0-->A15.
- Deux réseaux de données : Din, Dout.
- Quatre réseaux de sélection et contrôle : AR1, AR2, CSW*, WE*.

Un réseau d'interconnexion peut atteindre une capacité totale de 900 pF et une résistance locale élevée d'où la nécessité d'avoir des réseaux où ces paramètres sont minimum.

II.2. ETAPES GENERALES DE REALISATION

L'organisation virtuelle des mémoires 64k et des réseaux d'interconnexion nous donne une implantation idéale de la mémoire-WSI sur une tranche. Mais en réalité les mémoires et les réseaux utilisables (qui n'ont aucun défaut) pour la formation de la mémoire-WSI sont déterminés après une étape de test. On remarque que ces éléments utilisables ne sont pas voisins, mais dispersés aléatoirement sur la tranche, d'où la nécessité d'une stratégie intelligente de configuration de la mémoire-WSI.

Les différentes phases de réalisation de la mémoire 4.5 Mbits sont résumées dans l'organigramme suivant :



A chaque étape principale correspondent des contraintes dont on doit tenir compte lors de la réalisation de la mémoire 4.5 Mbits. Ces contraintes sont discutées dans le paragraphe qui suit.

III.CONTRAINTES LIEES AUX DIFFERENTES ETAPES DE REALISATION

III.1. CONTRAINTES LIEES A L'IMPLANTATION

III.1.1. La Technologie HCMOS3

Lors de l'étude de la mémoire 64k à Thomson (Eurotechnique), il a été décidé de choisir la technologie HCMOS3 développée et mise au point par

Thomson Semiconducteur et par le LETI. Cette technologie est à double caisson N^{++} et P^{++} implantés sur un substrat dopé P^- [Annexe 2.1]. Elle présente l'avantage de minimiser la surface occupée par les circuits (grâce à sa haute densité d'intégration), ainsi que les phénomènes de latch up.

Les principales caractéristiques de cette technologie sont la présence de deux niveaux de métal Alu1 et Alu2, une largeur minimale de grille de 1.2 μm , et une largeur maximale de connexion de 25 μm .

III.1.2. La photo-répétition

Il existe deux techniques pour réaliser un produit WSI : la photo-composition et la photo-répétition. La photo-composition consiste à réaliser des motifs différents sur la tranche. Actuellement, aucune usine de production n'a réussi à fabriquer des produits WSI en photo-composition qui soient rentables et qui puissent remplacer les circuits imprimés. Pour la réalisation de la mémoire-WSI on a été amené à utiliser la technique de photo-répétition qui elle, est mieux maîtrisée. Elle consiste à avoir uniquement un seul réticule "photo-répété" côte à côte sur la tranche de façon à réaliser une jonction naturelle entre les réticules.

Cette technique implique plusieurs contraintes ; la principale contrainte est l'impossibilité de réaliser des motifs (amplificateurs, croix d'alignement ou de laser, ...) en dehors du réticule photo-répété. D'où la règle générale: tous les motifs nécessaires pour former la mémoire doivent être inclus dans le réticule de photo-répétition. Cette règle est valable pour tous les motifs sauf pour les plots de soudure de la tranche qui sont réalisés en Alu2 par un masque à l'échelle 1.

La photo-répétition présente de nombreux avantages : elle est sûre et simple car elle permet de réduire le nombre de réticules ainsi que les problèmes d'alignement, elle permet également de "photo-répéter" des

surfaces qui peuvent atteindre $16 \times 16 \text{ mm}^2$. Par contre, les grands réticules causent une perte de surface sur le bord de la tranche (par exemple ; la surface perdue au bord d'une tranche de 4 pouces est égal à 35% de la surface de la tranche pour un réticule de $16 \times 16 \text{ mm}^2$, et 18% pour un réticule de $8 \times 8 \text{ mm}^2$), d'où la nécessité d'un compromis entre la taille d'un réticule et la surface perdue sur le bord de la tranche.

III.1.3. Dispositifs de reconfiguration

L'intégration à l'échelle de la tranche impose l'utilisation de puces de base à haut rendement (assuré par l'utilisation d'éléments redondants interconnectés par interrupteurs), et des réseaux reconfigurables. Ces objectifs peuvent être atteints par l'utilisation de dispositifs de commutation qui jouent le rôle d'interrupteurs. Actuellement on peut distinguer deux types de dispositifs; ceux programmables par un moyen physique externe et ceux programmables par logique à partir des plots externes.

**** Dispositifs directement programmables :***

Transistors à grille flottante

Il s'agit de transistors MOS dont la grille est isolée, programmés par un faisceau d'électrons (dont le courant et la haute tension sont contrôlés) qui apporte des charges négatives aux grilles [SHA84], [GIR86]. La déprogrammation se réalise par irradiation Ultra-Violet (UV). Il y a deux genres de transistors qui répondent aux conditions de reproductibilité de programmation/ déprogrammation, ainsi qu'à la rétention des charges sur la grille : les transistors MOS canal N à déplétion et les transistors MOS canal P enrichis. Les transistors canal N déplétés sont passants après fabrication et peuvent devenir bloqués par l'action localisée d'un faisceau d'électrons sur leur grilles (inversement pour le TMOS canal P enrichi).

TMOS	Après fabrication	Programmé	Déprogrammé
N déplété	Passant $\xrightarrow{+e^-}$	Bloqué $\xrightarrow{+UV}$	Passant
P enrichi	Bloqué $\xrightarrow{+e^-}$	Passant $\xrightarrow{+UV}$	Bloqué

Les avantages des TMOS à grilles flottantes sont :

- ils offrent la possibilité de reprogrammation,
- leur coût est faible du fait que leur réalisation ne nécessite pas d'étapes technologiques supplémentaires,
- la surface d'un transistor à grille flottante est comparable à celle d'un transistor normal.

Les inconvénients majeurs sont :

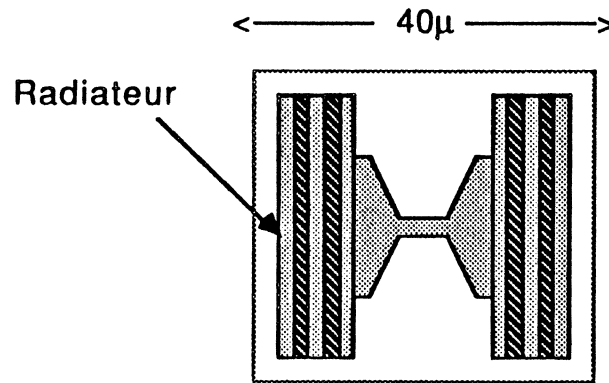
- actuellement le nombre de programmation/déprogrammation est limité à 3. Au delà, on a une détérioration (dégradation de la qualité) du transistor.
- actuellement la déprogrammation par UV n'est pas localisée, la tranche entière est insolée par UV, donc la déprogrammation voulue d'un transistor à grille flottante quelconque implique la déprogrammation non désirée des autres transistors à grille flottante existants sur la tranche.
- la rétention des charges sur la grille n'est pas certaine, elle est fonction du temps et de la température (le temps de rétention est de l'ordre de 10 ans à 50 °C, et de quelques heures à 150 °C) [SHA84].

Les fusibles et anti-fusibles

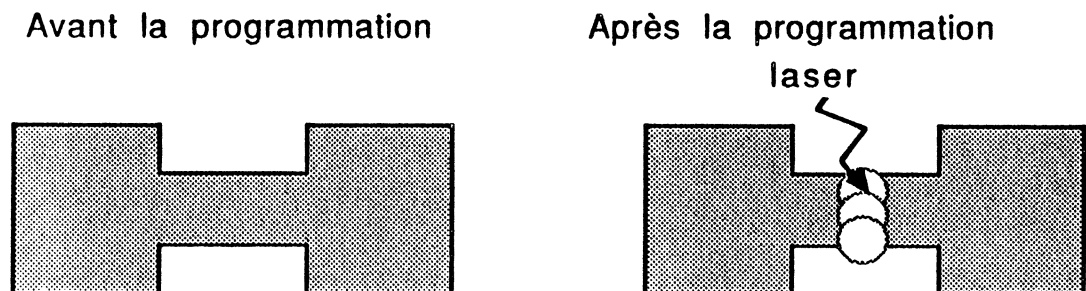
Un fusible est une connexion métallique que l'on fait fondre pour réaliser la déconnexion entre deux lignes. Les fusibles les plus utilisés peuvent être de deux types : en silicium polycristallin (poly-si) [MET83], ou en aluminium [NIC87].

Les fusibles en si-poly peuvent être programmés (coupés) de deux manières différentes : électriquement ou par laser.

La coupure électrique consiste à faire passer un fort courant électrique (de quelques dizaines de mA) qui fait fondre le fusible et qui provoque une augmentation localisée de la température. Cette méthode nécessite des radiateurs pour évacuer la chaleur (voir la figure suivante)



La coupure laser est une coupure directe qui consiste à diriger un faisceau laser de puissance précalculée sur une connexion en silicium polycristallin. Ceci provoque alors un trou circulaire de diamètre $d=1.5$ à $2\mu\text{m}$ dans le métal. Pour une ligne plus large que le faisceau laser, on fait plusieurs tirs translattés de $d/2$.



Les fusibles en aluminium sont souvent programmés par laser de la même manière que ci-dessus, dans ce cas on parle généralement de coupure de ligne.

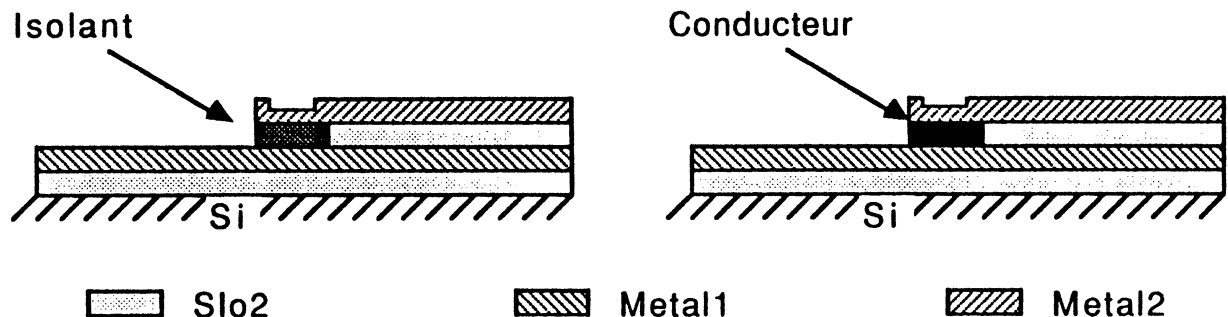
Les fusibles ont les avantages suivants :

- ils sont fiables,
- la chance de réussir leur programmation est $> 99\%$ [NIC87],
- ils sont très peu résistifs avant leur programmation et très résistifs après,
- ils sont peu encombrants,

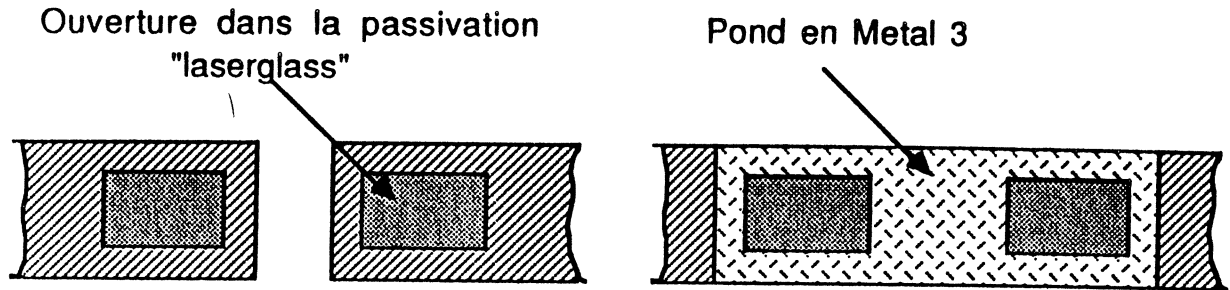
Par contre, ils présentent les inconvénients suivants :

- leur programmation est irréversible
- la programmation des fusibles entraîne une pulvérisation de matière qui risque de provoquer des courts-circuits entre les connexions.

Un "anti-fusible" est un dispositif qui permet de réaliser une connexion électrique entre deux lignes. Il a été développé au "Massachusetts Institute of Technology" une technique qui consiste à réaliser par laser un via dans lequel on diffuse de l'aluminium entre deux niveaux d'aluminium.



Un autre concept très utilisé, développé au LETI, consiste à connecter deux lignes d'aluminium par un pont métallique. Pour cela on réalise par laser deux trous dans la passivation. Après plusieurs étapes technologiques classiques (dépôt résine, insolation laser, développement, déposition de métal, élimination résine) on obtient l'anti-fusible. Ce procédé est appelé "lift off". On peut également faire une lithographie classique (dépôt métal, dépôt résine, insolation laser, développement, gravure métal).



Pour tous les fusibles, ainsi que pour les anti-fusibles, un effort de dimensionnement s'impose.

Les anti-fusibles ont les avantages suivants :

- ils sont fiables,
- la chance de réussir leur programmation est supérieure à 97% [NIC87],
- ils sont peu résistifs après leur programmations ($\approx 2\Omega$) [NIC87],
- ils sont peu encombrants (quelque dizaines de μm^2).

Par contre ils présentent les inconvénients suivants :

- leur programmation est irréversible,
- ils nécessitent des étapes technologiques supplémentaires.

* **Dispositifs programmables par logique**

Ce sont des dispositifs constitués par des interrupteurs à base de portes de transfert commandées par des décodeurs. Les valeurs nécessaires pour programmer les interrupteurs sont stockées dans des registres.

Ces dispositifs présentent les avantages suivants :

- ils sont programmables par les plots extérieurs (le laser et l'e-beam ne sont pas nécessaires),
- ils ne nécessitent pas d'étapes technologiques supplémentaires.

Par contre ils présentent les inconvénients suivants :

- ils occupent une surface relativement importante,
- pour passer de fort courant, ils nécessitent des portes de transfert qui dégradent les signaux qui les traversent, et par conséquent nécessitent des amplificateurs après quelques interrupteurs.

III.1.4. La mémoire SRAM 64K

La mémoire 64k présente des contraintes d'une part électriques (courant, tension des entrées/sorties) et d'autre part topologiques (surface, dimensions). Dans ce paragraphe on développera les points qui interviennent dans la conception et le test de la mémoire-WSI.

* Description générale de la mémoire 64k [FER85]

La mémoire 64K est un circuit de grande complexité (416700 transistors), elle est constituée de trois blocs fonctionnels [FER85] (voir figure 2.3):

- le plan mémoire,
- les décodeurs,
- la logique de sélection et d'écriture/lecture.

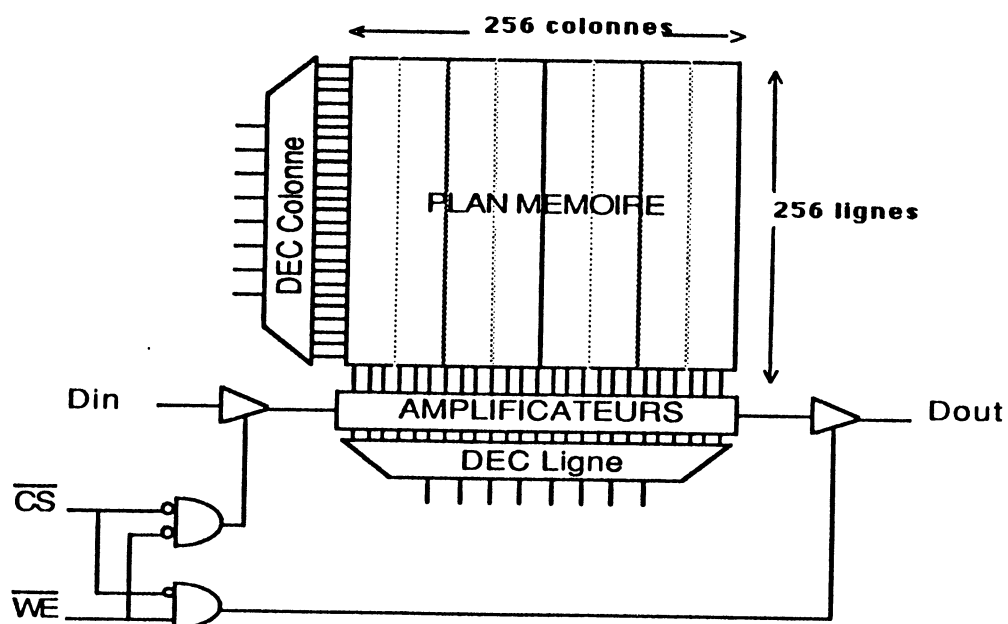


figure 2.3 : Organisation en blocs fonctionnels de la SRAM 64k

Le plan mémoire de la 64K SRAM occupe près de 80% de la surface de la mémoire. Il est réalisé avec des cellules élémentaires à 6 transistors en HCMOS3, et des lignes de bit BL et leurs compléments BL*. Il est organisé en deux demis plans distincts de 256 lignes et 132 colonnes séparés par le décodeur ligne. Il est possible de sélectionner un quart de plan (66 colonnes) selon la combinaison d'adresses (A0,A7) ; par l'adresse A0 (sélection d'un demi plan), puis par l'adresse A7 (sélection d'un quart de plan). Les adresses (A0,A7) : (0,0) (0,1) (1,0) (1,1) sélectionnent successivement de gauche à droite les quatre quarts de plan.

Le décodeur colonne a 8 adresses, une adresse A0 pour la sélection d'un demi plan mémoire et les 7 autres adresses A5-->A11 permettant de sélectionner une colonne parmi 128. Un autre décodeur commun aux deux demis plans sélectionne une ligne parmi 256 lignes, il comporte 8 adresses A1-->A4 et A12-->A15.

La mémoire est contrôlée par les signaux suivants :

CS* : sélection de la mémoire quand CS*=0

WE* : la mémoire est en mode lecture quand WE*=1
et en mode écriture quand WE*=0

F* : Commande de redondance utilisée uniquement pendant le test sur tranche, F*= 0 ordre de claquage d'un fusible, sinon
F*=Vcc

Les circuits d'entrée/sortie amplifient en puissance les données issues soit de l'extérieur (Din) pour charger les capacités des circuits d'écriture, soit de l'amplificateur de lecture (Dout) pour charger des capacités de l'ordre de 50pF. Les amplificateurs de sortie sont suffisamment puissants pour pouvoir utiliser cette mémoire en WSI.

*** Reconfiguration de la mémoire 64K**

La mémoire 64k est reconfigurable. Etant donné que pour la réalisation de la mémoire on utilise une technologie à très haute densité d'intégration, le rendement de fabrication est donc faible. Pour augmenter ce rendement et par conséquent diminuer le coût, il a été introduit, par quart de plan, 2 colonnes redondantes pour la correction d'un seul défaut. Le principe de réparation est le remplacement d'une colonne défailante par une autre saine [SMI81], [FLA86]. Lorsqu'une cellule ou une colonne défailante est repérée par son adresse, on peut donner l'ordre de griller un fusible (en silicium polycristallin) par la commande F^* ($F^*=0$), cet ordre sera transmis dans l'un des quatre quarts de plan en fonction de la combinaison (A0,A7).

Après le claquage du fusible, les données écrites/lues aux adresses de la colonne défailante sont déviées vers une colonne redondante.

L'avantage de cette méthode est de récupérer des mémoires ayant 1,2,3,4 défauts (sachant que l'on ne peut réparer qu'un seul défaut par quart de plan), ceci sans dégradation de leur performance. Par contre, la redondance implique l'ajout de la logique et par conséquent l'augmentation de la surface du circuit.

*** Paramètres électriques**

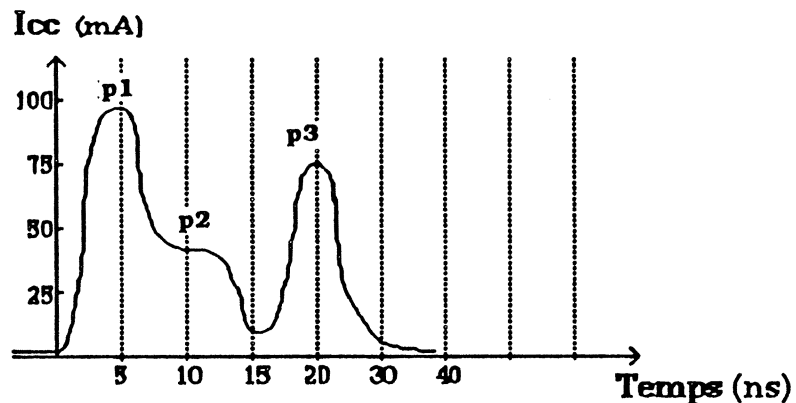
Il s'agit de paramètres admis ou fournis par les plots d'entrée/sortie de la mémoire.

- V_{ccmax} : la tension haute maximale d'alimentation = 7 volts
- V_{ccmin} : la tension haute minimale d'alimentation = 4.5 volts
- V_{ss} : la tension basse d'alimentation = 0 volt

- V_{olmax} : la tension maximale de la sortie Dout pour un état logique 0
(et un courant de test $I_{ol} = 8 \text{ mA}$) = 0.5 volt

- V_{ohmin} : la tension minimale de la sortie Dout pour un état logique 1
(et un courant de test $I_{oh} = -4$ mA) = 2.4 volts
- V_{ihmax} : la tension maximale de l'entrée Din pour un état logique 1
= 6 volts
- V_{ihmin} : la tension minimale de l'entrée Din pour un état logique 1
= 2.2 volts
- V_{ilmax} : la tension maximale de l'entrée Din pour un état logique 0
= 0.8 volts
- V_{ilmin} : la tension minimale de l'entrée Din pour un état logique 0
= -0.5 volts
- C_{in}, C_{out} : capacité d'entrée = 5 à 7 pF
- P_{tmax} : puissance moyenne dissipée = 1 Watt

La consommation de courant en lecture est illustrée par la courbe suivante:



- p1** : pic de précharge des lignes de bits
- p2** : pic de sortie des données sur les lignes de bits
- p3** : pic de sortie des données vers l'extérieur

On remarque que :

- I_{ccmax} : courant dynamique maximal ≈ 100 mA
- I_{ccmoy} : courant moyen ≈ 35 mA
- I_{ccstb} : courant de repos ≈ 20 nA

*** paramètres temporels**

- T_{ac} : temps d'accès ≈ 55 ns (0 à 70 °C)
 ≈ 70 ns (-55 à 125 °C)

Les caractéristiques dynamiques en lecture et en écriture ainsi que leurs chronogrammes sont expliqués en [Annexe 2.2] et [Annexe 2.3].

*** Paramètres topologiques**

Les dimensions géométriques de la mémoire 64k sont :

$$dx = 3.876 \text{ mm}$$

$$dy = 7.055 \text{ mm}$$

$$\text{sa surface est donc} = 27.345 \text{ mm}^2$$

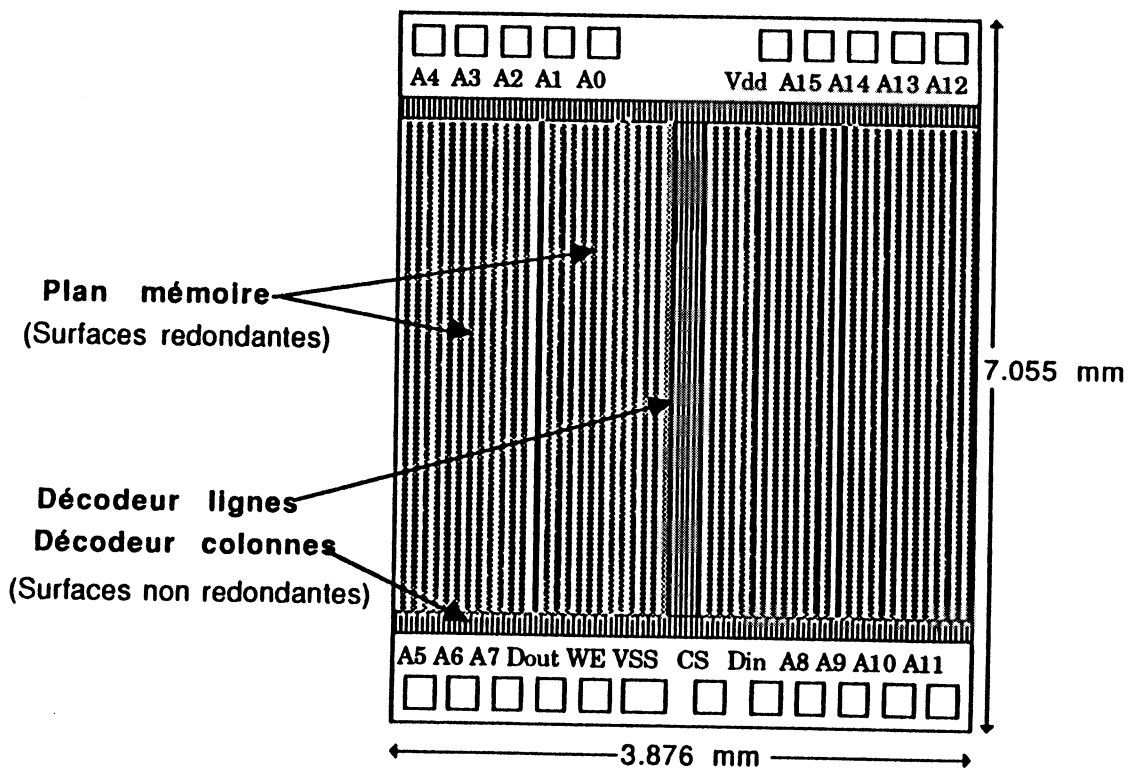


figure 2.4 : Aspect physique de la mémoire 64k

Les plots sont disposés sur les 2 petits côtés de la mémoire (voir figure 2.4).

Les données Din et Dout peuvent partager un plot commun, car elle ne sont pas actives simultanément.

*** Etude théorique du rendement de la mémoire 64k**

Modèles théoriques [HED82], [STA83], [KET85]

On sait que l'apparition d'un défaut sur une tranche est un phénomène aléatoire. Ce phénomène est décrit par différents modèles de probabilité, reliant la probabilité d'apparition de défaut à des facteurs tels que les étapes technologiques de fabrication, la densité de défaut, et la surface du circuit. Ces paramètres sont estimés ou tirés de mesures statistiques effectuées dans les chaînes de production. Seule la surface peut être connue.

Les premiers travaux menés sur le rendement de fabrication ont montré que la probabilité d'avoir un circuit de surface S sans défaut après une étape technologique critique i est modélisé par la "loi de Poisson".

D'où :

$$P_i = e^{-D_i \cdot S}$$

S = la surface du circuit (identique pour toutes les étapes)

D_i = densité de défauts.

D_i n'est pas une constante, elle varie sur une même tranche, et d'une tranche à une autre. Sa valeur est affectée d'une densité de probabilité p_i telle que:

$$\int_0^{\infty} p_i(D) d(D) = 1$$

Alors

$$P_i = \int_0^{\infty} p_i(D) e^{-D \cdot S} d(D)$$

Après n étapes technologiques critiques la probabilité finale d'avoir un circuit sans défaut est donc :

$$P = P_1 * P_2 * \dots P_n$$

Cette valeur (P) n'est que le rendement de fabrication d'un circuit de surface S.

En généralisant l'expression P_i à n étapes critiques par introduction d'une densité totale moyenne de défauts D_t on obtient :

$$P = P(D_t \cdot S) = P(n \cdot D \cdot S)$$

Avec $D_t = nD$

D = densité moyenne de défauts / étape critique

Quelques dizaines de fonction de distribution p_i ont été proposées dont l'intégration donne les probabilités suivantes :

$$\text{"Murphy"} \text{ ----> } P_i = - \frac{1 - e^{-2nSD}}{2nSD} \text{ ou } P_i = \left(- \frac{1 - e^{-nSD}}{nSD} \right)^2$$

$$\text{"Seeds"} \text{ ----> } P_i = e^{-\sqrt{nSD}}$$

$$\text{"Murphy - Seeds"} \text{ ----> } P_i = - \frac{\left(\frac{1 - e^{-nSD}}{nSD} \right)^2 + e^{-\sqrt{nSD}}}{2}$$

$$\text{"Bose - Einstein"} \text{ ----> } P_i = - \frac{1}{(1 + S \cdot D)^n}$$

Ces modèles peuvent être utilisés en fonction de la chaîne de fabrication du type de circuit, ou de la technologie. Le meilleur modèle est celui qui s'approche le plus de la mesure statistique effectuée en production.

Pour la technologie HCMOS3 l'expérience montre que les modèles les plus proches de la réalité sont :

- la moyenne des deux expressions "Murphy" et "Seeds" pour les valeurs faibles de Dt.S.
- le modèle de "Bose-Einstein" pour les valeurs élevées de Dt.S.

Il est donc recommandé de prendre l'expression suivante pour le rendement:

$$\text{REND}(n,D,S) = \inf (\text{Murphy-Seeds}, \text{Bose-Einstein})$$

L'étude théorique du rendement de la mémoire de base doit permettre l'estimation de la faisabilité de la mémoire- WSI.

*** Etude du rendement théorique de la mémoire 64k**

Lors de la conception de la mémoire 64k, il a été montré l'amélioration du rendement due à la redondance (colonne). Dans la suite, on va résumer la modélisation de cette influence.

La redondance va améliorer considérablement le rendement de fabrication, car les mémoires qui ont 1 défaut par quart de plan sont réparables et peuvent devenir fonctionnelles. Pour le calcul du rendement de la mémoire, on admet les hypothèses générales suivantes :

La surface totale de la mémoire ST est composée d'une surface redondante SR sans les colonnes redondantes et d'une surface non redondante SNR, donc :

$$\text{ST} = \text{SR} + \text{SNR}$$

La surface redondante est composée de p plans identiques de surface SP , et chaque plan est formé de k éléments réparables et de r éléments de réserves pour chacune des surfaces SE (voir figure 2.4). On a donc :

$$SR = p * SP$$

$$SP = (k+r) * SE$$

On appelle :

PRSE : La probabilité de remplacement d'un élément par un autre

$$(0.8 < PRSE < 1)$$

$P(S) = \text{REND}(n, D, S)$ la probabilité d'avoir 0 défaut sur une surface S .

On peut dire donc que :

La probabilité d'avoir 0 (zéro) défaut sur un circuit = la probabilité d'avoir {0 défaut sur SNR et [(0 défaut sur SR) ou (SR défectueuse et réparable)] }

Autrement dit :

$$P(ST) = P(SNR) * [P(SR) + \underline{PSRDR}] = \text{RDMEM (Rendement d'une mémoire)}$$

$P(SNR)$: probabilité d'avoir 0 défaut sur la surface non réparable (SNR),

$P(SR)$: probabilité d'avoir 0 défaut sur la surface réparable (SR),

$PSRDR$: probabilité d'avoir la surface SR défectueuse et réparable.

$$\text{Or } SR = p * SP$$

Donc $PSRDR$ d'après la loi binomiale est égale à :

$$PSRDR = \sum_{i=1}^p C_p^i \text{ prob}[(i \text{ plans défectueux et réparables}) * (p-i \text{ autres sains})]$$

La probabilité $PSRDR$ peut être également écrite de la façon suivante :

$$PSRDR = \sum_{i=1}^p C_i^j \text{PSPDR}^i * P[(p-i) * SP]$$

PSPDR = probabilité d'avoir la surface SP défectueuse et réparable

$$\text{Or } SP = (k+r) * SE$$

Donc PSPDR d'après la loi binomiale est égale à :

$$\text{PSPDR} = \sum_{j=1}^r C_{k+r}^j \text{prob}[(j \text{ éléments défectueux et réparables}) * (k+r-j \text{ autres sains})]$$

$$\text{PSPDR} = \sum_{j=1}^r C_{k+r}^j \text{PSEDR}^j * P[(k+r-j) * SE]$$

Or

PSEDR = (la probabilité pour que SE soit défectueuse) * (la probabilité pour que SE soit réparable)

$$\text{PSEDR} = [1 - P(SE)] * PRSE$$

D'autre part :

La densité de défauts de la surface non réparable est inférieure à celle de la surface redondante, ceci grâce aux règles de dessin large et aux faibles densités d'interconnexion.

Donc

$$P(SNR) = \text{REND}(DSNR, SNR) \text{ avec } (DSNR < D)$$

Pour exécuter ces formules, il a été réalisé un programme [Annexe 2.4] qui demande en entrée les données suivantes : D, DSNR, n, les dimensions d'une puce sans éléments de réserve XCEL et YCEL, POURC (pourcentage de surface réparable), p, k, PRSE, Et qui donne en sortie le rendement en fonction des éléments redondants, ainsi que le nombre des bons circuits.

On note que les autres paramètres sont fixes, tel que :

- le rayon utile de la tranche $R_u = 45\text{mm}$,
- l'espace entre puces : $dx = 0.64\text{ mm}$ et $dy = 1.79\text{ mm}$.
- la probabilité de remplacement d'un élément par un autre (PRSE) = 0.9
- la densité de défauts de la surface non réparable $DSNR = 0.5\text{ défaut/cm}^2$

Ce programme a été exécuté dans le cas de la mémoire 64k pour les valeurs suivantes :

$D = 1\text{ défaut/cm}^2$, $n = 6$ étapes critiques, $XCEL = 3.685\text{mm}$, $YCEL = 7.055\text{mm}$, $POURC = 80\%$, $p = 4$, $k = 64$, $PRES = 0.9$, et pour différentes valeurs de r (éléments de réserves)

les résultats obtenus nous ont permis de tracer la variation du rendement d'une mémoire 64k (RDMEM) en fonction des éléments de réserves r (voir figure 2.5).

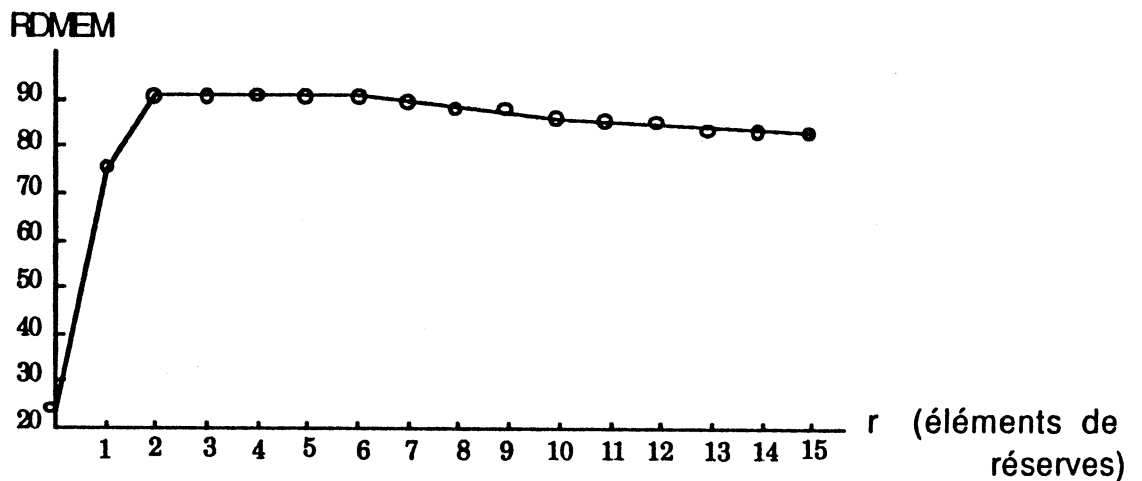


figure 2.5 : variation du rendement d'une mémoire (RDMEM) en fonction des éléments de réserves (r).

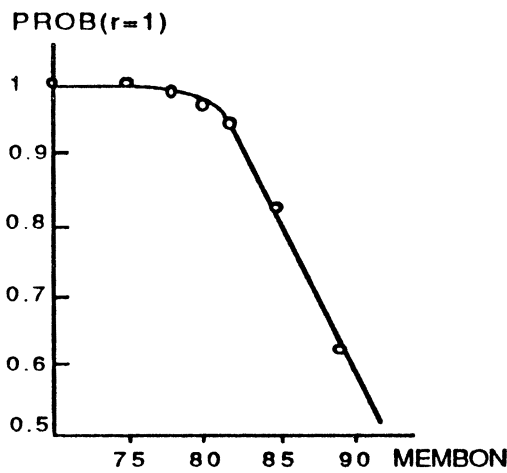
On remarque que le rendement croît rapidement, puis il décroît à cause de l'augmentation de la surface due à l'introduction excessif des éléments de réserves.

Ce qui importe pour nous c'est le rendement effectif obtenu (le nombre potentiel des mémoires bonnes en fin de réparation) par l'ajout des éléments de réserves.

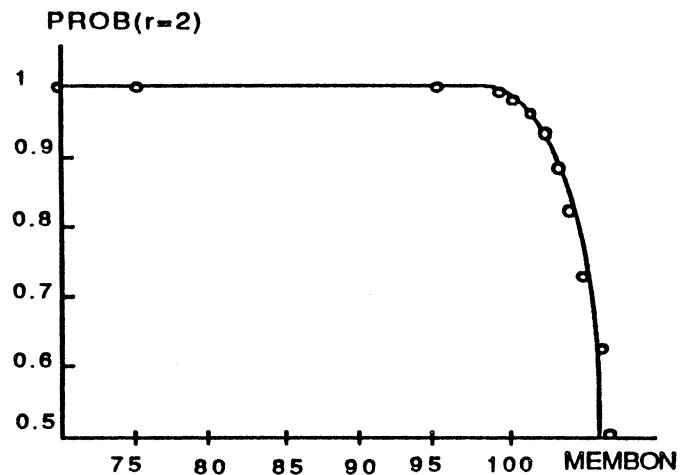
Pour cela on calcul la probabilité de trouver au moins MEMBON (=70, 71,... des bonnes mémoires parmi le nombre total des mémoires 64k (NBCEL(r)) intégrées sur la tranche. Cette probabilité est donnée par la formule :

$$\text{PROB}(r, \text{MEMBON}) = \sum_{i=1}^{\text{NBCEL}(r)-\text{MEMBON}} \text{C}_{\text{NBCEL}(r)}^{\text{MEMBON}+i} \text{RDMEM}(r)^{\text{MEMBON}+i} (1-\text{RDMEM}(r))^{\text{NBCEL}(r)-\text{MEMBON}-i}$$

Les courbes suivantes montrent la variation de probabilité en fonction du nombre de bonnes mémoires désirées.



(2.a)



(2.b)

figure 2.6 : probabilité de trouver au moins MEMBON bonnes mémoires parmi le nombre total de mémoires 64k (NBCEL(r)) intégrées sur la tranche.

fig (2.a) : pour un seul élément de réserve (r=1), et NBCEL=120

fig (2.b) : pour deux éléments de réserve (r=2), et NBCEL=120

On remarque que par l'ajout de deux éléments de réserve (r=1) (voir figure 2.6a), on trouvera (avec une probabilité de 0.98) environ 80 bonnes mémoires 64k ce qui est suffisant pour la réalisation d'une mémoire 4.5Mbits. Dans ce cas le rendement de la tranche est de l'ordre de 67%.

On trouvera en annexe [Annexe 2.5] les valeurs qui ont permis de tracer les courbes des figures 2.5, 2.6a et 2.6b.

III.1.5. Les lignes d'interconnexion

* Modélisation des lignes d'interconnexion

Une ligne intégrée sur un substrat en silicium Si (forcé à un potentiel 0 volt) peut être représentée électriquement comme une succession d'un certain nombre d'éléments RC en série (voir figure 2.7)

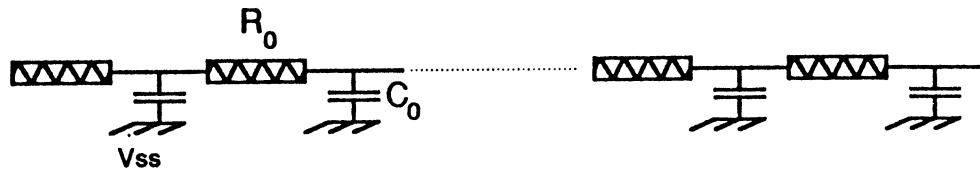


figure 2.7 : Représentation électrique d'une ligne d'interconnexion

La résistance totale R_L et la capacité totale C_L d'une ligne sont égales respectivement à la somme des résistances R_0 et à la somme des capacités C_0 linéiques par unité de longueur.

$$R_L = \sum R_0$$

$$C_L = \sum C_0$$

En réalité, une connexion intégrée sur le substrat est une bande métallique de conductivité ρ , de longueur L , de largeur W et de hauteur T , intégrée sur un isolant (SiO_2) de hauteur H et de permittivité ϵ (voir figure 2.8).

On appelle S la distance qui sépare deux connexions.

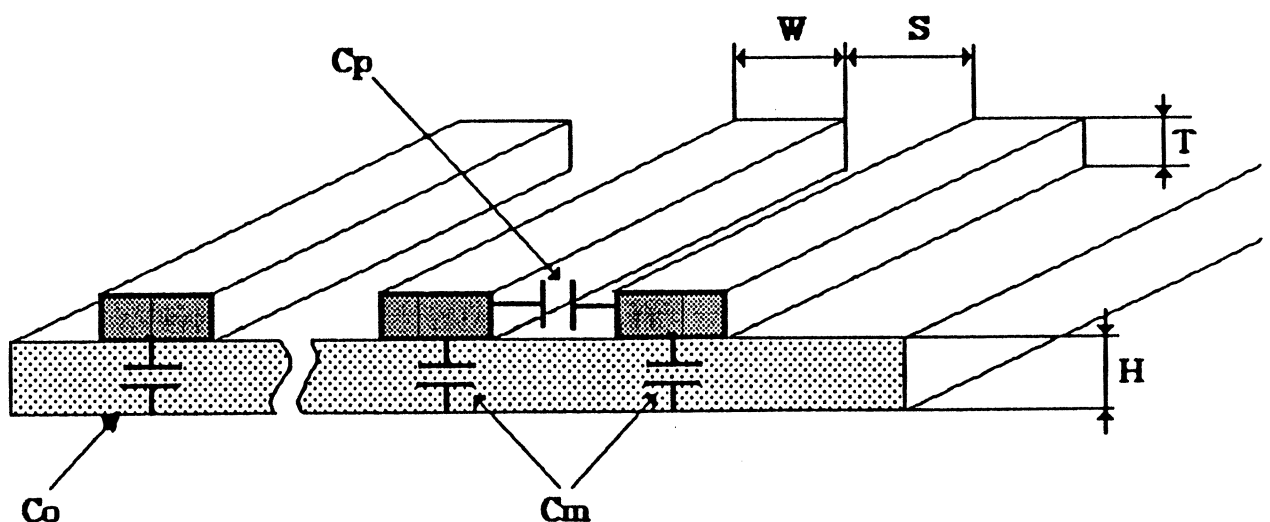


figure 2.8 : Représentation physique d'une ligne d'interconnexion

Les valeurs de W, T, H sont de l'ordre de quelques μm , sauf L qui peut atteindre quelques milliers de μm .

Il existe plusieurs modèles pour modéliser le phénomène de propagation des signaux sur une ligne

Modèle RC [SAK1-83]

Temps de propagation

Ce modèle propose une représentation d'une ligne en fonction de la capacité C qu'elle doit charger et de la résistance R de l'attaque de cette ligne. Cette ligne est illustrée dans la figure (2.9) suivante :

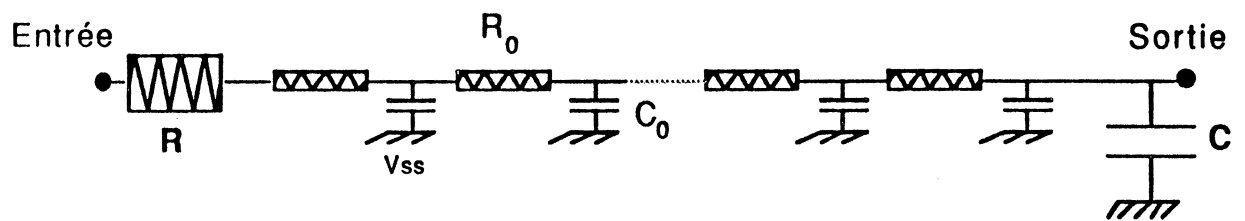


figure 2.9 : représentation électrique d'une ligne

On définit par : t_{mk} le temps de montée d'une tension de 0 volt à k volts.

Dans ce modèle le temps de montée de la sortie $t_{m4.5}$ pour une ligne d'interconnexion est donné par la formule suivante :

$$t_{m4.5} / RL * CL = 1.02 + 2.21 (Rn * Cn + Rn + Cn)$$

Avec

$$Rn = R / RL \text{ et } Cn = C / CL ; \text{ (coefficients de normalisation)}$$

Calcul des capacités d'une connexion

On rappelle que lorsqu'une seule connexion isolée est à un potentiel différent de celui du substrat, des charges de signes opposés vont se créer d'une part sur la surface de substrat et d'autre part sur la surface

W*L de connexion. Cette différence de charges va créer une capacité propre (Co) facilement calculable par des modèles assez simples. Par contre, dans le cas de plusieurs connexions adjacentes, des phénomènes de couplage capacitif commencent à apparaître, le calcul de la capacité devient alors sérieusement plus complexe.

On a adopté le modèle [SAK2-83] qui prend en compte les effets suivants:

- l'influence des effets de bord sur la capacité lineïque (C₀).
- l'influence du couplage capacitif (C_p).

Les formules qui permettent de calculer les capacités par unité de longueur sont :

$$C_0 = K * [1.15 * (W/H) + 2.8 * (T/H)^{0.222}]$$

$$K = 3.45 \text{ E } -5 ; \text{ fonction de permittivité}$$

$$DC_0 = K * 0.07 * (T/H)^{0.222} * (S/H)^{-1.34}$$

$$C_m = C_0 - DC_0 ; \text{ capacité lineïque modifiée par le phénomène de couplage}$$

$$C_p = K * [0.03 * (W/H) + 0.83 * (T/H)] * (S/H)^{-1.34}$$

Les résultats de simulation sont dans le tableau suivant :

W	S	C ₀ pF/ μ	DC ₀ pF/ μ	CP pF/ μ
5	5	2.11 E -4	4.73 E -7	4.60 E -6
10	5	3.35 E -4	4.73 E -7	5.29 E -6
15	5	4.59 E -4	4.73 E -7	7.05 E -6
20	5	5.83 E -4	4.73 E -7	8.10 E -6
25	5	7.07 E -4	4.73 E -7	9.15 E -6

tableau 2.1 : Calcul des différentes capacités

Dans le calcul des capacités, on prend comme épaisseur d'oxyde $H = 1.6\mu$, et comme hauteur de connexion $T = 1\mu$.

D'après le tableau ci-dessus, les capacités de couplage (C_p), ainsi que (DCo) sont négligeables devant (Co). Dans les simulations ultérieures (SPICE) pour les lignes de plus de 5μ de large et espacées de 5μ , il sera pris en compte uniquement la capacité propre des lignes (Co).

Les autres effets, tels que l'effet entre deux lignes croisées de deux niveaux différents, sont négligeables devant les capacités propres des lignes.

Donc la capacité d'une ligne $CL =$ la capacité linéique \times la longueur de ligne.

$$CL = Co * L$$

Calcul de la résistance linéique d'une connexion

La résistance d'une connexion dépend uniquement des paramètres propres à cette connexion, tels que :

Les dimensions de la connexion W, L, T .

La résistivité ρ qui est une fonction croissante de la température.

La formule la plus classique qui va permettre de calculer RL est :

$$RL = Ro * L/W \text{ avec}$$

$$Ro = (\rho / T) \quad ; \quad (\approx 30 \text{ m}\Omega/\mu \text{ pour la technologie HCMOS3, et à } 25 \text{ }^\circ\text{C})$$

Modèle "Mosaic system" [STO87]

Temps de propagation

Mosaic system présente un modèle de simulation caractérisé par les paramètres suivants:

- l'impédance $Z_o = 189 g * H/W$ avec $1/g = 1 + 2H/\pi W [1 + \ln(1 + \pi W/2H)]$

- la capacité élémentaire par mm, $C' = 0.035 W/gH$

- la résistance élémentaire par mm, $R' = 30/W(\mu\text{m}) T(\mu\text{m})$.
- la longueur critique, $dc = 1.4 Z_0/R'$.

La longueur critique est la longueur maximale d'une ligne pour obtenir un temps de propagation minimal.

Grâce à ces paramètres on peut calculer le temps de propagation pour une distance critique dc avec la formule suivante:

$$t^0 = Z^0 * C' * dc$$

Ce modèle n'est valable que pour des épaisseurs d'oxyde supérieures à $2\mu\text{m}$.

*** Simulation théorique d'une ligne d'interconnexion**

Dans le but de comparer les deux modèles précédents, une simulation a été faite pour une portion de ligne sans capacité de charge ni résistance d'attaque, et qui a les caractéristiques suivantes :

$$W = 10\mu\text{m}$$

$$H = 2 \mu\text{m}$$

$$T = 3 \mu\text{m}$$

La longueur L de ligne va être calculée par la longueur critique dc .

Avec le modèle "Mosaic System" on obtient :

$$1/g = 1.4$$

$$Z_0 = 26.9 \Omega$$

$$C' = 0.245 \text{ pF/mm}$$

$$R' = 1 \Omega/\text{mm}$$

$$dc = 37.66 \text{ mm} = L$$

pour une ligne de longueur dc

$$t_0 = 0.248 \text{ ns.}$$

Pour la même ligne les résultats de simulation par le modèle "RC" sont les suivants :

$$C_0 = 3.0 \text{ E-4 pF}/\mu\text{m} \rightarrow C' = 0.3 \text{ pF/mm}$$

$$R_0 = 1.0 \text{ E-3 } \Omega/\mu\text{m} \rightarrow R' = 1 \Omega/\text{mm}$$

$$t_{0m4.5} = 0.434 \text{ ns}$$

La simulation "SPICE" donne un résultat proche de celui du modèle RC :

$$t_{0m4.5} = 0.5 \text{ ns}$$

*Reconfiguration des lignes

Différentes méthodes de reconfiguration [ESP87]

Bien que la probabilité de panne sur les lignes larges (10μ) soit très faible, un défaut sur ces lignes (coupure, court-circuit, ...) menace le fonctionnement de toute la mémoire-WSI, et peut la rendre inexploitable.

La réparation d'un défaut sur une ligne est souvent délicate, d'où la nécessité d'utiliser une structure redondante. On discute dans la suite les avantages et inconvénients des différentes méthodes de redondance et de reconfiguration classiques afin de définir si l'une d'elles est utilisable pour la mémoire-WSI.

-1^{ère} METHODE : Décalage d'une Nappe de Lignes.

Cette méthode consiste à localiser et isoler la panne sur une portion de ligne, puis la contourner en décalant la nappe vers le haut (à partir de ligne défectueuse) tout le long de la portion (voir figure 2.10). Ceci nécessite une ligne redondante et l'intervention physique (laser, E.beam) sur les lignes voisines (par exemple dans le cas de court-circuit).

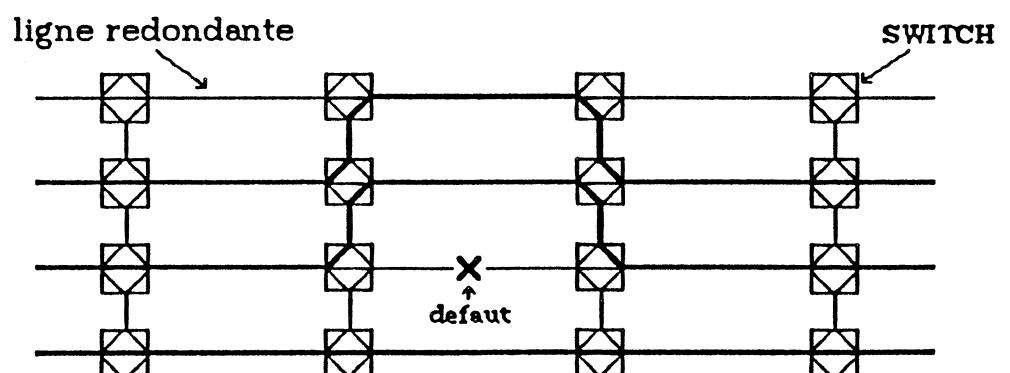


figure 2.10 : Décalage d'une nappe de lignes

- 2^{ème} METHODE : Décalage d'une ligne d'interconnexion.

Il s'agit de localiser et d'isoler la panne sur une portion de ligne, puis emprunter une portion semblable et saine d'une ligne redondante (voir figure 2.11).

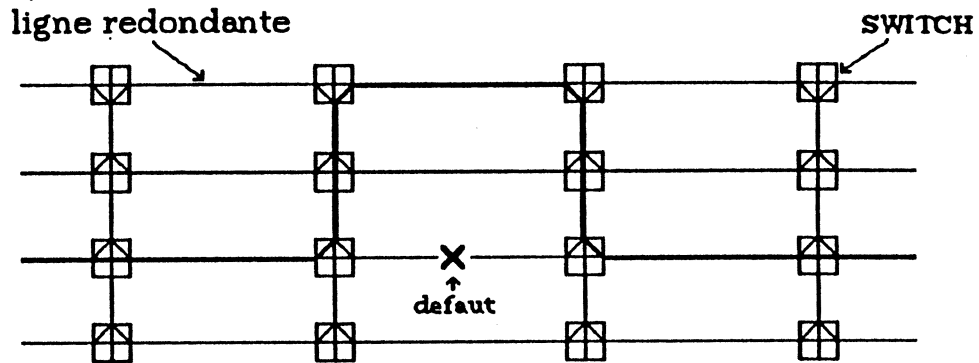
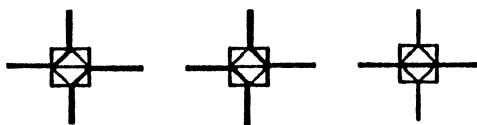


figure 2.11 : Décalage d'une ligne d'interconnexion

Dans les deux méthodes précédentes, le routage des lignes est réalisé grâce à des interrupteurs (switches) programmables, comportant des portes de transfert commandées par des grilles flottantes ou des fusibles. La programmation de ceux-ci doit permettre d'obtenir les trois configurations suivantes:

pour la 1^{ère} méthode

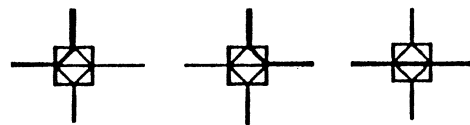


1.a

1.b

1.c

pour la 2^{ème} méthode



2.a

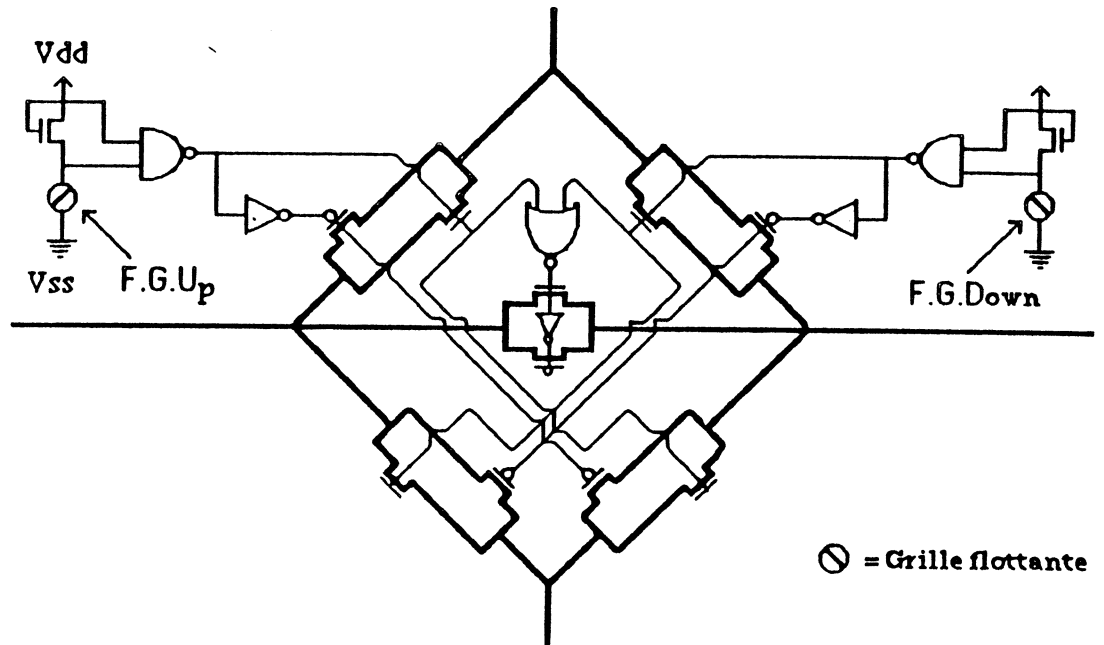
2.b

2.c

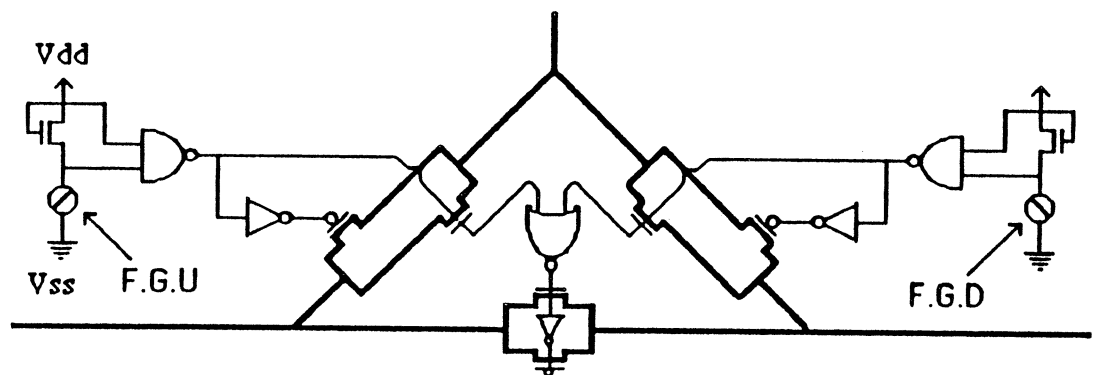
Après la fabrication, on aura les configurations 1.c et 2.c pour tous les interrupteurs. Un signal qui traverse un interrupteur ne passe que par une seule porte de transfert, ceci limite sa dégradation.

Ces spécifications nous conduisent à choisir pour les interrupteurs le schéma électrique suivant:

pour la 1^{ère} méthode



pour la 2^{ème} méthode



Pour diminuer le nombre de grilles flottantes à programmer, on peut pour la 1^{ère} méthode ajouter une logique de commande qui permet de décaler toutes les lignes vers le haut ou vers le bas à partir de lignes défailantes par programmation d'une seule grille flottante.

Avantages. Inconvénients

Dans les deux méthodes précédentes, un signal qui traverse l'interrupteur passe obligatoirement par une porte de transfert qui présente une charge résistive non négligeable. Pour limiter la dégradation éventuelle du signal il faut dimensionner correctement les portes de transfert en fonction d'une part, de leur nombre en série sur la ligne, et d'autre part du courant qui les traverse.

Une telle structure présente les avantages architecturaux suivants :

- elle peut être utilisée pour le réseau des adresses dans le but de reconfiguration et de réparation.
- pour le réseau des données dans le but de routage.

Par contre, elle présente les inconvénients électriques et topologiques suivants :

- la réalisation de chaque interrupteur demande environ 30 transistors dont 10 (pour les portes de transfert) doivent être surdimensionnés et donc occuperont une surface importante.
- l'augmentation du risque de panne des interrupteurs comportant des gros transistors.

*** Conclusion**

On retiendra quatre points concernant les lignes d'interconnexion :

- Il existe deux moyens de diminuer le temps de propagation sur une ligne d'interconnexion de longueur L et de largeur W fixes :

- diminuer la capacité linéique
- diminuer la résistance linéique

On remarque dans les formules des trois modèles précédents que pour diminuer la capacité d'une connexion, il faut augmenter l'épaisseur d'oxyde (SiO_2 dans le cas des circuits MOS) et donc le paramètre H . D'autre part pour diminuer la résistance linéique, il faut augmenter l'épaisseur de connexion T . La technologie HCMOS3 choisie pour la réalisation de la mémoire-WSI, présente quelques contraintes ; elle ne permet pas d'intégrer des lignes d'interconnexion sur une épaisseur d'oxyde supérieur à $1.6\mu\text{m}$, par conséquent la capacité linéique est limitée par une valeur inférieure. Par contre elle permet l'épaississement des connexions par le cuivre "copper tracking" qui se traduit par l'augmentation du paramètre T

dans les formules des modèles, et par conséquent la diminution de la résistance linéique.

- Dans la suite on utilisera le modèle "RC" pour le calcul des paramètres électriques R et C des réseaux d'interconnexion, ceci pour la raison suivante :

Les résultats pratiques du Circuit Test Techno (CTT) concernant les valeurs de capacités et résistances, sont très proches des valeurs calculées par le modèle "RC" (par exemple : pour une ligne $W=10\mu$ et $T=1\mu$ et $H=1.6\mu$; la valeur mesurée de $C_0 \approx 3.2610^{-4} \text{pF}/\mu\text{m}$ et la valeur calculée $\approx 3.3510^{-4} \text{pF}/\mu\text{m}$).

L'utilisation du modèle de "Mosaic system" impose d'une part l'intégration des lignes d'interconnexion avec une épaisseur d'oxyde de plus de $2 \mu\text{m}$, cette épaisseur est interdite dans la technologie HCMOS3.

- Dans le cas de la mémoire-WSI, on réalise la tranche par photo-répétition, les interrupteurs à base de portes de transfert nécessaires pour un réseau vont donc être photo-répétés une centaine de fois sur la tranche, et on aura pour une seule ligne de données ou d'adresses une dizaine d'interrupteurs. Ceci présente une charge résistive de l'ordre de 100Ω dans le meilleur des cas de dimensionnement des transistors CMOS des portes de transfert. Les solutions de reconfiguration à l'aide de dispositifs tels que les portes de transfert devront être évitées au maximum.

- L'utilisation des lignes de grande longueur peut faire craindre l'apparition d'un effet de couplage capacitif entre deux lignes voisines (diaphonie...) or, il a été montré dans le paragraphe "simulation

des réseaux", que pour des lignes espacées de 5μ , la capacité d'influence entre une ou plusieurs lignes voisines devient négligeable par rapport à la capacité propre des lignes. Les phénomènes de couplage capacitif entre les lignes s'avèrent donc négligeables.

III.2. CONTRAINTES DE TEST

III.2.1. Le temps de test

Le temps de test T nécessaire pour tester la mémoire 4.5 Mbits est essentiellement le temps nécessaire pour tester environ 120 mémoires 64k. Ce temps peut être très long, et peut être évalué de la manière suivante :

soient :

t_0 = le temps total de test d'une bonne mémoire 64k,

$k \cdot t_0$ = le temps maximal de test d'une mémoire 64k défailante, car pendant le test d'une mémoire défailante le test n'est pas achevé,

p = un coefficient de parallélisme de test qui représente le nombre de mémoires testées en parallèle,

$RDMEM_{max}$ = le rendement théorique maximal d'une mémoire 64k $\approx 90\%$ (voir chapitre III paragraphe III.1.4).

Donc le temps maximal, T_{max} est égal :

$$T_{max} \approx [120 \cdot 0.9 \cdot t_0 + 120 \cdot 0.1 \cdot k \cdot t_0] / p = (108 + 12 \cdot k) \cdot t_0 / p$$

Pour diminuer le temps T_{max} , il faudrait diminuer le temps t_0 et/ou augmenter le coefficient de parallélisme, k étant un paramètre aléatoire compris entre 0 et 1.

Or t_0 est fonction, d'une part, des paramètres temporels de la mémoire 64k et du choix de l'algorithme de test, tels que :

- la fréquence de test, qui est limitée par le temps d'accès de la mémoire (fréquence de test $\leq 1/\text{temps d'accès}$).
- de la complexité de l'algorithme de test qui est exprimée en fonction du nombre N des cellules mémoires élémentaires du plan mémoire.

Ces deux paramètres seront choisis et détaillés dans le chapitre "Stratégie de test et de configuration de la mémoire WSI".

D'autre part, T_{max} dépend des paramètres qui sont fonction de la stratégie de test et par conséquent fonction de l'architecture de la mémoire-WSI, tels que : le parallélisme de test p, qui sera le sujet des paragraphes suivantes.

III.2.2. Test en parallèle

Il y a deux possibilités de tester la mémoire WSI en parallèle :

-soit par un test intégré, l'algorithme pour test est généré par un ou plusieurs circuits spéciaux intégrés sur la tranche, et envoyé sur les mémoires 64k en parallèle.

-soit de l'extérieur, le programme de test est généré par un testeur extérieur, et envoyé en parallèle dans plusieurs mémoires 64k.

**** Test intégré [KIN86], [BEN87]***

- Approche autotest (BIST : Built In Self Test) ; Il s'agit d'intégrer un circuit de test dans chaque mémoire 64k (voir figure 2.12).

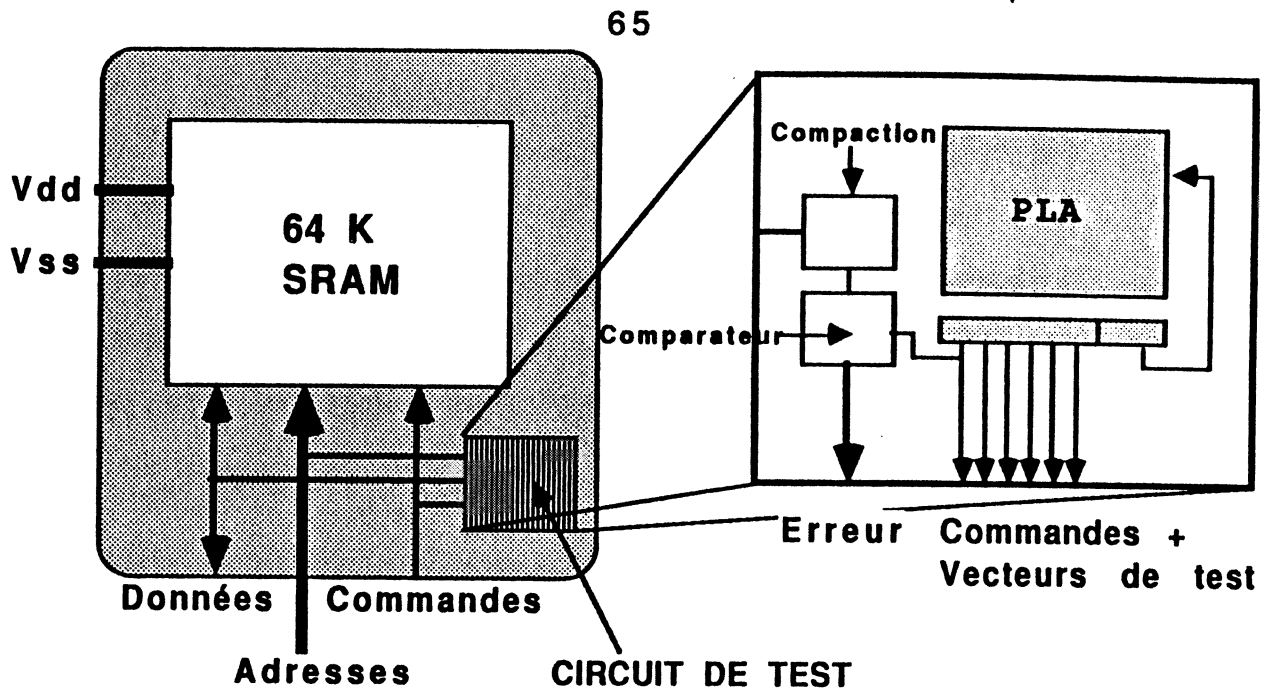


figure 2.12 : Circuit d'autotest intégré dans la mémoire 64k

Ce circuit est constitué par les dispositifs suivants :

- un PLA (programmable logic Array) qui commande la mémoire 64k, il contient les commandes (CS : sélection de la 64k, écriture/lecture), les vecteurs de test, ainsi que les commandes nécessaires pour le séquençage du programme de test.
- un dispositif de signature pour la compaction des résultats de test (les données lues dans la mémoire)
- un comparateur capable de comparer les résultats de signature et les bonnes valeurs prédéterminées (valeurs écrites dans la mémoire).

Le lancement du test peut se faire par une commande spéciale de l'extérieur ou par faisceaux d'électrons (e-beam).

Cette approche présente des avantages et des inconvénients :

Avantages :

- elle permet le test de toutes les mémoires 64k intégrées sur tranche en parallèle.
- elle teste à la fréquence normale d'utilisation.

inconvenients :

- l'algorithme de test implanté sera figé.
- l'intégration d'un automate de test augmentera la surface de la mémoire de 15%, il diminue donc le rendement de fabrication.
- le mécanisme de réparation par redondance sera difficile à réaliser, car ceci nécessite la localisation de défaut qui n'est pas possible par un test intégré.

*** Test de l'extérieur**

Les séquences de test sont générées entièrement par le testeur extérieur. Le testeur génère et écrit les données dans la mémoire, puis compare les données lues dans la mémoire avec les données précédemment écrites. Le parallélisme dans le test de la mémoire 4.5 Mbits apparaît ici dans la manière d'accéder aux mémoires 64k en parallèle.

On propose deux méthodes d'accès aux mémoires 64k :

- via les réseaux d'interconnexion par une carte à pointes à l'échelle de la tranche.
- par accès direct ou par l'intermédiaire d'une carte à pointes à l'échelle de quelques puces mémoires.

*** Via les réseaux d'interconnexion**

Il s'agit de tester les différentes parties de la mémoire-WSI (mémoires 64k, réseaux d'interconnexion,...) par une carte à pointes à l'échelle de la tranche, via les réseaux d'interconnexion. Le test des mémoires 64k est réalisé par connexions successives des groupes de mémoires, et test en parallèle des mémoires de chaque groupe.

L'avantage de cette méthode est qu'elle permet de tester jusqu'à 16 mémoires 64k en parallèle avec un algorithme de test au choix généré par un testeur extérieur.

Par contre elle présente des inconvénients graves :

- nécessité d'utilisation des éléments de configuration (grilles flottantes, fusibles, anti-fusibles) sur les lignes de données afin de pouvoir connecter et déconnecter les mémoires pour les tester en parallèle. Ceci n'est pas possible à réaliser actuellement pour deux raisons :

i) l'utilisation des transistors à grille flottante ajoute des résistances R_{on} de transistor assez élevées sur le chemin de données. D'autre part, pour des problèmes technologiques la programmation d'une grille flottante est limitée à 3 fois, elle est réalisée par bombardement direct et local de la grille par un faisceau d'électrons mais la déprogrammation ou la neutralisation des charges négatives est réalisée par insolation UV non localisée de la tranche, par conséquent, on ne peut pas tester plus de 3 groupes de mémoires via les réseaux.

ii) l'utilisation des fusibles ou des anti-fusibles n'offre pas la réversibilité de programmation.

- le test sera réalisé à basse fréquence, car les vecteurs de test vont se propager sur les lignes d'interconnexion

* Accès direct en parallèle

Il s'agit d'accéder et de tester directement les 2 ou 4 mémoires par une carte à pointes spéciale.

Avantages de cette méthode :

- elle offre la possibilité de tester les mémoires 64k par accès direct indépendamment des autres éléments constituant la mémoire 4.5 Mbits
- elle permet la réparation des 64k.

Inconvénients :

- Il n'est pas possible de tester plus de 4 mémoires en parallèle, car le nombre de pointes est limité sur une carte.

*** Conclusion**

Notre choix entre les différentes méthodes de test dépend d'une part des avantages et des inconvénients que présente chacune d'elles, et d'autre part du coût d'adaptation des matériels disponibles.

Dans notre cas, le but est de tester et de réparer le maximum de mémoires 64k. Pour atteindre ce but, il faut avoir un accès direct à chaque mémoire 64k pour la réparation. Le test intégré ne permet pas la localisation des défauts et par conséquent ne permet pas de diagnostiquer et de réparer. D'autre part il augmente la surface des puces et donc baisse le rendement. On a vu que la solution de test via les réseaux d'interconnexion est inutilisable. La méthode d'approche adoptée sera donc le test par accès direct moyennant une carte à pointes qui sera développée dans le chapitre "stratégie de test". Les modifications architecturales seront fonction de cette stratégie de test.

IV. CONCLUSION

Les points essentiels à prendre en considération dans la suite sont les suivants :

- * La mémoire-WSI doit être réalisée en technologie HCMOS3 par photo-répétition d'un seul réticule qui comporte deux mémoires 64K.
- * Il est très pénalisant d'utiliser des dispositifs de reconfiguration à base de portes de transfert. On utilisera dans la suite
 - des transistors à grilles flottantes qui serviront uniquement pendant la phase de test,
 - des fusibles et des anti-fusibles pour figer la configuration finale.

- * Les simulations électriques des lignes d'interconnexion ainsi que des autres dispositifs formant la mémoire-WSI, seront faites en "SPICE".
- * Enfin la mémoire sera testée par une carte à pointes spéciale à l'échelle de 2 ou 4 mémoires 64k. La stratégie de test sera détaillée dans le chapitre 4.



CHAPITRE III

REALISATION DE LA MEMOIRE

4.5 MBITS



I. INTRODUCTION

Ce chapitre est dédié à l'architecture et à la réalisation de la mémoire 4.5 Mbits. L'architecture et la réalisation de la mémoire-WSI obéissent aux différentes contraintes et choix de réalisation définis dans le chapitre précédent.

II. ARCHITECTURE ET REALISATION DE LA MEMOIRE 4.5 Mbits

II.1. Environnement de la mémoire 64k

II.1.1. Dispositif de commutation

Les dispositifs de commutation consistent à connecter/déconnecter les réseaux d'interconnexion aux plots de la mémoire 64k suivants :

- adresses A_0 --> A_{15}
- contrôle WE^* et CS^*
- données Din et Dout
- Alimentation Vdd

Le plot V_{ss} est toujours connecté au réseau V_{ss} . Le plot CS^* sera connecté à la sortie du décodeur.

Les impératifs de test et de configuration nécessitent la connexion (déconnexion) des mémoires 64k aux(des) réseaux d'interconnexion. En effet pendant la phase de test, les mémoires 64k doivent être déconnectées des réseaux pour réaliser leur test individuel, et pendant la phase finale de configuration les mémoires utilisées doivent être connectées définitivement aux réseaux. Il est donc nécessaire d'avoir des dispositifs d'une part facilement et rapidement (re)programmables, et d'autre part pouvant assurer un bon transfert des signaux.

Pour cela nous avons adopté une solution à base de portes de transfert commandées par deux commandes C et C^* . Les transistors MOS à canal P

(PMOS) assurent le transfert de la valeur logique 1 (5 volts) sans dégradation de la tension. Les transistors MOS à canal N (NMOS) sont eux de bons transmetteurs de valeurs logique 0. Par conséquent les portes de transfert sont de bons transmetteurs des valeurs logiques 0 et 1. On les utilise donc pour transférer dans la mémoire les signaux d'adresses, de contrôles, de données qui peuvent être 0 ou 1. L'alimentation Vdd (5 volts) est donc transférée au plot Vdd par un transistor PMOS commandé par C*. Les commandes C et C* sont générées par une logique de commande qui comporte des amplificateurs et un transistor N déplété à grille flottante programmable par faisceaux d'électrons. On illustre dans la figure suivante les différents dispositifs de commutation et la logique de commande.

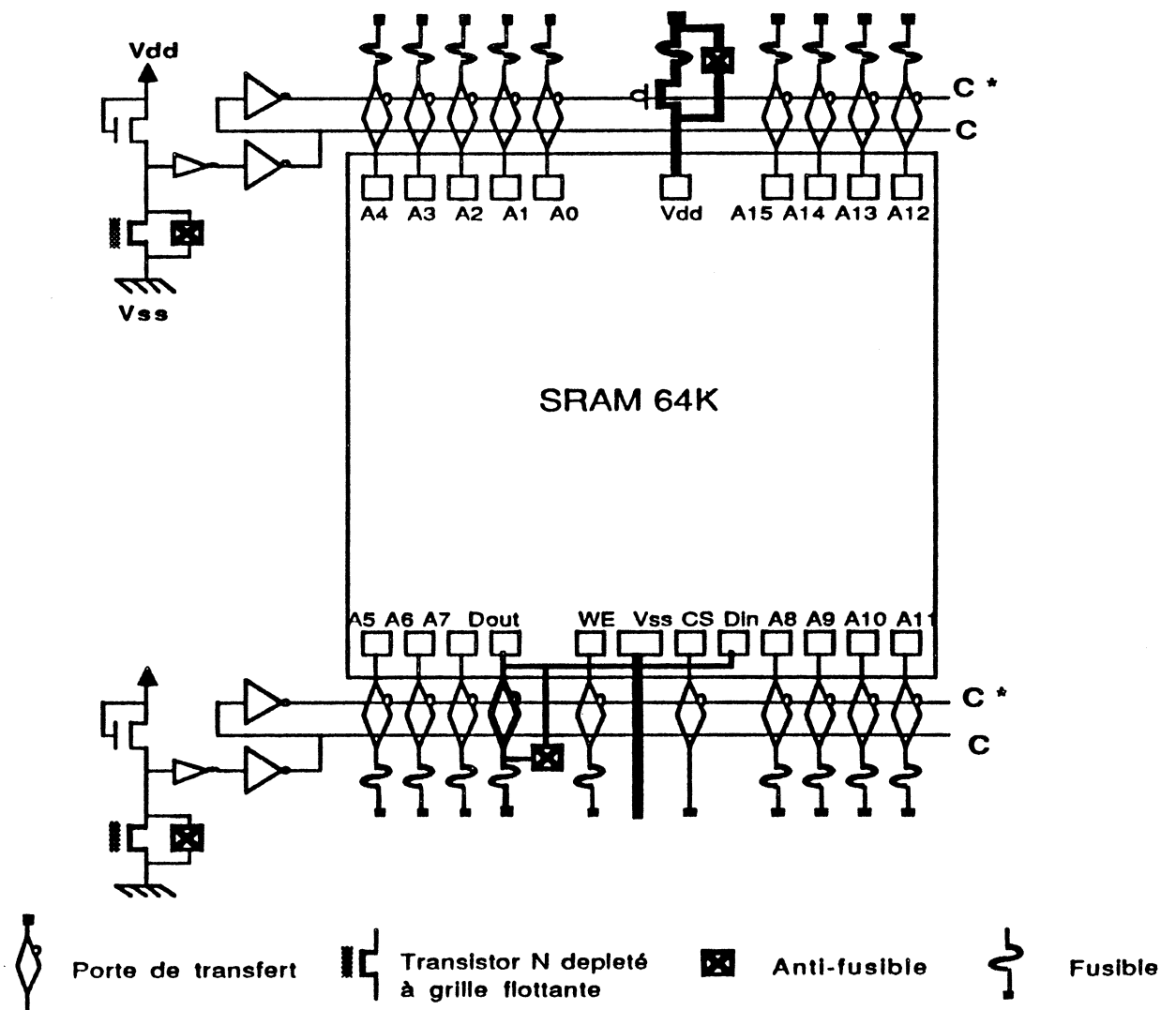


Figure 3.2 : dispositifs de commutation et de commande d'une mémoire 64k

Les portes de transfert de données Din/Dout ainsi que le transistor PMOS d'alimentation doivent être surdimensionnés. En effet, on a vu dans le chapitre précédent que la mémoire 64k fournissait en sortie un courant I_{oh} de l'ordre de 4 mA, et que sa consommation dynamique maximale était de l'ordre de 100 mA. Pour que la traversée des signaux de données et d'alimentation à travers les portes de transfert Din/Dout et à travers le transistor PMOS d'alimentation n'induisse pas une chute de tension importante, il faut que la résistance R_{on} de ces portes et de ce transistor soit la plus faible possible.

Or $R_{on} \approx \alpha L/W$ (α est une constante technologique, L et W représentant la longueur et la largeur de la grille d'un transistor). Ces deux paramètres sont contrôlables. D'après l'équation de R_{on} , pour diminuer la résistance R_{on} il faut augmenter la largeur du transistor W, car L est limitée à $1.2\mu\text{m}$.

Des anti-fusibles seront utilisés dans la phase finale de configuration pour, d'une part court-circuiter les portes de transfert de données et le transistor PMOS d'alimentation afin d'éviter leurs résistances R_{on} et d'autre part, pour court-circuiter les deux transistors N déplétés à grille flottante des deux logiques de commande afin de figer la connexion de la mémoire avec les réseaux d'interconnexion.

Les fusibles utilisés sont en aluminium 2, ils peuvent être coupés par laser et servent à isoler électriquement la mémoire 64k dans le cas d'une défaillance de la logique de commande.

Simulation électrique et réalisation

Une simulation électrique "SPICE" a été réalisée pour le dispositif de commutation et de la logique de commande. Pour cela deux schémas électriques ont été simulés, l'un pour les dispositifs situés en haut d'une mémoire 64k [Annexe3.1], et l'autre pour ceux situés en bas [Annexe3.2].

Dans les deux cas les tailles des transistors sont :

pour les portes de transfert d'adresses :

PMOS : largeur $W = 100\mu\text{m}$, longueur $L = 1.3\mu\text{m}$

NMOS : largeur $W = 100\mu\text{m}$, longueur $L = 1.3\mu\text{m}$

Ron : la résistance équivalente de ces portes $\approx 300\Omega$

pour les portes de transfert de données :

PMOS : largeur $W = 600\mu\text{m}$, longueur $L = 1.2\mu\text{m}$

NMOS : largeur $W = 400\mu\text{m}$, longueur $L = 1.2\mu\text{m}$

Ron : la résistance équivalente de ces portes $\approx 50\Omega$

pour le transistor de transfert d'alimentation :

PMOS : largeur $W = 2400\mu\text{m}$, longueur $L = 2\mu\text{m}$

Ron : la résistance équivalente de ce transistor $\approx 10\Omega$

La simulation montre que le transfert des signaux par les portes de transfert se réalise:

pour les adresses en 3ns [Annexe3.3]

pour les données en 5ns [Annexe3.3]

pour l'alimentation en 6ns [Annexe3.3]

II.1.2. Décodeur de rangée

On a vu dans le chapitre précédent, paragraphe "Organisation virtuelle", que la mémoire 4.5 Mbits est organisée en 18 blocs de 4 mémoires 64k *1bit chacun. Cette organisation nécessite un décodeur qui sélectionne une mémoire 64k parmi 4. Ce décodeur comporte une porte "NON OU" dont les entrées sont programmables par des transistors N déplétés à grille flottante, et d'autres portes simples. Ce décodeur a pour entrées deux adresses de rangée AR1 et AR2 et un signal de sélection de la mémoire 4.5

Mbits CSW* . Il a pour sortie Sdec reliée au plot CS* de la mémoire 64k à travers une porte de transfert.

La figure suivante illustre ce décodeur.

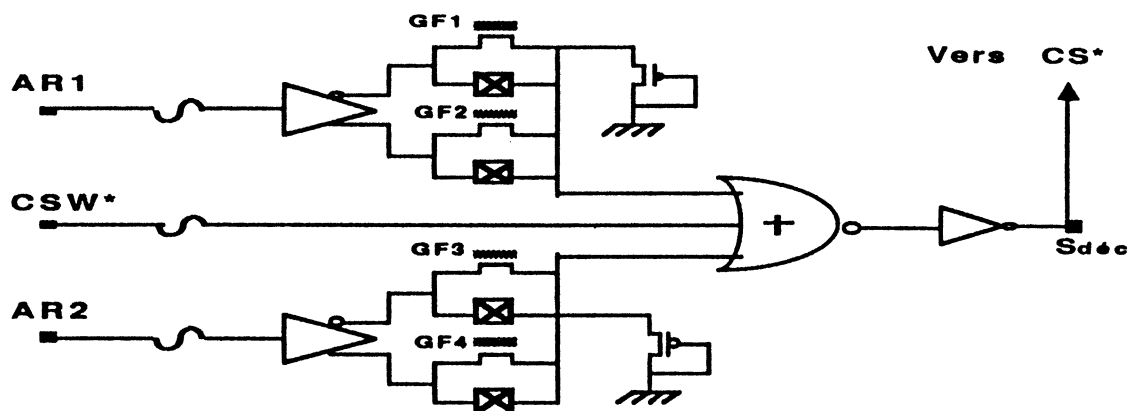


Figure 3.2 : décodeur de rangée

Les transistors à grille flottante sont utilisés uniquement pendant les phases de test et de configuration, car bien que présentant l'avantage de la réversibilité de programmation, leur programmation n'est pas permanente. Ils seront court-circuités par des anti-fusibles pendant la phase de configuration finale .

Pour éviter la modification de la mémoire 64k et pour faciliter le test du décodeur, les décodeurs ont été implantés à proximité des mémoires 64k. La figure suivante présente une mémoire 64k et son environnement (dispositif de commutation, logique de commande, décodeur).

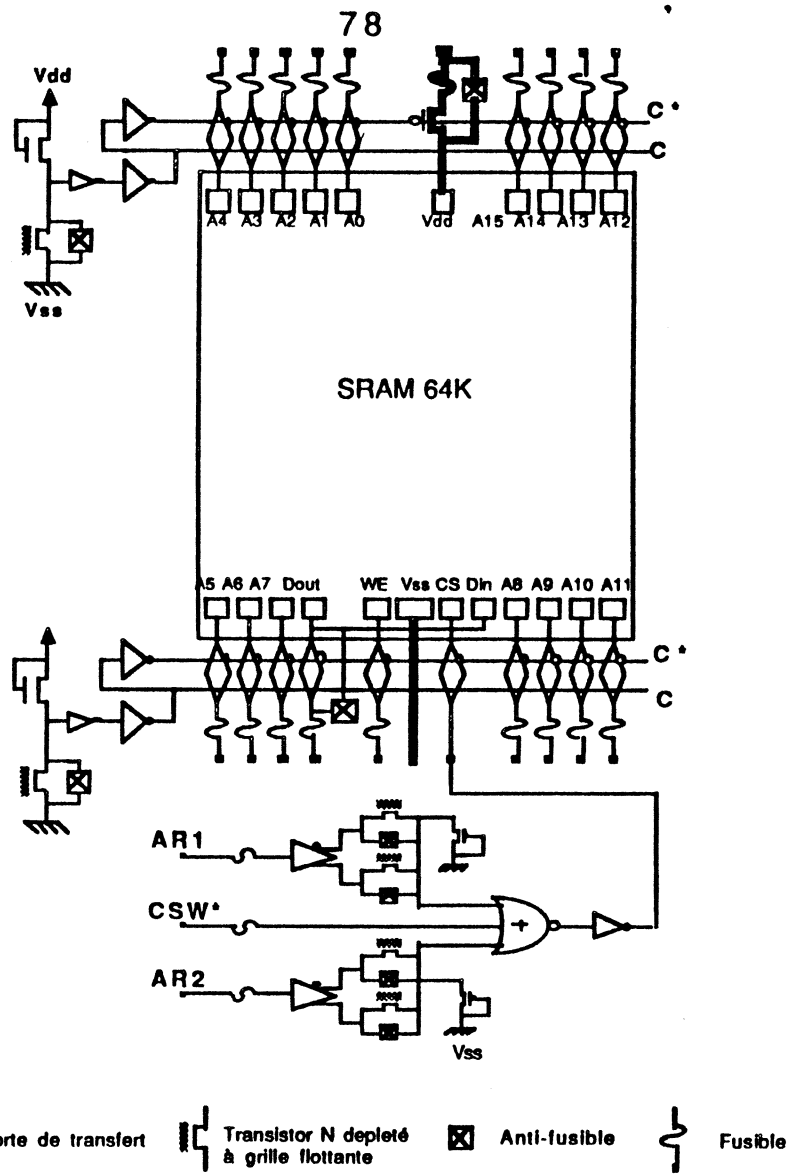


Figure 3.3 : environnement d'une mémoire 64k

configuration de décodeurs

Les configurations possibles des quatre décodeurs (des quatre mémoires) du même bloc sont présentées dans les tableaux suivants :

AR1	AR2	SCW*	GF1	GF2	GF3	GR4	Sdéc
0	0	0	Bloqué	Passant	Bloqué	Passant	0
0	1	0	=	=	=	=	1
1	0	0	=	=	=	=	1
1	1	0	=	=	=	=	1
x	x	1	x	x	x	x	1

AR1	AR2	SCW*	GF1	GF2	GF3	GR4	S _{déc}
0	0	0	Bloqué	Passant	Passant	Bloqué	1
0	1	0	=	=	=	=	0
1	0	0	=	=	=	=	1
1	1	0	=	=	=	=	1
x	x	1	x	x	x	x	1

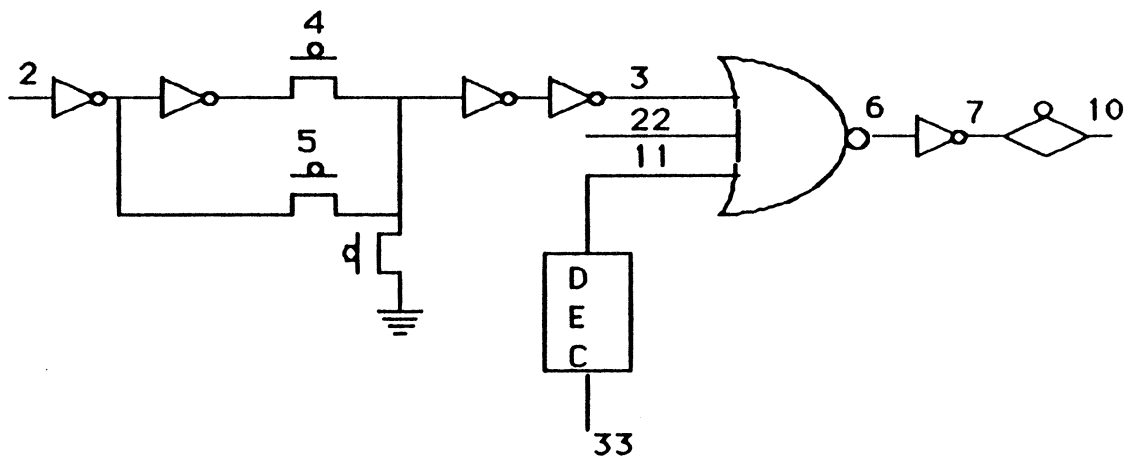
AR1	AR2	SCW*	GF1	GF2	GF3	GR4	S _{déc}
0	0	0	Passant	Bloqué	Bloqué	Passant	1
0	1	0	=	=	=	=	1
1	0	0	=	=	=	=	0
1	1	0	=	=	=	=	1
X	X	1	X	X	X	X	1

AR1	AR2	SCW*	GF1	GF2	GF3	GR4	S _{déc}
0	0	0	Passant	Bloqué	Passant	Bloqué	1
0	1	0	=	=	=	=	1
1	0	0	=	=	=	=	1
1	1	0	=	=	=	=	0
X	X	1	X	X	X	X	1

Tableaux 3.1 : Configurations possibles des quatre décodeurs

Simulation électrique et réalisation du décodeur

Le schéma électrique du décodeur a été simulé en "SPICE" [Annexe 3.4] Afin de déterminer la dimension des transistors et des grilles flottantes :



Les résultats de la simulation ont montré qu'une grille flottante de $30\mu\text{m} \times 10\mu\text{m}$ est suffisante pour une bonne programmation du décodeur. D'autre part le temps de traversée du décodeur (du noeud 3 jusqu'au noeud 10) est de l'ordre de 10ns [Annexe3.5]. Cette simulation a permis le dessin du décodeur (sur une station de travail CAO " DELTA").

II.2. Choix de la forme d'un réticule

La surface utilisable d'une tranche de 4" est égale à la surface d'un cercle de rayon 45 mm, donc égale à $\pi(45)^2 \approx 6362 \text{ mm}^2$. Les impératifs de fabrication (chapitre2) nous imposent d'avoir un réticule photo-répété sur la surface utile de la tranche. Cette surface utile est fonction de la dimension et de la forme d'un réticule.

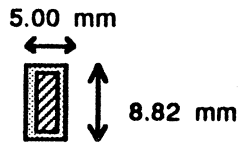
Pour calculer approximativement la surface d'un réticule on admet l'hypothèse suivante :

Pour un réticule constitué de N_x mémoires 64k dans le sens des x , et N_y dans le sens des y , les motifs tels que décodeur, dispositifs de commutation, lignes d'interconnexion ont pour dimensions 30% de Δx et 25% de $(N_y * \Delta y)$. Δx et Δy étant les dimensions de la mémoire 64k.

La surface d'un réticule S est donc approximativement :

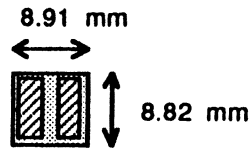
$$S = [(N_x + 0.3) * \Delta x] [(1 + 0.25) * N_y * \Delta y]$$

D'où les différentes surfaces du réticule en fonction des formes et des dimensions approximatives du réticule:



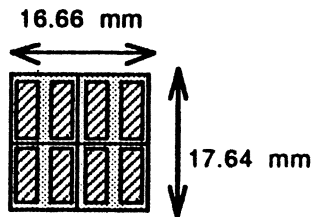
$N_x = 1, N_y = 1$

Surface de réticule $\approx 44.1 \text{ mm}^2$



$N_x = 2, N_y = 1$

Surface de réticule $\approx 78.58 \text{ mm}^2$



$N_x = 4, N_y = 2$

Surface de réticule $\approx 293.88 \text{ mm}^2$

Grâce à un programme qui optimise le placement des réticules sur la surface utile de la tranche (de 4") et qui donne le nombre maximal de mémoires intégrées ainsi que le pourcentage de surface occupée par le réticule [Annexe3.6], on peut décider du choix de la forme du réticule.

On note : $N = N_x * N_y$, le nombre total de mémoires 64k par réticule.

Exemples de placement pour les deux premiers cas :

$N = 1 \rightarrow N_x = 1, N_y = 1$

$N = 2 \rightarrow N_x = 2, N_y = 1$

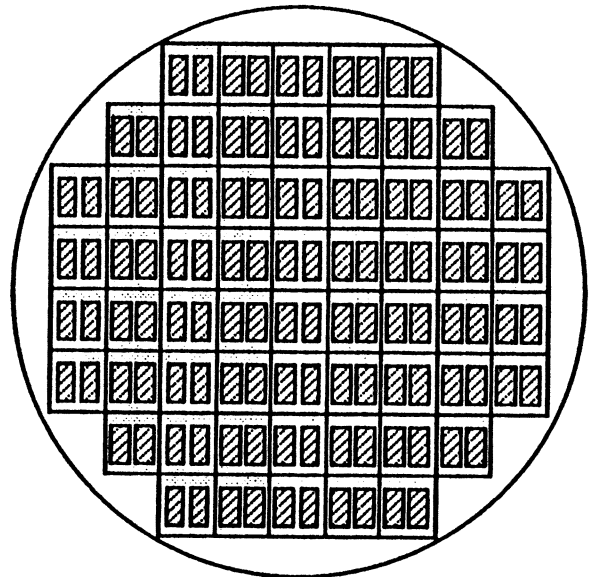
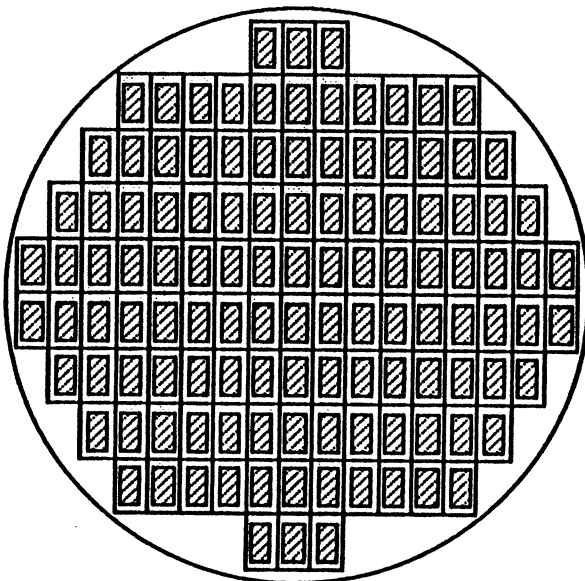


Figure 3.4 : Placement des différents réticules sur la tranche

Dans le tableau suivant on exprime le pourcentage d'occupation de la surface utile, ainsi que le nombre de mémoire 64k intégrées en fonction des différentes formes des réticules.

N = Nombre de mémoire 64k par réticule	Nombre de réticule par tranche	Nombre de mémoires par tranche	Pourcentage de surface occupée par les réticules
1	118	118	81.0 %
2	60	120	74.1 %
8	12	96	55.4 %

Le réticule avec deux mémoires 64k ($N = 2$, $N_x = 2$, $N_y = 1$) sera choisi pour les raisons suivantes :

Il permet d'intégrer le plus grand nombre de mémoires.

Il assure une surface utile de forme à peu près carrée, ceci pourrait permettre la découpe droite du bord de la tranche souvent contaminée par la dernière couche d'aluminium.

II.3. Réseaux d'interconnexion

II.3.1. Introduction

L'intégration des circuits à l'échelle de la tranche (WSI) met en évidence des effets qui pouvaient être considérés négligeables en VLSI.

Ces effets essentiellement dûs à la longueur des lignes d'interconnexion utilisées ($\approx 90\text{mm}$) sont de deux ordres:

- Effets d'ordre électrique relatifs à l'importance de la capacité et de la résistance des lignes. Ces effets se traduiront par une augmentation du temps de propagation des signaux.
- Effets d'ordre physique dûs en grande partie à la technique de photo-répétition utilisée (coupure, court-circuit,...).

Le réseau d'interconnexion de la mémoire-WSI 4.5Mbits a pour but de permettre l'acheminement des données, des adresses, et des signaux de contrôle et d'alimentation des plots périphériques de la tranche vers chacune des mémoires 64 kbits utilisée.

Des problèmes d'ordre électrique vont donc se poser ;

i) Pour les réseaux d'adresses, de commandes et de données ce sera le délai de transmission des signaux.

ii) Pour les réseaux d'alimentations, c'est la quantité de courant transmissible par les lignes qui retiendra notre attention.

Chacun de ces points sera discuté et détaillé dans les paragraphes suivants pour chaque type de réseaux utilisé sur la tranche (réseaux de données, d'adresses, d'alimentations...), de plus des simulations électriques SPICE ont été faites en tenant compte des résultats du Circuit de Test Technologique (CTT) de ESPRIT824-WSI. Ceci dans le but d'estimer le temps de propagation sur les différents réseaux ainsi que les caractéristiques électriques de la mémoire-WSI.

Les effets d'ordre physique sont tels qu'ils peuvent amener des défauts entraînant un non fonctionnement de la mémoire (coupure, court-circuit,.....). Il est donc intéressant d'avoir une structure de réseau tolérante aux défauts.

II.3.2. Epaisseur en Cuivre (copper tracking).

L'un des problèmes majeurs réside dans la résistivité assez élevée des lignes. En effet, à la fréquence normale de fonctionnement de la mémoire-WSI, les appels de courant de chaque mémoire 64K peuvent atteindre des pics de 100 mA. La diminution de la résistivité des lignes devient donc indispensable. Pour cela deux possibilités :

- Un accroissement de la largeur de ligne qui diminuera leur résistivité, mais augmentera leur capacité, le produit RC restant constant.
- Un épaissement avec le cuivre qui permet de conserver la largeur des lignes. Leur capacité restera donc constante, par contre leur résistivité deviendra 3 à 4 fois plus faible. Une étude de faisabilité de ce procédé est réalisé à l'université de CORK [BAR86], et elle sera utilisée pour les lignes en Alu2 après la phase de reconfiguration finale de la mémoire.

II.3.3. Réseaux d'adresses et de signaux de contrôles

Représentation d'un Réseau

D'après les caractéristiques électriques de la mémoire 64K, on peut conclure, que pour maintenir une performance dynamique, et un temps d'accès inférieur à 100 ns pour la mémoire 4.5 Mbits après encapsulation, le temps de propagation d'un signal sur un réseau d'adresse ne doit pas dépasser 25 ns. De plus, ce réseau doit respecter les conditions électriques suivantes :

- Il doit amener pour un niveau logique "1" une tension minimale de 2.5 Volts à chaque mémoire 64K quelle que soit sa position sur la tranche.
- Il doit véhiculer un courant dynamique maximal de 2 mA pour chaque plot d'adresse.
- Il doit être tolérant aux défauts.

Pour répondre aux exigences précédentes, ce réseau ne doit pas contenir des charges capacitives et résistives (telles qu'une série d'interrupteurs) en plus de leurs charges. D'autre part, les plots de la mémoire 64K sont localisés horizontalement, 10 en haut et 13 en bas. Ces contraintes ont conduit à choisir des lignes parallèles et horizontales en Alu2 pour

accéder aux 18 adresses et aux 2 signaux de contrôles. On dispose d'une nappe de 11 lignes en bas des mémoires 64K pour les signaux A5 --> A7, CSW*, AR1, AR2, WE, A8 --> A11, et une autre de 9 lignes en haut des mémoires 64K pour les adresses A0 --> A4, et A12 --> A15.

La photo-répétition ne permet pas d'interconnecter au bord de la tranche les lignes horizontales des différents réticules, d'où la nécessité d'une nappe verticale de 20 lignes dans chaque réticule (sinon il aurait fallu au bord de la tranche un plot pour chaque ligne horizontale). Cette structure est illustrée dans la figure suivante.

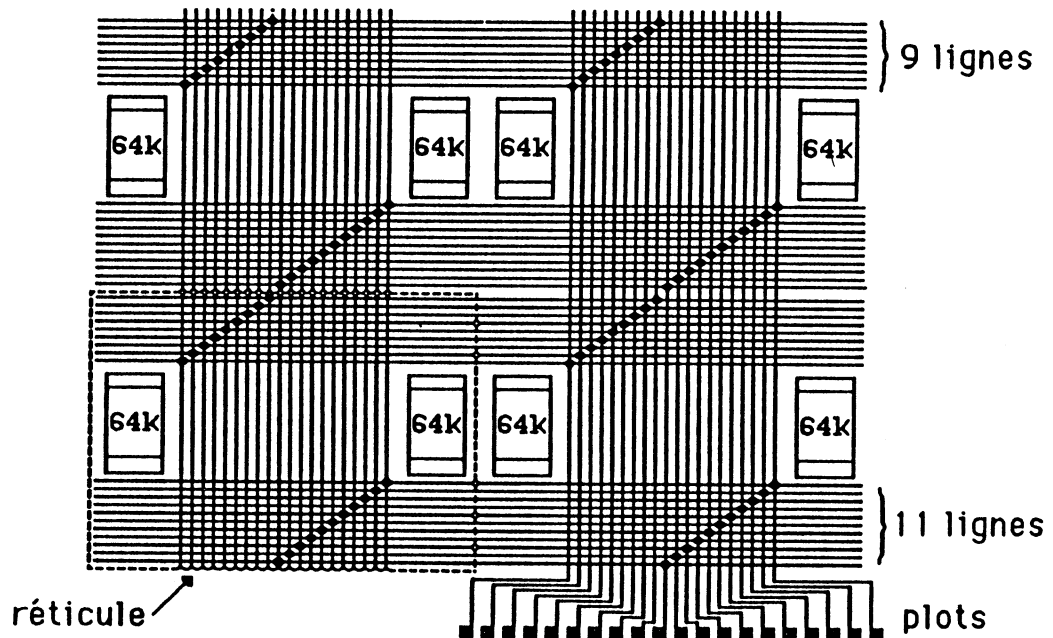


Figure 3.5 : Structure partielle du Réseau d'Adresses

Une ligne horizontale est donc interconnectée avec les équipotentiels des autres réticules grâce à une ligne verticale. Sur les bords de la tranche, quelques lignes horizontales et verticales peuvent être connectées aux plots correspondants de la tranche.

Les lignes horizontales et verticales seront de 10μ de large, espacées de 5μ , et en Alu2 (du fait de sa faible résistivité de $30\text{m}\Omega/\mu^2$), et pour réaliser le croisement des lignes horizontales et verticales, des portions

de lignes verticales en Alu1 seront utilisées. Ces choix répondent aux recommandations précédentes, ils minimisent donc l'inter-influence entre les lignes et les couplages capacitifs, ainsi que le risque de court-circuit ou coupure dû à un léger chevauchement ou rotation du pas de photo-répétition qui est de l'ordre de $0.2 \mu\text{m}$ (voir figure 3.6).



Figure 3.6 : conséquences d'un chevauchement ou d'une rotation d'un réticule

D'autre part, du fait de sa structure en grille, cette solution offre une possibilité simple et efficace de réparation et de reconfiguration car elle tolère des pannes simples du genre coupure, et permet la réparation des courts-circuits, par contre elle est inefficace pour quelques mémoires situées au bord de la tranche (voir figure 3.7).

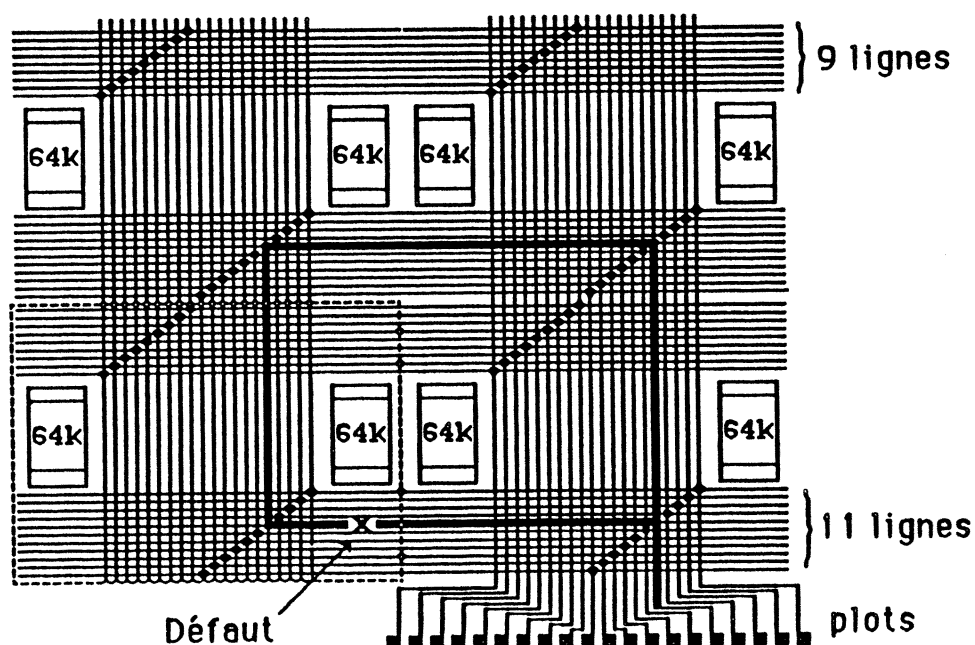


Figure 3.7 : Modèle de réparation d'une ligne d'adresses

Afin de diminuer la capacité du réseau d'adresse vu par un plot extérieur ou de la périphérie de la tranche, on divisera physiquement le réseau des adresses en 4 parties indépendantes. chaque réseau adresse donc seulement un quart des mémoires intégrées sur la tranche (voir figure 3.8).

Plots d'adresses :

A4-->Ao, A15-->A12, WE, AR1, CSW, AR2, A5-->A11

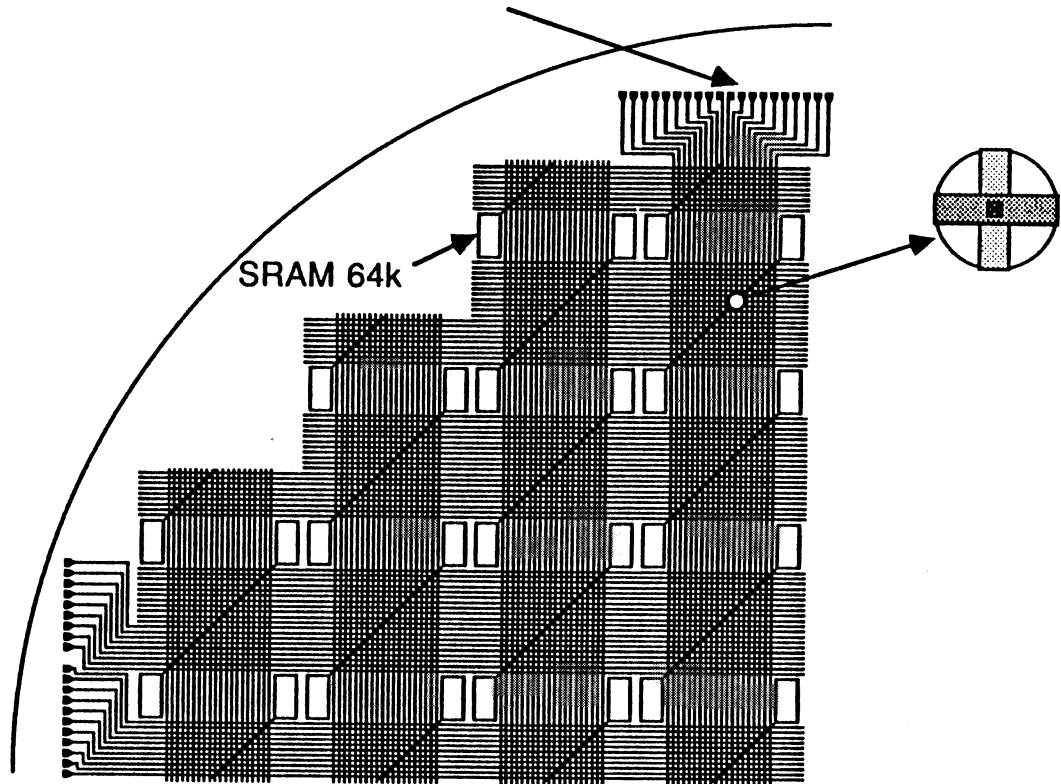


Figure 3.8 : Réseau d'Adresses sur un Quart de Tranche

Chaque réseau d'adresses admet deux groupes de plots, d'où une multiplication par huit du nombre de plots d'adresses sur la périphérie de la tranche.

Simulation électrique d'un réseau d'adresses

D'après l'architecture adoptée, on a établi un modèle pour le réseau d'adresses (figure 3.9), sur lequel on simule le phénomène d'adressage pour une seule adresse et dans un quart de tranche.

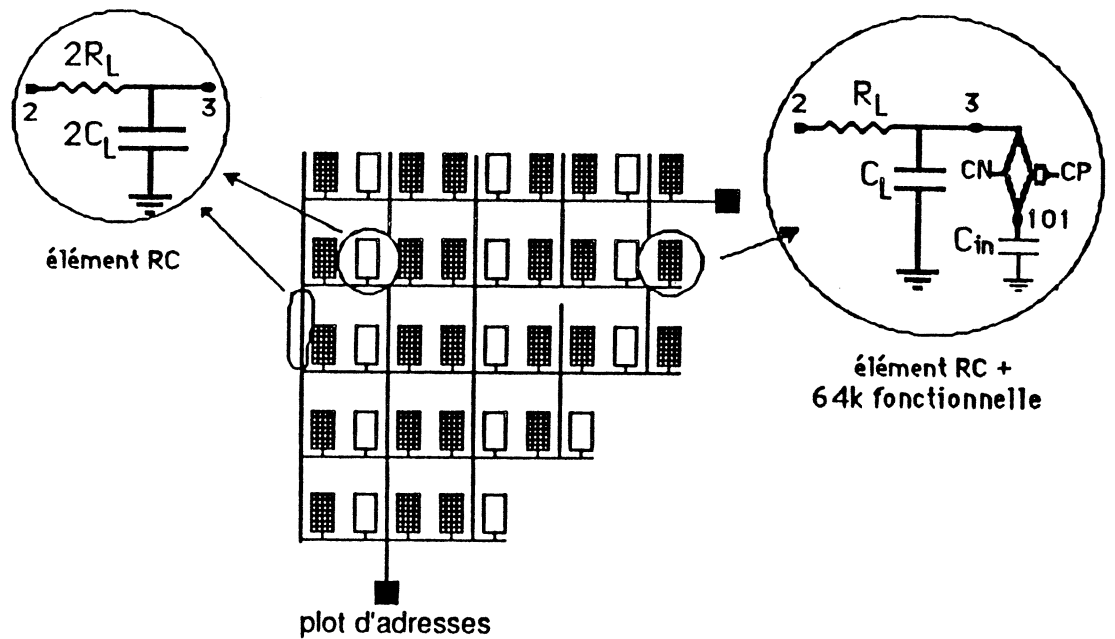


Figure 3.9 : modèle de simulation d'un réseau d'adresses

Dans ce modèle on présente une mémoire fonctionnelle par une porte de transfert dont la résistance équivalente R_{on} des deux transistors NMOS et PMOS est de l'ordre de 300Ω , et par une capacité de plots d'adresses C_{in} égale à $5pF$. Les mémoires non utilisées sont isolées électriquement des réseaux d'adresses. Dans un quart de tranche il peut y avoir 20 mémoires fonctionnelles, et 10 non utilisées.

Les réseaux d'adresses peuvent se représenter électriquement comme une succession en série et en parallèle des éléments RC. Un élément RC est le couple (Résistance = 13.34Ω , Capacité = $1.48pF$) d'une portion de ligne de longueur $4450\mu m$ et de $10\mu m$ de largeur, soit, le couple (R, C) par demi réticule. Dans la simulation on prendra en compte les paramètres électriques suivantes :

- la consommation en courant transitoire d'un plot d'adresses est de $1mA$, et la chute de tension maximale permise à son entrée $dV=2.5$ volts.
- le courant qui passe par le plot d'adresse au bord de la tranche ne doit pas dépasser $10mA$, sinon, la connexion plot /réseau de $10\mu m$

de largeur risquerait de fondre. Or ce courant est contrôlé selon la formule :

$I = C \, dV/dt$, C étant la capacité du réseau et du plot d'adresse, dV la variation de tension, et dt le temps de montée d'un signal d'adresse. Or $C \approx 220\text{pF}$ et $dV = 5$ volts sont fixés, il est donc nécessaire d'augmenter le temps de montée dt d'une adresse.

Les paramètres électriques calculés sont injectés dans le fichier SPICE de simulation [Annexe3.7], dont l'exécution nous donne les résultats suivants :

- Le temps de propagation d'un signal sur le réseau d'adresse est de l'ordre de 40ns en phase de test [Annexe3.8], et de l'ordre de 10ns en phase normale de fonctionnement (après épaissement du réseau avec le cuivre) [Annexe3.9].
- Le réseau d'adresse est capable de véhiculer un courant dynamique de l'ordre de 0.6 mA pour chaque plot d'adresses d'une mémoire 64k fonctionnelle en phase de test [Annexe3.8], et de l'ordre de 2 mA après épaissement du réseau avec le cuivre [Annexe3.9].

Ces simulations nous ont permis de dessiner les portes de transfert et le réseau d'adresses dans un réticule [Annexe3.18]. Les dimensions des transistors PMOS et NMOS de porte de transfert de commutation sont :

- pour le transistor PMOS : $W=100\mu\text{m}$, et $L=1.3\mu\text{m}$
- pour le transistor NMOS : $W=100\mu\text{m}$, et $L=1.3\mu\text{m}$

II.3.4. Réseaux de données

Représentation d'un Réseau

La mémoire-WSI est réalisée avec 18 blocs de 256KX1bit. Le réseau des données doit permettre de configurer ces blocs quelle que soit la distribution des bonnes mémoires sur la tranche. Ce réseau est formé des lignes horizontales situées en bas des mémoires 64K à proximité des plots de données Din et Dout. Le nombre de lignes a été choisi de la manière suivante :

En tenant compte du rendement théorique des mémoires 64k sur la tranche qui est supérieur à 60%, on peut espérer trouver en moyenne 10 à 12 bonnes mémoires parmi les 18 intégrées sur le diamètre de la tranche. Trois lignes horizontales sur le diamètre sont donc nécessaires pour la formation des blocs car les 12 bonnes mémoires vont faire partie de 3 blocs différents. Pour des raisons de sécurité et du fait d'un rendement possible supérieur à 60%, une ligne redondante supplémentaire est prévue (voir figure 3.10).

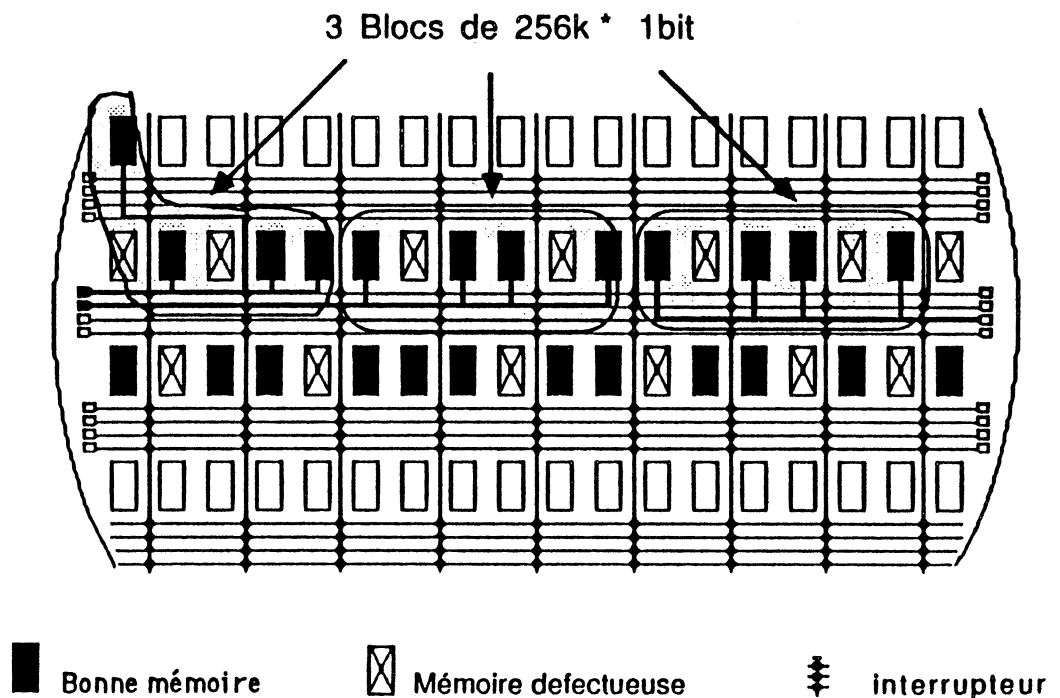


figure 3.10 : Modèle de formation des blocs

Les mémoires 64k d'un même bloc peuvent se trouver sur deux niveaux horizontaux différents, d'où la nécessité d'une ligne verticale par réticule pour interconnecter ces mémoires.

En fonctionnement normal de la mémoire-WSI (lecture/écriture), les lignes de données doivent véhiculer un courant transitoire qui peut atteindre jusqu'à 80 mA, cependant, pour maintenir une bonne performance de la mémoire, on doit minimiser les charges capacitives et résistives sur ces lignes. Cet aspect sera pris en compte dans l'implantation des interrupteurs. Deux solutions sont proposées.

a) On utilise des lignes horizontales en Alu2 de 25 μ m de large, espacées de 5 μ m. Le routage des données est réalisé grâce à des anti-fusibles intégrés directement sur les lignes horizontales.

b) La deuxième solution est semblable à la première, sauf que le routage des données est réalisé grâce à une porte de transfert par réticule et par ligne verticale. Ces portes de transfert sont commandées par des dispositifs à grille flottante. A la sortie de chaque porte de transfert est intégré un amplificateur bidirectionnel (commandé par le signal WE*) capable de charger les lignes de données horizontales également bidirectionnelles. La porte de transfert et l'amplificateur peuvent être intégrés entre les deux mémoires du réticule (voir figure 3.11).

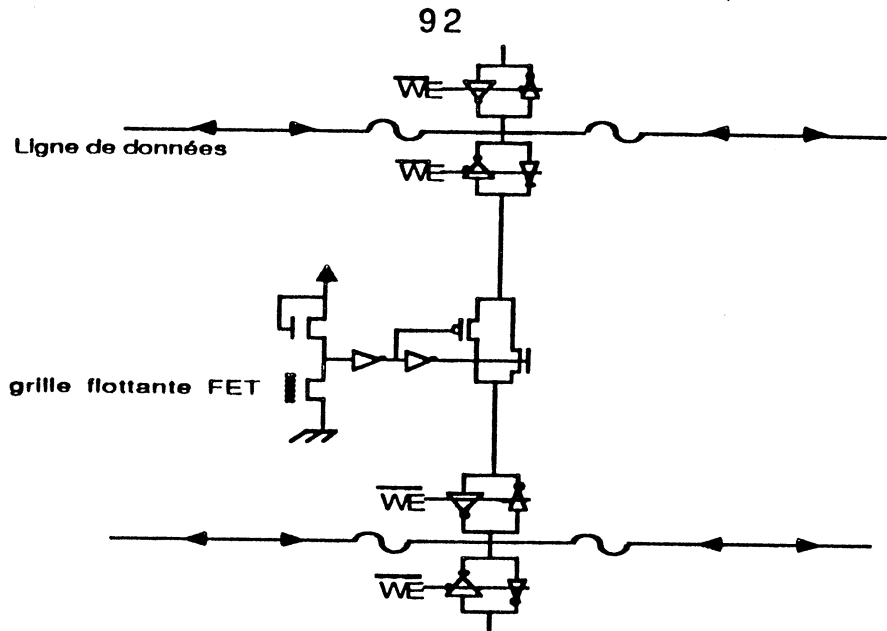


figure 3.11 : Possibilité de Réseau de Données

Simulation électrique d'un réseau de données

En suivant une démarche semblable à celle du réseau d'adresses, on établit des modèles de simulation (pour le pire cas d'une ligne de données) :

Pour la solution a)

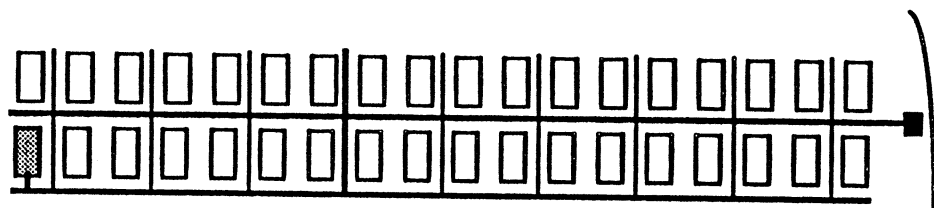


figure 3.12.1 : Une ligne de données pendant la phase de test

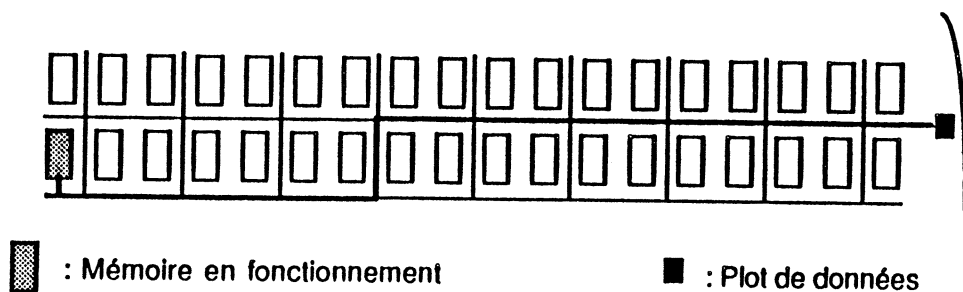


figure 3.12.2 : Une ligne de données après la configuration finale

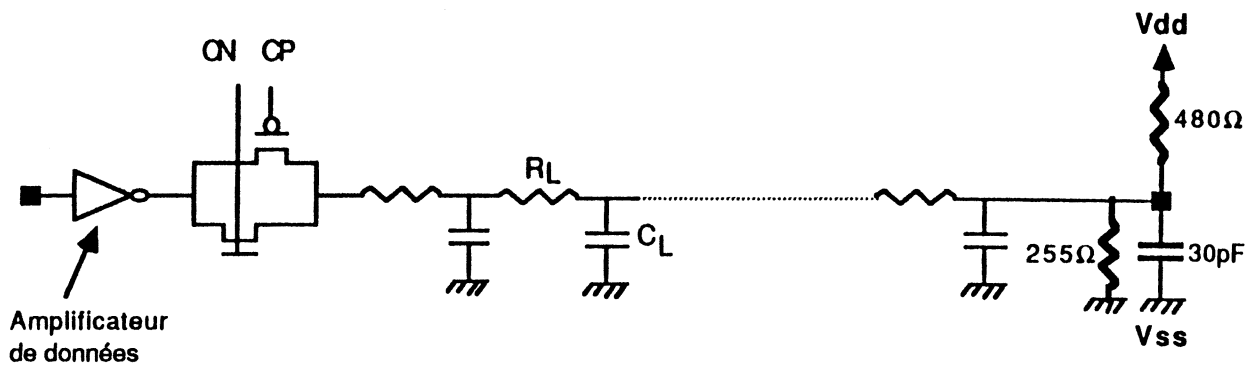


Figure 3.12.3 : Représentation électrique d'une ligne de données en lecture

En fonction des caractéristiques électriques d'un plot de données, on peut déterminer les paramètres électriques de simulation de la plus grande ligne parcourue par un signal de données. En effet, en fonction du courant dynamique maximal I , de l'ordre de 80mA fourni par le plot de données, et en fonction de la chute de tension maximale permise à l'entrée d'un plot de données ($dV=2.5$ volts), on peut déterminer la résistance de ligne (R_t) entre un plot de données de la tranche et celui de la mémoire 64k ($R_t=dV/I$). La résistance de cette ligne est la somme des résistances suivantes:

- R_l : résistance de la ligne d'interconnexion (en Alu2)
- R_{on} : résistance équivalente d'un transistor en mode saturé, pour la porte de transfert qui isole le plot de données.
- R_{conf} : résistance des éléments de reconfiguration et des connexions, déconnexions (fusibles, anti-fusible, G.F.....)

Après le calcul de la résistance d'une ligne $R_l = R_t - R_{on} - R_{conf}$, on peut déduire la largeur de ligne $l = \rho L/R_l$. (L étant la longueur maximale de ligne).

Cette démarche de calcul a été programmée en utilisant le modèle (pour le calcul des capacités et des résistances) étudié dans le chapitre 2. L'exécution de ce programme avec $dV = 2.5\text{volts}$, $R_{on} = 50\Omega$, et le nombre de mémoires 64K (NBCEL), donne les résultats suivants :

- | | |
|--|------|
| - la largeur de ligne | I |
| - la résistance théorique de ligne/mémoire 64k | RL |
| - la capacité théorique de ligne/ mémoire 64k | CL |
| - la capacité théorique propre/ μm | CLPU |
| - la capacité théorique d'influence/ μm | CLPU |

Les paramètres électriques calculés sont injectés dans le fichier SPICE de simulation [Annexe3.10], dont l'exécution nous donne les résultats suivants:

- Le temps de propagation d'un signal sur une ligne de données est de l'ordre de 40 ns en phase de test [Annexe3.11], et 15 ns en phase normale de fonctionnement (après épaissement des lignes, donc la résistance RL dans le fichier SPICE est remplacée par RL/4) [Annexe3.12].
- Une ligne de données configurée peut véhiculer un courant dynamique de l'ordre de 25 mA en phase de test [Annexe3.11], et 45 mA après épaissement [Annexe3.12].
- les dimensions des transistors PMOS et NMOS des portes de transfert de commutation des données (voir le dispositif de commutation) sont les suivantes :
longueur du canal $L=1.3\mu\text{m}$, largeur $WP=600\mu\text{m}$, $WN=600\mu\text{m}$ [Annexe3.10].

Pour la solution b) [Annexe3.13]:

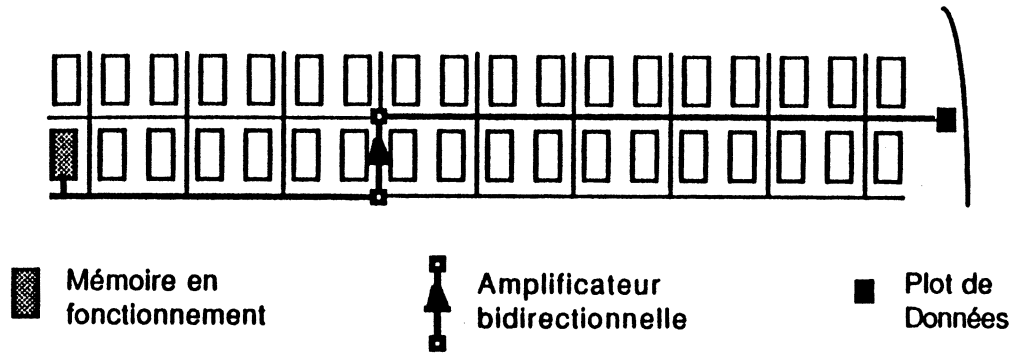


figure 3.13.1 : Une ligne de données pendant la phase de test

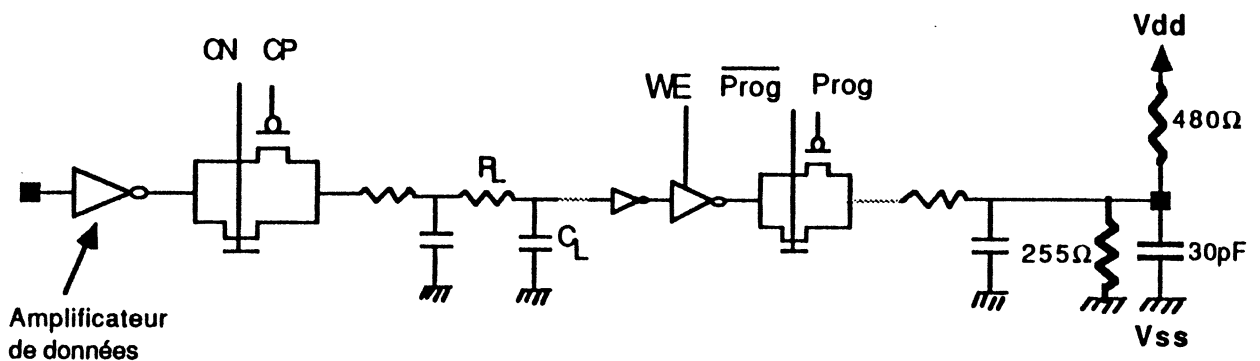


figure 3.13.2 : Ligne de données amplifiée en lecture

- Le temps de propagation d'un signal amplifié sur une ligne de données est de l'ordre de 30 ns en phase de test [Annexe3.14].
- Une ligne de données amplifiée et configurée peut véhiculer un courant dynamique de l'ordre de 35 mA en phase de test [Annexe3.14].
- l'amplificateur et la porte doivent être surdimensionnés (par exemple: la largeur de transistor P de l'amplificateur doit être égale à 600 μ m) pour garantir une bonne performance de la mémoire WSI.

Cette solution bien qu'intéressante du point de vue rapidité, est très

pénalisante du point de vue surface de silicium car elle double la surface verticale entre deux mémoires. Ceci implique une réduction de 6.5% du nombre de mémoires 64k intégrées sur la tranche. Pour cette raison on adopte la solution a) pour le réseau de données. Le réseau de données et le dispositif de commutation sont dessinés au niveau du réticule [Annexe3.18].

II.3.5. Réseaux d'alimentation

Représentation du Réseau

Une grande proportion de la surface allouée aux réseaux sera occupée par les réseaux d'alimentation Vss et Vdd. Ces derniers doivent satisfaire aux exigences suivantes :

- La tolérance de chute de tension aux entrées des plots Vdd est de 1.5 volts en phase de test (car la tension d'alimentation à la périphérie de la tranche est portée à 6 volts en phase de test), et de 0.5 volt en phase normale de fonctionnement (Les mémoires 64K devraient avoir une tension minimale d'alimentation de 4.5 volts).
- Le courant dynamique maximal qui doit être véhiculé dans ce réseau est de 100 mA pour une mémoire 64k active.

Une solution qui satisfait ces exigences consiste à implanter un réseau d'alimentation dans chaque quart de tranche (voir figure 3.14). Vu les contraintes technologiques de la photo-répétition et la nécessité d'avoir une seule capacité de découplage par quart de tranche, la structure des réseaux d'alimentation sera en grille de lignes parallèles en Alu1 et Alu2. Dans le pire des cas, une ligne horizontale en Alu2 alimente 2 mémoires en même temps, et fournit un courant de 100mA pour chacune (pendant la phase de test à basse fréquence), une largeur de 100µm suffirait donc pour ces lignes. L'Alu1 de largeur 100µm est utilisé uniquement dans la région

de croisement. Les portes de transfert et les anti-fusibles ne sont pas utilisés dans le réseau d'alimentation, car ceux ci présentent une charge résistive trop importante.

Simulation du Réseau d'Alimentation

Il s'agit d'une grille de lignes horizontales et verticales qui peuvent être représentées par le modèle suivant :

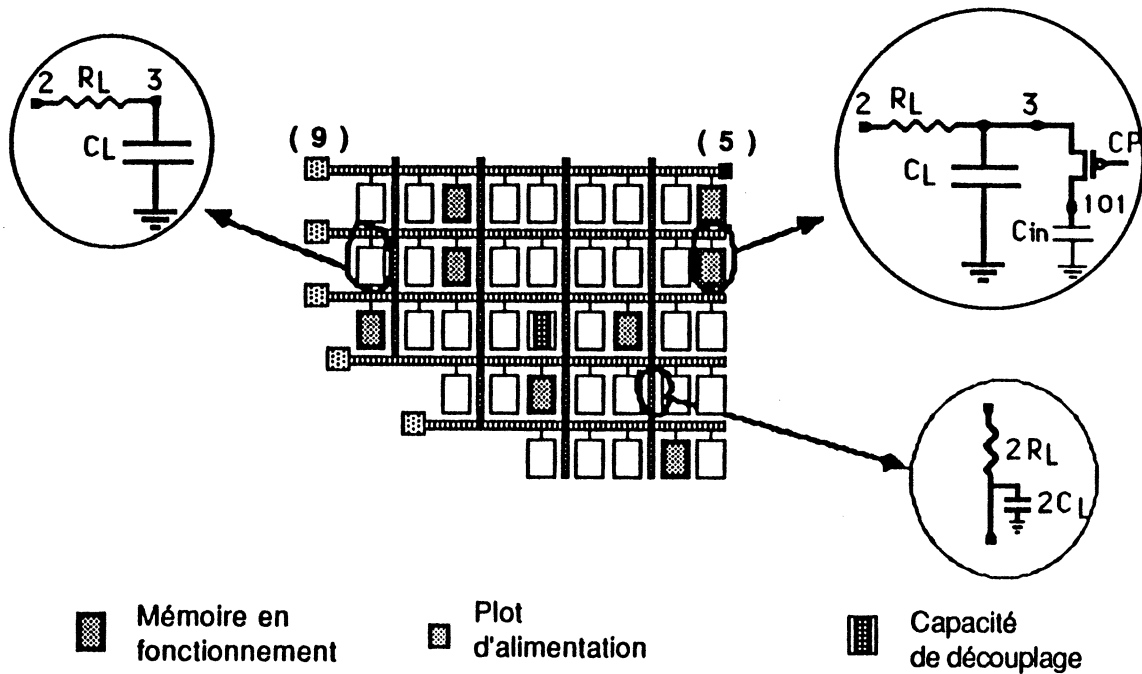


figure 3.14 : Réseau d'Alimentation

Dans la simulation on prend en compte les caractéristiques électriques suivantes :

- La consommation dynamique maximale d'un plot d'alimentation peut atteindre 100 mA.
- La chute de tension (dV) à l'entrée d'une mémoire 64k ne doit pas dépasser 0.5 volt pendant l'appel de courant.
- On alimente les plots de la tranche avec 6 volts en phase de test, et avec 5 volts en phase normale.

Ces paramètres ont été introduits dans le fichier de simulation [Annexe3.15] pour simuler les phénomènes suivants :

- la charge et la décharge du réseau d'alimentation doivent être lentes (temps de montée de l'ordre de 350 ns en phase de test [Annexe3.16], et de 160 ns en phase normale [Annexe3.17]).
- l'appel du courant dynamique par les mémoires 64k (5 mémoires) de l'ordre de 100 mA fait chuter la tension du réseau de 0.5 volts en phase de test [Annexe3.16], et de 0.1 volt en phase normale [Annexe3.17]).
- le transistor PMOS de commutation admet 2400 μ m de largeur, sa résistance R_{on} est de l'ordre de 10 Ω , il est dessiné avec les lignes de réseau d'alimentation au niveau du réticule [Annexe3.18].

II.4. Architecture du réticule

Un réticule est constitué de trois parties principales :

- Deux mémoires 64k
- de dispositifs de commutation et de deux décodeurs de rangée (1 parmi 4) (portes de transfert, dispositif de commande, fusibles, anti-fusibles, grilles flottantes FETs)
- des lignes d'interconnexion pour les différents réseaux

leur placement dans le réticule est illustré dans la figure suivante :

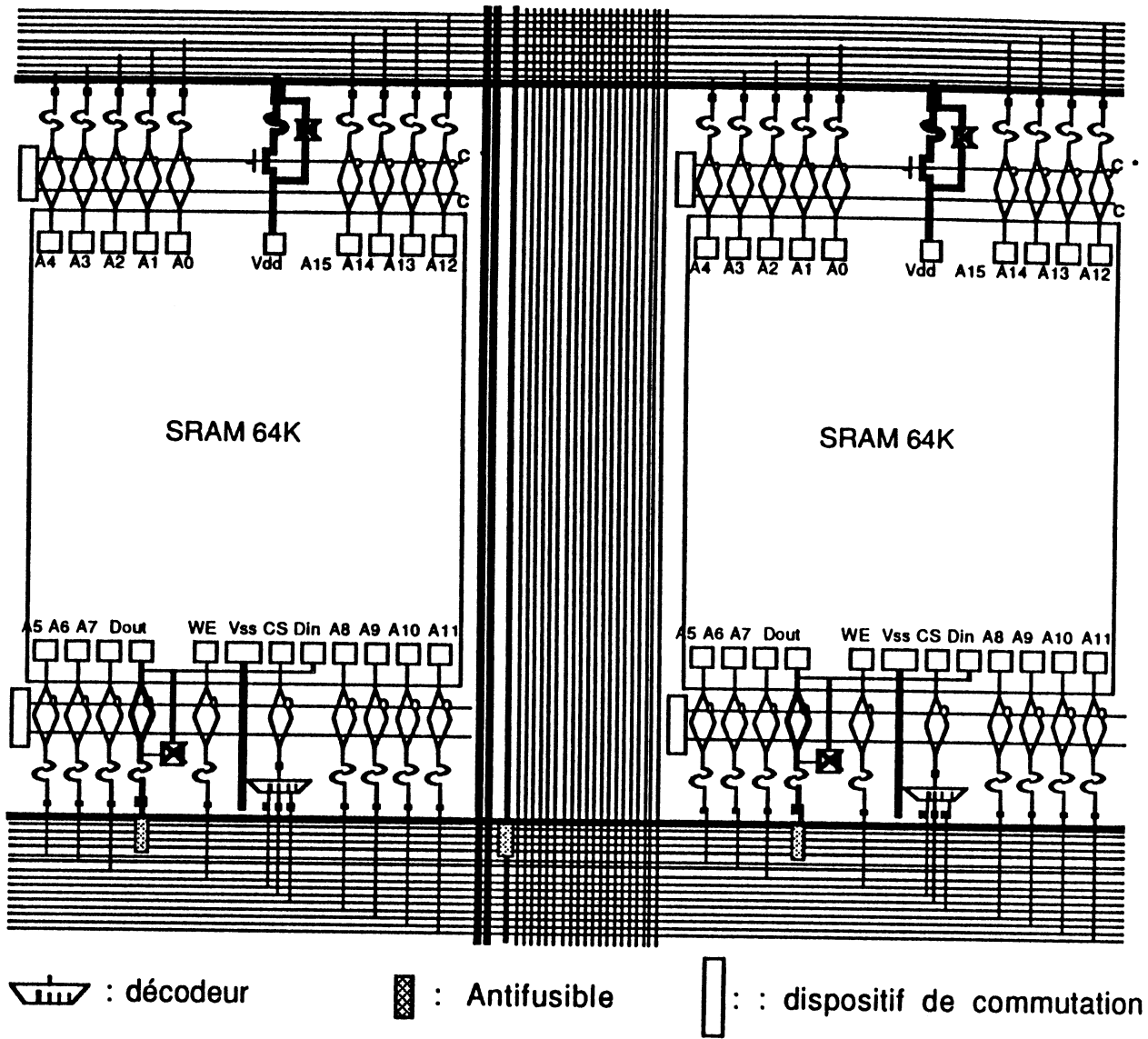
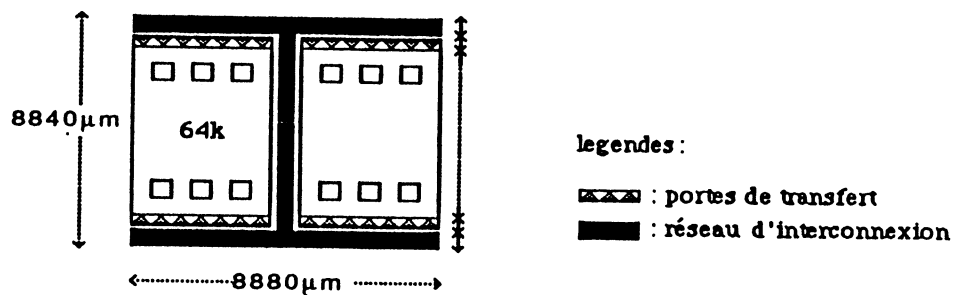


figure 3.15 : schéma électrique d'un réticule

On trouvera en annexe [Annexe3.18] le dessin au μ de ce réticule.

Les dimensions définitives d'un réticule est $\Delta x = 8880\mu\text{m}$ et $\Delta y = 8840\mu\text{m}$.



Un réticule comporte d'autres motifs dessinés tout autour tels que : les croix d'alignement, laser et faisceaux d'électrons. Ainsi que des motifs de test paramétriques.

II.5. Solution choisie sur la tranche

II.5.1. Présentation

Moyennant un programme qui optimise le placement des puces sur une tranche ([Annexe3.6] , on obtient le placement et le nombre maximal des reticules intégrés sur la surface utile (circonférence de rayon 45mm) de la tranche de 4" (voir figure3.16).

Les deux demi-tranches inférieures et supérieures admettent chacune les caractéristiques suivantes:

- Elles sont adressées par 2 réseaux d'adresses, WE, CSW, indépendants.
- Elles sont alimentées par 2 réseaux indépendants.

Le réseau de données sera le même pour les deux demi-tranches, mais il est envisageable de le couper en deux parties pour une meilleure flexibilité d'utilisation de la mémoire-WSI, soit comme une mémoire de 256k*18bits soit comme deux fois 256k*9 bits. Cette solution peut être adoptée en fonction du rendement de la tranche .

La séparation physique des deux moitiés peut se faire par coupure laser.

Au niveau du nombre de plots, cette structure demande sur la tranche :

128 plots d'adresses	A0 ----> A15
16 plots d'adresses rangée	AR1, AR2
16 plots de contrôle	WE, CSW
32 plots d'alimentation	Vdd, Vss
64 plots totaux de données Din, Dout,	dont seulement 18 plots utilisés

Soit un nombre total de plots sur la tranche de 256 dont seulement 210

sont câblés. On réalise ces plots en Alu2 par un masque à l'échelle 1 sur la surface disponible au bord de la tranche. L'extrémité de chaque ligne est connectée à un plot. Cependant ces connections peuvent être coupées par laser après la configuration de la mémoire-WSI. Ces plots sont disposés comme illustré dans la figure suivante :

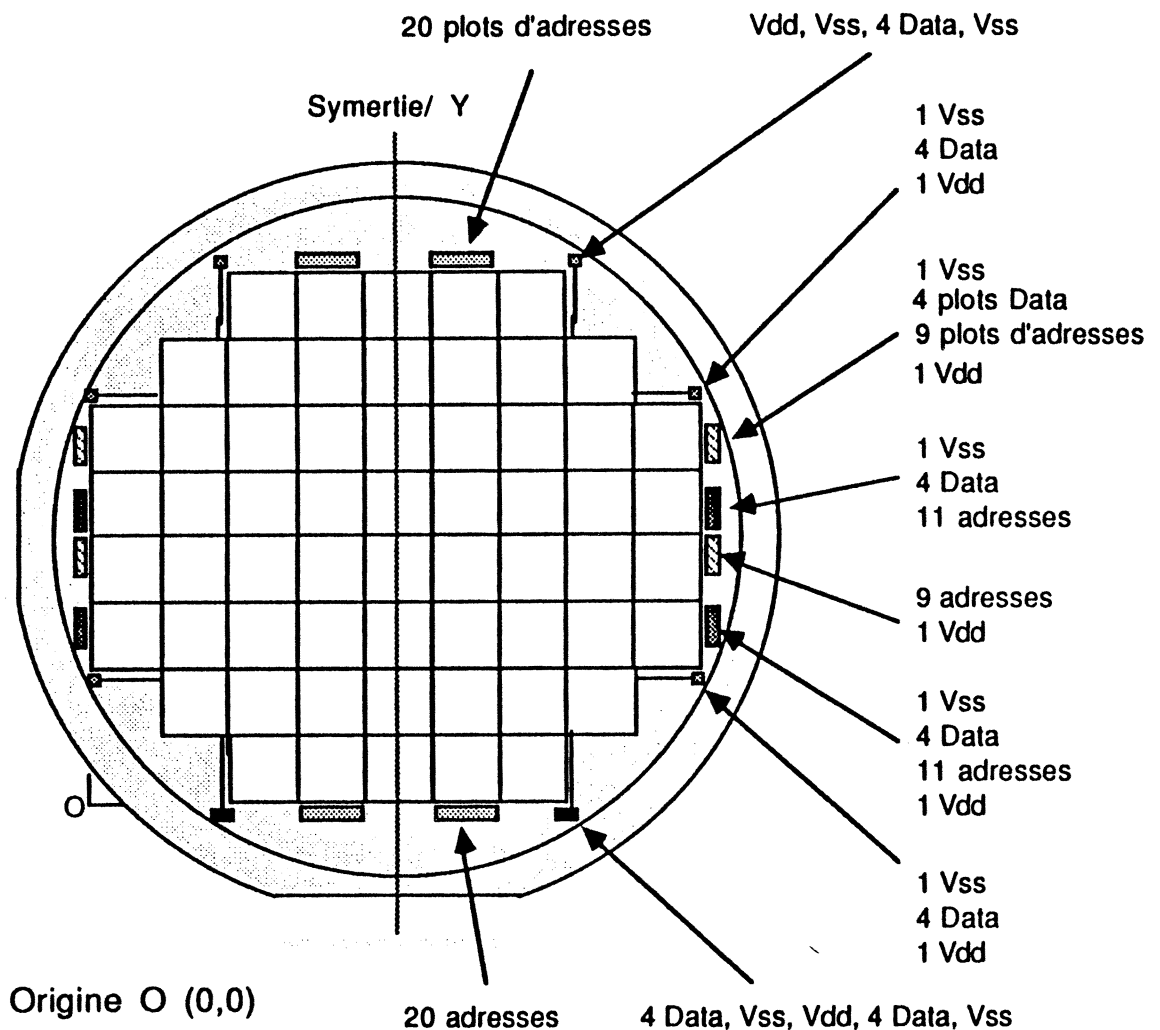


figure 3.16 : disposition des plots et des réticules sur la tranche

On trouvera en annexe [Annexe3.19] l'implantation sur tranche de la mémoire 4.5 Mbits.

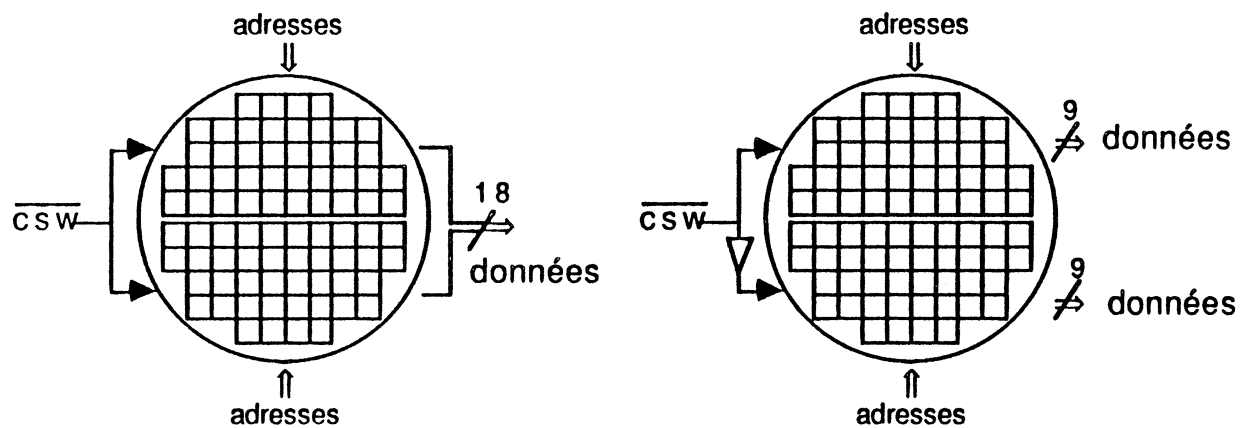
L'épaississement des lignes du réseau d'interconnexion peut être effectué pour l'ensemble des lignes horizontales des différents réseaux ainsi que pour les différentes connexions réseaux/plots de la tranche :

- . lignes de 10 μm pour les signaux d'adresse et de contrôle,
- . lignes de 25 μm pour les signaux de données,
- . lignes de 100 μm pour la distribution de puissance.

Cette solution est intéressante pour des raisons surtout électriques. Elle présente les avantages et les inconvénients suivants :

Avantages:

- les charges capacitatives et résistives totales vues de l'extérieur des réseaux sont divisées par 4 pour le réseau d'adresses. Ce fait est confirmé dans les simulations des réseaux,
- La possibilité d'utiliser deux mémoires de 256Kx9bits séparément ou les deux ensembles comme (256Kx18bits).



- Utilisation de lignes moins larges, car seulement le quart de la tranche est adressé par le même réseau. D'où la possibilité d'avoir moins de courant,
- Meilleure alimentation des mémoires,
- Localisation des pannes sur les lignes par quart de tranche.

Inconvénients

- Nécessité d'avoir plus de plots sur la tranche,
- Nécessité d'avoir le même rendement minimum (60%) dans les deux moitiés de la tranche (256Kx9bits).

II.5.2. Influence du rendement des mémoires 64k sur la solution choisie

La séparation en deux demi-tranches à été choisie pour une meilleure flexibilité d'utilisation de cette mémoire. Par contre, ce choix est coûteux en terme de rendement vis à vis de la configuration des blocs de 4 mémoires de 64k et plus particulièrement dans le cas d'un tranche de faible rendement.

Nous allons modéliser ici les divers cas possibles de découpage de la tranche en deux demi-tranches indépendantes d'égale importance. Nous évaluons par la suite la probabilité de configuration conjointe des deux demi-tranche obtenues.

Nous établirons aussi les relations entre le rendement global de la tranche et la probabilité de configuration de la tranche scindée en deux parties. Adoptons le modèle probabiliste suivant :

La tranche globale est considérée comme une urne opaque contenant des mémoires bonnes, B , et des mémoires défectueuses, D.

Le nombre total de mémoires sur la tranche est $T = B + D$.

Nous définissons un tirage aléatoire d'une partition binaire de la tranche, comme étant le tirage aléatoire de t mémoires ($t = T/2$) de l'urne. La première demi-tranche est constituée des T/2 mémoires tirées au hasard et la seconde demi-tranche est constituée des T-t (soit T/2) restantes dans l'urne.

Une tranche est dite configurable si et seulement si après chaque tirage effectué, on obtient dans chaque demi tranche au moins un nombre critique t_c de mémoires bonnes.

Pour un tirage de t mémoires de l'urne, on a la répartition suivante :

$$t = tb + td \quad \text{ou} \quad \begin{array}{l} tb \text{ est le nombre de mémoires bonnes et} \\ td \text{ est le nombre de mémoires defectueuses} \end{array}$$

On constitue une demi tranche si $t = T/2$, alors la seconde demi-tranche est constituée de :

$$\begin{aligned} t &= tb' + td' \quad \text{ou} \quad tb' = B - tb \\ &\quad \text{et} \quad td' = D - td = T - B - td \end{aligned}$$

D'où la probabilité d'avoir au moins tc mémoires bonnes dans chaque demi-tranche.

$$\text{Prob } [tb \geq tc \quad \text{et} \quad tb' \geq tc]$$

Etant donné que les tirages se font sans reprise, nous sommes donc en présence d'une "loi hypergéométrique" ("loi de Pascal") définie par :

$$\text{Prob } [tb, tb'] = C_B^{tb} C_D^{td} / C_T^t$$

On démontre que la formule précédente est égale à :

$$\text{Prob } [tb, tb'] = C_t^{tb} C_{T-t}^{B-tb} / C_T^B$$

Démonstration :

$$\begin{aligned} C_B^{tb} C_D^{td} / C_T^t &= B! D! (T-t)! t! / (B-tb)! tb! (D-td)! td! T! \\ &= B! (T-B)! (T-t)! t! / (B-tb)! tb! (T-B-td)! td! T! && \text{car } D = T-B \\ &= B! (T-B)! (T-t)! t! / (B-tb)! tb! (T-B-t+tb)! (t-tb)! T! && \text{car } td = t-tb \\ &= B! (T-B)! t! t! / (B-tb)! tb! (t-(B-tb))! (t-tb)! T! && \text{car } t = T/2 \\ &= [t!/tb! (t-tb)!] * [t!/(t-(B-tb))!(B-tb)!] * [B! (T-B)! T!] \\ &= C_t^{tb} C_{T-t}^{B-tb} / C_T^B \end{aligned}$$

C'est la probabilité de tirer t cellules parmi les T cellules dans l'urne de façon que les bonnes cellules soient dans la première demi-tranche et donc que $B-tb$ bonnes mémoires soient dans la seconde demi-tranche. D'où la probabilité de configuration des deux demi-tranches.

$$\begin{aligned} \text{Prob [tb} \geq \text{tc et tb}' \geq \text{tc]} &= \text{Prob [tb} \geq \text{tc et B-tb} \geq \text{tc]} \\ &= \text{Prob [tc} \leq \text{tb} \leq \text{B-tc}] \\ &= \sum_{\text{tc} \leq \text{tb} \leq \text{B-tc}} C_{t}^{\text{tb}} C_{t}^{\text{B-tb}} / C_{T}^{\text{B}} \quad \text{Avec} \end{aligned}$$

$$\begin{aligned} C_n^p &= n!/(n-p)!p! && \text{si } n \leq p ; n \text{ et } p \text{ entier} \\ &= 0 && \text{sinon} \end{aligned}$$

Pour l'application qui nous concerne, on pose $T=120=2t$, $tc=36$, $B=R*T$ où R est le rendement de la tranche entière.

Appelons $P_c(R)$ la probabilité de configuration conjointe des deux demi-tranches, pour un rendement R de la tranche entière.

$$P_c(R) = \sum_{36 \leq \text{tb} \leq [120 \cdot R] - 36} C_{60}^{\text{tb}} C_{60}^{[120 \cdot R] - \text{tb}} / C_{120}^{[120 \cdot R]}$$

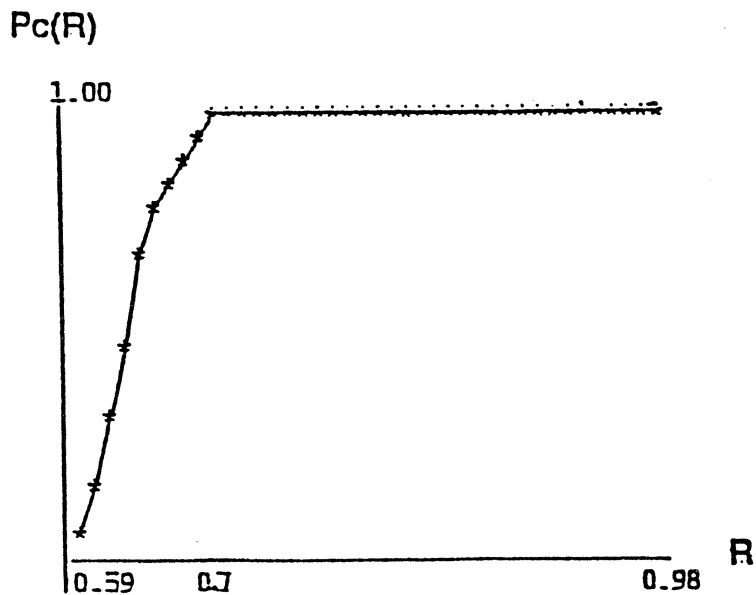
Avec la notation $[120 \cdot R] =$ partie entière de $120 \cdot R$ (plus grand entier inférieur ou égale à $120 \cdot R$).

L'exécution du programme qui calcule P_c en fonction de R , nous a permis de déterminer les valeurs suivantes :

ITERATION	RENDMENT TRANCHE	PROB. CONFIG
10	0.5900000	0.0000000
11	0.6000000	0.1476994
12	0.6100000	0.2914068
13	0.6200000	0.4255048
14	0.6300000	0.553746
15	0.6400000	0.67467594
16	0.6500000	0.7899782
17	0.6600000	0.8768991
18	0.6700000	0.9297742
19	0.6800000	0.9696157
20	0.6900000	0.9829336
21	0.7000000	0.9909033
22	0.7100000	0.9954574
23	0.7200000	0.9978907
24	0.7300000	0.9996480
25	0.7400000	0.9998766
26	0.7500000	0.9999617
27	0.7600000	0.9999893
28	0.7700000	0.9999973
29	0.7800000	0.9999994
30	0.7900000	0.9999996
31	0.8000000	0.9999997
32	0.8100000	0.9999998
33	0.8200000	0.9999999
34	0.8300000	0.9999999
35	0.8400000	0.9999999
36	0.8500000	0.9999999
37	0.8600000	0.9999999
38	0.8700000	0.9999999
39	0.8800000	0.9999999
40	0.8900000	0.9999999
41	0.9000000	0.9999999
42	0.9100000	0.9999999
43	0.9200000	0.9999999
44	0.9300000	0.9999999
45	0.9400000	0.9999999
46	0.9500000	0.9999999
47	0.9600000	0.9999999
48	0.9700000	0.9999999
49	0.9800000	0.9999999

Tab : Probabilité de configuration en fonction du rendement de la tranche

On peut tracer la courbe Pc(R) pour chaque valeur de R.



Courbe : probabilité de configuration en fonction du rendement de la tranche

Nous constatons qu'il faut un rendement de la tranche $R \geq 70\%$ dans le cas de la tranche divisée, pour avoir une probabilité de configuration $\geq 98\%$.

Par conséquent, la séparation de la tranche en deux demi-tranches complètement indépendantes (coupure des lignes de données) n'est pas rentable pour des rendements de la tranche inférieur à 70%.

II.5.3. Rendement de la périphérie des mémoires 64k dans un réticule [DON86]

La périphérie de deux mémoires 64k d'un réticule occupe 30% de la surface d'un réticule (les lignes d'interconnexion occupent environ 25%, et les autres dispositifs 5%). Comme dans le chapitre 2 (dans le paragraphe "Etude théorique du rendement de la mémoire"), on peut calculer la probabilité de n'avoir aucun défaut sur la périphérie (de surface s) de deux mémoires 64k par la loi de :

$$\text{"Murphy - Seeds"} \rightarrow P_i = \frac{\left(\frac{1 - e^{-S.D}}{S.D}\right)^2 + e^{-\sqrt{S.D}}}{2}$$

D , étant la densité moyenne de défauts par cm^2 .

L'application de la formule pour $s=23\text{mm}^2$, et $D=0.5$ défaut/ cm^2 , donne une probabilité

$P_1 = 80\%$, pour que la périphérie soit sans défauts.

II.6. Evaluation de la performance de la mémoire-WSI

Le temps d'accès :

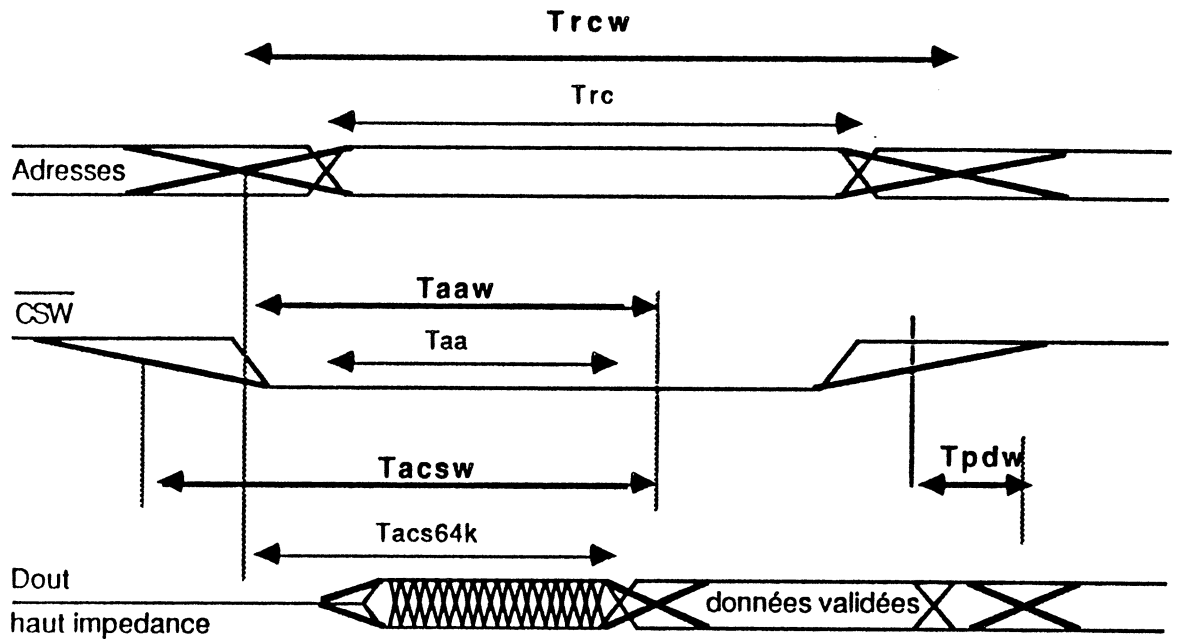
Le temps d'accès de la mémoire WSI, est la somme des deux temps suivants:

- le temps de propagation des données et des adresses sur les lignes,
- le temps d'accès de la mémoire SRAM 64 kbits (≈ 55 ns pour $V_{cc} = 4.5$ volts),

Or dans les simulations précédentes on a estimé le temps maximal de propagation :

- des données (entrantes/sortantes) à 40 ns en phase de test, et à 15ns en phase normale,
- de stabilisation des adresses à l'entrée des mémoires 64k après 30ns en phase de test, et après 10ns en phase normale.

Ces temps de propagation modifient le chronogramme de lecture d'une mémoire 64k de la façon suivante :



Donc le temps d'accès de la mémoire-WSI Tacs64k par rapport aux CSW* est égal à :

$$T_{acs64k} = T_{cs64k} + 40 + 40 = 135\text{ns en phase de test,}$$

$$T_{acs64k} = T_{cs64k} + 10 + 15 = 80\text{ns en phase normale de fonctionnement.}$$

Consommation

La consommation statique de la mémoire WSI sera de l'ordre de 0.5mA.

la consommation dynamique peut atteindre 1.5 Ampère.

Courant de sortie

Le courant de sortie est de l'ordre de 4 mA.

II.7. CONCLUSION

Les résultats obtenus par la simulation valident les spécifications définies dans l'étude de faisabilité (introduction du chapitre II), et seront pris en compte dans la suite.

CHAPITRE IV

*STRATEGIE DE TEST ET DE
CONFIGURATION DE LA MEMOIRE
WSI*



I. INTRODUCTION

Il s'agit de tester les différentes parties de la mémoire-WSI dans un ordre déterminé de façon à assurer un gain de temps de test, et de façon à minimiser les étapes de test, les changements de machine et de site de test.

La stratégie de test de la mémoire-WSI est fortement liée à son architecture. La mémoire-WSI est constituée de mémoires 64k (120 mémoires), de réseaux d'interconnexion, ainsi que de dispositifs de communication qui réalisent d'une part la connexion/déconnexion des mémoires aux réseaux et d'autre part le routage des données sur les lignes d'interconnexion (voir chapitre III).

L'étude menée dans le chapitre II et l'influence entre le test et l'architecture nous ont conduit à établir la stratégie de test suivante :

Test par Carte à Pointes Réticule (CPR) + Carte à Point Tranche (CPT)

Les phases générales de test sont les suivantes :

- Test de la périphérie par une CPR
- Test de 2 mémoires 64K en parallèle par CPR
- Réparation des 64K par fusible électrique (ou Laser+CPR)
- Test des réseaux d'interconnexion -> Réparation par Laser+CPT
- Cartographie des mémoires 64k bonnes
- Configuration + Test de consommation statique par Laser + CPT
- Test de 4.5 Mbits par CPT

Ces différentes phases sont détaillées dans les paragraphes suivants

Pour déterminer une stratégie de test il est important d'étudier les pannes qui peuvent survenir sur ces différentes parties de la mémoire et les erreurs qu'ils peuvent causer.

II. DEFAUTS PHYSIQUE DE FABRICATION

Le bon fonctionnement des différents dispositifs constituant un circuit intégré (transistors, diodes, lignes d'interconnexion...) dépend du soin apporté à leur fabrication. Les défauts essentiels de fabrication d'un circuit peuvent être les suivants :

II.1. Défauts paramétriques

Ce type de défauts est le résultat d'un problème lors du processus de fabrication. En effet, les paramètres électriques d'un transistor sont fortement liés aux étapes de fabrication, telles que les diffusions, implantations ioniques, formation d'isolant. Lors de ces étapes sont formés les sources, les drains, et l'oxyde mince des grilles des transistors. Un mauvais contrôle de la température, des ions déposés... pendant ces étapes va entraîner une modification des paramètres électriques des transistors tels que la tension de seuil (V_t), la résistance passante (R_{on}). Ces anomalies vont apparaître sur tous (ou sur une grande partie) les transistors de la tranche qui impliquera le rejet de la tranche. On peut grouper avec les défauts paramétriques tous les phénomènes de claquage de diélectrique, des effets thermomécaniques, des contaminations ioniques.

Ces défauts sont détectés par le test des motifs dit "motifs de test paramétrique" sur lesquels on intègre des dispositifs semblables à ceux des circuits intégrés tels que les transistors, les diodes,... de taille limitée "ou exagérée". Le bon fonctionnement de ces dispositifs écarte les hypothèses de pannes dues aux défauts paramétriques lors du test des circuits de la tranche.

II.2. Défauts aléatoires

Ces défauts sont la cause principale du rejet des circuits. Comme leur nom l'indique, ce sont des défauts aléatoires qui peuvent arriver sur n'importe

quel niveau. Il s'agit principalement de particules microscopiques (poussière,...) déposées sur la tranche lors des étapes de fabrication, et qui entraînent une mauvaise insolation, un trou dans l'oxyde, ou des motifs manquants ou rajoutés dans les circuits. Ces défauts peuvent se manifester principalement lors de pannes tels que les court-circuits ou les coupures.

III. ERREURS DUES A DES PANNES PHYSIQUES.

III.1. Sur les réseaux

Il s'agit d'étudier les hypothèses de panne et leurs manifestations logiques et électriques, qui peuvent intervenir sur des lignes d'interconnexion en Alu2 et en ALu1 parallèles et intégrées sur le substrat (dopé P) en silicium. Les principales pannes sont les suivantes :

- * **Court-Circuit** : Les courts-circuits possibles sur ces lignes peuvent être :
 - Court-circuit d'une ligne avec une autre voisine de même niveau ou de deux niveaux différents. Ce type de panne induit des erreurs d'influence ; soit deux lignes i et j en court-circuit, le changement de niveau logique $0 \uparrow 1$ ou $1 \downarrow 0$ (ou une transition) sur la ligne i entraîne le même changement de niveau logique sur la ligne j .
 - Court-circuit d'une ligne avec le substrat. Ces court-circuits se produisent pour diverses raisons : un défaut de gravure, trou dans l'oxyde, ... Ce type de panne induit un collage de ligne à un niveau logique zéro ou 1, or dans notre cas les lignes passent uniquement sur le substrat $p+$ polarisé à zéro, donc seul le collage à zéro est possible ; soit une ligne i collée à zéro, cette ligne reste à zéro quel que soit l'état des autres lignes voisines.

* **Couplage capacitif, diaphonie** : Les lignes d'interconnexion sont suffisamment espacées pour négliger les erreurs dues aux influences de couplage capacitif et de diaphonie entre les lignes.

III.2. Dans le dispositif de commutation.

Dans l'implantation du dispositif de commutation, les règles de dimensionnement et de gardes ont été largement respectées, afin de diminuer le risque de coupure et de court-circuit. Cependant, il faut établir une liste des défaillances physiques qui entraînent les erreurs suivantes :

α) mauvaise rétention de la charge sur la grille flottante de la logique de commande, ceci peut induire en phase de test un blocage des portes de transfert correspondantes.

β) des défaillances se manifestant au niveau des transistors (de la logique de commande) telles que coupure sur drain, source, ou court circuit entre drain-source peuvent induire un collage à 0 ou à 1 (selon le transistor N ou P) sur la commande C ou C* de la logique de commande.

χ) il existe d'autres défaillances dans la logique de commande telles que coupure d'une grille, court-circuit entre grille-source, grille-drain qui donnent lieu à des niveaux intermédiaires (entre 0 et 1) sur les commandes C ou C*.

δ) les pannes sur les portes de transfert sont induites soit par des défaillances telles que α, β, χ sur la logique de commande, soit par une défaillance de la porte elle même. En effet, si une défaillance α se présente on verra que toutes les portes de transfert correspondantes restent bloquées, et les défaillances β ou χ induisent une mauvaise transmission de niveau logique via les portes de transfert, car il y aurait

dans ce cas un transistor N ou P bloqué dans chaque porte ; d'autre part, les défaillances de type β et χ , peuvent arriver sur les portes de transfert elles mêmes, ceci donnera les mêmes symptômes que dans la logique de commande.

ε) Les défaillances physiques classiques (coupure, court_circuit,...) dans le décodeur sont fatales pour l'utilisation de la mémoire correspondante, exceptés les défauts sur les transistors à grille flottante qui peuvent être corrigés grâce à des anti-fusibles.

III.3. Sur les mémoires 64k

Définition : Une cellule mémoire élémentaire i est dans l'état x ($x=0$ ou $x=1$) si la lecture à l'adresse i donne la valeur x .

Des nombreuses recherches ont été effectuées dans ce domaine pour déterminer les erreurs de fonctionnement des mémoires [MAR82], [BEH84], on se contente ici de citer les erreurs les plus classiques que l'on peut tester sur la mémoire :

- collage d'une ou plusieurs cellules mémoires à 0 ou à 1; une cellule i est collée à l'état x si, quelle que soit la valeur (x ou x^*) écrite à l'adresse i , la cellule reste toujours à l'état x quelque soit l'état des autres cellules de la mémoire.
- transition de 0 à 1 ou de 1 à 0 d'une ou plusieurs cellules ; Il y a transition de la cellule i si la cellule passe de l'état x à l'état x^* .
- influences idempotentes : la transition (1 à 0 ou 0 à 1) de la cellule j a une influence idempotente sur la cellule i si une transition de j provoque une transition de i seulement quand cette dernière est dans un certain état x . Si i est à l'état x^* la transition de j n'a pas d'effet sur i .

Ces erreurs sont souvent dues aux défauts (court-circuits, coupures..), soit sur les transistors des cellules mémoires, soit sur les lignes de sélection ou de bit de plan mémoire. Il a été démontré [THA77] que les défauts sur le décodeur et sur la logique d'écriture/lecture ont des manifestations fonctionnelles identiques à celles des défauts de plan mémoire telles que les collages, les transitions, les influences des transitions,.....

Certaines défaillances dans le dispositif de commutation (portes de transfert, décodeurs, logique de commande) d'une mémoire 64k peuvent compromettre son utilisation, et d'autres dans la mémoire elle même, tel qu'un court-circuit entre les alimentations Vdd et Vss, peuvent avoir des conséquences graves sur les réseaux d'interconnexion, il est donc important de commencer par le test de la périphérie et le bon isolement des mémoires 64k

IV. TEST DU DISPOSITIF DE COMMUTATION

IV.1. début de test

On débute le test par une phase de validation de technologie et de conception. Il s'agit de faire un test préliminaire grossier sur différentes mémoires 64k et vérifier s'il existe des défauts graves sur toutes les mémoires 64k intégrées. Ce type de test est réalisé sur une seule tranche. La carte à pointes est ensuite posée successivement sur chaque réticule. Les 46 pointes (2 X 23pointes) de cette carte vont servir pour le test des périphéries de 2 mémoires 64k Gauche (G) et Droite (D) en parallèle. Il faut seulement 38 pointes car d'une part, les plots des data Din et Dout sont déconnectés par des Anti-fusibles en fin de fabrication, et d'autre part, le signal CS* est connecté à la sortie du décodeur et non au réseaux. Donc on ne peut tester que les commutations des adresses A0...A15, de signal WE*, et des alimentation Vss ,Vdd.

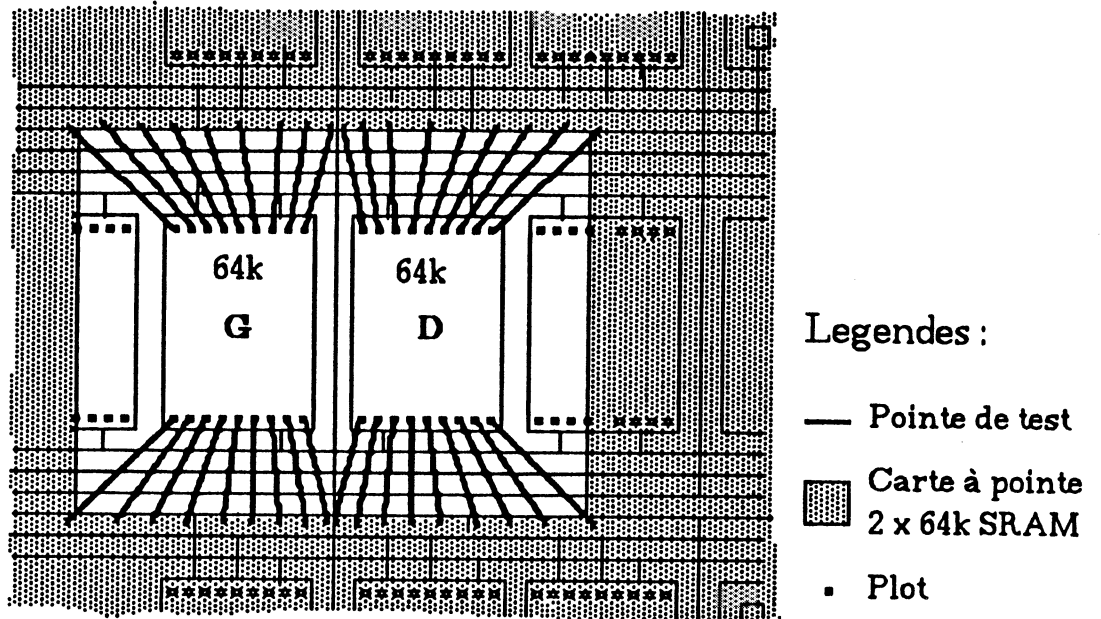


figure 4.1 : Position de la carte à pointes sur deux mémoires 64k

Le commande des dispositifs de commutation est effectuée par le forçage à 0 ou à 1 de deux petits plots directement liés aux commandes C et C* générées par le dispositif de commande. En effet, pendant cette phase de test la mémoire-WSI ne sera pas alimentée, donc le dispositif de commande non plus.

IV.2. Test des portes de transfert

IV.2.1 Chemin de test

le test de ces portes revient à tester le chemin illustré dans la figure suivante :

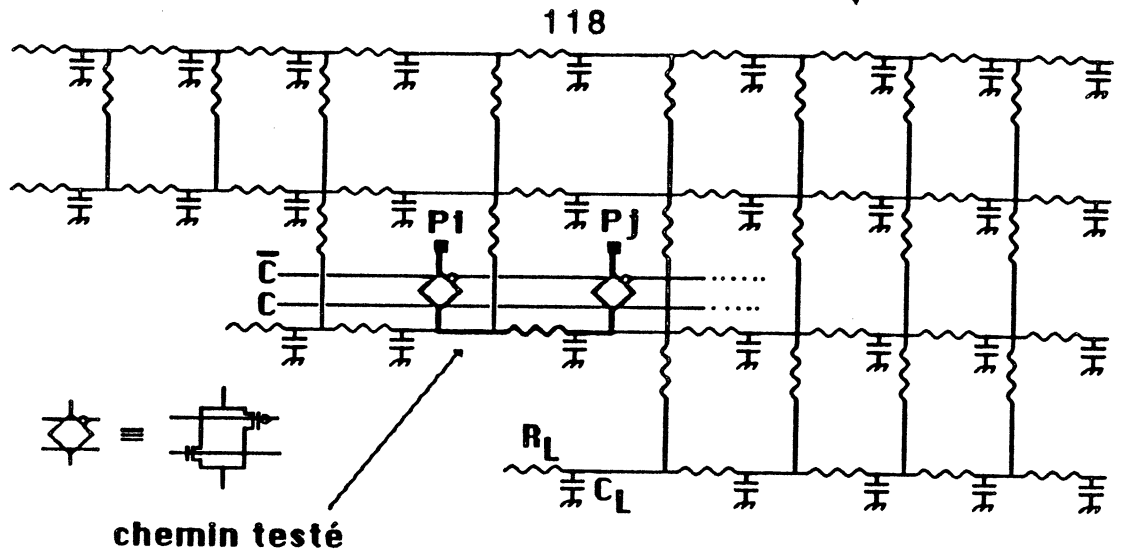


figure 4.2 : le chemin testé entre deux plots de deux mémoires i et j

Le test des portes de transfert est à la fois paramétrique et fonctionnel. Par exemple la variation de tension et de courant, ainsi que le temps de propagation entre deux plots, donnent une information sur le fonctionnement des portes de transfert en question.

On propose un algorithme de test des dispositifs de commutation des mémoires 64k de complexité $O(n)$, n : étant le nombre de plots des mémoires G ou D

$$n = \sum_{i=1}^{19} P_i = 19 \quad i=1 \dots 19$$

IV.2.2 Algorithme de test

Après déchargement des plots P, on étudie l'influence des plots P_i avec $DEP \leq i \leq DEP+18$ des mémoires G ou D forcées à x (0 ou 1) sur les autres plots P_j avec $DEP \pm 19 \leq j \leq DEP+18 \pm 19$.

on note :

MEM_TEST = G ; test de mémoire Gauche

DEP ; plot de départ

k ; paramètre de frontière des plots de test

DECH Pi ; décharge du plot i, ou initialiser le plot i à 0
 TYP_VECT ; type de vecteur de test (XXX ou XYX)
 VAL ; valeur forcée
 FORCE Pi = x ; forçage du plot i à une valeur x
 TEST Pj = x ; tester que le niveau logique sur le plot Pj = x

Si MEM_TEST = G alors DEP := 1; k := 1

Si MEM_TEST = D alors DEP := 20; k := -1

DECH Pi, Initialisation de tous les Pi

Si TYP_VECT = XXX alors

Pour VAL = 0 jusqu'à 1 faire

Pour i=DEP jusqu'à DEP+18 faire FORCE Pi = VAL

Pour j=DEP+19.k jusqu'à DEP+18+19.k faire TEST Pj = 0

FinPour

Sinon

Pour VAL := 0 jusqu'à 1 faire

Pour i=DEP jusqu'à DEP+18 faire FORCE Pi := (i+VAL) mod 2

Pour j=DEP+19.k jusqu'à DEP+18+19.k faire TEST Pj = (j+1+VAL) mod 2

FinPour

IV.2.3. Phases de test

Phase 1

- connexion de la mémoire D au réseau d'interconnexion par forçage des commandes C et C* sur 1 et 0, ceci pour tester la déconnexion de la mémoire G.
- lancement de l'algorithme de test pour les données suivantes :

MEM_TEST = G, TYP_VECT = XXX, DEP = 1

Phase 2

- connexion de la mémoire G et déconnexion de la mémoire D.

- lancement de l'algorithme de test pour les données suivantes :

MEM_TEST = D, TYP_VECT = XXX, DEP = 20

phase 3

- connexion de la mémoire D aux réseaux .
- lancement de l'algorithme de test pour :

MEM_TEST = G, TYP_VECT = XYX, DEP = 1

IV.2.4 Diagnostic de chaque phase et mémorisation.

Dans la phase 1 la mémoire G est normalement déconnectée, donc le forçage à 1 de ses plots P_i ($1 < i < 19$) ne doit pas influencer l'état des autres plots, on doit donc lire 0 sur les plots P_j ($20 \leq j \leq 38$). En conséquence le but de la phase 1 est de tester l'isolement de la mémoire G. La phase 2 a pour but similaire de tester l'isolement de la mémoire D. Les deux phases 1 et 2 nous permettent de détecter des erreurs sur les portes de transfert des mémoires G et D. Ces erreurs peuvent être de deux types :

- de type β ou χ (voir III.1), si elles se manifestent sur des plots voisins, car ce sera du à un défaut de commande C ou C*.
- ou de type δ si elles se manifestent sur un plot quelconque.

Dans la phase 3, les mémoires G et D sont connectées avec les réseaux d'interconnexion. Le forçage à 1 (ou à 0) des plots P_i d'indice impair et à 0 (ou à 1) ceux d'indice pair, doit forcer à 1 (ou à 0) uniquement les plots $P_{(i+19)}$ d'indice pair et à 0 (ou à 1) ceux d'indice impair. La phase 3 permet de détecter des erreurs de type α .

A la détection d'une erreur, le plot ainsi que le chemin soupçonné sont mémorisés, afin de pouvoir les réparer (par Laser) en phase de réparation.

V. TEST DES MEMOIRES 64k

En gardant la carte à pointes sur le réticule testé, un test industriel de fin de fabrication est lancé pour 2 mémoires 64K en parallèle, il s'agit des tests préliminaires, fonctionnels, et paramétriques.

V.1. Test préliminaire

Le but de ce test est de décider le plus vite possible si la mémoire est utilisable ou pas. Il consiste à tester la continuité des connexions entre les pointes et les plots, les défauts graves, tels que les courts-circuits surtout entre Vss et Vdd, les coupures, les consommations excessives, les courants de fuite. La présence d'un défaut de ce genre va nous permettre d'arrêter le test, et juger une mémoire mauvaise.

V.2. Test fonctionnel

Le test fonctionnel consiste à tester le fonctionnement de la mémoire en fournissant des vecteurs de test (en entrées) et en étudiant les sorties qui en résultent, il s'agit de test déterministe des mémoires 64k. Pour le test ainsi que pour les réparations (grâce à la redondance dans les colonnes) de ces mémoires, il est important de connaître leur architecture.

V.2.1 Rappel sur l'architecture de la mémoire 64K

Nous avons vu en III.1.4 que La mémoire 64K est constituée de trois blocs fonctionnels :

- le plan mémoire,
- les décodeurs,
- la logique de sélection et d'écriture/lecture.

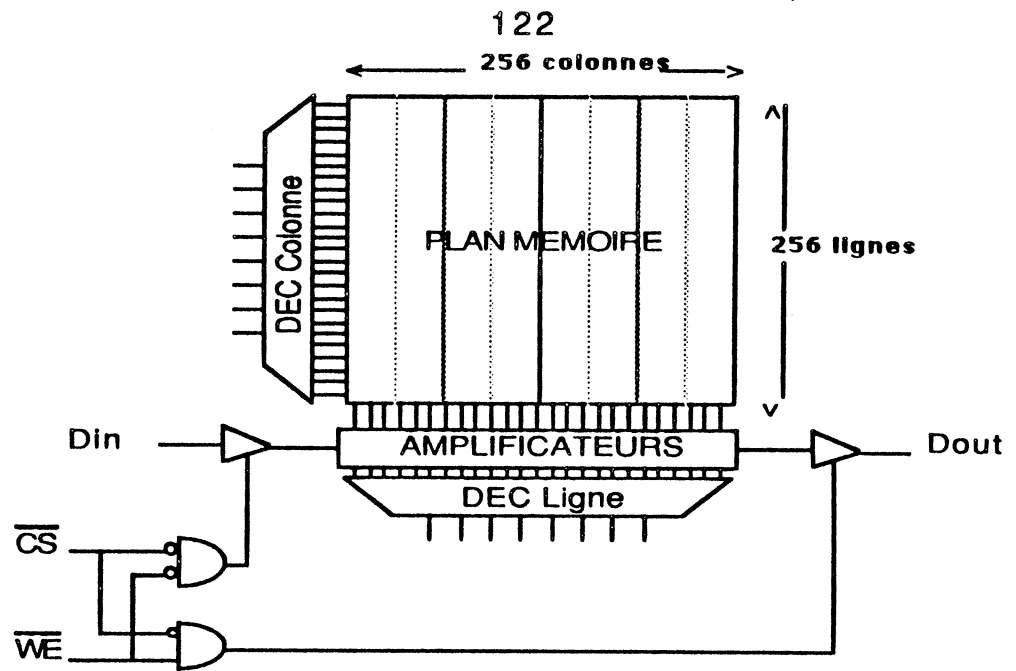


figure 4.3 : schéma des blocs fonctionnels

Le plan mémoire de la mémoire 64K SRAM est réalisé avec des cellules élémentaires à 6 transistors (HCMOS), il est divisé en deux demi plans distincts. 4 colonnes du plan mémoire sont redondantes, en effet on peut remplacer une colonne défaillante par une bonne dans chaque quart de plan. Le décodeur colonne à 8 adresses, 7 adresses permettant de sélectionner une colonne parmi 128, et une 8^{ème} adresse sélectionne l'un ou l'autre demi plan mémoire. Un décodeur ligne commun aux deux demi plans sélectionne une ligne parmi 256 lignes.

Ces renseignements sur l'architecture de la mémoire 64K, nous permettent de ne considérer que les types d'erreurs classiques citées dans le paragraphe III.2 (collages, transitions, influences des transitions,.....).

Les familles des algorithmes Read/Write, Marching, Walking, Galoping [HEN76], [MAR82], [BEN84] nous permettent de tester ces types d'erreurs, ils sont très utilisés dans l'industrie pour le test des mémoires, leur complexité varie entre $O(n)$ et $O(n^2)$.

V.2.2. Les critères de choix d'un algorithme.

On peut juger un algorithme d'après les critères suivants :

- la complexité qui s'exprime en nombre d'accès mémoire, et qui détermine le temps de test.
- la couverture des pannes détectées par l'algorithme.
- la facilité d'implantation, et les matériels de test nécessaires.

Une analyse des différents algorithmes nous a permis de choisir un algorithme de test Damier ($4n$) qui répond aux critères ci dessus.

V.2.3 Algorithme Damier

Le principe de cet algorithme consiste à écrire un damier dans la mémoire testée, puis de le lire entièrement, cette séquence est répétée avec le damier complémentaire. Cet algorithme admet les phases suivantes :

- initialisation du damier
- lecture de la mémoire
- initialisation du damier complémentaire
- lecture du damier complémentaire

La complexité de cet algorithme est donc de $4n$. n étant le nombre de cellules mémoires élémentaires. L'algorithme détecte uniquement les collages à 1 ou à 0, les transitions $0 \rightarrow 1$ ou $1 \rightarrow 0$ des éléments mémoires. L'algorithme de test en Damier est complété par des motifs simples tels que :

- écriture/lecture des 0 plein champs
- écriture/lecture des 1 plein champs

V.2.4. Réalisation du test

a) Ecriture d'un damier

Pour écrire un damier topologique dans une mémoire 64k, on a besoin avant tout de connaître la correspondance entre les adresses logiques des cellules mémoires élémentaires, et leur emplacement physique dans le plan mémoire ou leurs adresses topologiques. En effet, les adresses logiques sont générées séquentiellement moyennant un compteur/décompteur qui réalise en fonction du signal Up/Down, soit le comptage de 0 à 2^{16} soit le décomptage de 2^{16} à 0. Ces adresses sont injectées successivement dans le testeur qui par une opération de conversion (en anglais SCRABLING) retrouve les adresses topologiques des cellules mémoires élémentaires et affecte à chacune d'elles une valeur 0 ou 1 afin de construire le damier. Le phénomène de conversion nécessite de connaître parfaitement le décodage lignes/colonnes de la mémoire 64K.

b) Test ou lecture du damier

La génération des séquences de test peut se faire de deux manières différentes :

b1) Génération des séquences pipelinées

La génération pipe line se fait à une grande vitesse (de l'ordre de 10 MHz). elle consiste à lire complètement la mémoire sans interruption du test. La mémorisation des erreurs ou précisément des comptes rendus de la comparaison entre les valeurs lues dans la mémoire et les valeurs attendues, peut se réaliser en utilisant deux dispositifs :

- un drapeau (Flag)
- une mémoire rapide de saisie (Fast Capture Memory).

* **Utilisation d'un Flag** ; dans le cas de détection d'une erreur, le testeur envoie un signal qui change l'état du Flag. A l'achèvement du test

cet état nous indique la présence ou pas d'une (ou plusieurs) panne dans la mémoire testée.

Si la mémoire est déclarée bonne on continue les autres étapes de tests.

Sinon on lance une étape de configuration et de réparation dans **chaque quart de plan mémoire**, cette étape comporte plusieurs phases :

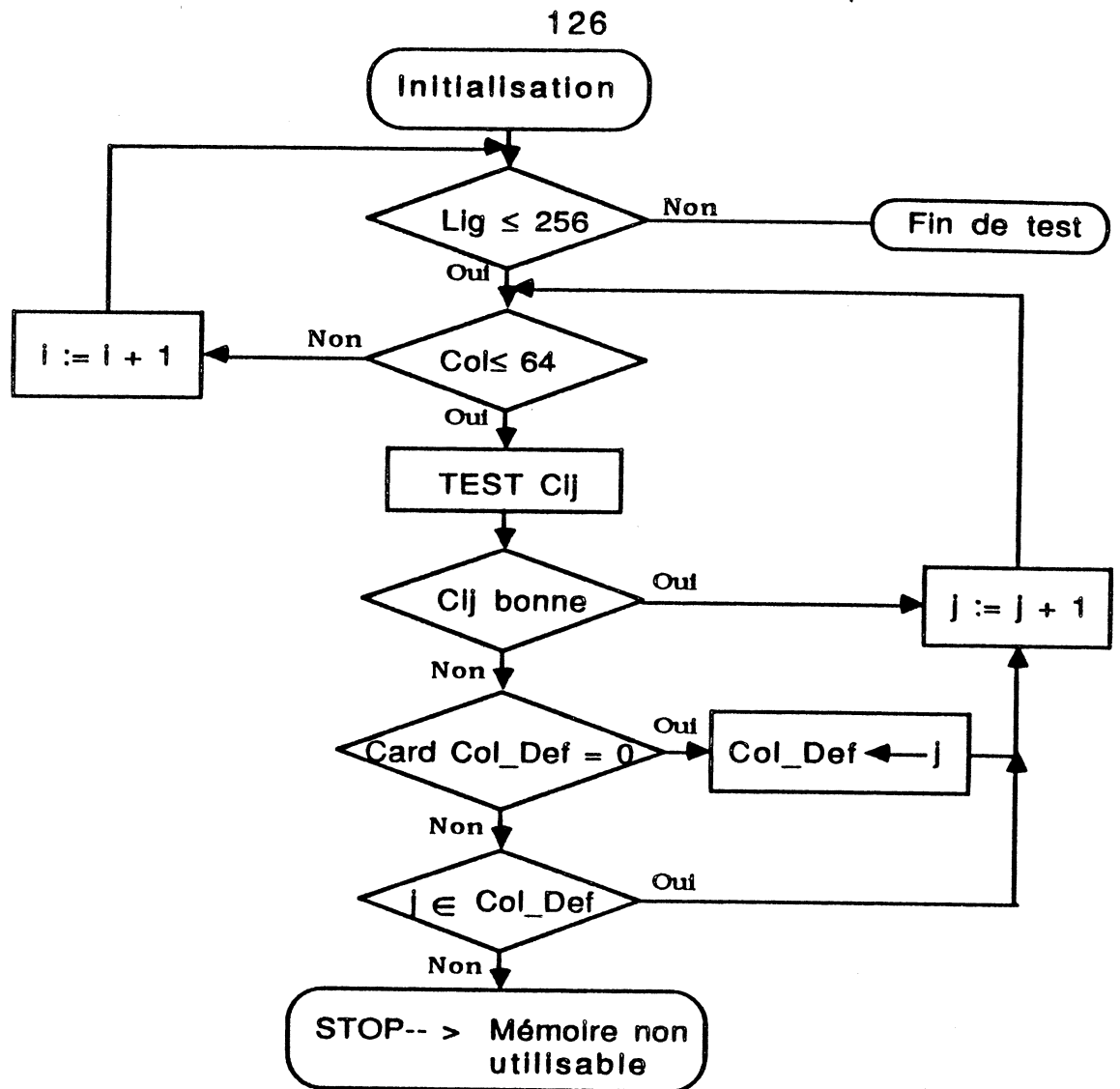
Phase 1 : décision de réparation

Elle consiste à exécuter l'algorithme interprétatif suivant :

Pour chaque ligne i sélectionnée on explore toutes les colonnes j du quart de plan et on teste toutes les cellules C_{ij} qui se trouvent sur la même ligne et sur des colonnes différentes. A la présence d'une erreur sur C_{ij} on mémorise la colonne défaillante j , si c'est la première erreur rencontrée. Sinon deux cas sont possibles :

- la colonne j est déjà jugée défaillante -- > on continue le test
- la colonne j est une deuxième colonne défaillante --> on stoppe le test, et la mémoire 64k est jugée mauvaise car on ne peut pas réparer plus d'une colonne par quart de plan.

Organigramme de la phase 1 :



Phase 2 : réparation (utilisation de la redondance de la mémoire)

La phase de décision nous indique si la mémoire est réparable ou pas. Si oui, on répare la colonne défailante par une saine et ceci par claquage électrique (voir chapitre II, paragraphe "la mémoire SRAM 64K).

* **Utilisation d'une mémoire de saisie;** cette mémoire est capable mémoriser les contenus de plusieurs plans mémoires. Dans notre cas elle peut être utilisée pour restituer l'image du damier et du damier complémentaire écrits dans la mémoire testée, ainsi que les résultats de test présentés comme une matrice contenant dans les cases i, j des 0 si les cellules C_{ij} sont bonnes et 1 si elles sont mauvaises (voir figure 4.4).

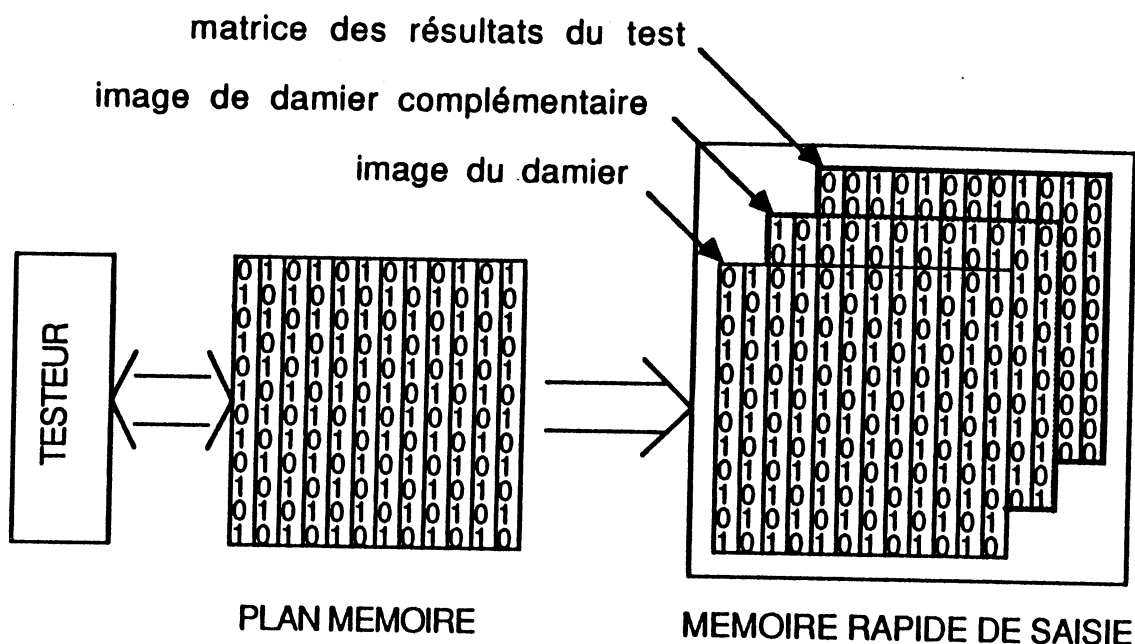


figure 4.4 : Utilisation d'une mémoire de saisie

Dans le cas où un élément de la matrice est égal à 1 la mémoire testée est mauvaise, mais peut être réparable. Pour cela on lance deux phases.

Phase 1 : décision de réparation

Comme précédemment, cette phase consiste à lancer l'algorithme précédent en exploitant les éléments de la matrice de résultats, mais ici on teste le résultat du test d'une cellule mémoire élémentaire et non la mémoire elle même.

Phase 2 : réparation ; elle est la même que précédemment.

b2) Génération des séquences pas à pas

Pour chaque lecture de la mémoire, l'adresse de la cellule mémoire élémentaire lue est maintenue jusqu'à l'apparition des résultats de comparaisons, deux cas sont alors possibles :

- absence d'erreur -----> .génération de l'adresse suivante
- soit présence d'erreur --> .l'adresse courante est maintenue
 .réparation de la colonne défailante
 .incrémentatation du compteur réparation

L'inconvénient de cette méthode est que le test doit se dérouler à basse fréquence pour éviter l'écrasement des adresses (ou le vol des cycles) pendant l'attente du résultat de comparaison. D'autre part, la réparation des colonnes se fait à la volée, donc avec cette méthode on ne peut décider si la mémoire est utilisable ou pas qu'après 4 réparations (ou après 4 incrémentations du compteur réparation)

V.3. Test dynamique

Il s'agit de mesurer le temps de propagation des différents signaux, les largeurs, ainsi que les fronts des impulsions, les chronogrammes des entrées/sorties, et le temps d'accès de la mémoire.

V.4. Test paramétrique

Il consiste à mesurer l'effet de variation de tension et de courant ainsi que leurs limites, les paramètres des sorties V_{ol} I_{ol} V_{oh} I_{oh}, ainsi que I_{cc} dynamique, I moyen pendant une période donnée de fonctionnement, et vérifier si ces paramètres sont conformes aux spécifications de la mémoire 64K. Un test dit logique, avec des contraintes dynamiques est nécessaire, il nous permet de confirmer que la mémoire fonctionne correctement dans les limites normales de son utilisation telle que la tension d'alimentation et la température.

VI. TEST DES RESEAUX D'INTERCONNEXION

VI.1. Présentation des réseaux

Sur toute la tranche on dispose des réseaux d'interconnexion indépendants :

72 réseaux d'adresses (A0---> A15).

8 réseaux de contrôles (CSW, WE*).

16 réseaux d'alimentations (Vss,Vcc).

32 réseaux de données (D1--->D18) dont 18 utilisés dans la phase de configuration.

On appelle un réseau un ensemble de lignes d'interconnexion horizontales et verticales (en Alu1 et Alu2) interconnectées entre elles, elles ont donc le même potentiel. Chaque réseau d'adresse ou de contrôle est connecté à deux plots et celui d'alimentation est connecté à quatre plots, par contre un réseau de donnée est connecté à deux plots seulement en phase de test et à un plot après la phase de configuration finale. On verra plus loin les figures qui représentent ces réseaux. La tranche est divisée en deux demi tranches complètement indépendantes.

VI.2. Environnement de test

Une coupure ou un court-circuit sur un réseau peuvent être détectés mais pas localisés, d'où la nécessité de l'observation des défauts physiques. Par conséquence l'utilisation du Microscope Electronique à Balayage (MEB) est indispensable. D'autre part, l'accès direct aux réseaux nécessite l'équipement du MEB en carte à pointes tranche. Il est possible de tester soit la totalité de la tranche soit les deux demie tranches indépendamment, ceci dépendra des contraintes imposées par le MEB (par exemple le nombre de connecteurs maximum sur la carte à pointes). On développera dans la suite le test d'une demi tranche.

VI.3. Test des réseaux d'adresses

VI.3.1 Présentation des réseaux

Il s'agit de tester d'une part le temps de propagation et la présence d'un signal sur le réseau, et d'autre part l'influence d'un réseau sur un autre. Il s'agit donc d'un test paramétrique et fonctionnel à la fois. Le type d'algorithme détaillé dans le paragraphe "test des portes de transfert" répond bien à notre besoin.

On envoie les vecteurs de test sur deux types de plots nécessaires pour le test (voir figure 4.5) :

- les plots des entrées de réseaux; E1, E2, E3,..., E40.
- les plots des sorties (ou de test) des réseaux; S1,S2, S3,...,S40.

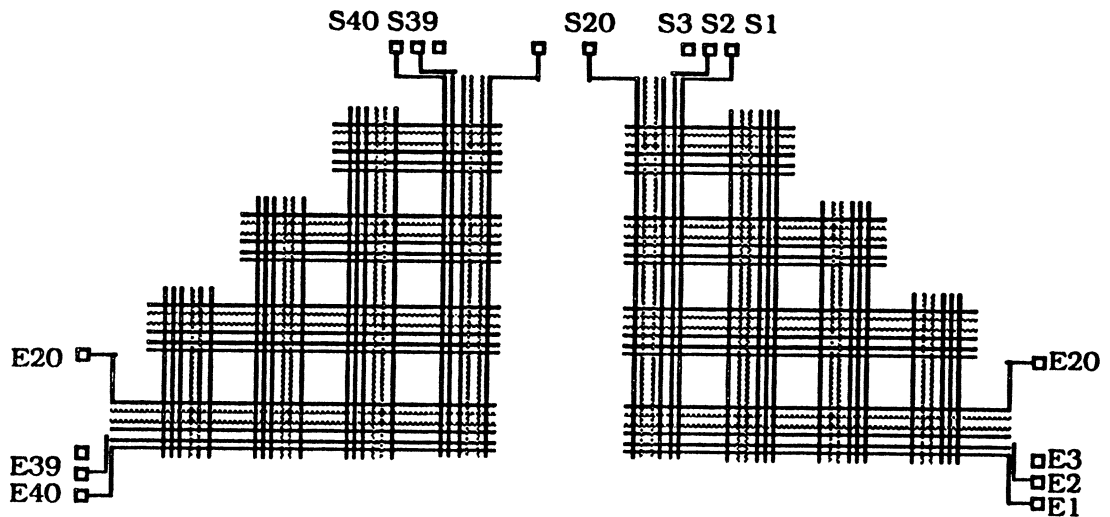


figure 4.5 : réseaux d'adresses

Une simulation SPICE a été faite pour mesurer le temps de propagation entre un plot E_i et S_i , ce temps est de l'ordre de 20 ns (pour imposer un niveau logique 1 = 4.5 volts).

On se contentera ici de résumer les séquences de test générées par l'algorithme.

VI.3.2 Algorithme de test

Après déchargement des plots E et S, on étudie l'influence des plots E_i avec $DEP \leq i \leq ARR$ des mémoires G ou D forcées à x (0 ou 1) sur les autres plots de sortie S_j avec $DEP \leq j \leq ARR$.

on note :

- DECH P_i ; décharge du plot i, ou initialiser le plot i à
- VAL ; valeur forcée
- DEP ; plot de départ
- ARR ; plot d'arrivée
- FORCE $P_i := x$; forçage du plot i à une valeur x

TEST $S_i = x$; tester que le niveau logique de plot $S_i = x$

DECH P_i , Initialisation de tous les P_i

Pour VAL := 0 jusqu'à 1 faire

Pour $i=DEP$ jusqu'à ARR faire FORCE $E_i := (i+VAL) \bmod 2$

Pour $j=DEP$ jusqu'à ARR faire TEST $S_j = (j+VAL) \bmod 2$

FinPour

*** Phases de test :**

On lance cet algorithme deux fois. La première pour tester les plots $E_1...E_{20}$, $S_1...S_{20}$ donc pour $DEP=1$ et $ARR=20$, et la deuxième pour tester les plots $E_{20}..E_{40}$, $S_{20}...S_{40}$ donc pour $DEP=20$ et $ARR=20$. les séquences de test sont lancées pas à pas. Le testeur doit être capable de comparer les valeurs des plots S_i et E_i , ainsi que le temps de propagation entre ces plots. En cas d'anomalie le testeur s'arrête au pas courant. Ensuite on lance donc une phase de diagnostic et de mémorisation.

VI.4. Test des réseaux de données et des alimentations

De même que pour le test du réseau d'adresses, on utilise le même algorithme de test et on suit les mêmes étapes de test, mais ici les plots et les temps de propagation ne sont pas les mêmes.

Les réseaux et les plots de données testés, ainsi que les plots et les réseaux d'alimentation sont illustrés sur les figure 4.6a et la figure 4.6b.

VI.5. Diagnostic et mémorisation

A l'arrêt du testeur au pas $i=k$, on recommence le test du plot E_i forcé à 1, et son influence sur les plots S_j , deux cas sont possibles :

- le temps de propagation entre E_i et S_i est grand, et le E_i forcé à 1 influence sur un (ou plusieurs) plot(s) $S_j \neq i$, donc il existe un (ou plusieurs) court-circuit(s) entre le réseau i et le(s) réseau(x) j .
- le temps de propagation entre E_i et S_i est grand, et il n'y a pas d'influence entre le plot E_i et les plots $S_j \neq i$, donc il existe sur le réseau i une coupure ou un court-circuit avec le substrat.

Pour départager ces deux hypothèses on utilisera le MEB par visualisation du contraste de potentiel. *A ce stade on doit exploiter les renseignements acquis et mémorisés (en phase de test des portes de transfert) concernant les lignes d'interconnexion ainsi que les portes de transfert.* Les types et coordonnées des défauts sont ensuite mémorisés pour la phase de réparation.

VI.6. Réparation des réseaux

En fonction du type et des coordonnées des défauts mémorisés précédemment, un programme de pilotage de laser est lancé afin d'effectuer les réparations nécessaires sur les réseaux. Les réparations typiques sur les réseaux sont illustrées dans les figures suivantes :

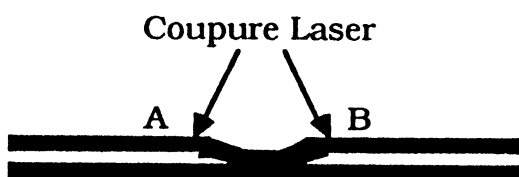


figure 4.7a : Court-circuit
entre deux lignes



figure 4.7b : Court-circuit entre
une ligne et le substrat

VII. TEST DES DECODEURS ET DE LA LOGIQUE DE COMMANDE

A ce stade on aura des réseaux d'interconnexions sains, l'alimentation de décodeurs et des logiques de commandes est donc possible par ces réseaux

VII.1. Test des décodeurs

Les entrées du décodeur intégré à proximité d'une mémoire 64k sont accessibles uniquement par les plots de la tranche via les réseaux d'interconnexion, donc par une carte à pointe à l'échelle de la tranche. Par contre la sortie du décodeur n'est pas accessible par cette carte. Par conséquent, le test de sortie du décodeur ne peut se faire que par observation du contraste de potentiel (MEB).

La programmation des décodeurs est limitée à trois fois (car la programmation des grilles flottantes est limitée à 3 fois). Le test des quatre configurations des décodeurs n'est donc pas possible. Par contre il est possible de tester une configuration en envoyant sur les entrées des décodeurs CSW*, AR1, AR2 les vecteurs de test 000, 001, 010, 011, 100, 101, 110, 111.

La réparation du décodeur n'est pas prévue dans sa conception, un défaut dans le décodeur condamne l'utilisation de la mémoire 64k correspondante.

VII.2. Test de la logique de commande

Par forçage de l'entrée de cette logique à 0 ou à 1, on peut facilement tester le fonctionnement de cette logique. Il existe des plots à l'entrée et à la sortie de chaque dispositif destiné à cet effet.

VIII. CARTOGRAPHIE DES RESSOURCES

La synthèse des étapes précédentes nous permet d'établir une cartographie des mémoires 64k bonnes et accessibles ainsi qu'une liste des coordonnées utilisables en phase de configuration de la mémoire, pour piloter l'E-beam ou le Laser.

IX. CONFIGURATION DE LA MEMOIRE

IX.1. Introduction

Le problème de la configuration de la mémoire consiste à former 18 groupes de 4 cellules chacun, soit $18 \times (256 \times 1 \text{bits}) = 4.5 \text{Mbits}$, parmi les cellules bonnes. Une cartographie de la mémoire permet de localiser les cellules défectueuses. Le problème revient donc à immerger la mémoire virtuelle (figure 4.8) au sein des ressources bonnes de la mémoire réelle. On note qu'un groupe de 4 mémoires 64k = un bloc de 256kx1bit. Et à chaque instant, seulement une mémoire par groupe est en fonctionnement.

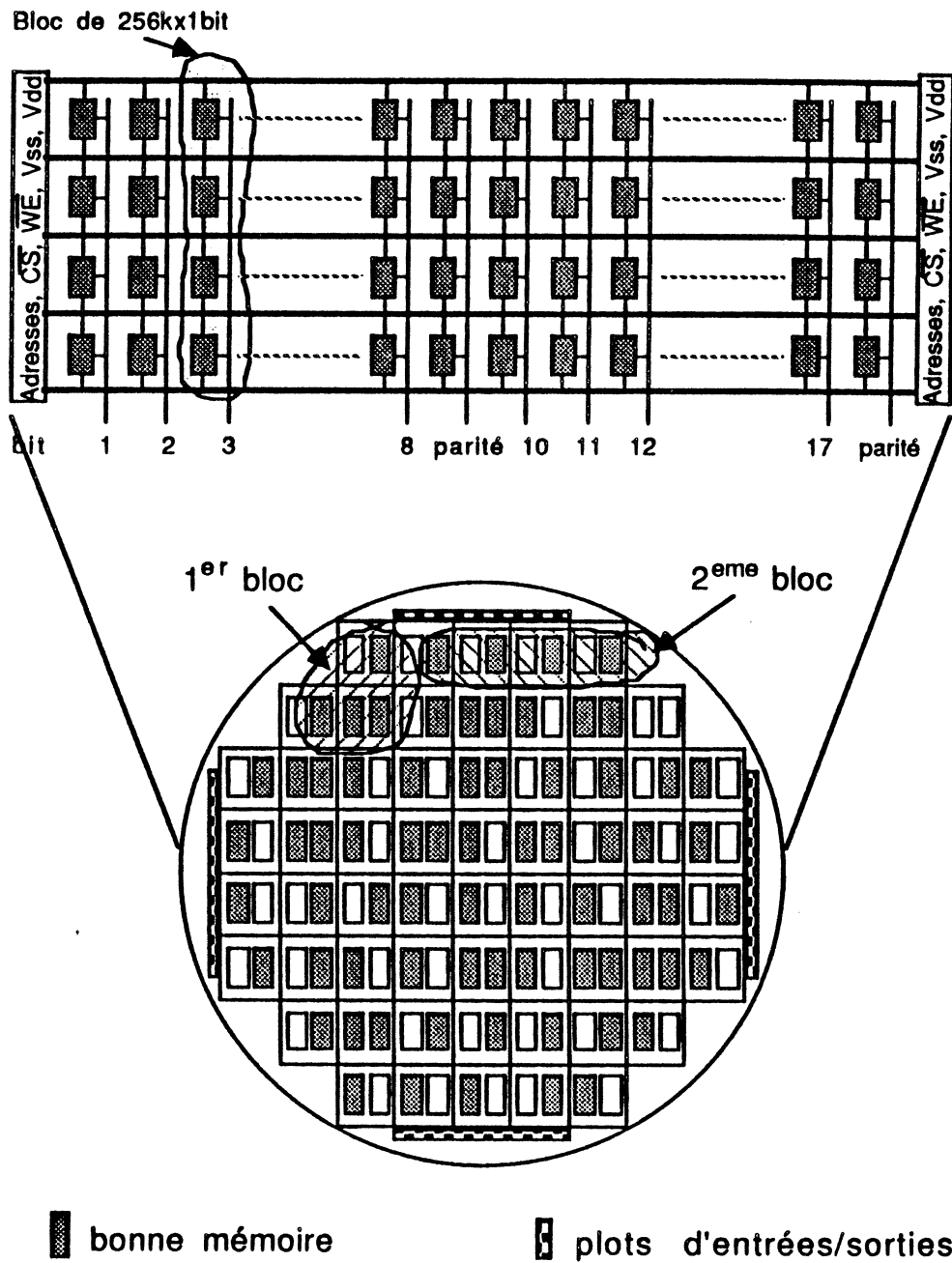


figure 4.8 : immersion de la mémoire virtuelle sur la tranche

Les différentes phases de configuration de la mémoire sont illustrées dans la figure suivante :

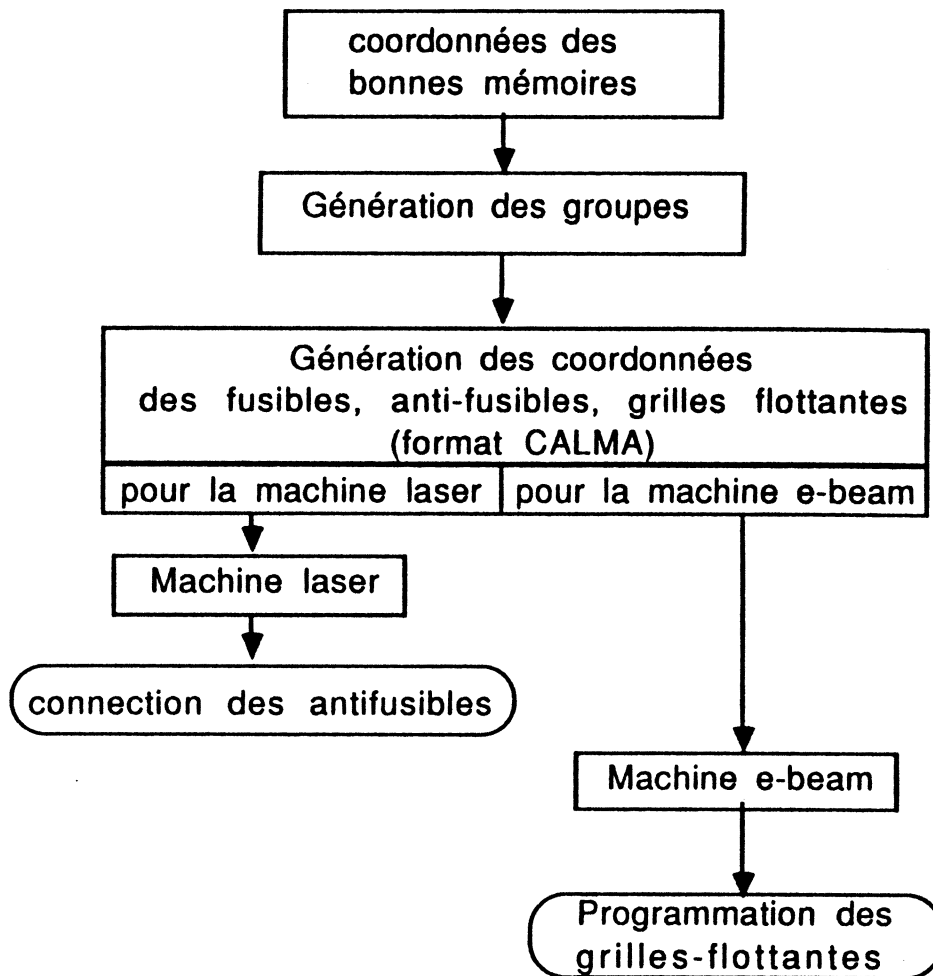


figure 4.9 : Organigramme de configuration

IX.2. Génération des groupes

La formation des groupes de cellules est soumise à certaines contraintes :

- Les cellules au sein d'un même groupe doivent être les plus proches possibles de façon à minimiser la longueur des connexions entre cellules et les plots de la tranche.
- Dans la formation des groupes les connexions verticales coûtent plus chères que les connexions horizontales. En effet, une connexion verticale nécessite la traversée de deux anti-fusibles ; ceux-ci peuvent être défectueux avant, pendant et après la reconfiguration de la mémoire.

- Chaque groupe doit être connecté à un plot d'entrée/sortie de données de la mémoire et cette connexion devrait être la plus courte possible.

Les avantages de cette stratégie de groupement sont les suivants :

- une bonne Alimentation des mémoires ; avec cette solution on obtient une meilleure alimentation des cellules en fonctionnement, car dans un quart de tranche on trouvera sur la même ligne horizontale d'alimentation, 4 à 5 cellules 64k bonnes, mais seulement 1 à 2 cellules qui fonctionnent en même temps.
- bonne évacuation de la chaleur ; la consommation des puces en fonctionnement (statique, dynamique,..) se transforme en chaleur (par effet joule) qui se propage dans le semi-conducteur et puis est évacuée par le semi-conducteur (substrat). L'évacuation de chaleur sera de plus en plus facile et rapide si les sources de chaleur (ici mémoire 64k en fonctionnement) sont distribuées uniformément sur une surface de semi-conducteur.

IX.2.1 Critères d'évaluation

Pour fournir des solutions réalisables optimales, nous devons définir un ensemble de critères d'évaluation de chaque solution proposée. Nous considérerons par la suite qu'une connexion horizontale entre deux cellules consécutives est de longueur unité tandis qu'une connexion verticale entre deux mémoires consécutives vaut $3 \cdot k$; k est un coefficient multiplicateur qui est fonction de la qualité de l'anti-fusible utilisé. Cela nous conduit aux fonctions de coûts suivantes qui détermineront les algorithmes de résolution utilisés.

a) Coûts intrinsèques des cellules

Du simple fait de sa localisation géométrique sur la tranche, un coût intrinsèque peut être associé à une mémoire.

*** Coût relatif à la ligne médiane de la tranche**

Pour minimiser la longueur des connexions liant les plots périphériques aux groupes de cellules associés, il est recommandé d'éviter de former des groupes chevauchant la ligne médiane verticale de la tranche. De ce fait un coût indiquant l'éloignement d'une mémoire de la ligne médiane est associé à chaque mémoire (à l'inverse ce coût indique la proximité de la mémoire du plot périphérique le plus proche).

*** Coût relatif aux plots périphériques**

Un repère arbitraire peut être défini sur la tranche de sorte qu'on peut associer une paire de coordonnées à chaque plot et à chaque mémoire sur la tranche. A la mémoire i de la tranche on associe les coordonnées (x_i, y_i) et au plot périphérique j on associe la paire de coordonnées (x_j, y_j) .

La contribution de la mémoire i au coût de connexion de son groupe d'appartenance au plot j peut alors être estimée par :

$$C_{ij} = |x_i - x_j| + 3 * k * |y_i - y_j|$$

Ainsi, nous pouvons associer à chaque mémoire i un vecteur de coût :

$$\zeta_i = (C_{i1}, C_{i2}, C_{i3}, \dots, C_{ij}, \dots, C_{in})$$

Ceci permet d'évaluer globalement la contribution de la mémoire i au coût d'un groupe quelconque la contenant lorsque ce groupe est affecté à un plot périphérique donné.

b) Coût d'un groupe

Un groupe est constitué de quatre cellules bonnes interconnectées. Chaque groupe doit être affecté à un plot périphérique.

Soit le groupe G_{ijkl} constitué des cellules C_i , C_j , C_k et C_l . Rappelons que si le groupe G_{ijkl} est affecté au plot périphérique p alors la contribution de la mémoire i au coût du groupe est C_{ip} . Notons V_{ijkl} le nombre total de liaisons verticales nécessaires à l'interconnexions des cellules du groupe G_{ijkl} . Alors la fonction de coût suivante peut être retenue pour évaluer un groupe affecté au plot p :

$$f(G_{ijkl})_p = \max_{s=i,j,k,l} C_{sp} + g.V_{ijkl}$$

g est un poids servant à pénaliser les groupes impliquant de nombreuses liaisons verticales. Le choix de cette fonction conduit à la formation de groupes proches de k périphéries et induisant peu de liaisons verticales.

c) Coût d'une configuration

Une configuration est constituée de 18 groupes affectés à 18 plots périphériques. Comme les débuts de propagation des signaux sont essentiellement pénalisés par les longues connexions, il est justifié d'illustrer ce fait par le choix de la fonction de coût suivante :

Soit une configuration consistant en les groupes $K = G^1, G^2, \dots, G^{18}$ alors le coût de la configuration, $F(K)$, est donné par :

$$F(K) = \max_{1 \leq i \leq 18} (f(G^i))$$

C'est le coût du groupe le plus cher au sein de la configuration. Nous devons rendre ce coût minimum, d'où le problème

$$\text{minimiser } (\max_{1 \leq i \leq 18} (f(G^i)))$$

IX.2.2 Formulation mathématique du problème de la configuration

Soit I le nombre total des cellules bonnes et soit J le nombre total de plots périphériques bons sur la mémoire.

Considérons la variable indicatrice X_{ij} qui vaut 1 si nous retenons dans la configuration finale l'attribution de la mémoire i au groupe de cellules affecté au plot j, et $X_{ij}=0$ sinon.

D'après les critères d'évaluation évoqués précédemment, nous pouvons formuler le problème de la configuration comme suit :

$$\text{Minimiser } (\text{MAX } (C_{ij} + g \cdot V_{i1i2i3i4}) \\ 1 \leq j \leq J ; i \in \{i_1, i_2, i_3, i_4\})$$

$$X_{i_1j} = X_{i_2j} = X_{i_3j} = X_{i_4j} = 1$$

$$0 \leq \sum_{i=1}^I X_{ij} \leq 4 \quad (\text{pas plus de 4 mémoires sont affectées à un plot pour former un groupe})$$

$$0 \leq \sum_{j=1}^J X_{ij} \leq 1 \quad (\text{une mémoire est au plus affectée à un groupe partition de groupes})$$

$$\sum_{i,j} X_{ij} \leq 72 \quad (\text{le résultat final retiendra 72 mémoires bonnes réparties en paquets de 4, soit 18 groupes au total})$$

Cette formulation nous conduit à un programme linéaire combinatoire en nombre entier. La mise en oeuvre des algorithmes classiques de recherche

opérationnelle tels que Le branch and bound ou la méthode des groupes de comery conduit à des temps de résolution exorbitants. Nous nous sommes orienté vers le développement d'algorithmes heuristiques fournissant des solutions acceptables en temps raisonnable(de l'ordre de quelques minutes). Trois algorithmes heuristiques sont proposés :

- La méthode du serpent
- La méthode de tri
- La méthode semi-énumérative

IX.2.3 Algorithmes heuristiques de reconfiguration

Nous exposons ici les trois algorithmes retenus pour résoudre le problème de la configuration.

a) La méthode du serpent [LEI86]

**** Méthode du serpent classique***

Cette méthode exploratoire est utilisée pour donner rapidement une indication approximative du coût de la faisabilité de la configuration pour un état de la mémoire donné.

La méthode consiste à parcourir la tranche mémoire par mémoire en suivant les rangées de cellules. L'exploitation se fait sans discontinuité, de sorte qu'après avoir parcouru une rangée suivante en sens inverse en partant de la mémoire qui se trouve juste sur la verticale de la dernière mémoire consultée sur la rangée précédente (voir figure 4.10).

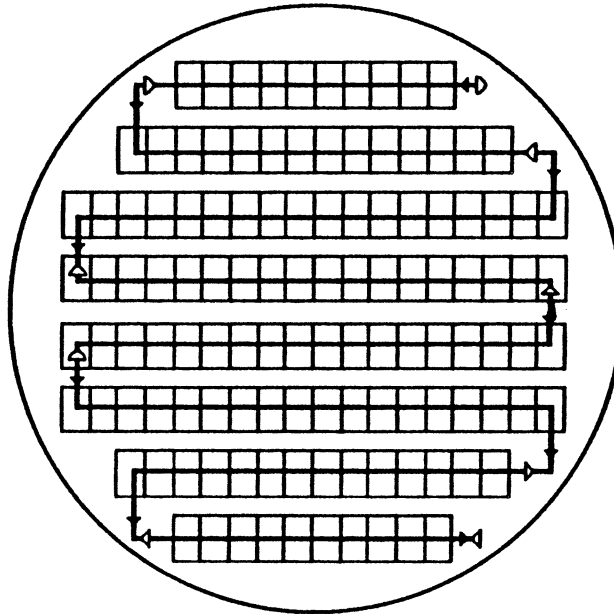


Figure 4.10 : Méthode du serpent

Les groupes sont formés en regroupant les cellules par paquets de quatre au fur et à mesure que les bonnes cellules sont rencontrées sur le parcours.

Cette méthode a l'avantage d'être extrêmement rapide et de fournir toujours une solution pour peu que le nombre de cellules bonnes soit au moins égale à 72.

Cependant, elle induit des connexions horizontales assez longues. La plus part des connexions verticales se font dans le changement de sens de parcours, c'est à dire sur la périphérie de la tranche. Le coût des groupes n'est pas contrôlé durant la formation de ceux-ci. L'évaluation n'est faite qu'après le traitement. La complexité de l'algorithme est linéaire $O(n)$.

Remarquons qu'il y a 4 façons de "serpenter" la tranche .

* *Méthode du serpent partiel*

Au lieu de serpenter sur les rangées entières, cette méthode parcourt des demi rangées (voir Figure 4.11).

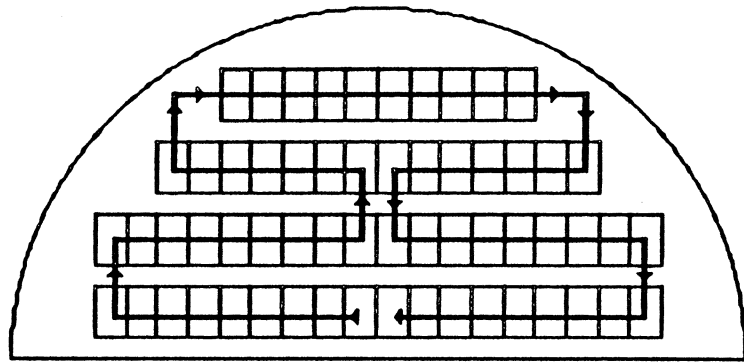


figure 4.11: Méthode du serpent partiel dans une demi tranche

L'avantage de cette méthode est d'éviter la formation de groupes qui se chevauchent sur l'axe, mais par contre elle introduit plus des connexions verticales.

b) La méthode à base de tri

Cette méthode part de l'idée suivante : les groupes les plus proches de la périphérie de la tranche entraînent un coût d'affectation aux plots faibles. Pour cela on associe à chaque mémoire i le vecteur $C_i = (C_{i1}, C_{i2}, \dots, C_{ip})$, indiquant sa contribution globale au coût d'un groupe quelconque le contenant.

A tout plot p on associe la liste des cellules bonnes sur la tranche. Cette liste est ordonnée suivant le coût C_{ip} croissant. Soit L_p cette liste, pour un plot p .

L'algorithme procède comme suit :

1^{ère} étape : On regroupe au sein de chaque liste L_p les quatre cellules donnant le coût le moins cher vis à vis de leur

affectation au plot p . Cela donne un groupe G_p . On évalue le coût de la création de ce groupe $f(G_p)$

2^{ème} étape : On cherche sur toutes les listes L_p celui qui donne le groupe de coût minimum $\min f(G_p)_{1 \leq p \leq J}$. Soit p_0 le plot entraînant la formation du groupe de coût minimum relativement à sa liste.

3^{ème} étape : Le groupe G_{p_0} est formé et affecté au plot p_0 . Toutes les cellules appartenant au groupe G_{p_0} sont retirées de toutes les listes L_p . Ces dernières sont remises à jour.

4^{ème} étape : On arrête si 18 groupes sont obtenues ou s'il reste moins de cellules bonnes qu'il n'en faut pour former un groupe. Sinon les itérations repartent en 1 avec les nouvelles listes L_p .

Cette méthode conduit à un algorithme de complexité $O(n \log_2 n)$. Le temps d'exécution est raisonnable (moins de 3 minutes). Les groupes sont formés en priorité au voisinage de la périphérie de la tranche. Il est possible de contrôler la création des groupes en agissant sur les fonctions de coût.

Les performances, en qualité, de la méthode baissent rapidement pour des rendements inférieurs à 65% (moins de 78 cellules bonnes).

Pour remédier à cet inconvénient, un prétraitement des cellules critiques est affecté avant de lancer l'algorithme de tri proprement dit. Les cellules critiques sont celles qui sont le plus proches de la ligne médiane de la tranche. Le prétraitement consiste à affecter ces cellules aux groupes entraînant un coût de la solution finale minimum. Ces cellules imposent le coût de la configuration finale. Une fois ces cellules agrégées, l'algorithme de tri forme les groupes résiduels sans remettre en cause les groupes où

ont été affectées les cellules critiques. L'expérience montre que le prétraitement permet à l'algorithme de maintenir de bonnes performances même pour des rendements faibles.

c) Algorithme semi-énumératif

Cet algorithme ne vise pas l'énumération de toutes les combinaisons de groupes possibles. Ce serait trop onéreux. L'idée est de former un grand nombre de groupes dont on sait à priori que le coût n'excédera pas un certain seuil fixé à l'avance. Pour cela, la mémoire est découpée en zone rectangulaire : des pavés. Un pavé est constitué d'un ensemble de cellules contingentes inscrites entre une rangée R1 et une rangée R2. Le pavé s'étend d'une colonne C1 à une colonne C2. Les rangées et les colonnes au sein d'un même pavé sont consécutives. Il s'en suit que tout groupe créé au sein d'un pavé donné n'induit pas des longueurs de connexion horizontales supérieures à $|C2-C1|$ et une connexion verticale au sein du pavé n'excédera pas $|R2-R1|$ (voir figure4.12).

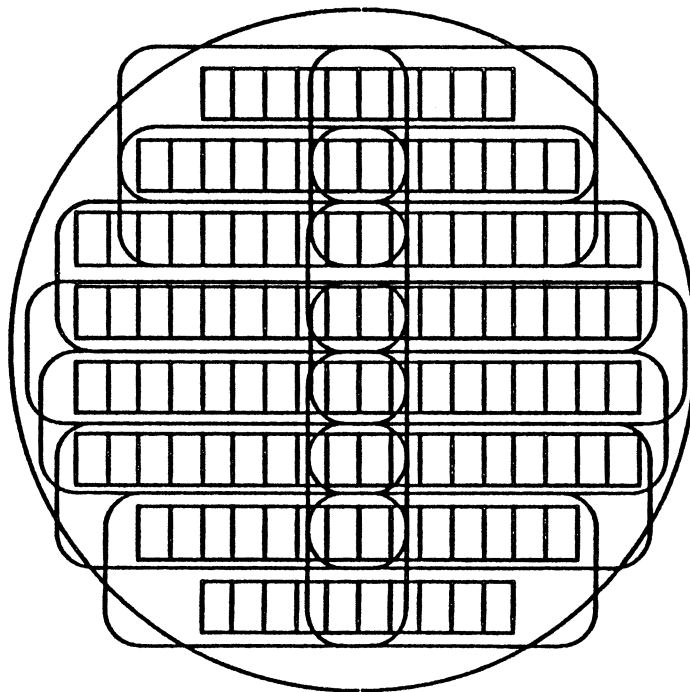


Figure 4.12 : Partitionnement de la tranche en 14 pavés

L'algorithme procède comme suit :

1^{ère} étape : la tranche est découpée en pavés. Les pavés peuvent se recouvrir sur deux rangées au plus et deux colonnes au plus.

2^{ème} étape : Toutes les combinaisons de cellules par paquets de quatre sont générées au sein de chaque pavé. Ensuite, on construit la liste des groupes en regroupant les différentes combinaisons (les combinaisons apparaissant dans plusieurs pavés y figurant une seule fois). Cette liste des groupes ainsi obtenue peut contenir des groupes non disjoints.

Ensuite pour chaque groupe de la liste, on calcul le coût minimum du groupe par rapport aux plots de la tranche. Une fois ce minimum trouvé, on associe le plot correspondant au groupe.

3^{ème} étape : A chaque groupe est associé un coût et un plot avec lequel il doit être connecté. Il s'agit maintenant d'extraire de l'ensemble des cellules obtenues à la 2^{ème} étape, 18 groupes disjoints dont le coût du groupe le plus cher est minimum parmi toutes les choix de 18 groupes disjoints possibles. A chaque agrégation de 18 groupes disjoints est associé le coût du groupe le plus cher dans l'agrégation.

Pour résoudre le problème posé à la 3^{ème} étape nous adoptons le formalisme suivant issu de la théorie des graphes.

On définit le graphe $\mathcal{G}_r = (G, A)$

où $G = \{g_1, g_2, \dots, g_n\}$ est l'ensemble des groupes formés à la 2^{ème} étape. Ce seront les sommets du graphe \mathcal{G}_r .

A est l'ensemble des arêtes du graphe. On met une arête entre le sommet i et le sommet j si et seulement si les groupes g_i et g_j ont au moins une mémoire commune ($g_i \cap g_j \neq \emptyset$) ou si g_i et g_j sont affectés au même plot (voir figure 4.13).

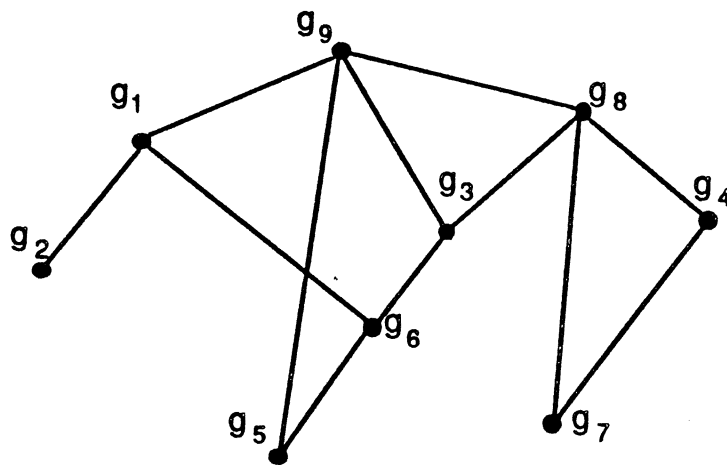
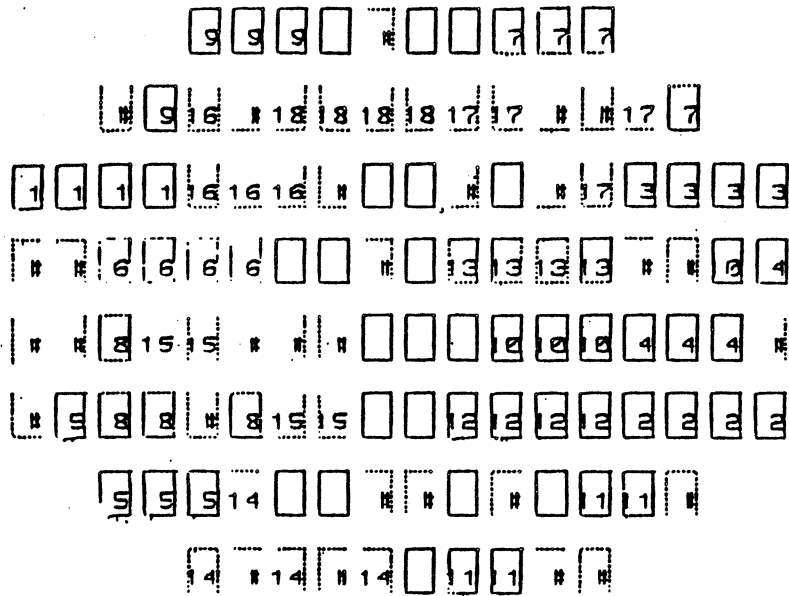


figure 4.13 : graphe $\mathcal{G}_r = (G, A)$ pour 9 groupes

A tout sommet g_i est affecté un poids qui est égale au coût du groupe i . Chaque groupe est affecté au plot lui attribuant son coût minimal.

En définitif le problème de la configuration de coût minimum revient donc à rechercher dans le graphe \mathcal{G}_r un stable de cardinalité 18 et de poids minimum. Un stable est un ensemble de sommets du graphe ne partageant aucune arête.

A l'inverse si l'on définit le graphe $\mathcal{G}_r^* = (G, A^*)$ où G est l'ensemble des sommets de \mathcal{G}_r , et maintenant l'arc $(g_i, g_j) \in A^* \Leftrightarrow (g_i, g_j) \notin A$ (c'est à dire qu'on relie deux groupes disjoints) alors le même problème revient à trouver un sous-graphe complet (une clique) de cardinalité 18 et de coût minimum. Le problème vu sous cet angle, est soluble d'autant plus qu'il existe des algorithmes classiques de recherche de composante connexe



ENCORE UN ESSAI (0/1) (rendement 76% algo TRI/ESPRIT LXC 824)

figure 4.14 b : exécution de l'algorithme de tri pour un rendement de 76%

Dans cet exemple en remarque la formation de groupes près du bord.

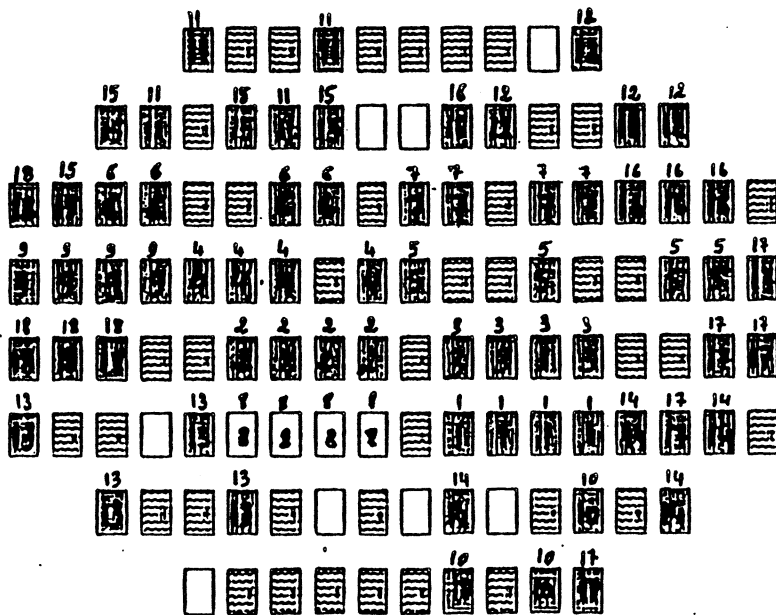
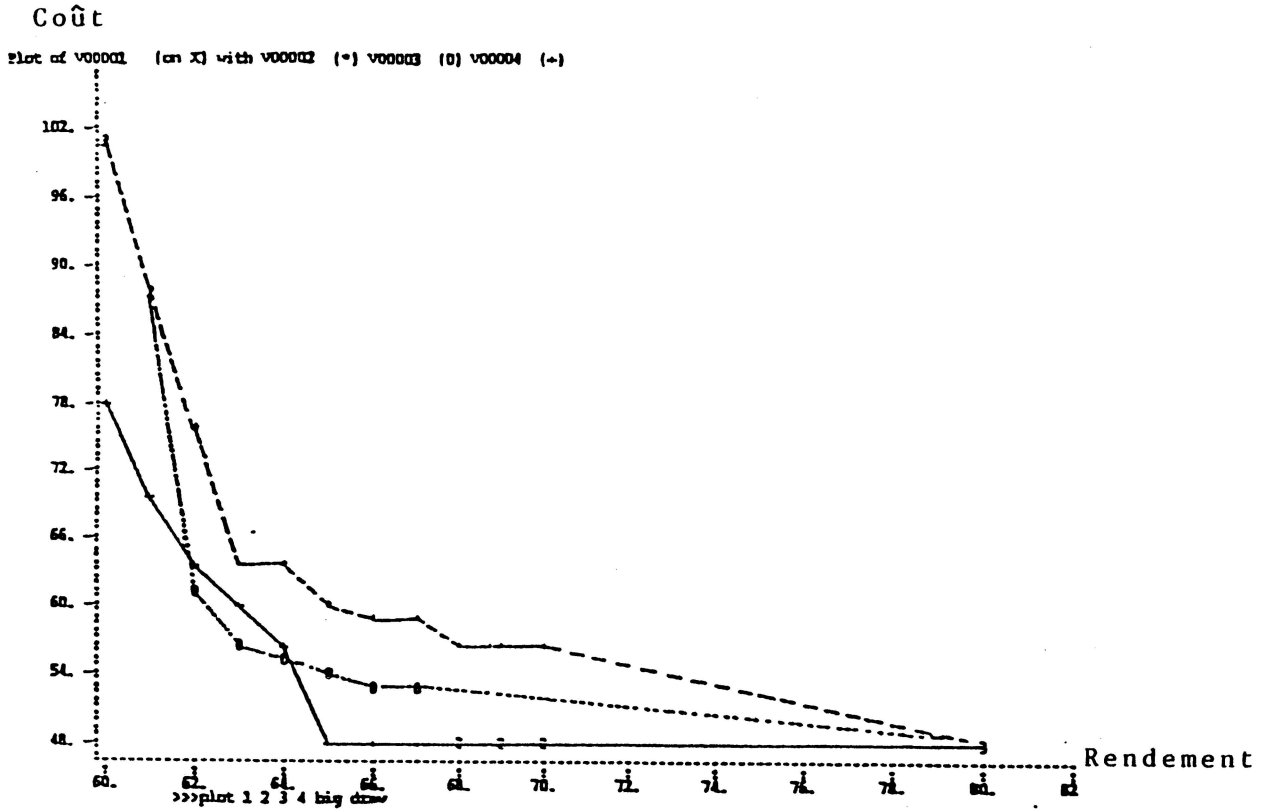


figure 4.14 c : algorithme semi-énumératif pour un rendement de 67%

De ces simulations, on a pu tirer les courbes d'évaluation de performance qui expriment le coût de configuration en fonction du rendement, ces courbes sont présentées en figure 4.15.



- - - Algorithme de tri » temps moyen de calcul \approx 50 sec.
- Algorithme de tri avec prétraitement » temps moyen de calcul \approx 40sec.
- ___ Algorithme semi-énumératif » temps moyen de calcul \approx 25min

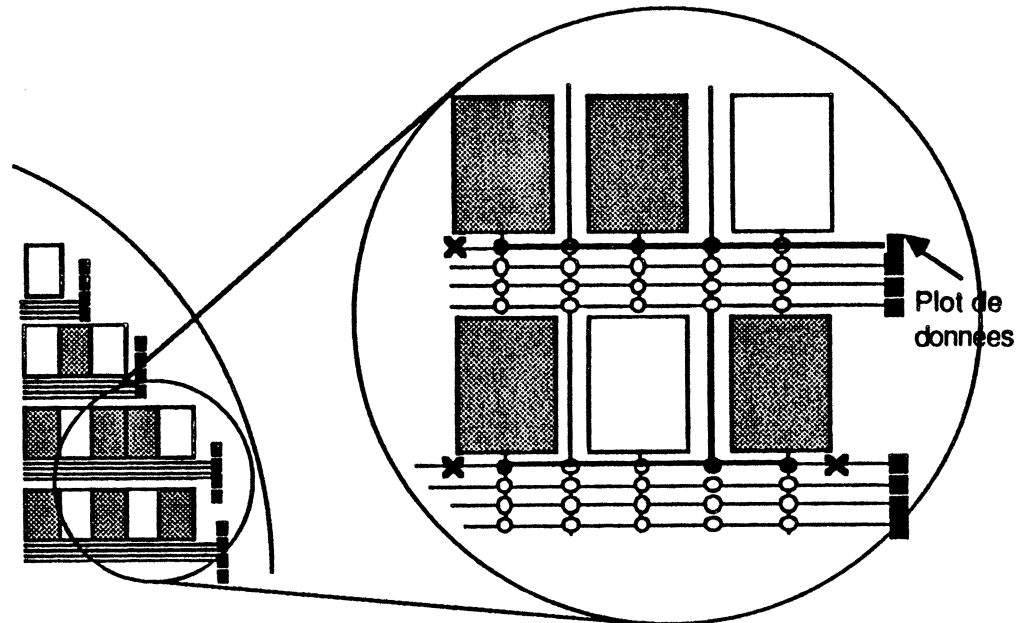
figure 4.15 : le coût de configuration en fonction du rendement

IX.3. Configuration physique de la mémoire-WSI

IX.3.1 Configuration du chemin de données

Le programme de formation de groupe va être exécuté sur un micro-ordinateur "HP 9000", qui à partir des coordonnées des mémoires de chaque groupe va piloter d'une part une machine laser et d'autre part un Microscope Electronique à Balayage (MEB).

Avec la machine laser on va pouvoir programmer les anti-fusibles de chaque groupe dans le but, d'une part de connecter les mémoires 64k de chaque groupe à la ligne de données qui leur est destinée, et d'autre part de réaliser la (ou les) connexion verticale(s) (voir figure 4.16)



- anti-fusible non programmé
- anti-fusible programmé
- ✕ coupure laser
- mémoire défailante
- bonne mémoire 64k

figure 4.16 : Groupe configuration by laser beam

IX.3.2 Configuration des décodeurs de chaque groupe

On a étudié dans le chapitre II la configuration de quatre décodeurs de quatre mémoire d'un groupe. Cette configuration consiste à programmer 1 ou 2 grilles-flottantes dans chaque décodeur pour qu'en fonction des entrées (AR1 et AR2) des décodeurs, une seule sortie soit à 0.

IX.3.3 Programmation des portes de transfert

A priori, toutes les cellules 64k sont isolées des réseaux d'interconnexion. Il s'agit donc de programmer les grilles-flottantes des circuits "logique de commande" des portes de transferts, afin de connecter les cellules utilisées aux réseaux d'interconnexion.

X. Test de la mémoire 4.5 Mbits à basse fréquence

Il s'agit de tester globalement la mémoire 4.5 Mbits après sa configuration par programmation des grilles flottantes. Les mémoires 64k utilisées pour la configuration sont supposées saines, elles ne sont donc pas l'objet d'une panne. Par l'utilisation de l'algorithme de damier pour le test de la mémoire 4.5 Mbits, on réduit les hypothèses de panne de la mémoire-WSI aux hypothèses de panne des réseaux et des dispositifs de commutation. Ces derniers ont été testés individuellement, par conséquent, les erreurs qui peuvent arriver sur la mémoire-WSI seront dues d'une part aux pannes de configuration, et d'autre part aux incidents inattendus de manipulation. Le test de la mémoire-WSI revient à tester les 4 rangées de 64kX18bits sélectionnées par le décodeur (1/4), donc pour chaque rangée on lance les phases de test suivantes:

Phase 1 : écriture du damier

L'écriture du damier dans une rangée revient à écrire alternativement les vecteurs de 18 bits 111...1 et 000...0 dans les 64kX18bits (voir figure4.17).

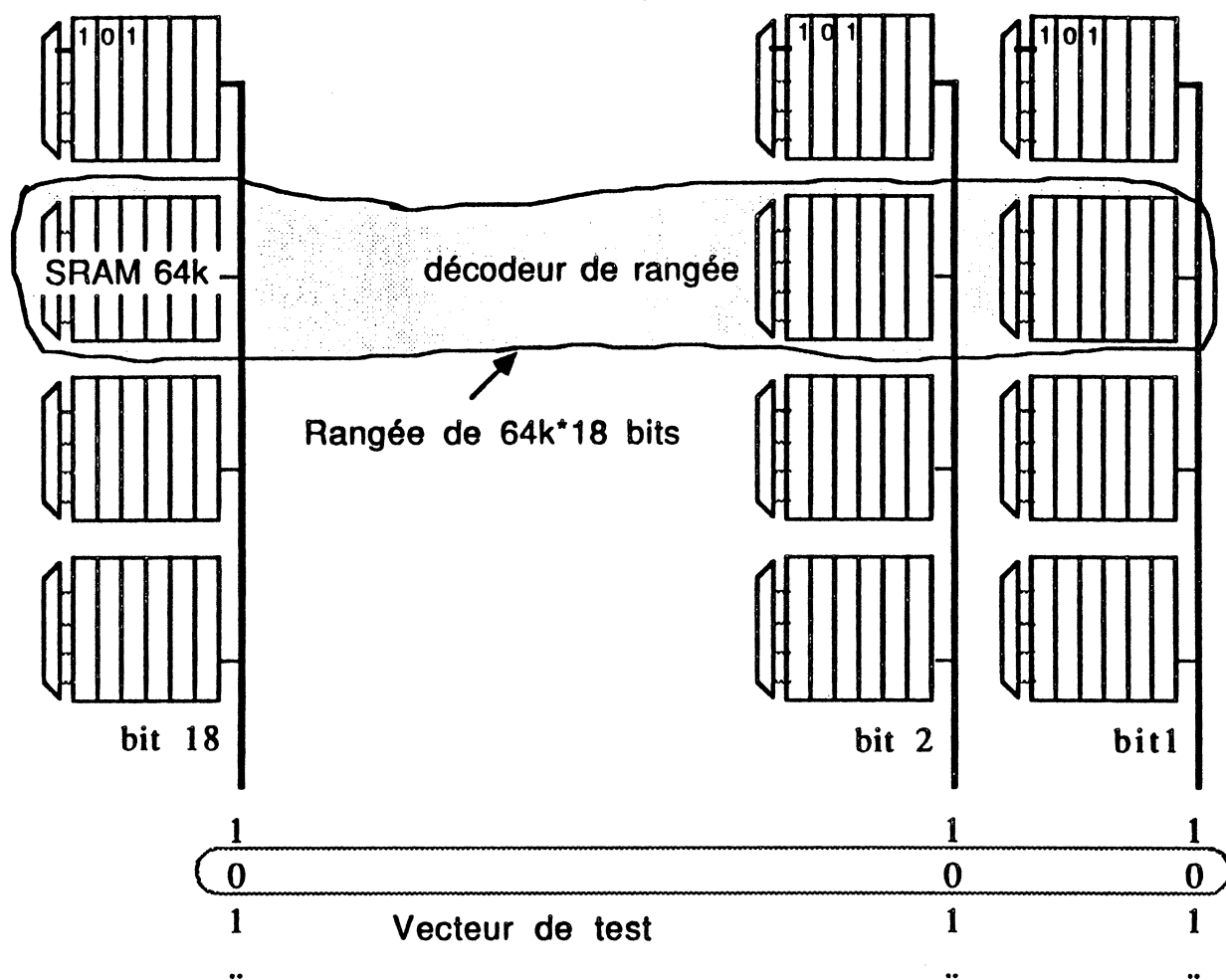


figure 4.17 : test de la mémoire-WSI

Comme pour l'écriture d'un damier dans une mémoire 64k, à la génération des adresses, on doit connaître la correspondance entre les adresses logiques et les adresses topologiques des cellules mémoires élémentaires des mémoires 64k.

Phase 2 : lecture du damier

En lecture la génération des adresses est la même que celle d'écriture, on doit donc lire alternativement (sur les 18 bits de la rangée testée) les vecteurs de test 111...1 et 000...0. en cas d'anomalie sur un bit de donnée, on vérifie si ces bits appartiennent à une liste de bits défailants, sinon on mémorise ce bit, Par exemple si on a :

adresse courante	donnée lue	donnée attendue	bit mémorisé
000...00	1 <u>1</u> 10...10	1 <u>0</u> 10...10	17

000...01	0101...01	0101...01	aucun
000...10	1110...01	1010...10	aucun

Phase 3 : On recommence la phase 1 en écrivant alternativement 000...0 et 111...1.

Phase 4 : On recommence la phase 2, mais en testant les vecteurs écrits en phase 3.

Phase 5 : Diagnostic, Localisation des pannes, Réparation.

Après les phases 1,2,3,4 on aura une liste des bits défectueux. Pour chaque bit on doit localiser la panne, pour cela on suit les étapes suivantes :

a) sans adresser la rangée, en forçant successivement à (00..)0 et (11..)1 le (ou les) chemin(s) de(s) bit(s) défectueux, et on le(s) visualise(s) par le MEB. Dans cette étape on teste : la programmation des anti-fusibles qui configurent le chemin de bit, ou si une ligne de bit est collée à 0 ou à 1.

b) en adressant la rangée avec l'adresse défectueuse, on visualise par le MEB l'adresse courante sur les entrées de la mémoire 64k qui correspond à la ligne de bit soupçonnée, dans cette étape on teste l'adressage de cette mémoire.

Si on remarque un défaut réparable dans l'étape a ou b, on lance une étape de réparation semblable à celle étudiée en phase de test des réseaux d'interconnexion.

Sinon la mémoire 64k correspondante est donc endommagée, alors suivant la cartographie on recommence la configuration ou pas. Par exemple si il existe une mémoire bonne (non utilisée dans la première configuration) à côté d'une mémoire endommagée, on continue la configuration définitive (programmation des anti-fusibles) en tenant compte des résultats du test

de la mémoire-WSI, sinon on rejette la tranche.

Toutes les phases d'écriture/ lecture seront réalisées à 6MHz, ce qui correspond à un temps d'accès de la mémoire-WSI de 160ns.

XI. Reconfiguration et test de la mémoire 4.5 Mbits

Grâce à l'utilisation des grilles-flottantes programmables, il est possible de revenir sur une configuration. En se servant de la cartographie établie précédemment qui cette fois est mise à jour, on recommence la phase de configuration de la mémoire-WSI, mais cette fois on utilise seulement les anti-fusibles pour effectuer dans chaque mémoire d'un groupe les tâches suivantes :

- court-circuitage des TMOS grilles-flottantes de la logique de commande et des décodeurs.
- court-circuitage des portes de transfert de données Din et Dout et des transistors PMOS de l'alimentation Vdd.

A ce stade la configuration devient donc figée. Un test semblable à celui du paragraphe XI est effectué. On aura deux résultats de test possibles :

- mémoire-WSI fonctionnelle ---> donc on continue les prochaines phases
- mémoire-WSI défailante ---> donc rejet de la tranche.

XII. Epaisseur en cuivre

Pour les raisons électriques détaillées dans le chapitre III, on prévoit l'épaississement des lignes d'alimentations et de données par le cuivre (jusqu'à 4 fois leurs épaisseurs).

XIII. Capacité de découplage

Une capacité de découplage sera greffée pour chaque quart de tranche prêt de son baricentre. Une mémoire 64k sera sacrifiée pour cet effet, car dans le plan de cette mémoire on réalise un trou pour pouvoir coller

(directement sur la tranche) la capacité de découplage de dimension inférieure à celle de 64k (voir figure 4.18).

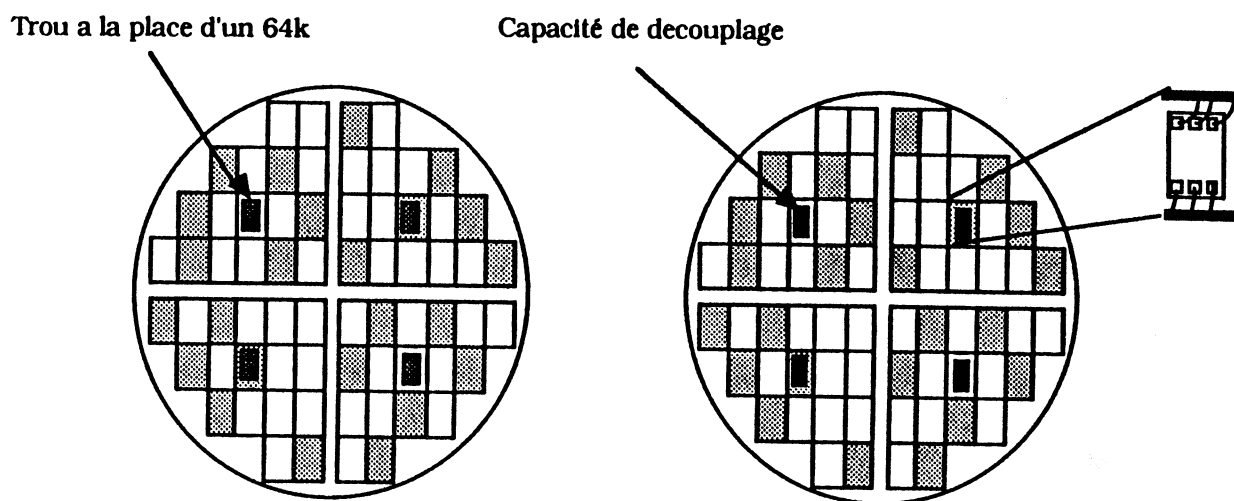


figure 4.18 : position des capacités de découplage sur la tranche

La liaison entre les plots de la capacité de découplage et les alimentations Vdd et Vss se fait directement par thermocompression sur les lignes Vdd et Vss, car celles-ci font environ 200μ de largeur.

XIV. Test final sur tranche de la mémoire-WSI

Il s'agit de tester les paramètres spécifiques de la tranche tels que la consommation, le courant de fuite....., ainsi que le bon fonctionnement de la mémoire, c'est à dire le bon adressage, la sélection des rangées, la propagation des données sur une ligne....

Après ce test, deux cas sont possibles :

- La mémoire ne fonctionne pas correctement, elle est donc rejetée car après cette étape aucune réparation n'est possible.
- La mémoire fonctionne correctement, on envisage alors sa mise en boîtier.

XV. Encapsulation (Packaging)

Cet aspect ainsi que les capacités de découplage sont étudiés en détail à CIMSA-SINTRA (THOMSON). Au niveau de l'emballage en WSI, il faut résoudre des problèmes majeurs tels que la dissipation de la chaleur, le bruit, la distribution de l'alimentations. En effet, dans le cas de la mémoire-WSI la puissance dissipée dans le circuit, qui doit être évacuée par la tranche de surface 78 cm^2 est environ de 20 à 25 watts. Le rapport puissance/surface de dissipation (ici tranche), est égale à 0.3 watt/cm^2 , ce qui est tout à fait acceptable, donc une encapsulation relativement peu sophistiquée peut être utilisée pour la mémoire-WSI. Le boîtier doit être en plus amagnétique pour éviter les problèmes de déviation du faisceau électronique utilisé pour la programmation des grilles flottantes.

XVI. Test final en boîtier de la mémoire-WSI

Il s'agit de tests paramétriques et fonctionnels dans des conditions normales de fonctionnement. Ce test va permettre de déterminer les caractéristiques électriques et fonctionnelles définitives de la mémoire-WSI, ainsi que son temps d'accès.

CONCLUSION



Nous avons montré, dans cette thèse, comment surmonter les deux principales difficultés liées à l'intégration sur tranche entière et ceci pour un démonstrateur type. Rappelons que ces problèmes sont d'une part le problème de la baisse de rendement due à l'augmentation de la taille des puces, d'autre part la résolution du problème de transfert de signal sur des distances de plusieurs centimètres avec des impédances de lignes plus importantes que dans le cas d'un circuit imprimé. Pour surmonter le premier problème, on a utilisé en fin de fabrication, des éléments en réserve pour remplacer les éléments défectueux. La déconnexion des éléments mauvais et la connexion des bons éléments étant faites par laser ou par faisceaux d'électrons. Pour la seconde, des réseaux d'interconnexions intégrés sur une épaisseur d'oxyde de 1.6μ sont utilisés; ils sont ensuite rendus plus épais (de 4 à 5 fois) grâce au cuivre.

La mémoire 4.5Mbits est actuellement en cours de fabrication en fonderie à SGS Thomson microelectronics. On prévoit un rendement de 60% et un temps d'accès de la mémoire de l'ordre de 100ns.

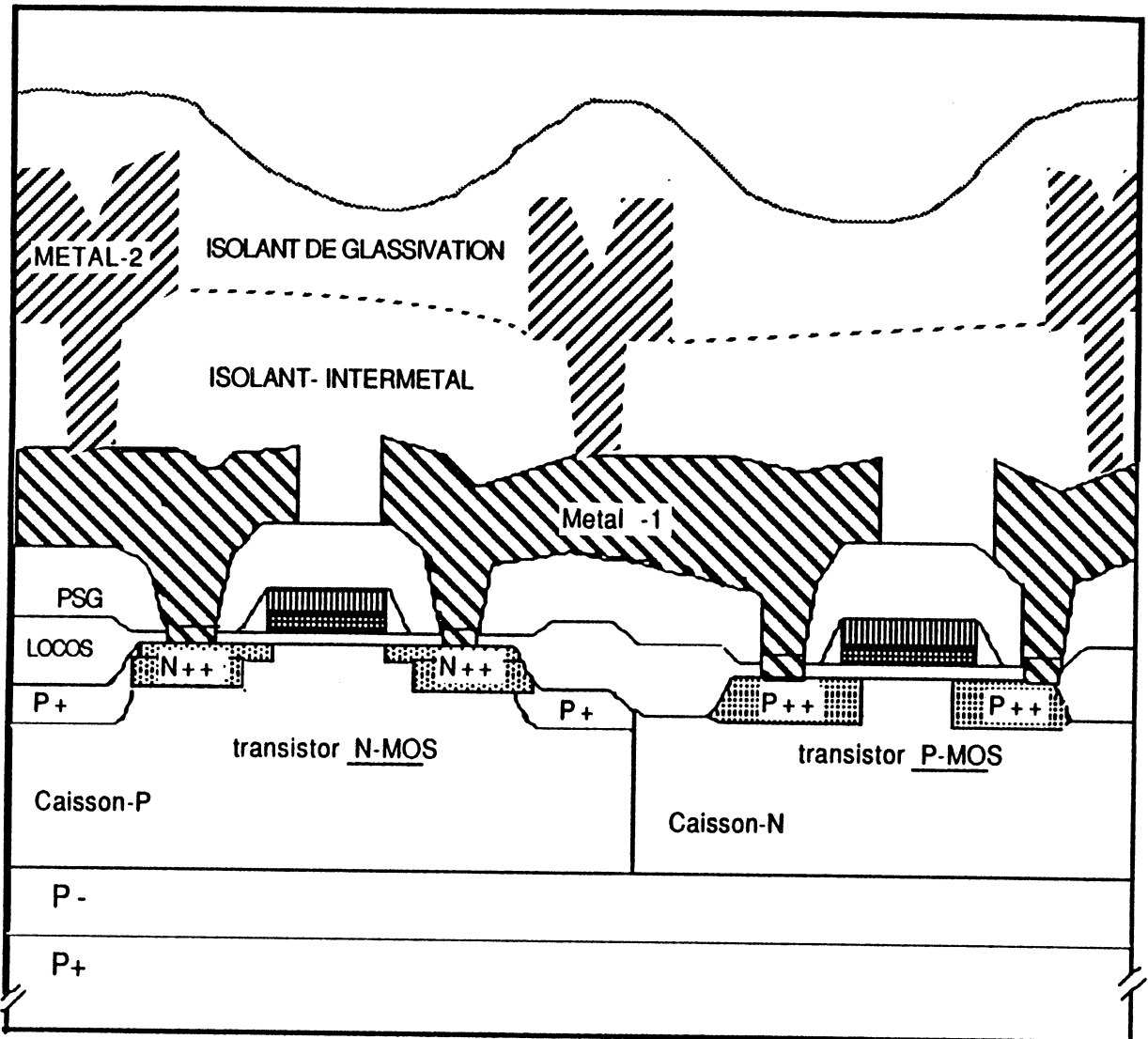
Un prolongement de cette étude consisterait à étudier la possibilité d'augmentation de la capacité de la mémoire en utilisant des puces de taille réduite et de grande capacité (des puces de 1-Mbit SRAM et de taille $7.6 \times 12.4 \text{mm}^2$ [SHI88] sont actuellement étudiée), avec la possibilité de récupérer 1/8, 1/4 ou 1/2 d'une puce jugée défectueuse. D'autre part, il est envisagée d'utiliser des interrupteurs programmables par faisceaux d'électrons plus fiables pour les réseaux d'interconnexion.



ANNEXES



[Annexe 2.1] : Différentes étapes de réalisation d'un circuit en "HCMOS3"



Après la réalisation :

- des Caissons N et P,
- des diffusions N et P
- des grilles (polysilicium)

on effectue les étapes suivante :

- | | | |
|--|---|---|
| <p>(1) Dépôt Metal-1(Al-Si-Cu)
Photo M1
Gravure M1
Elimination Resine
Dépôt Isolant Intermetal</p> | <p>(3) Dépôt Metal-2
Photo Metal-2
Gravure M-2 et Eli. Resine
Recuit</p> | <p>(5) TEST PARAMETRIQUE
TEST FONCTIONNEL</p> |
| <p>(2) Photo Vias
Gravure Vias
Elimination Resine</p> | <p>(4) Dépôt de Glassivation
Photo Plots
Gravure Plots
Elimination Resine
Contrôle visuelle</p> | |

[Annexe 2.2] : Chronogramme de lecture de la mémoire SRAM**CARACTERISTIQUES DYNAMIQUES EN LECTURE**

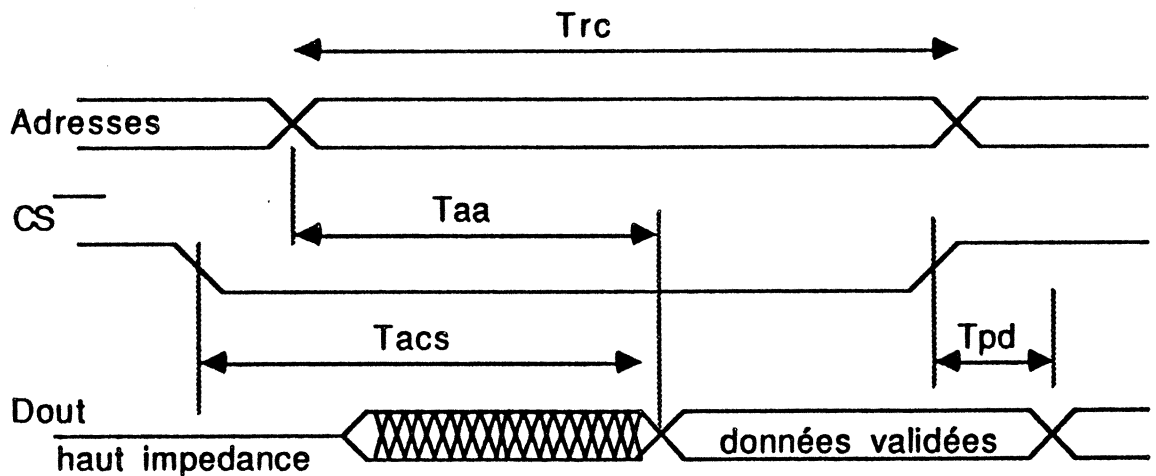
CARACTERISTIQUES	Symbole	64k-55		64k-70		Unites
		Min	Min	Min	Max	
Temps de cycle de lecture	Trc	55	-	-	70	ns
Temps d'accès par les adresses	Taa	-	55	-	70	ns
Temps d'accès par l'entrée CS	Tacs	-	55	-	70	ns
Temps de désélection	Tpd	-	30	-	30	ns

Remarque : 1) l'entrée \overline{WE} est a l'état haut durant tout le cycle de lecture

2) les valeurs min sont pour $T_a = -55\text{ °C}$ et $V_{cc} = 7\text{ volts}$

3) les valeurs typ sont pour $T_a = 25\text{ °C}$ et $V_{cc} = 5\text{ volts}$

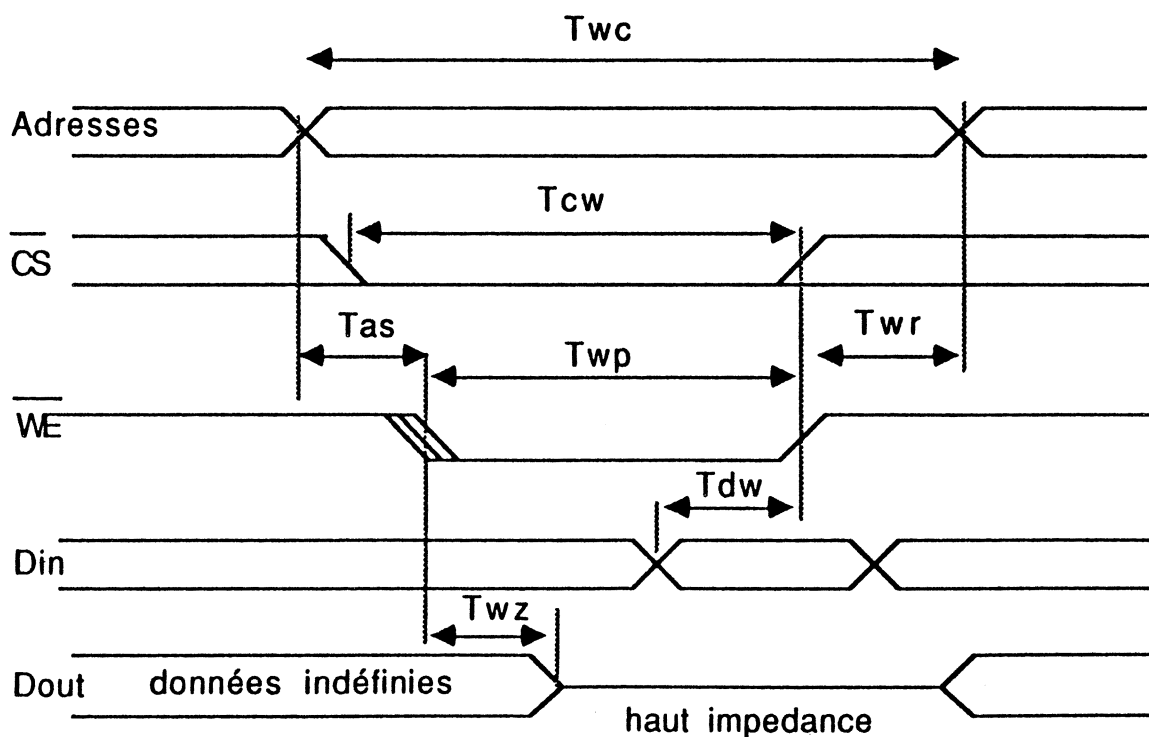
4) les valeurs max sont pour $T_a = 125\text{ °C}$ et $V_{cc} = 4.5\text{ volts}$



[Annexe 2.3] : Chronogramme d'écriture de la mémoire SRAM 64K**CARACTERISTIQUES DYNAMIQUES EN ECRITURE**

CARACTERISTIQUES	Symbole	64k-55		64k-70		Unites
		Min	Max	Min	Max	
Temps de cycle d'écriture	Twc	55	-	70	-	ns
Durée de l'impulsion de \overline{CS}	Tcw	50	-	60	-	ns
Durée de l'impulsion d'écriture	Twp	20	-	35	-	ns
Temps de maintien des adresses	Twr	0	-	0	-	ns
Temps de maintien des données	Tdw	15	-	20	-	ns
Sélection par rapport aux adresses	Tas	0	-	0	-	ns
Autorisation écriture par rapport aux données Dout en HZ	Twz	0	15	0	20	ns

- Remarque 1) les valeurs min sont pour $T_a = -55\text{ °C}$ et $V_{cc} = 7$ volts
 2) les valeurs max sont pour $T_a = 125\text{ °C}$ et $V_{cc} = 4.5$ volts



[Annexe 2.4] : Programme de calcul du rendement

```
PROGRAM rendement (input, output) ;
```

```
CONST
```

```
    PRSE = 0.8 ;
```

```
VAR
```

```
    surfout          : text ;
```

```
    D, DSNR, S, ST, SR, SNR, SE, SP, PSRD, PSPD, PSED : real ;
```

```
    RDWAF, PST, A, B, XCEL, YCEL, XE, YE : real ;
```

```
    I, J, REREND, REPROG, NP, K, R, MASQUE, NBCEL, POURC : integer ;
```

```
FUNCTION prob (S, D : real) : real ;
```

```
VAR
```

```
    pbe, pms          : real ;
```

```
BEGIN
```

```
IF S = 0 THEN prob := 1
```

```
ELSE
```

```
begin
```

```
    pbe := 1 / exp (MASQUE * ln (1 + S * D * 1E-2)) ;
```

```
    pms := sqrt ((1 - exp (-S*MASQUE*D*1E-2))/(S*D*MASQUE*1E-2)) ;
```

```
    pms := pms + exp (- sqrt (MASQUE * S * D * 1E-2)) ;
```

```
    pms := pms / 2 ;
```

```
    IF pbe <= pms
```

```
    THEN prob := pbe
```

```
    ELSE prob := pms ;
```

```
end;
```

```
END ;
```

```
FUNCTION CNP (N, P : integer) : real ;
```

```
VAR
```

```
    I : integer ;
```

```
    CAL : real ;
```

```
BEGIN
```

```
    CAL := 1 ;
```

```
    FOR I := 1 TO P DO CAL := CAL*(N-(I-1))/I ;
```

```
    CNP := CAL ;
```

```
END ;
```

```
PROCEDURE bassam (a, b : real ; VAR nbcel : integer) ;
```

```
CONST
```

```
    RAY = 45 ;
```

```
VAR
```

```
    rang          : ARRAY [1..30] OF integer ;
```

```
    d, x, y, su   : real ;
```

```
    i, s, c, l    : integer ;
```

```
BEGIN
```

```

d := 2 * RAY / a - trunc (2 * RAY / a) ;
x := RAY - d * a / 2 ;
y := b ;
l := 1 ;
c := 0 ;
BEGIN
  IF sqrt (y * y + x * x) <= RAY
  THEN rang [1] := trunc (x / a + 1 / 10)
  ELSE
  BEGIN
    x := x - a / 2 ;
    rang [1] := trunc (x / a + 1 / 10)
  END ;
END ;
(* open (surfout, 'SURFOUT', new) ; *)
(* rewrite (surfout) ; *)
IF sqrt (y * y + x * x) <= RAY THEN
BEGIN
  y := y + b ;
  s := rang [1] ;
  WHILE x > a / 2 + 1 / 10 DO
  BEGIN
    IF sqrt (y * y + x * x) <= RAY THEN
    BEGIN
      l := l + 1 ;
      rang [l] := trunc (x / a + 1 / 10) ;
      y := y + b ;
      s := s + rang [l]
    END ELSE
    BEGIN
      x := x - a ;
      c := c + 1
    END
  END ;
  c := 2 * c ;
  IF (x > a / 2 - 1 / 10) AND (x < a / 2 + 1 / 10) THEN
  BEGIN
    FOR i := 1 TO l DO rang [i] := 2 * rang [i] + 1 ;
    s := 4 * s + 2 * l ;
    c := c + 1 ;
  END ELSE
  BEGIN
    FOR i := 1 TO l DO rang [i] := 2 * rang [i] ;
    s := 4 * s
  END ;
  l := 2 * l ;
  nbcel := 2 * s ;
  su := a * b * s * 100 / (3.1416 * RAY * RAY) ;
END ELSE
BEGIN
  writeln ('COTE X = ', a : 3 : 2) ;
  writeln ('COTE Y = ', b : 3 : 2) ;
  writeln ('RAYON R = ', RAY) ;
  writeln ;
  writeln (surfout, 'COTE X = ', a : 3 : 2) ;
  writeln (surfout, 'COTE Y = ', b : 3 : 2) ;
  writeln (surfout, 'RAYON R = ', RAY) ;
  writeln (surfout, ' ');
  writeln ('PUCE TROP GRANDE') ;

```

```

        writeln (surfout, 'PUCE TROP GRANDE')
    END ;
END ; (* ----- PROGRAMME PRINCIPAL ----- *)

BEGIN
open (surfout, 'SURFOUT', new) ;
rewrite (surfout) ;
REPEAT
writeln (' ');
writeln (surfout, ' ');
writeln ('CALCUL DU RENDEMENT : ');
writeln (surfout, 'CALCUL DU RENDEMENT : ');
writeln (' ');
writeln (surfout, ' ');
writeln ('Donnez le default (default/cm2) ');
readln (D) ;
writeln ('Donnez le nombre de masque critique ');
readln (MASQUE) ;
writeln ('cote x de 64k/mm = ? ');
readln (XCEL) ;
writeln ('cote y de 64k/mm = ? ');
readln (YCEL) ;
writeln ('pourcentage de surface reparable d,une memoire 64k/mm = ? ');
readln (POURC) ;
writeln ('plans identiques de surface reparable (NP) = ? ');
readln (NP) ;
writeln ('elements identiques / plans (K) = ? ');
readln (K) ;
S:= XCEL*YCEL;
SR:=POURC*S/100;
XE:=XCEL/(K*NP);
YE:=SR/XCEL;
SE:=XE*YE;
REPEAT
writeln ('nombre de elements de reserves par plan (R) = ? ');
readln (R) ;
SP :=(K+2*R)*SE;
SR :=SP*NP;
SNR:= (K+2*R)*NP*XE*(YCEL-YE);
S:= (K+2*R)*NP*XE*YCEL;
DSNR:= S/(SR*(5/3)+SNR);
bassam (2*((K+2*R)*NP*XE+0.64), YCEL+1.79, NBCEL) ;
writeln ;
IF R >= 1 THEN
BEGIN
PSED:=(1-PROB(SE,D))*PRSE ;
PSPD:=0 ;
PSRD:=0 ;
FOR J:=1 TO R DO
begin
PSPD:=PSPD+CNP (K,J)*(EXP(J*LN(PSED))*EXP((K+R-J)*LN(1-PSED)));
end;
FOR I:=1 TO NP DO
begin
PSRD:=PSRD+CNP(NP,I)*(EXP(I*LN(PSPD))*EXP((NP-I)*LN(1-PSPD)));
end;
PST:=PROB(SNR,DSNR)*(PROB(SR,D)+PSRD);
END ELSE

```

```

PST := PROB(SNR, DSNR)*PROB (SR, D) ;
writeln (surfout,'NBCEL =',NBCEL:4);
NBCEL:=TRUNC(NBCEL*0.9);
FOR J:=65 TO NBCEL DO
BEGIN
RDWAF:=0;
FOR I:=0 TO NBCEL-J DO
BEGIN
RDWAF:=RDWAF+CNP(NBCEL,J+I)*(EXP((J+I)*LN(PST))*EXP((NBCEL-J-I)*LN(1-PST)
));
END;
writeln (surfout,'r =',R:4,' RDMEM =',PST*100:4:2,' PROB =',RDWAF:4:2,' MEMBON =
',TRUNC(J+J*0.1):4);
writeln ('R =',R:4,' RDMEM =',PST*100:4:2,' RDWAF =',RDWAF:4:2,'
MEMBON=',TRUNC(J+J*0.1):8);
writeln ;
END;
writeln ('Voulez vous recommencer ? (1 si OUI, 0 si NON) ');
readln (REREND) ;
UNTIL (REREND = 0) ;
writeln ('voulez vous recommencer le programme ? (1 si OUI, 0 si NON)') ;
readln (REPROG) ;
UNTIL (REPROG = 0) ;
END.

```


[Annexe 2.5] : Table des valeurs des figures 2.5, 2.6a et 2.6b**CALCUL DU RENDEMENT :**

NBCEL = 128

r = 0 RDMEM =23.45 PROB =0.00 MEMBON = 71

.....

NBCEL = 120

r = 1 RDMEM =75.77 PROB =1.00 MEMBON = 71

r = 1 RDMEM =75.77 PROB =1.00 MEMBON = 72

r = 1 RDMEM =75.77 PROB =1.00 MEMBON = 73

r = 1 RDMEM =75.77 PROB =1.00 MEMBON = 74

r = 1 RDMEM =75.77 PROB =1.00 MEMBON = 75

r = 1 RDMEM =75.77 PROB =1.00 MEMBON = 77

r = 1 RDMEM =75.77 PROB =0.99 MEMBON = 78

r = 1 RDMEM =75.77 PROB =0.99 MEMBON = 79

r = 1 RDMEM =75.77 PROB =0.98 MEMBON = 80

r = 1 RDMEM =75.77 PROB =0.97 MEMBON = 81

r = 1 RDMEM =75.77 PROB =0.95 MEMBON = 82

r = 1 RDMEM =75.77 PROB =0.92 MEMBON = 83

r = 1 RDMEM =75.77 PROB =0.88 MEMBON = 84

r = 1 RDMEM =75.77 PROB =0.83 MEMBON = 85

r = 1 RDMEM =75.77 PROB =0.78 MEMBON = 86

r = 1 RDMEM =75.77 PROB =0.70 MEMBON = 88

r = 1 RDMEM =75.77 PROB =0.62 MEMBON = 89

r = 1 RDMEM =75.77 PROB =0.54 MEMBON = 90

r = 1 RDMEM =75.77 PROB =0.45 MEMBON = 91

r = 1 RDMEM =75.77 PROB =0.36 MEMBON = 92

r = 1 RDMEM =75.77 PROB =0.28 MEMBON = 93

r = 1 RDMEM =75.77 PROB =0.21 MEMBON = 94

r = 1 RDMEM =75.77 PROB =0.15 MEMBON = 95

r = 1 RDMEM =75.77 PROB =0.10 MEMBON = 96

r = 1 RDMEM =75.77 PROB =0.06 MEMBON = 97

.....

NBCEL = 120

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 71

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 72

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 73

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 74

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 75

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 77

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 78

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 79

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 80

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 81

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 82

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 83

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 84

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 85

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 86

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 88

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 89

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 90

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 91

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 92

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 93

r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 94
 r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 95
 r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 96
 r = 2 RDMEM =90.13 PROB =1.00 MEMBON = 97
 r = 2 RDMEM =90.13 PROB =0.99 MEMBON = 99
 r = 2 RDMEM =90.13 PROB =0.98 MEMBON = 100
 r = 2 RDMEM =90.13 PROB =0.96 MEMBON = 101
 r = 2 RDMEM =90.13 PROB =0.93 MEMBON = 102
 r = 2 RDMEM =90.13 PROB =0.89 MEMBON = 103
 r = 2 RDMEM =90.13 PROB =0.82 MEMBON = 104
 r = 2 RDMEM =90.13 PROB =0.73 MEMBON = 105
 r = 2 RDMEM =90.13 PROB =0.62 MEMBON = 106
 r = 2 RDMEM =90.13 PROB =0.50 MEMBON = 107
 r = 2 RDMEM =90.13 PROB =0.37 MEMBON = 108
 r = 2 RDMEM =90.13 PROB =0.25 MEMBON = 110
 r = 2 RDMEM =90.13 PROB =0.15 MEMBON = 111
 r = 2 RDMEM =90.13 PROB =0.08 MEMBON = 112
 r = 2 RDMEM =90.13 PROB =0.04 MEMBON = 113

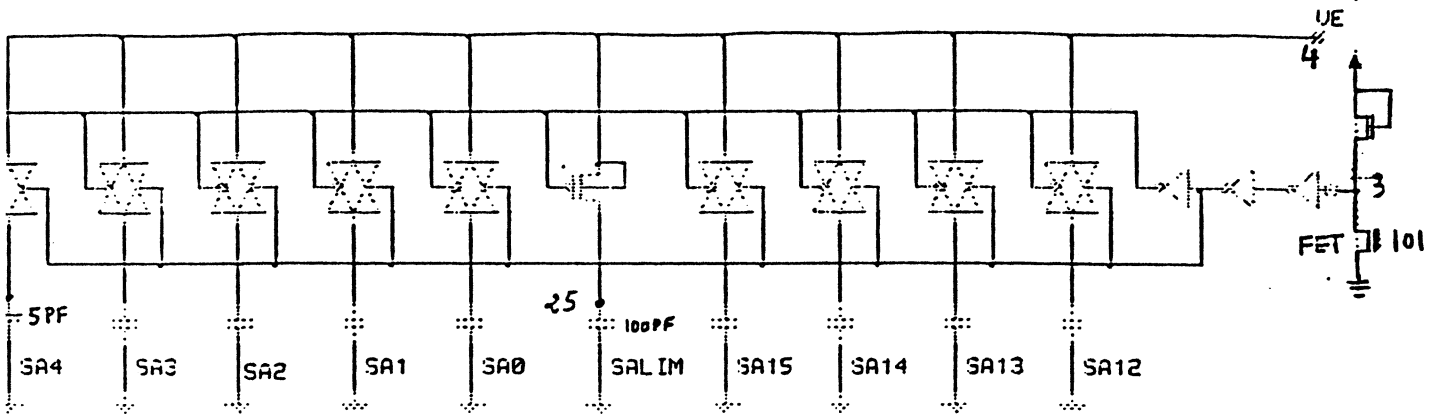
.....

NBCEL = 112

r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 71
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 72
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 73
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 74
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 75
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 77
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 78
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 79
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 80
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 81
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 82
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 83
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 84
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 85
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 86
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 88
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 89
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 90
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 91
 r = 3 RDMEM =91.45 PROB =1.00 MEMBON = 92
 r = 3 RDMEM =91.45 PROB =0.99 MEMBON = 93
 r = 3 RDMEM =91.45 PROB =0.98 MEMBON = 94
 r = 3 RDMEM =91.45 PROB =0.95 MEMBON = 95
 r = 3 RDMEM =91.45 PROB =0.92 MEMBON = 96
 r = 3 RDMEM =91.45 PROB =0.85 MEMBON = 97
 r = 3 RDMEM =91.45 PROB =0.77 MEMBON = 99
 r = 3 RDMEM =91.45 PROB =0.65 MEMBON = 100
 r = 3 RDMEM =91.45 PROB =0.51 MEMBON = 101
 r = 3 RDMEM =91.45 PROB =0.37 MEMBON = 102
 r = 3 RDMEM =91.45 PROB =0.24 MEMBON = 103
 r = 3 RDMEM =91.45 PROB =0.13 MEMBON = 104
 r = 3 RDMEM =91.45 PROB =0.06 MEMBON = 105



ANNEXE 3.1 : Dispositifs situés en haut d'une mémoire 64k



COMUTH

.MODEL N *BIB:SPI\$BIB:HCMOS3C.ENRN3.TYP
 .MODEL P *BIB:SPI\$BIB:HCMOS3C.ENRP3.TYP

.....
 * CAPACITORS

C3 3 0 .020PF
 * total=.020 PF gate=.020

C15 15 0 2.739P
 * total=PF gate=2.739

C16 16 0 13.204P
 * total=PF gate=13.204

C17 17 0 5.000P

C18 18 0 5.000P

C19 19 0 .042P
 * total=.042 PF gate=.042

C20 20 0 5.000P

C21 21 0 5.000P

C22 22 0 5.000P

C23 23 0 5.000P

C24 24 0 5.000P

C25 25 0 100.000P

C26 26 0 5.000P

C27 27 0 5.000P

* TRANSISTORS

```

*
M1  17  16  4  1  P  L=1.30U  W=100.00U
M2  17  15  4  0  N  L=1.30U  W=100.00U
M3  0   19  15  0  N  L=1.30U  W=5.00U
M4  15  19  1  1  P  L=1.30U  W=9.00U
M5  20  16  4  1  P  L=1.30U  W=100.00U
M6  20  15  4  0  N  L=1.30U  W=100.00U
M7  21  16  4  1  P  L=1.30U  W=100.00U
M8  21  15  4  0  N  L=1.30U  W=100.00U
M9  22  16  4  1  P  L=1.30U  W=100.00U
M10 22  15  4  0  N  L=1.30U  W=100.00U
M11 23  16  4  1  P  L=1.30U  W=100.00U
M12 23  15  4  0  N  L=1.30U  W=100.00U
M13 24  16  4  1  P  L=1.30U  W=100.00U
M14 24  15  4  0  N  L=1.30U  W=100.00U
M15 0   15  16  0  N  L=1.30U  W=15.00U
M16 16  15  1  1  P  L=1.30U  W=30.00U
M17 25  16  4  4  P  L=2.00U  W=2400.00U
M18 26  16  4  1  P  L=1.30U  W=100.00U
M19 26  15  4  0  N  L=1.30U  W=100.00U
M20 0   3   19  0  N  L=1.30U  W=3.40U
M21 19  3   1  1  P  L=1.30U  W=3.40U
M22 18  16  4  1  P  L=1.30U  W=100.00U
M23 18  15  4  0  N  L=1.30U  W=100.00U
M24 27  16  4  1  P  L=1.30U  W=100.00U
M25 27  15  4  0  N  L=1.30U  W=100.00U
M26 1   1   3  0  N  L=5.00U  W=2.00U
M27 3   101 0  0  NDEP L=10.00U W=20.0U

```

```

* .....
* VOLTAGE SOURCES

```

```

VDD 1 0 5
VCOM 101 0-5
VE 4 0 PULSE (0 5 5NS 2NS 2NS 30NS 40NS)

```

```

.OPTION NOLIST NOMOD NOOP LIMPTS=1000
.TRAN 1NS 40NS
.PLOT TRAN V(3) V(4) V(17) V(25) (0,5)
.PLOT TRAN I(VDD)

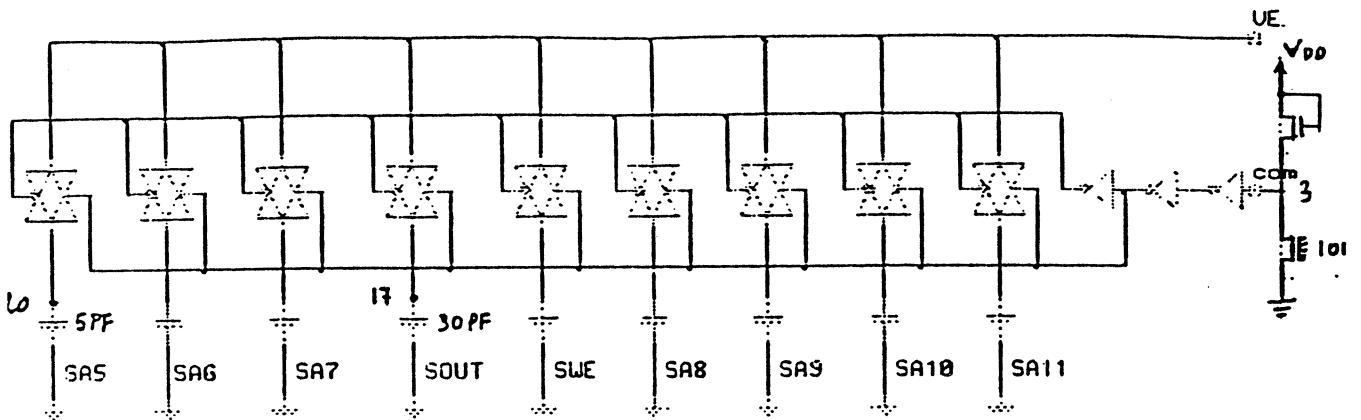
```

```

.END

```

ANNEXE 3.2 : Dispositifs situés en haut d'une mémoire 64k



COMUTB

.MODEL N *BIB:SPI\$BIB:HCMOS3C.ENRN3.TYP
 .MODEL P *BIB:SPI\$BIB:HCMOS3C.ENRP3.TYP
 .MODEL NDEP *BIB:SPI\$BIB:HCMOS3C.ENRP3.TYP

.....
 * CAPACITORS

C3 3 0 .020PF

* total=.020 PF gate=.020

C15 15 0 4.186PF

* total=4.186 PF gate=4.186

C16 16 0 4.262PF

* total=PF gate=4.262

C25 25 0 .042PF

* total=.042PF gate=.042

C17 17 0 30.00P

C14 14 0 5.000P

C18 18 0 5.000P

C19 19 0 5.000P

C20 20 0 5.000P

C21 21 0 5.000P

C22 22 0 5.000P

C23 23 0 5.000P

C24 24 0 5.000P

.....
 * TRANSISTORS

```

*
M1  17  16  4  1  P  L=1.30U  W=600.00U
M2  17  15  4  0  N  L=1.30U  W=600.00U
M3  20  16  4  1  P  L=1.30U  W=100.00U
M4  20  15  4  0  N  L=1.30U  W=100.00U
M5  21  16  4  1  P  L=1.30U  W=100.00U
M6  21  15  4  0  N  L=1.30U  W=100.00U
M7  22  16  4  1  P  L=1.30U  W=100.00U
M8  22  15  4  0  N  L=1.30U  W=100.00U
M9  23  16  4  1  P  L=1.30U  W=100.00U
M10 23  15  4  0  N  L=1.30U  W=100.00U
M11 18  16  4  1  P  L=1.30U  W=100.00U
M12 18  15  4  0  N  L=1.30U  W=100.00U
M13 14  16  4  1  P  L=1.30U  W=100.00U
M14 14  15  4  0  N  L=1.30U  W=100.00U
M15 19  16  4  1  P  L=1.30U  W=100.00U
M16 19  15  4  0  N  L=1.30U  W=100.00U
M17 24  16  4  1  P  L=1.30U  W=100.00U
M18 24  15  4  0  N  L=1.30U  W=100.00U
M19 0  15  16  0  N  L=1.30U  W=15.00U
M20 16  15  1  1  P  L=1.30U  W=30.00U
M21 0  25  15  0  N  L=1.30U  W=5.00U
M22 15  25  1  1  P  L=1.30U  W=9.00U
M23 0  3  25  0  N  L=1.30U  W=3.40U
M24 25  3  1  1  P  L=1.30U  W=3.40U
M25 1  1  3  0  N  L=5.00U  W=2.00U
M26 3  101 0  0  NDEP L=10.00U W=20.0U

```

```

*
```

```

* *****
```

```

* VOLTAGE SOURCES
```

```

*
```

```

VDD 1 0 5
YCOM 101 0 -5
YE 4 0 PULSE (0 5 5NS 5NS 5NS 30NS 40NS)

```

```

*
```

```

.OPTION NOLIST NOMOD NOOP LIMPTS=1000
.TRAN INS 40NS
.PLOT TRAN V(3) V(4) V(17) V(20) (0,5)
.PLOT TRAN I(YE)

```

```

*
```

```

.END

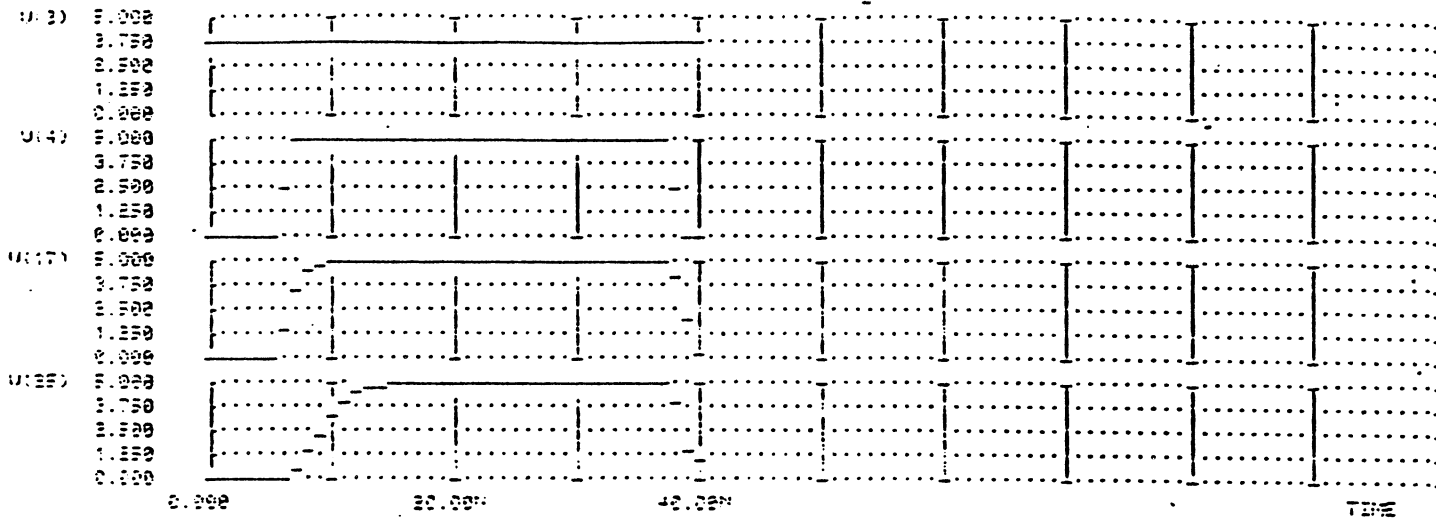
```

ANNEXE 3.3 : Résultats de simulation

Pour un dispositif situé en haut d'une mémoire 64k

V(17) : plot d'adresses d'une mémoire 64k

V(25) : plot d'alimentation Vdd d'une mémoire 64k

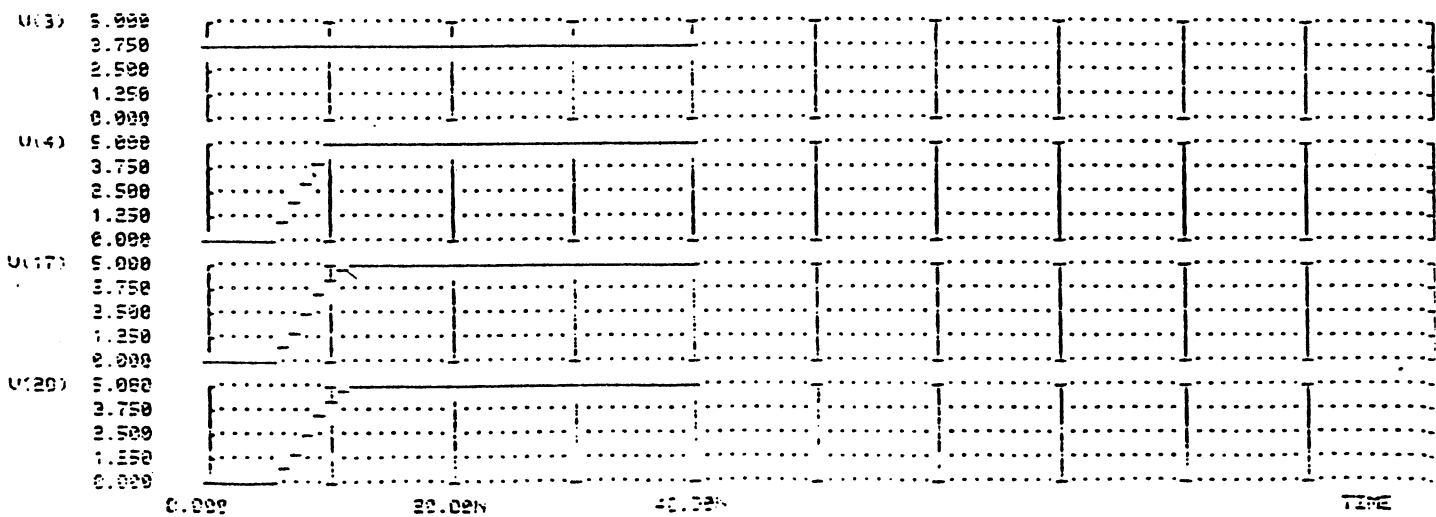


V(3) ---> Le temps nécessaire pour transférer une adresse = 3ns

V(25)---> Le temps nécessaire pour transférer l'alimentation Vdd = 6ns

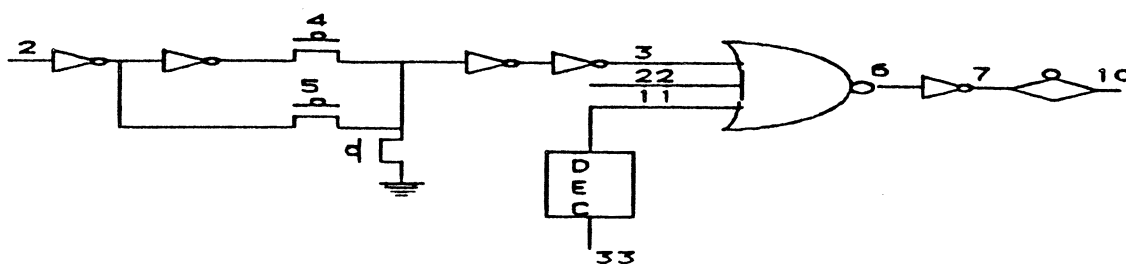
Pour un dispositif situé en bas d'une mémoire 64k

V(20) : plot de données d'une mémoire 64k



V(20) ---> Le temps nécessaire pour transférer une donnée = 5ns

ANNEXE 3.4 : Décodeur de rangée



DECODEUR

.MODEL MP *BIB:spi\$bib:HCMOS3C.ENRP3.PCC

.MODEL MN *BIB:spi\$bib:HCMOS3C.ENRN3.PCC

**

***** LIGNE DE DONNEES DE 25U *****

.SUBCKT INV 1 2 3

M2 3 2 0 0 MN W=3.4U L=1.3U

M1 3 2 1 1 MP W=3.4U L=1.3U

.ENDS

.SUBCKT NOR 1 2 3 4 5

M1 5 2 0 0 MN W=4U L=1.3U

M2 5 3 0 0 MN W=4U L=1.3U

M3 5 4 0 0 MN W=4U L=1.3U

M4 1 2 6 1 MP W=12U L=1.4U

M5 6 3 7 1 MP W=12U L=1.4U

M6 7 4 5 1 MP W=12U L=1.4U

.ENDS

.SUBCKT DECO 1 2 3 4 5 11

X1 1 2 3 INV

X2 1 6 7 INV

M1 7 4 11 0 NDEP W=30U L=10U

M2 6 5 11 0 NDEP W=30U L=10U

M3 11 0 0 1 MP W=3.4U L=1.4U

X3 1 11 22 INV

X4 1 22 3 INV

.ENDS

.SUBCKT PORT 1 2 3 4 5

M2 2 4 5 0 MN W=100U L=1.3U

M1 2 3 5 1 MP W=100U L=1.3U

.ENDS

.SUBCKT INV1 1 2 3

M2 3 2 0 0 MN W=20.2U L=1.3U

M1 3 2 1 1 MP W=60U L=1.4U

.ENDS

X1 1 2 3 4 5 40 DECO

X2 1 33 11 44 55 50 DECO

X3 1 3 22 11 6 NOR

X4 1 6 7 INV1

X5 1 7 8 9 10 PORT

VCC 1 0 5

VP 8 0 0

VN 9 0 1

VAR1 2 0 PWL 0 0 10NS 0 12NS 5 30NS 5 32NS 0 50NS 0

VAR2 33 0 0

VGF1 4 0 1

VGF2 5 0 -5

VGF3 44 0 -5

VGF4 55 0 -5

VCSW 22 0 0

.TRAN 1NS 50NS

.OPTION NOMOD NOOP NOLIST LVLTIM=1 LDIF=3.7U

.PLOT TRAN V(2) V(10) (0.5)

END

ANNEXE 3.5 : Résultats de la simulation

***** DECODEUR DE RANGEE *****

LEGEND:

*: V(2) ENTREE DE DECODEUR (ADRESSES DE RANGEE)

+: V(10) SORTIE DE DECODEUR (ENTREE DE CS DE 64K)

TIME

X(+)--	0.0	1.25	2.5	3.75	5.0
0.000E+00	*	.	.	.	+
1.000E-09	*	.	.	.	+
2.000E-09	*	.	.	.	+
3.000E-09	*	.	.	.	+
4.000E-09	*	.	.	.	+
5.000E-09	*	.	.	.	+
6.000E-09	*	.	.	.	+
7.000E-09	*	.	.	.	+
8.000E-09	*	.	.	.	+
9.000E-09	*	.	.	.	+
1.000E-08	*	.	.	.	+
1.100E-08	.	.	*	.	+
1.200E-08	X
1.300E-08	X
1.400E-08	+
1.500E-08	.	.	.	+	*
1.600E-08	.	.	+	.	*
1.700E-08	.	+	.	.	*
1.800E-08	*
1.900E-08	+	.	.	.	*
2.000E-08	+	.	.	.	*
2.100E-08	+	.	.	.	*
2.200E-08	+	.	.	.	*
2.300E-08	+	.	.	.	*
2.400E-08	+	.	.	.	*
2.500E-08	+	.	.	.	*
2.600E-08	+	.	.	.	*
2.700E-08	+	.	.	.	*
2.800E-08	+	.	.	.	*
2.900E-08	+	.	.	.	*
3.000E-08	+	.	.	.	*
3.100E-08	+	.	*	.	.
3.200E-08	X
3.300E-08	X
3.400E-08	*	+	.	.	.
3.500E-08	*	.	+	.	.
3.600E-08	*	.	.	+	.
3.700E-08	*	.	.	.	+
3.800E-08	*	.	.	.	+
3.900E-08	*	.	.	.	+
4.000E-08	*	.	.	.	+

* Temps de sélection ou de désélection d'une rangée est de l'ordre de 10ns

ANNEXE 3.6 : Programme de placement des réticules

```

PROGRAM SURF(INPUT,OUTPUT);
VAR SURFOUT: TEXT;
VAR RANG : ARRAY [1..10] OF INTEGER;
    D,R,A,B,X,Y,SU:REAL; I,S,C,L,N:INTEGER;
BEGIN
    WRITELN ('NOMBRE DES MEMOIRE 64K/RETICULE N = ?'); READ(N);
    WRITELN ('COTE X = ?'); READ(A);
    WRITELN ('COTE Y = ?'); READ(B);
    WRITELN ('RAYON R = ?'); READ(R);
    D:=2*R/A-TRUNC(2*R/A);
    X:=R-D*A/2; Y:=B; L:=1; C:=0;
    BEGIN
    IF SQRT(Y*Y+X*X)<=R
    THEN RANG[1]:=TRUNC(X/A+1/8)
    ELSE
    BEGIN
    X:=X-A/2; RANG[1]:=TRUNC(X/A+1/8)
    END
    END;
    OPEN (SURFOUT, 'SURFOUT, NEW);
    REWRITE (SURFOUT);
    IF SQRT(Y*Y+X*X)<=R THEN
    BEGIN
    Y:=Y+B;
    S:=RANG[1];
    WHILE X>5*A/8 DO
    BEGIN
        IF SQRT(Y*Y+X*X)<=R THEN
        BEGIN
            L:=L+1;
            RANG[L]:=TRUNC(X/A+1/8);
            Y:=Y+B; S:=S+RANG[L]
        END
        ELSE
        BEGIN
            X:=X-A; C:=C+1
        END
    END;
    C:=2*C;
    IF(X>3*A/8) AND (X<5*A/8)
    THEN
    BEGIN
    FOR I:=1 TO L DO RANG[I]:=2*RANG[I]+1;
    S:=4*S+2*L; C:=C+1;
    END
    ELSE
    BEGIN
    FOR I:=1 TO L DO RANG[I]:=2*RANG[I];
    S:=4*S
    END;
    WRITELN ('NOMBRE DE MEMOIRES 64K/RETICULE N =',N:3);
    WRITELN ('COTE X =',A:3:3);
    WRITELN ('COTE Y =',B:3:3);
    WRITELN ('RAYON R =',R:3:3);
    WRITELN;
    WRITELN (SURFOUT,'NOMBRE DE MEMOIRES 64K/RETICULE N =',N:3);
    WRITELN;
    WRITELN (SURFOUT,'COTE X = ',A:5:3);
    WRITELN (SURFOUT,'COTE Y = ',B:5:3);
    WRITELN (SURFOUT,'RAYON R = ',R:5:3);

```

```

WRITELN (SURFOUT,' ');
FOR I:=1 TO L DO
BEGIN
WRITELN(SURFOUT,'RANGEE',I:2,'=':2,RANG[I]:3);
WRITELN('RANGEE',I:2,'=':2,RANG[I]:3)
END;
WRITELN (SURFOUT,' ');
L:=2*L;
SU:=A*B*S*100/(3.1416*R*R);
WRITELN ('X FINAL=',X:3:3);
WRITELN ('Nb DE LIGNE=',L:3);
WRITELN ('Nb DE COLONNES=',C:3);
WRITELN ('Nb DE RETICULES/TRANCHE =',S:4);
WRITELN ('Nb DE MEMOIRES 64K/TRANCHE =',N*S:4);
WRITELN (' ');
WRITELN ('SURFACE UTILE/SURFACE TOTALE =',SU:4:1,'%':2);
WRITELN ('RENDEMENT MINI DE FABRICATION =',100*72/(N*S):4:1,'%':2);
WRITELN (SURFOUT,'Nb DE LIGNE = ',L:3);
WRITELN (SURFOUT,'Nb DE COLONNES = ',C:3);
WRITELN (SURFOUT,'Nb DE RETICULES/TRANCHE = ',S:4);
WRITELN (SURFOUT,'Nb DE MEMOIRES 64K/TRANCHE = ',N*S:4);
WRITELN (SURFOUT,' ');
WRITELN (SURFOUT,'SURFACE UTILE/SURFACE TOTALE = ',SU:4:1,'%':2);
WRITELN (SURFOUT,'RENDEMENT MINI DE FABRICATION= ',100*72/(N*S):4:1,'%':2)
END
ELSE
BEGIN
WRITELN ('COTE X =',A:3:3);
WRITELN ('COTE Y =',B:3:3);
WRITELN ('RAYON R =',R:3:3);
WRITELN;
WRITELN (SURFOUT,'COTE X =',A:3:3);
WRITELN (SURFOUT,'COTE Y =',B:3:3);
WRITELN (SURFOUT,'RAYON R =',R:3:3);
WRITELN (SURFOUT,' ');
WRITELN ('RETICULE TROP GRANDE');
WRITELN (SURFOUT,'RETICULE TROP GRANDE')
END
END.

```

Exemple d'execution de programme

NOMBRE DE MEMOIRES 64K/RETICULE N = 1
 COTE X = 5000.000
 COTE Y = 8820.000
 RAYON R = 45000.000

RANGEE 1 = 17
 RANGEE 2 = 15
 RANGEE 3 = 13
 RANGEE 4 = 11
 RANGEE 5 = 3

Nb DE LIGNE = 10
 Nb DE COLONNES = 17
 Nb DE RETICULES/TRANCHE = 118
 Nb DE MEMOIRES 64K/TRANCHE = 118

SURFACE UTILE/SURFACE TOTALE = 81.8 %
 RENDEMENT MINI DE FABRICATION= 61.0 %

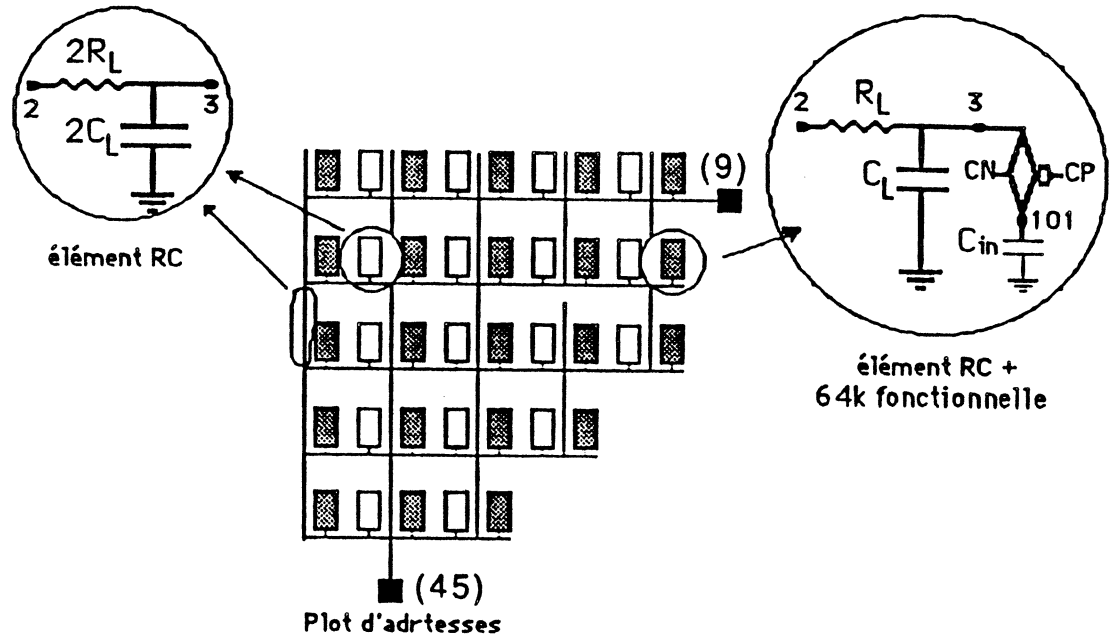
NOMBRE DE MEMOIRES 64K/RETICULE N = 1
 COTE X = 8910.000
 COTE Y = 8820.000
 RAYON R = 45000.000

RANGEE 1 = 9
 RANGEE 2 = 9
 RANGEE 3 = 7
 RANGEE 4 = 5

Nb DE LIGNE = 8
 Nb DE COLONNES = 9
 Nb DE RETICULES/TRANCHE = 60
 Nb DE MEMOIRES 64K/TRANCHE = 120

SURFACE UTILE/SURFACE TOTALE = 74.1 %
 RENDEMENT MINI DE FABRICATION= 60.0 %

ANNEXE 3.7 : Réseaux d'adresses



```

RESADD
.
.MODEL MP *BIB:spi$bib:HCMOS3C.ENRP3.PCC
.MODEL MN *BIB:spi$bib:HCMOS3C.ENRN3.PCC
.
*****CIRCUIT*****
**
*****LIGNE 10 U DE LARGE*****
**
** ELEMENT RC AVEC PORTE DE TRANSFERT **
.
*ELEMRC1      VDD ENTE SORTIE COMP COMN
.SUBCKT ELEMRC1 1 2 3 4 5
RL 2 3 13.32
CL 3 0 1.48PF
M1 3 4 6 1      MP W=100U L=1.3U
M2 3 5 6 0      MN W=100U L=1.3U
CP 6 0 3PF
.ENDS ELEMRC1
** ELEMENT RC AVEC POSSI DE MESURE DE COURANT**
.
*ELEMRC2      VDD ENTE SORTIE COMP COMN MESUR
.SUBCKT ELEMRC2 1 2 3 4 5 6
RL 2 3 13.32
CL 3 0 1.48PF
M1 3 4 6 1      MP W=100U L=1.3U
M2 3 5 6 0      MN W=100U L=1.3U
.ENDS ELEMRC2
.
** ELEMENT RC **
.
*ELEMRC0      ENTE SORTIE
.SUBCKT ELEMRC0 2 3
RL 2 3 13.32
CL 3 0 1.48PF
.ENDS ELEMRC0
** LIGNE DE 9 ELEM RC **
.
*MLIGRC9      VDD N N N N N CP CN CP CN CP CN CP CN CP CN MESUR

```

```
.SUBCKT MLIGRC9 1 5 7 9 12 14 50 51 52 53 54 55 56 57 58 59 100
XRC1 1 4 5 50 51 100      ELEMRC2
XRC2 5 6                  ELEMRC0
XRC3 1 6 7 52 53         ELEMRC1
XRC4 7 8                  ELEMRC0
XRC5 1 8 9 54 55         ELEMRC1
XRC6 9 11                 ELEMRC0
XRC7 1 11 12 56 57       ELEMRC1
XRC8 12 13               ELEMRC0
XRC9 1 13 14 58 59       ELEMRC1
CS  14 0                  0.1PF
CE  4 0                   0.1pF
.ENDS MLIGRC9
```

```
** LIGNE DE 9 ELEM RC **
```

```
*LIGRC9      VDD N N N N N CP CN CP CN CP CN CP CN CP CN
.SUBCKT LIGRC9 1 5 7 9 12 14 50 51 52 53 54 55 56 57 58 59
XRC1 1 4 5 50 51      ELEMRC1
XRC2 5 6              ELEMRC0
XRC3 1 6 7 52 53     ELEMRC1
XRC4 7 8              ELEMRC0
XRC5 1 8 9 54 55     ELEMRC1
XRC6 9 11            ELEMRC0
XRC7 1 11 12 56 57   ELEMRC1
XRC8 12 13          ELEMRC0
XRC9 1 13 14 58 59   ELEMRC1
CS  14 0              0.1PF
CE  4 0               0.1pF
.ENDS LIGRC9
```

```
** LIGNE DE 7 ELEM RC **
```

```
*LIGRC7      VDD N N N N CP CN CP CN CP CN CP CN
.SUBCKT LIGRC7 1 5 7 9 12 50 51 52 53 54 55 56 57
XRC1 1 4 5 50 51      ELEMRC1
XRC2 5 6              ELEMRC0
XRC3 1 6 7 52 53     ELEMRC1
XRC4 7 8              ELEMRC0
XRC5 1 8 9 54 55     ELEMRC1
XRC6 9 11            ELEMRC0
XRC7 1 11 12 56 57   ELEMRC1
CS  12 0              0.1PF
CE  4 0               0.1pF
.ENDS LIGRC7
```

```
** LIGNE 5 ELEM RC **
```

```
*LIGRC5      VDD N N N CP CN CP CN CP CN
.SUBCKT LIGRC5 1 5 7 9 50 51 52 53 54 55
XRC1 1 4 5 50 51      ELEMRC1
XRC2 5 6              ELEMRC0
XRC3 1 6 7 52 53     ELEMRC1
XRC4 7 8              ELEMRC0
XRC5 1 8 9 54 55     ELEMRC1
CS  9 0               0.1PF
CE  4 0               0.1pF
.ENDS LIGRC5
```

```
** RESEAU RC **
```

```
XRES1 1 5 6 7 8 9 0 1 0 1 0 1 0 1 0 1 100 MLIGRC9
XRES2 1 15 16 17 18 19 0 1 0 1 0 1 0 1 0 1 LIGRC9
```

```

XRES3 1 25 26 27 28 29 0 1 0 1 0 1 0 1 0 1    LIGRC9
XRES4 1 35 36 37 38 0 1 0 1 0 1 0 1    LIGRC7
XRES5 1 45 46 47 0 1 0 1 0 1    LIGRC5

```

```

VMESU 100 101 0
CPM 101 0 3PF

```

```

** LIGNES VERTICALES **

```

```

RLV5 5 15 26.32
RLV15 15 25 26.32
RLV25 25 35 26.32
RLV35 35 45 26.32
RLV6 6 16 26.32
RLV16 16 26 26.32
RLV26 26 36 26.32
RLV36 36 46 26.32
RLV7 7 17 26.32
RLV17 17 27 26.32
RLV27 27 37 26.32
RLV37 37 47 26.32
RLV8 8 18 26.32
RLV18 18 28 26.32
RLV28 28 38 26.32
RLV9 9 19 26.32
RLV19 19 29 26.32

```

```

CLV5 5 0 2.92PF
CLV15 15 0 2.92PF
CLV25 25 0 2.92PF
CLV35 35 0 2.92PF
CLV6 6 0 2.92PF
CLV16 16 0 2.92PF
CLV26 26 0 2.92PF
CLV36 36 0 2.92PF
CLV7 7 0 2.92PF
CLV17 17 0 2.92PF
CLV27 27 0 2.92PF
CLV37 37 0 2.92PF
CLV8 8 0 2.92PF
CLV18 18 0 2.92PF
CLV28 28 0 2.92PF
CLV9 9 0 2.92PF
CLV19 19 0 2.92PF

```

```

** SOURCES DE TENSION **

```

```

*VDD 1 0 5

```

```

VEPD 9 0 PWL 0 0 10NS 0 60NS 5 70NS 5
VEPB 45 0 PWL 0 0 10NS 0 60NS 5 70NS 5

```

```

** OPTIONS **

```

```

.OPTIONS NOMOD NOLIST NOOP NOPAGE OPTS
.OPTIONS CPTIME=50000 ITL5=0 ITL1=500 LIMPTS=500
*.OPTIONS GMIN=1.0E-10 VNTOL=1.0E-5 ABSTOL=1.0E-12 RELTOL=1.0E-2
.TRAN 2N 70N

```

```

** SORTIES **

```

```

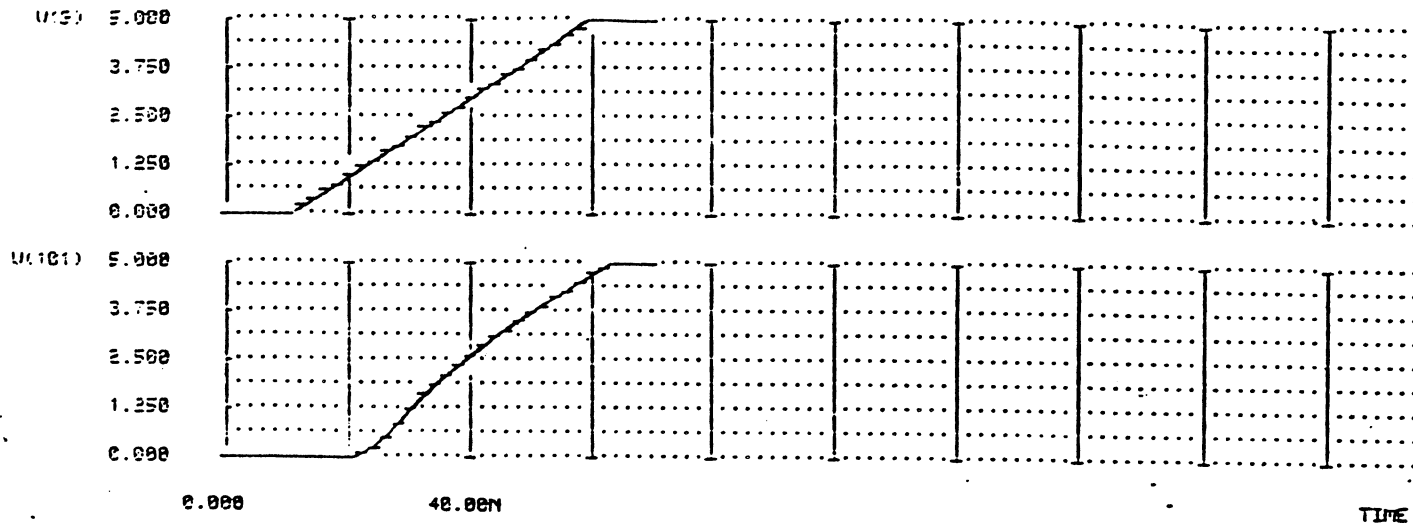
.PLOT TRAN V(9) V(101) (0,5)
.PLOT TRAN I(VMESU) I(VEPD)
END

```

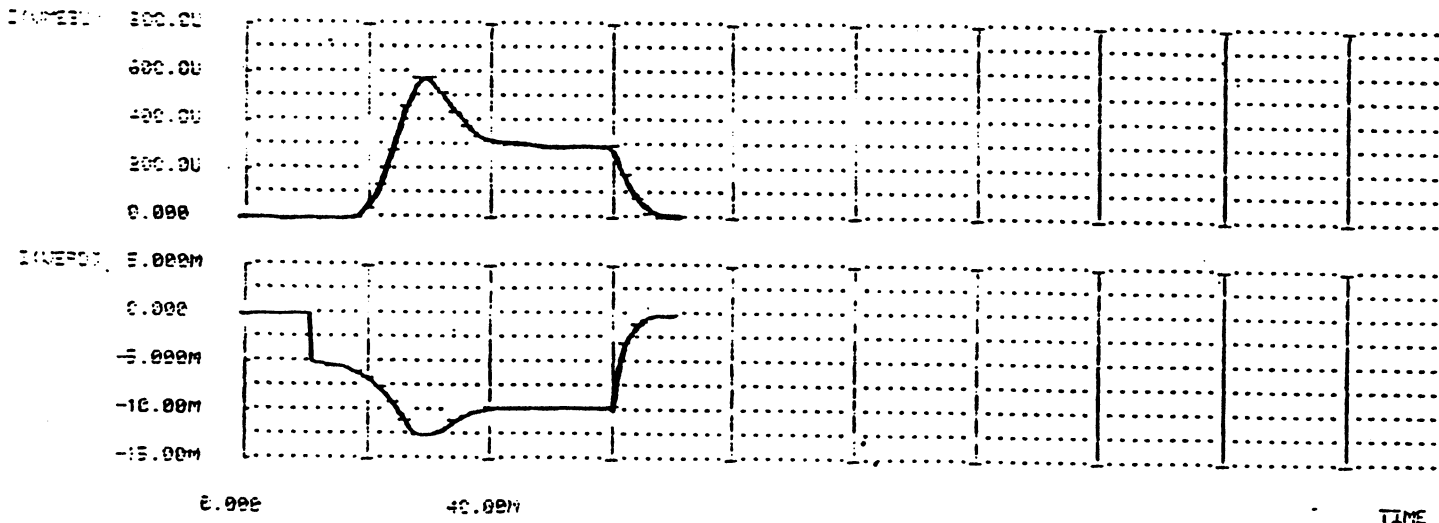
ANNEXE 3.8 : Résultats de la simulation en phase de test

V(9) : plot d'adresses de la tranche

V(101) : plot d'adresses d'une mémoire 64k



V(101) ---> le temps de transit d'adresses de la tranche jusqu'à la mémoire 64k est de l'ordre de 40ns



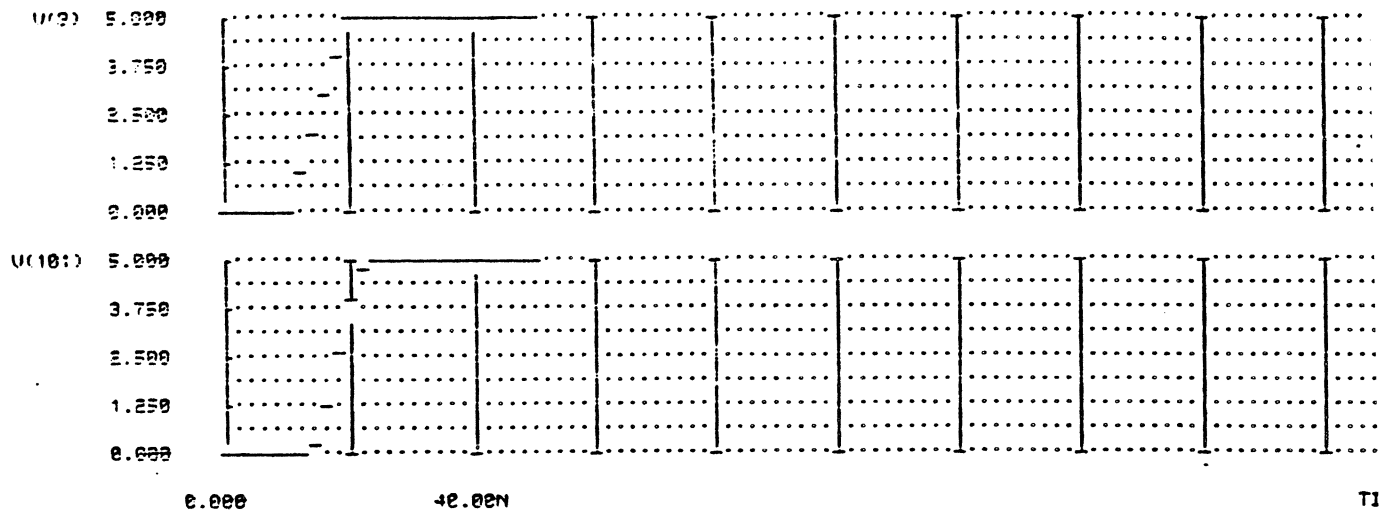
I(VMESU) : le courant fourni à une mémoire 64k est de l'ordre de 0.6mA

I(VEPD) : le courant fourni par un plot d'adresses de la tranche est de l'ordre de 12mA

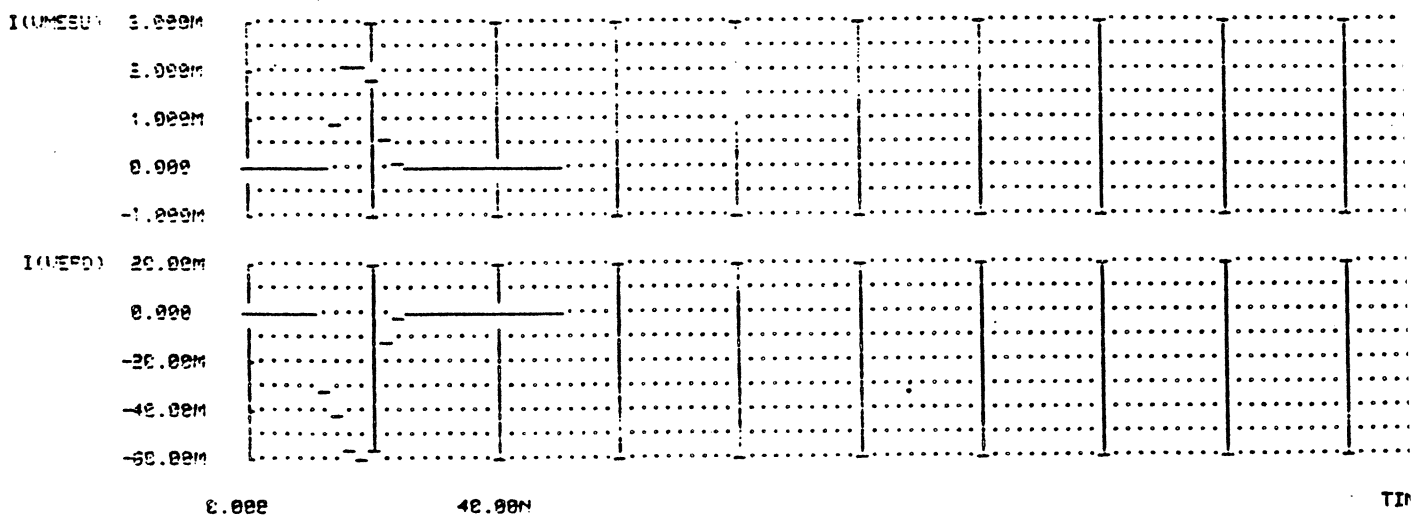
ANNEXE 3.9 : Résultats de la simulation en phase finale

V(9) : plot d'adresses de la tranche

V(101) : plot d'adresses d'une mémoire 64k



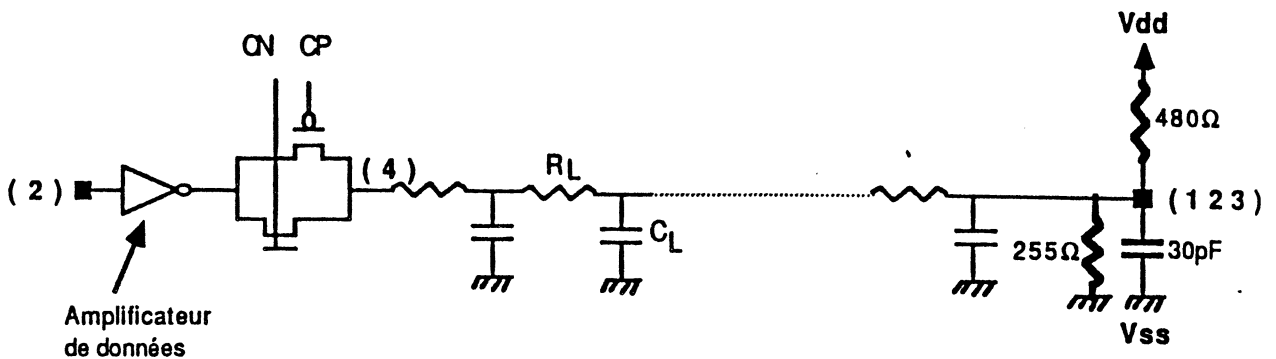
V(101) ---> le temps de transert d'adresses de la tranche jusqu'à la mémoire 64k est de l'ordre de 10ns



I(VMESU) : le courant fourni à une mémoire 64k est de l'ordre de 2mA

I(VEPD) : le courant fourni par un plot d'adresses de la tranche est de l'ordre de 60mA

ANNEXE 3.10 : Une ligne de données en lecture pendant la phase de test



```

RDN
.MODEL MP *BIB:spi$bib:HCMOS3C.ENRP3.PCC
.MODEL MN *BIB:spi$bib:HCMOS3C.ENRN3.PCC
*
*****CIRCUIT*****
**
***** LIGNE DE DONNEES DE 25U *****
**
*[DEPN3.EST] HCMOS_IIC -DEP-N MOSTEP07032 PL31 25 MAY 87
.MODEL NDEP
*
** ELRCIN      VDD ENTE SORTE SORTPORT
.SUBCKT ELRCIN 1 2 3 4
RL 2 3          5.33
CL 3 0          3.13PF
*PORTOUT VDD ENT COMP COMN SORT
XPORTOUT 1 3 0 1 4 PORTOUT
CIN 4 0 5PF
.ENDS ELRCIN
*
*ELEMRC0      ENTE SORTE
.SUBCKT ELEMRC0 2 3
RL 2 3          5.33
CL 3 0          3.13PF
.ENDS ELEMRC0
*
*BUFF         VDD ENT SORT
.SUBCKT BUFF 1 2 3
M1 3 2 1 1      MP W=600U L=1.2U
M2 3 2 0 0      MN W=400U L=1.2U
.ENDS
*
*PORTOUT      VDD ENT COMP COMN SORT
.SUBCKT PORTOUT 1 2 3 4 5
M1 2 3 5 1      MP W=600U L=1.3U
M2 2 4 5 0      MN W=600U L=1.3U
.ENDS
*
** LIGNE RC 25U DE LARGE **
*
.SUBCKT LIGRC 4 13 23
CE 4 0 0.1PF
XRC1 4 5        ELEMRC0
XRC2 5 6        ELEMRC0
XRC3 6 7        ELEMRC0

```

```

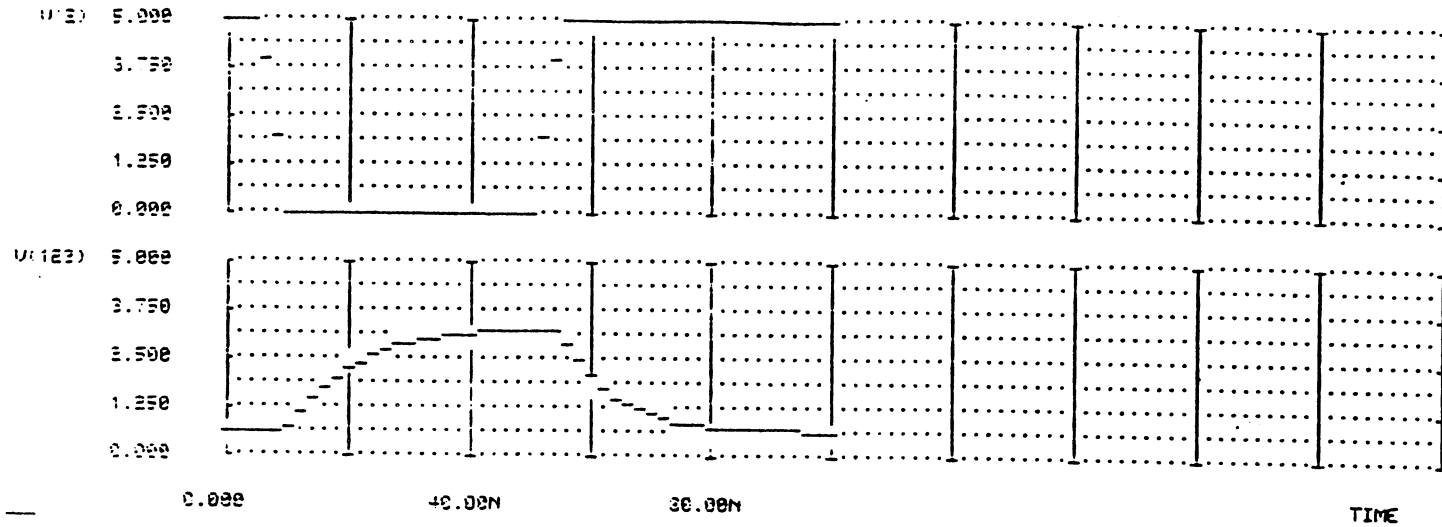
XRC4 7 8      ELEMRC0
XRC5 8 9      ELEMRC0
XRC6 9 11     ELEMRC0
XRC7 11 12    ELEMRC0
XRC8 12 13    ELEMRC0
XRC9 13 14    ELEMRC0
XRC10 14 15   ELEMRC0
XRC11 15 16   ELEMRC0
XRC12 16 17   ELEMRC0
XRC13 17 18   ELEMRC0
XRC14 18 19   ELEMRC0
XRC15 19 20   ELEMRC0
XRC16 20 21   ELEMRC0
XRC17 21 22   ELEMRC0
XRC18 22 23   ELEMRC0
CS 23 0 0.1PF
.ENDS LIGRC
.
** RESEAU RC **
.
XBUFF 1 2 3      BUFF
XPORT 1 3 0 1 4  PORTOUT
VMESUR 4 101 0
XRES1 101 13 23  LIGRC
XRES2 201 113 123 LIGRC
RCONV 13 113 10.66
CCONV 13 0 6.26PF
.
* DISPOSITIF DE MESURE, PIN ELECTRONIQUE *
CPE 123 0 30PF
RALIM 123 1 600
RMASS 123 0 300
*RALIM 123 1 480
*RMASS 123 0 255
.
VDD 1 0 5
VE 2 0 PWL 0 5 5NS 5 10NS 0 50NS 0 55NS 5 100NS 5
.TRAN 2NS 100NS
** OPTIONS **
.OPTIONS NOMOD NOLIST NOOP NOPAGE OPTS
.OPTIONS CPTIME=50000 ITL5=0 ITL1=500 LIMPTS=500
*.OPTIONS LVLTIM=1
.PLOT TRAN V(2) V(123) (0,5)
.PLOT TRAN I(VMESUR)
END

```

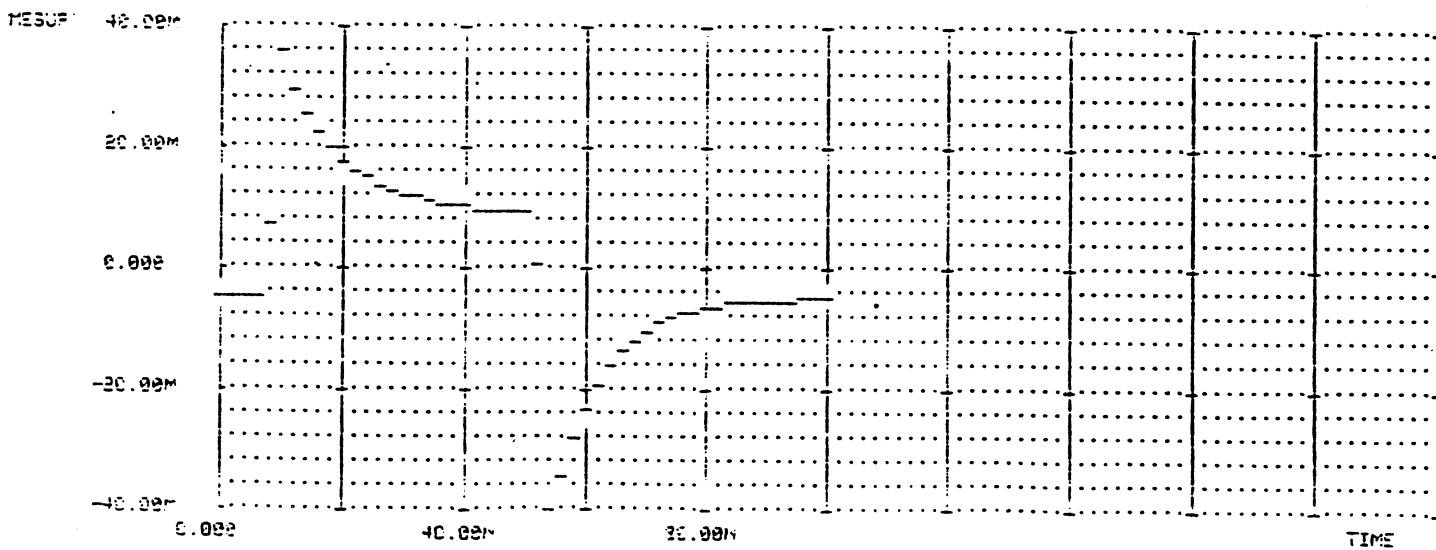
ANNEXE 3.11 : Résultats de la simulation en phase de test

V(2) : plot de données d'une mémoire 64k

V(123) : plot de données de la tranche



V(123) ---> le temps de transert de données d'une mémoire 64k jusqu'à un plot de données de la tranche est de l'ordre de 40ns

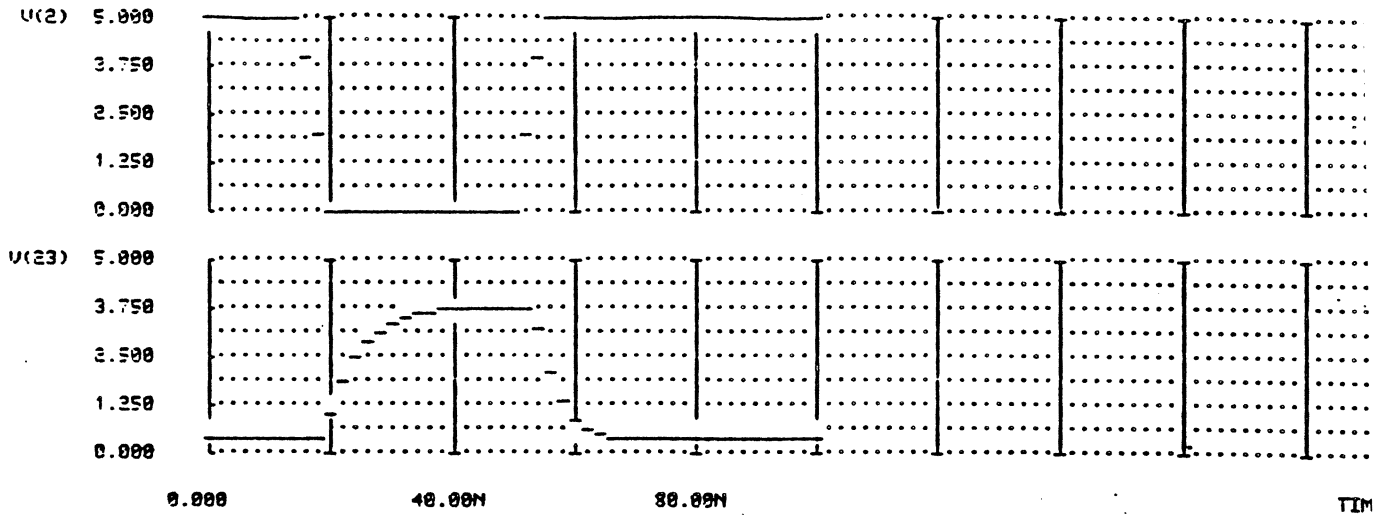


I(VMESU) : le courant fourni par une mémoire 64k est de l'ordre de 25mA

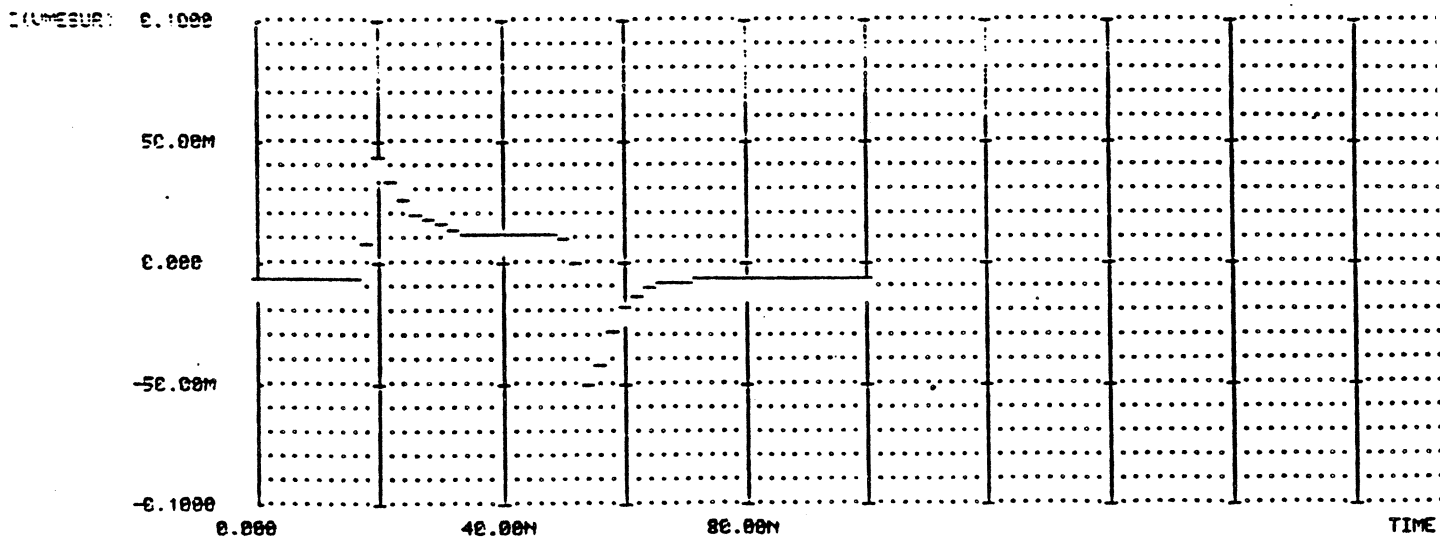
ANNEXE 3.12 : Résultats de la simulation en phase finale

V(2) : plot de données d'une mémoire 64k

V(23) : plot de données de la tranche

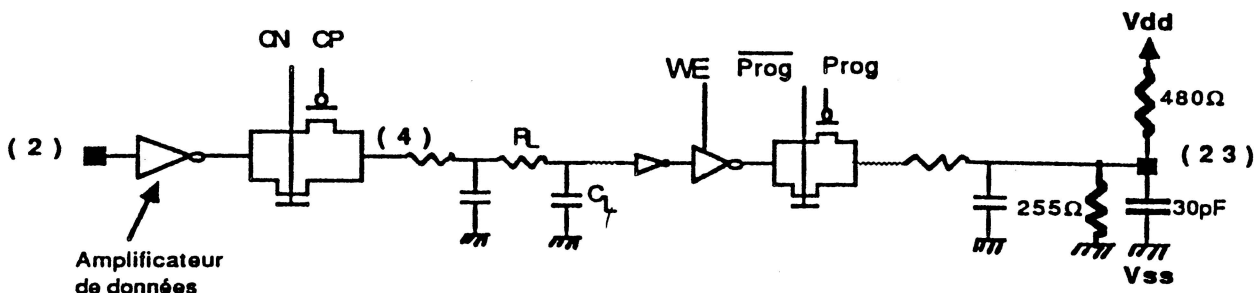


V(23) ---> le temps de transert de données d'une mémoire 64k jusqu'à un plot de données de la tranche est de l'ordre de 15ns



I(VMESU) : le courant fourni par une mémoire 64k est de l'ordre de 45mA

ANNEXE 3.13 : Une ligne de données amplifiée en lecture pendant la phase de test



FDCN

* NAND3 HCMOS3 WP=10 WN=6 CAS TYPIQUE TECHNO 5V 25 C

* evaluation DIFFERENTS LEVEL DE H3SPICE

.MODEL MP *BIB:spi\$bib:HCMOS3C.ENRP3.PCC

.MODEL MN *BIB:spi\$bib:HCMOS3C.ENRN3.PCC

*

.....CIRCUIT.....

**

..... LIGNE DE DONNEES DE 25U AMPLIFIER

**

*[DEPN3.EST] HCMOS_IIC -DEP-N MOSTEP07032 PL31 25 MAY 87

.MODEL NDEP

+ NMOS	LEVEL=3		
+ DMUT = 1.500	DVT = -0.002		
+ EOX = 20.0N	DL = -0.068U	DW = 0.236U	
+ VT0 = -2.189	KB1 = 1.462	KB2 = 0.483	
+ PHI1 = 3.23	VC = 2.500		
+ K0 = 0.361E-04	TG = 0.007	RACC = 466.5U	TW = -0.015U
+ A1 = -0.241U	B1 = -0.112U	A2 = -0.193U	B2 = 0.330U
+ A3 = -0.305U	B3 = 0.230U	DEL = 0.189	KE = 0.266E+08
+ GL = 1.453U	GW = 0.020U	L0 = 13.506U	EPS = 0.081U
+ N1 = 1.000	N2 = 1.000		
+ CJS = 0.350E-03	MJS = 0.38	CJP = 0.180E-9	MJP = 0.24
+ CJC = 0.150E-09	REC = 0.240U	RECL = 0.170U	

**ELRCIN VDD ENTE SORTIE SORTPORT

.SUBCKT ELRCIN 1 2 3 4

RL 2 3 5.33

CL 3 0 3.13PF

*PORTOUT VDD ENT COMP COMN SORT

XPORTOUT 1 3 0 1 4 PORTOUT

CIN 4 0 5PF

.ENDS ELRCIN

*

*ELEMRC0 ENTE SORTIE

.SUBCKT ELEMRC0 2 3

RL 2 3 5.33

CL 3 0 3.13PF

.ENDS ELEMRC0

*

*BUFF VDD ENT SORT

.SUBCKT BUFF 1 2 3

M1 3 2 1 1 MP W=600U L=1.2U

M2 3 2 0 0 MN W=400U L=1.2U

.ENDS

*

*INV VDD ENT SORT

.SUBCKT INV 1 2 3

M1 3 2 1 1 MP W=20U L=2U

M2 3 2 0 0 MN W=10U L=2U

.ENDS

*PORTOUT VDD ENT COMP COMN SORT

.SUBCKT PORTOUT 1 2 3 4 5

M1 2 3 5 1 MP W=600U L=1.3U

M2 2 4 5 0 MN W=600U L=1.3U

.ENDS

** LIGNE RC 25U DE LARGE **

.SUBCKT LIGRC 4 13 23

CE 4 0 0.1PF

XRC1 4 5 ELEMRC0

XRC2 5 6 ELEMRC0

XRC3 6 7 ELEMRC0

XRC4 7 8 ELEMRC0

XRC5 8 9 ELEMRC0

XRC6 9 11 ELEMRC0

XRC7 11 12 ELEMRC0

XRC8 12 13 ELEMRC0

XRC9 13 14 ELEMRC0

XRC10 14 15 ELEMRC0

XRC11 15 16 ELEMRC0

XRC12 16 17 ELEMRC0

XRC13 17 18 ELEMRC0

XRC14 18 19 ELEMRC0

XRC15 19 20 ELEMRC0

XRC16 20 21 ELEMRC0

XRC17 21 22 ELEMRC0

XRC18 22 23 ELEMRC0

CS 23 0 0.1PF

.ENDS LIGRC

** RESEAU RC **

XBUF1 1 2 3 BUFF

XPOR0 1 3 0 1 4 PORTOUT

VMESUR 4 101 0

XRES1 101 13 23 LIGRC

XRES2 201 113 123 LIGRC

XINV 1 13 50 INV

XBUF2 1 50 55 BUFF

XPOR2 1 55 0 1 113 PORTOUT

* DISPOSITIF DE MESURE, PIN ELECTRONIQUE *

CPE 123 0 30PF

RALIM 123 1 600

RMASS 123 0 300

*RALIM 123 1 480

*RMASS 123 0 255

VDD 1 0 5

VE 2 0 PWL 0 5 5NS 5 10NS 0 50NS 0 55NS 5 100NS 5

.TRAN 2NS 100NS

** OPTIONS **

.OPTIONS NOMOD NOLIST NOOP NOPAGE OPTS

.OPTIONS CPTIME=50000 ITL5=0 ITL1=500 LIMPTS=500

*.OPTIONS LVLTIM=1

.PLOT TRAN V(2) V(123) (0,5)

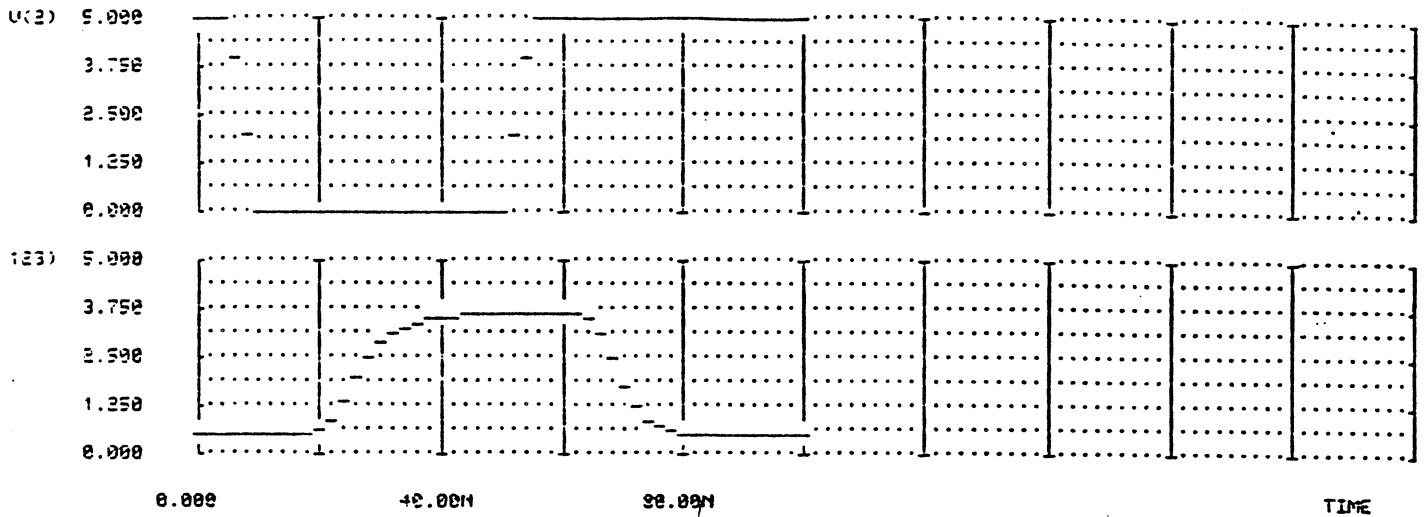
.PLOT TRAN I(VMESUR)

END

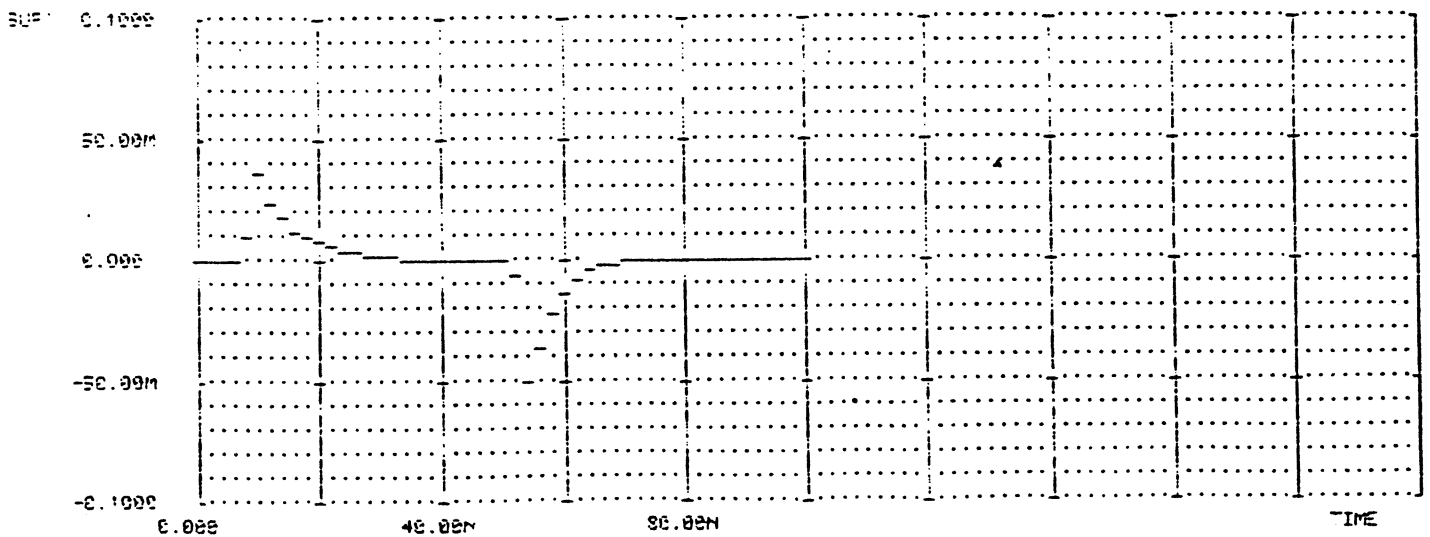
ANNEXE 3.14 : Résultats de la simulation en phase de test

V(2) : plot de données d'une mémoire 64k

V(123) : plot de données de la tranche

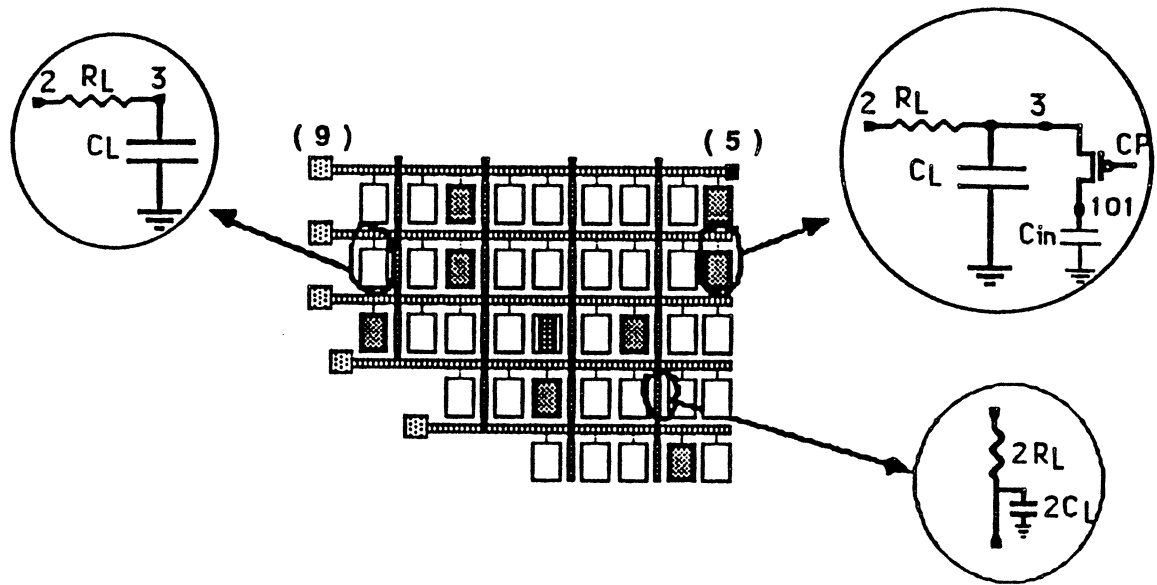


V(123) ---> le temps de transert de données d'une mémoire 64k jusqu'à un plot de données de la tranche est de l'ordre de 30ns



I(VMESU) : le courant fourni par une mémoire 64k est de l'ordre de 35mA

ANNEXE 3.15 : Réseaux d'alimentation



RALIM

.

.MODEL MP *BIB:spi\$bib:HCMOS3C.ENRP3.PCC

.MODEL MN *BIB:spi\$bib:HCMOS3C.ENRN3.PCC

.

*****CIRCUIT*****

**

***** LIGNE 100 U DE LARGE *****

**

** ELEMENT RC AVEC PORTE DE TRANSFERT **

.

*ELEMRC1 VDD ENTE SORTE COMP

.SUBCKT ELEMRC1 1 2 3 4

RL 2 3 1.33

CL 3 0 11.39PF

M1 3 4 6 1 MP W=2400U L=2U

CP 6 0 300PF

.ENDS ELEMRC1

** ELEMENT RC AVEC POSSI DE MESURE DE COURANT**

.

*ELEMRC2 VDD ENTE SORTE COMP MESUR-COURANT

.SUBCKT ELEMRC2 1 2 3 4 6

RL 2 3 1.33

CL 3 0 11.39PF

M1 3 4 6 1 MP W=2400U L=2U

.ENDS ELEMRC2

.

** ELEMENT RC **

.

*ELEMRC0 ENTE SORTE

.SUBCKT ELEMRC0 2 3

RL 2 3 1.33

CL 3 0 11.39PF

.ENDS ELEMRC0

** LIGNE DE 9 ELEM RC **

.

*MLIGRC9 VDD N N N N N CP CP CP CP CP MESUR-COURANT

.SUBCKT MLIGRC9 1 5 7 9 12 14 50 52 54 56 58 100

XRC1 1 4 5 50 100

ELEMRC2

XRC2 5 6

ELEMRC0

```

XRC3 1 6 7 52      ELEMRC1
XRC4 7 8           ELEMRC0
XRC5 1 8 9 54     ELEMRC1
XRC6 9 11         ELEMRC0
XRC7 1 11 12 56   ELEMRC1
XRC8 12 13        ELEMRC0
XRC9 1 13 14 58   ELEMRC1
CS  14 0          0.1PF
CE  4 0           0.1pF
.ENDS MLIGRC9

```

** LIGNE DE 9 ELEM RC **

```

*LIGRC9  VDD N N N N CP CP CP CP CP
.SUBCKT LIGRC9 1 5 7 9 12 14 50 52 54 56 58
XRC1 1 4 5 50      ELEMRC1
XRC2 5 6           ELEMRC0
XRC3 1 6 7 52     ELEMRC1
XRC4 7 8           ELEMRC0
XRC5 1 8 9 54     ELEMRC1
XRC6 9 11         ELEMRC0
XRC7 1 11 12 56   ELEMRC1
XRC8 12 13        ELEMRC0
XRC9 1 13 14 58   ELEMRC1
CS  14 0          0.1PF
CE  4 0           0.1pF
.ENDS LIGRC9

```

** LIGNE DE 7 ELEM RC **

```

*LIGRC7  VDD N N N N CP CP CP CP
.SUBCKT LIGRC7 1 5 7 9 12 50 52 54 56
XRC1 1 4 5 50      ELEMRC1
XRC2 5 6           ELEMRC0
XRC3 1 6 7 52     ELEMRC1
XRC4 7 8           ELEMRC0
XRC5 1 8 9 54     ELEMRC1
XRC6 9 11         ELEMRC0
XRC7 1 11 12 56   ELEMRC1
CS  12 0          0.1PF
CE  4 0           0.1pF
.ENDS LIGRC7

```

** LIGNE 5 ELEM RC **

```

*LIGRC5  VDD N N N CP CP CP
.SUBCKT LIGRC5 1 5 7 9 50 52 54
XRC1 1 4 5 50      ELEMRC1
XRC2 5 6           ELEMRC0
XRC3 1 6 7 52     ELEMRC1
XRC4 7 8           ELEMRC0
XRC5 1 8 9 54     ELEMRC1
CS  9 0           0.1PF
CE  4 0           0.1pF
.ENDS LIGRC5

```

** RESEAU RC **

```

XRES1 1 5 6 7 8 9 50 1 1 56 1 100  MLIGRC9
XRES2 1 15 16 17 18 19 60 1 1 66 1  LIGRC9
XRES3 1 25 26 27 28 29 1 72 1 1 78  LIGRC9
XRES4 1 35 36 37 38 1 1 84 1  LIGRC7
XRES5 1 45 46 47 1 92 1  LIGRC5

```

** CONFIGURATION, COMMANDE DES TRANSISTORS D'ALIM **

VCP10 50 0 PWL 0 0 25NS 0 30NS 5 450NS 5 455NS 0 500NS 0
 VCP16 56 0 PWL 0 0 25NS 0 30NS 5 450NS 5 455NS 0 500NS 0

VCP20 60 0 PWL 0 0 25NS 0 30NS 5 450NS 5 455NS 0 500NS 0
 VCP26 66 0 PWL 0 0 25NS 0 30NS 5 450NS 5 455NS 0 500NS 0

VCP32 72 0 PWL 0 0 25NS 0 30NS 5 450NS 5 455NS 0 500NS 0
 VCP38 78 0 PWL 0 0 25NS 0 30NS 5 450NS 5 455NS 0 500NS 0

VCP44 84 0 PWL 0 0 25NS 0 30NS 5 450NS 5 455NS 0 500NS 0

VCP52 92 0 PWL 0 0 25NS 0 30NS 5 450NS 5 455NS 0 500NS 0

VMESU 100 101 0
 CPM 101 0 300PF
 CDECOUP 17 0 50NF

** LIGNES VERTICALES **

RLV5 5 15 2.66
 RLV15 15 25 2.66
 RLV25 25 35 2.66
 RLV35 35 45 2.66
 RLV6 6 16 2.66
 RLV16 16 26 2.66
 RLV26 26 36 2.66
 RLV36 36 46 2.66
 RLV7 7 17 2.66
 RLV17 17 27 2.66
 RLV27 27 37 2.66
 RLV37 37 47 2.66
 RLV8 8 18 2.66
 RLV18 18 28 2.66
 RLV28 28 38 2.66
 RLV9 9 19 2.66
 RLV19 19 29 2.66

CLV5 5 0 22.78PF
 CLV15 15 0 22.78PF
 CLV25 25 0 22.78PF
 CLV35 35 0 22.78PF
 CLV45 45 0 22.78PF
 CLV6 6 0 22.78PF
 CLV16 16 0 22.78PF
 CLV26 26 0 22.78PF
 CLV36 36 0 22.78PF
 CLV46 46 0 22.78PF
 CLV7 7 0 22.78PF
 CLV17 17 0 22.78PF
 CLV27 27 0 22.78PF
 CLV37 37 0 22.78PF
 CLV47 47 0 22.78PF
 CLV8 8 0 22.78PF
 CLV18 18 0 22.78PF
 CLV28 28 0 22.78PF
 CLV9 9 0 22.78PF
 CLV19 19 0 22.78PF

** SOURCES DE TENSION **

* ALIMENTATION *

VDD 1 0 5

VEPD1 9 0 PWL 0 0 35NS 0 400NS 6 500NS 6
VEPD2 19 0 PWL 0 0 35NS 0 400NS 6 500NS 6
VEPD3 29 0 PWL 0 0 35NS 0 400NS 6 500NS 6
VEPD4 38 0 PWL 0 0 35NS 0 400NS 6 500NS 6
VEPD5 47 0 PWL 0 0 35NS 0 400NS 6 500NS 6

.NODESET V(101)=0

** OPTIONS **

.OPTIONS NOMOD NOLIST NOOP NOPAGE OPTS
.OPTIONS CPTIME=50000 ITL5=0 ITL1=500 LIMPTS=1000
.TRAN 10N 500N

** SORTIES **

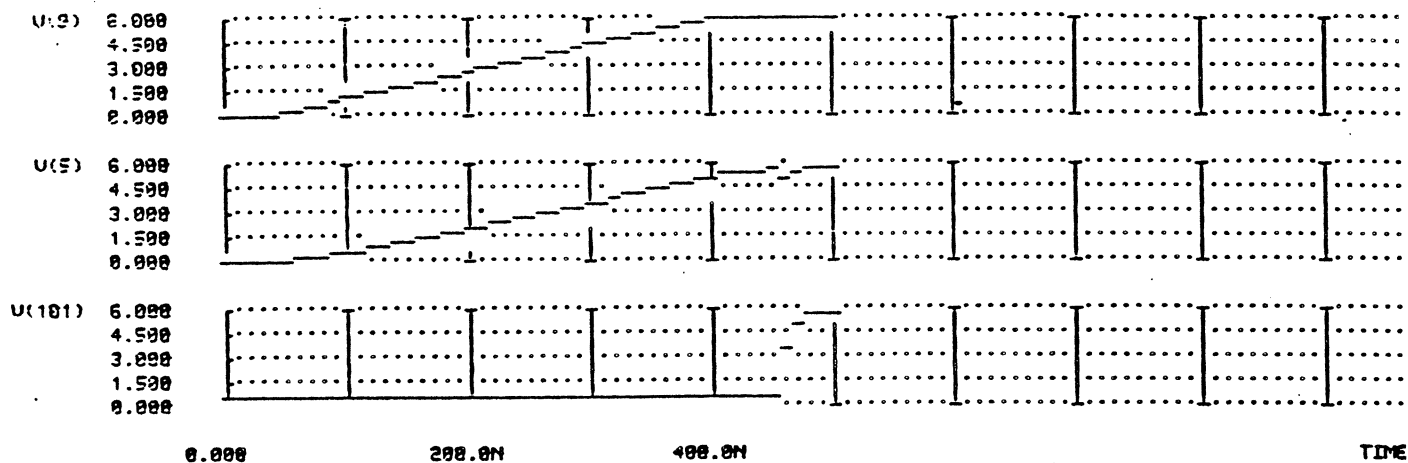
.PLOT TRAN V(9) V(5) V(101) (0,6)
.PLOT TRAN I(VMESU) I(VEPD1)
END

ANNEXE 3.16 : Résultats de la simulation en phase de test

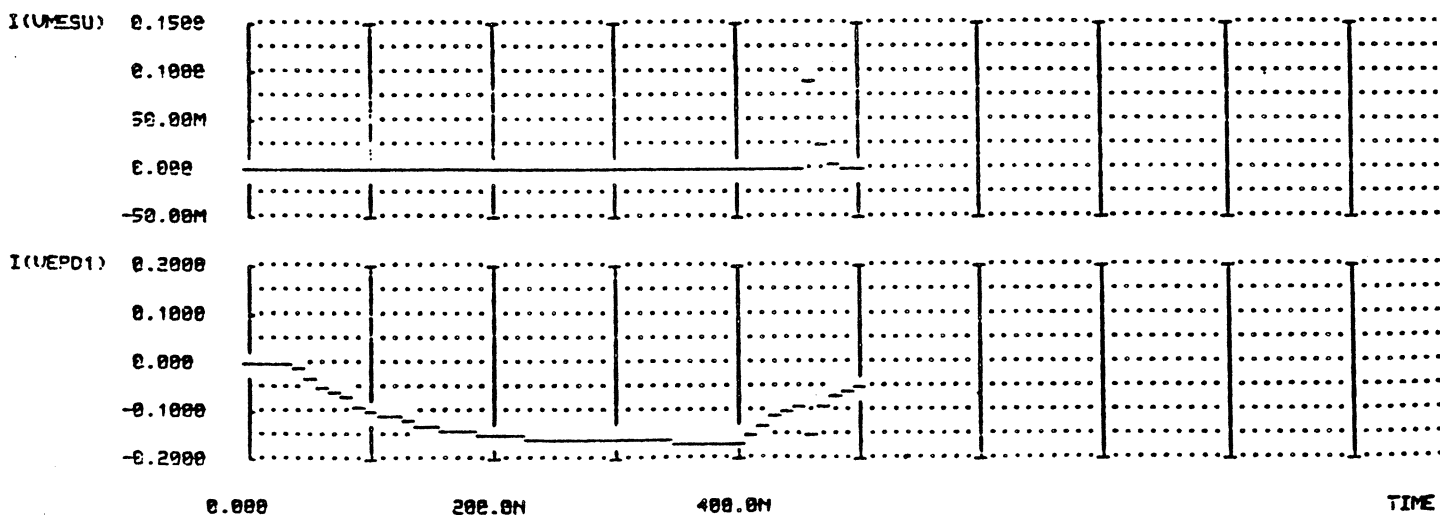
V(9) : plot d'alimentation de la tranche

V(5) : chargement de réseau d'alimentation

V(101) : plot d'alimentation d'une mémoire 64k



V(5) ---> le temps de chargement de réseau d'alimentation est de l'ordre de 350ns



I(VMESU) : le courant fourni a un plot d'alimentation d'une mémoire 64k est de l'ordre de 90mA

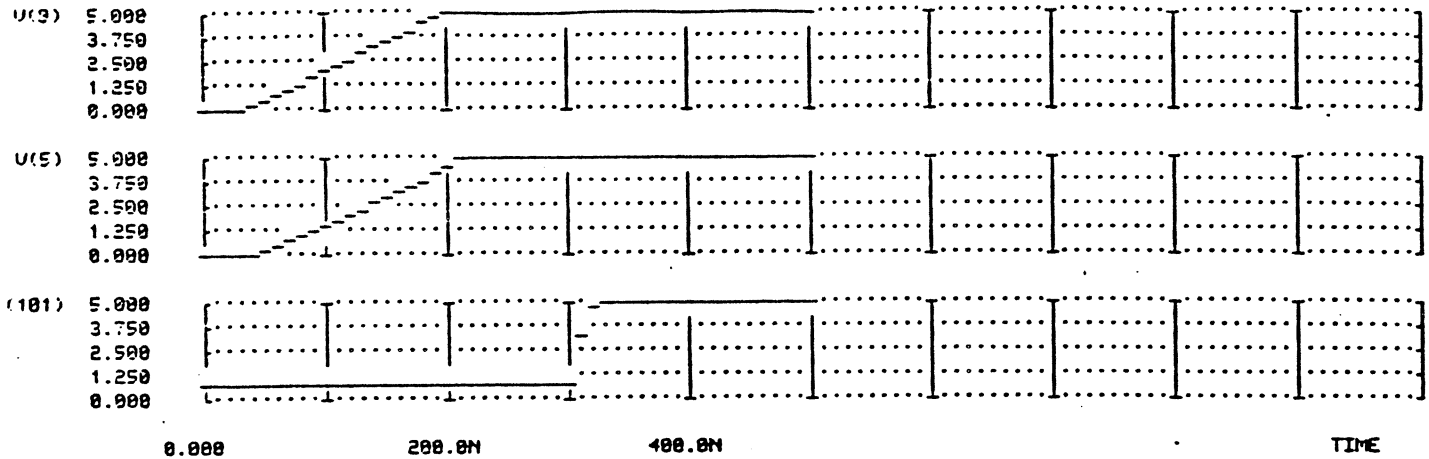
I(VEPDI) : le courant fourni par un plot d'alimentation de la tranche est de l'ordre de 160mA

ANNEXE 3.17 : Résultats de la simulation en phase finale

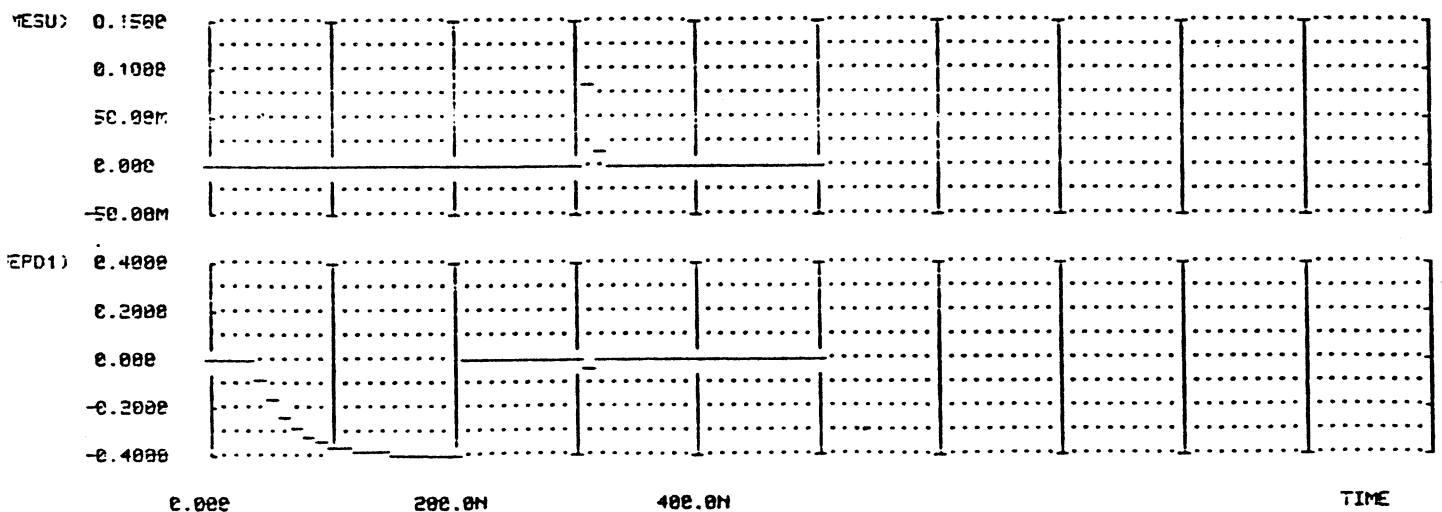
V(9) : plot d'alimentation de la tranche

V(5) : chargement de réseau d'alimentation

V(101) : plot d'alimentation d'une mémoire 64k

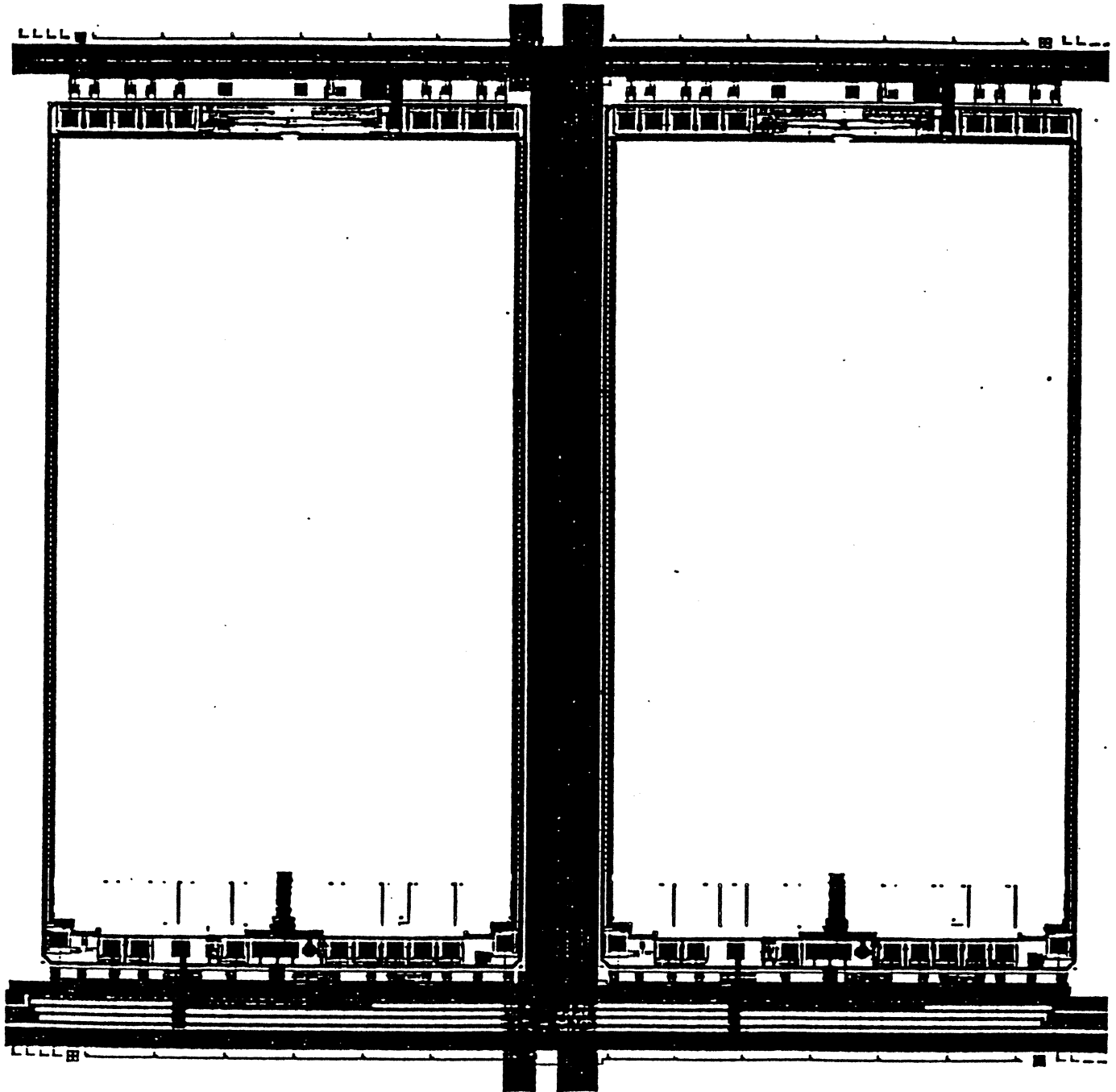


V(5) ---> le temps de chargement de réseau d'alimentation est de l'ordre de 160ns

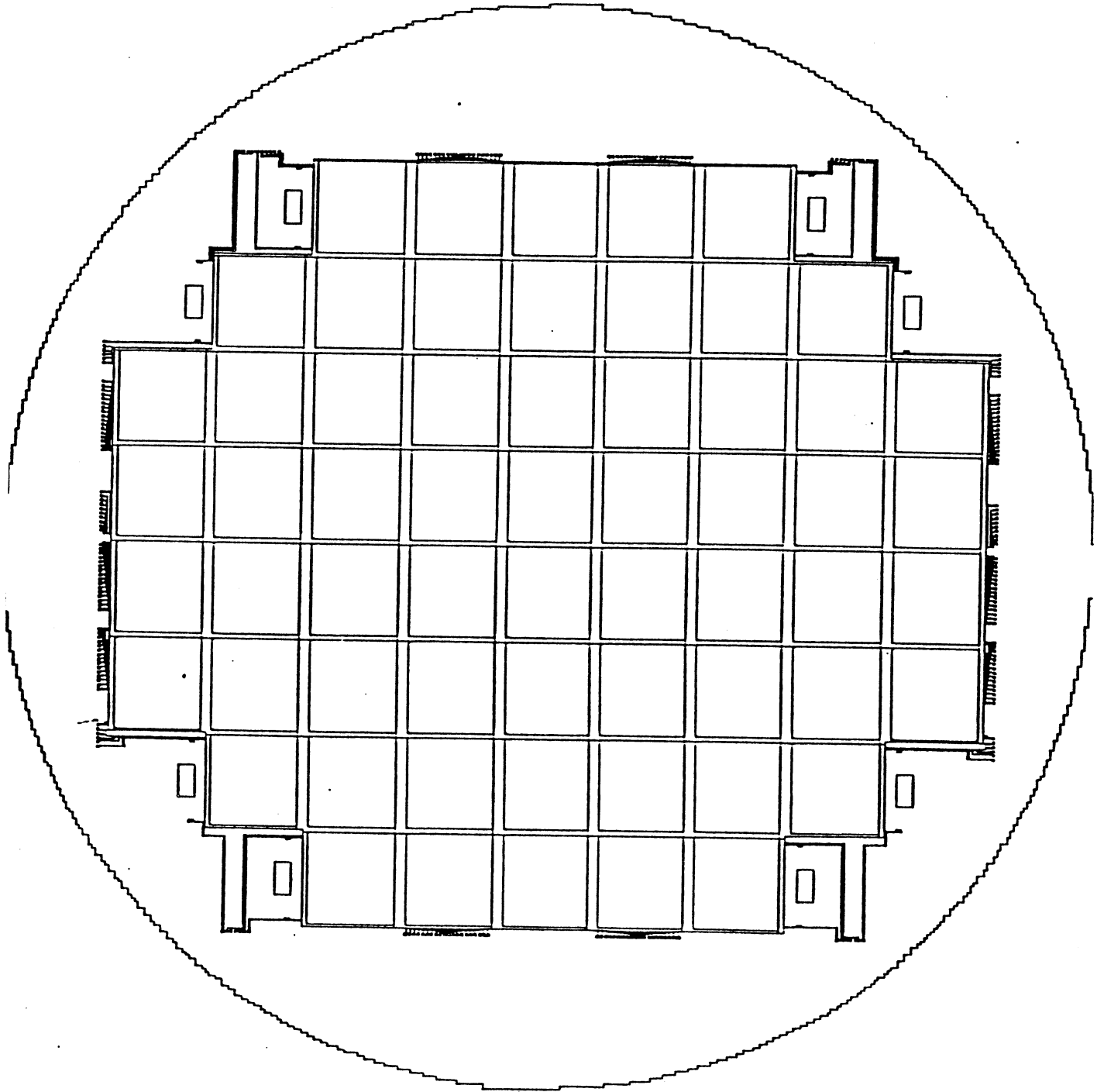


I(YMESU) : le courant fourni a un plot d'alimentation d'une mémoire 64k est de l'ordre de 90mA

I(YEPDI) : le courant fourni par un plot d'alimentation de la tranche est de l'ordre de 400mA

ANNEXE 3.18 : dessin au μ d'un réticule

ANNEXE 3.19 : Implantation sur la tranche de la mémoire 4.5Mbits





REFERENCES

BIBLIOGRAPHIQUES



- [AUB78] R. AUBUSSON, I. CATT
"Wafer Scale Integration - A Fault Tolerant Procedure"
IEEE Journal Solid-State Circuits . Vol.SC-13-3, pp. 339-344
June 1978
- [BAR86] J. BARRETT, E. SMITS, P.L. MORAN
"A copper tracking technique for wafer scale integration"
Proceeding of the IFIP-Workshop on WSI,INPGrenoble Mars 1986
- [BEN86] E. BENNETT
"Inova makes its move in risky Wafer Scale Work"
Electronic -Dec 2 -pp. 57-58, Dec 1985
- [BEN87] P.S. BENNETT, R.P. DIXON, and K.C. OERTLE
"Embedded RAM in Gate Arrays : Configurability and Testability"
VLSI SYSTEMS DESIGN, pp. 60-66, November 1987
- [CHA85] G.H. CHAPMAN
"Laser Linking Technology for RVLSI"
International Workshop on Wafer Scale Integration
Univ Southampton, July 1985
- [DAN86] A. DANDACHE
"Conception de PLA CMOS"
Thèse de doctorat de l'I.N.P.G, I.N.P.Grenoble, Juil 1986
- [DON86] J.F.McDONALD, C. DONLAN,
"Yield of Wafer-Scale Interconnections"
VLSI SYSTEMS DESIGN, pp. 62-66, December 1986

- [ESP87] B. NASREDDINE, D. MARRON
"Architecture of the Wafer Scale Memory"
Deliverable D5-Raport - ESPRIT824 - TaskA2/2.May1987
- [FER85] R. FERRANT
"Fast and low power asynchronous static RAM"
Centre d'étude et de fabrication - Eurotechnique, Mars 1985
- [FLA86] S. FLANAGAN, P.A. REED
"Two 13-ns 64k CMOS SRAM's with very low active power and improved asynchronous circuit techniques"
IEEE Journal of Solid state Circuits, Vol SC 21,N°5, pp 692-702
Oct 1986
- [GEN87] Ph. GENESTIER
"Conception de microprocesseurs à haut rendement"
Thèse de doctorat de l'I.N.P.G, I.N.P.Grenoble, Juil 1987
- [GIR86] P. GIRARD, F.M. ROCHE, B. PISTOULET
"Electron Beam effects on VLSI MOS conditions for testing and reconfiguration"
Proc of the IFIP-Workshop on WSI,I.N.P.Grenoble Mars 1986
- [HAW85] R. HAWKINS
"Wafer-Scale manufacturers must cut their teeth on simple memory circuits"
Electronic Design, p. 66, Oct. 3, Oct. 3. 1985

- [HAY75] J. HAYES
"Detection of Pattern-Sensitive Faults in Random-Access Memories"
IEEE Transactions on Computers, pp. 150-157, Feb 1975
- [HED82] K. HEDLUNG
"Wafer scale intergation of parallel processors"
PhD Tesis -Purdue University, West lafayette, IN, Nov 1982
- [HEN76] J. HENK
"Moving Inversions Test Pattern is Thorough, Yet Sppeedy"
Computer Design, May 1976
- [HOC87] B. HOCHET
"Conception de VLSI applications au calcul numerique"
Thèse de doctorat de l'I.N.P.G, I.N.P.Grenoble 12 janv 1987
- [KET85] M.B. KETCHEN
"Point Defect Yield Model for Wafer Scale Integration"
IEEE Circuits and devices magazine, pp 24-34, Juillet 1985
- [KIN86] K. KINOSHITA
"Built-In Testing of Memory Using an On-Chip Compact Testing Scheme"
IEEE Transactions on Computers, Vol.C-35,N°.10,p.862, Oct 1986
- [LEA85] R. LEA
"WASP : A WSI Associative String Processor for Structured Data Processing"
Proc of the International Workshop on Wafer Scale Intergation
Southampton Univ, July 1985

- [LEI85] T. LEIGHTON and C. E. LEISERSON
" Wafer-Scale Integration of Systolic Arrays"
IEEE Transactions on Computers, VOL. C-34, NO. 5,
pp. 448-461, May 1985
- [LEI86] T. LEIGHTON and C. E. LEISERSON
"A Survey of Algorithms for Integrating Wafer-Scale Systolic
Arrays"
Proceeding of the IFIP-Workshop on WSI,INPGrenoble Mars 1986
- [LEW85] L. LEWYN, J. D. MEINDL
"Physical limits of VLSI dRAM's"
IEEE Journal of Solid-state Circuits, vol SC 20, N° 1, Feb 1985
- [MAR82] M. MARINESCU
"Simple and Efficient Algorithms for Functional RAM Testing"
IEEE, Test Conference, pp. 236-239, May 1982
- [MET83] L. R. METZGER
"A 16 k CMOS PROM with polysilicon fuses links"
IEEE Journal of Solid-State Circuits . Vol. SC.18, N°.5, Oct 1983
- [MOO86] W. MOORE (Univ of Oxford)
"Yield Enhancement of Large Memories"
IFIP, Workshop on Wafer Scale Integration, Ses B1
I.N.P.Grenoble Mars 1986
- [MOR86] S.G. MORTON, E. ABREU
"The Dynamically Reconfigurable CAP Array Chip I"
IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. sc-21, NO. 5,
pp. 820-826, Octobre 1986

- [NAS86] B. NASREDDINE, C. JAY, Y. GIOMLAI
"Architecture of the Wafer Scale memory"
Intermediate report RAM/WSI-ESPRIT Project 824-Task A2/1
October 1986
- [NIC87] G. NICOLATS
"Statistics on fuse and antifuse" and "Aluminium repair on test
vehicle"
Deliverable D3- ESPRIT 824-Activity A1/5 and A 4/2, Oct 1987
- [SAK86] M. SAKAROVITCH
Hermann, Editeurs des Sciences et des Arts.
"OPTIMISATION COMBINATOIRE, Programmation Discrete"
- [SAK1-83] T. SAKURAI
"Approximation of Wiring Delay in mosfet LSI"
IEEE Journal of Solid-State Circuits . Vol. SC.18.N°.4, Aug 1983
- [SAK2-83] T. SAKURAI
"Simple Formules for 2 and 3 Dim Capacitance"
IEEE Trans. on Electron Devices . Vol. ED. 30. Feb 1983
- [SHA84] D.C. SHAVER
"Electron Beam Customization, Repair and Testing of
Wafer-Scale Circuits"
Solid State Technology, pp. 135-139, Feb 1984
- [SHI88] H. SHIMADA
"A 46-ns 1Mbit CMOS SRAM"
IEEE Journal of Solid-state Circuits, vol 23, N° 1, pp. 53-57
Feb 1988

- [SMI81] R. SMITH, J.D. CHLIPALA
"Laser programmable redundancy and yield improvement in :
64k DRAM"
IEEE Journal of Solid state Circuits, Vol SC 16, p 506, Oct 1981
- [STA83] C.H. STAPPER, F.M. ARMSTRONG
"Integrated Circuit Yield Statics"
Proceedings of the IEEE, VOL. 71, NO. 4, April 1983
- [STO87] H. STOPPER
"Transmission lines for Wafer Scale Integration"
Proceeding of the IEEE, pp. 551-554, Apr 1987
- [SYS87] P. HURAT, E.F. KOUKA, J.L. PATRY, G. SAUCIER
"The ASAP architecture"
Deliverable D20 - ESPRIT 824 - Task B, Section I. May 1987
- [THO88] Y. THOSHIO
"A 4-Mbit DRAM with 16-bit concurrent ECC"
IEEE Journal of Solid-State Circuits, vol, 23, No. 1, Feb 1988
- [TRI88] J. TRILHE
"Wafer Scale Integration : Mythe ou Réalite"
3^{ème} Colloque national de circuits intergés à la demande
Grenoble (Alpes Congrès) Janv 1988
- [UEO84] Y. UEOKA, C. MINAGAWA
"A Defect-Tolerant Design for Full-Wafer Memory LSI"
IEEE Journal Solid-State Circuits . Vol.SC.19, N°.3 pp. 319-324
June 1984

[VAL86] C. VAL

"Wafer Scale Integration packaging"

Proceeding of the IFIP - Workshop on WSI, Session k.3

Grenoble (INPG) March 1986

[YAM88] T. YAMADA, H. KOTANI

"A 4-Mbits DRAM whih 16-bit concurrent ECC"

IEEE Journal of Solid-State Circuits . Vol. 23, N°.1, pp. 20-25

Feb 1988



TABLE DES MATIERES



Sommaire

Introduction	13
 Chapitre 1 : ARCHITECTURE WSI.....	 19
I. INTRODUCTION	21
II. LA TOLERANCE AUX DEFAUTS.....	21
III. ARCHITECTURES INTEGRABLES SUR TRANCHE.....	22
III.1. L'APPROCHE "HYBRIDE".....	22
III.2. L'APPROCHE "WSI"	22
III.2.1 Les systèmes WSI non répétitifs.....	23
III.2.2 Les systèmes WSI répétitifs	24
 Chapitre 2 : ETUDE DE FAISABILITE ET CONTRAINTES.....	 29
I. INTRODUCTION.....	31
II. DIFFERENTES ETAPES DE REALISATION	31
II.1. ORGANISATION VIRTUELLE	31
II.2. ETAPES GENERALES DE REALISATION	33
III. LES CONTRAINTES LIEES AUX DIFFERENTES ETAPES DE REALISATION..	34
III.1. LES CONTRAINTES LIEES A L'IMPLANTATION.....	34
III.1.1. La technologie HCMOS3.....	34
III.1.2. La photo-répétition.....	35
III.1.3. Dispositifs de reconfiguration.....	36
III.1.4. La mémoire SRAM 64k.....	41
III.1.5. Les lignes d'interconnexion.....	53

III.2. CONTRAINTES DE TEST.....	63
III.2.1. Le temps de test.....	63
III.2.2. Test en parallèle.....	64
IV. CONCLUSION	68
Chapitre 3 : REALISATION DE LA MEMOIRE 4.5Mbits	7.1
I. INTRODUCTION.....	73
II. ARCHITECTURE ET REALISATION DE LA MEMOIRE 4.5 Mbits.....	73
II.1. Environnement de la mémoire 64k.....	73
II.1.1. Dispositif de commutation.....	73
II.1.2. Décodeur de rangée.....	76
II.2. Choix de la forme d'un réticule.....	80
II.3. Réseaux d'interconnexion.....	82
II.3.1. Introduction.....	82
II.3.2. Epaisseur en cuivre (copper tracking).....	83
II.3.3. Réseaux d'adresses et des signaux de contrôles.....	84
II.3.4. Réseaux de données.....	90
II.3.5. Réseaux d'alimentation.....	96
II.4. Architecture d'un réticule.....	98
II.5. Solution choisie sur la tranche.....	100
II.5.1. Présentation.....	100
II.5.2. Influence du rendement des mémoires 64K sur la solution choisie.....	103
II.5.3. Rendement de la périphérie des mémoires 64K dans un réticule.....	107

II.6. Evaluation de la performance de la mémoire-WSI.....	108
II.7. Conclusion.....	108
Chapitre 4 : STRATEGIE DE TEST ET DE CONFIGURATION DE LA MEMOIRE -WSI.....	109
I. INTRODUCTION.....	111
II. DEFAUTS PHYSIQUES DE FABRICATION.....	112
II.1 Défauts paramétriques.....	112
II.2 Défauts aléatoires.....	112
III. ERREURS DUES A DES PANNES PHYSIQUES.....	113
III.1. Sur le réseaux.....	113
III.2. Dans les dispositifs de commutation.....	114
III.3. Sur les mémoires 64k.....	115
IV. TEST DU DISPOSITIF DE COMMUTATION.....	116
IV.1. début de test.....	116
IV.2. Test des portes de transfert.....	117
IV.2.1 Chemin de test.....	117
IV.2.2 Algorithme de test.....	118
IV.2.3 Phases de test.....	119
IV.2.4 Diagnostic de chaque phase et mémorisation.....	120
V. TEST DES MEMOIRES 64k.....	121
V.1. Test préliminaire.....	121
V.2. Test fonctionnel.....	121
V.2.1 Rappel sur l'architecture de la mémoire 64K.....	121
V.2.2 Les critères de choix d'un algorithme.....	123

V.2.3 Algorithme "damier".....	123
V.2.4 Réalisation du test.....	124
V.3. Test dynamique.....	128
V.4. Test paramétrique.....	128
VI. TEST DES RESEAUX D'INTERCONNEXION.....	128
VI.1. Présentation des réseaux.....	128
VI.2. Environnement de test.....	129
VI.3. Test des réseaux d'adresses.....	129
VI.3.1 Présentation des réseaux.....	129
VI.3.2 Algorithme de test.....	130
VI.4. Test des réseaux de données et des alimentations.....	131
VI.5. Diagnostic et mémorisation.....	133
VI.6. Réparation des réseaux.....	133
VII. Test des décodeurs et de la logique de commande.....	134
VII.1 Test des décodeurs.....	134
VII.2 Test de la logique de commande.....	134
VIII. Cartographie des ressources.....	135
IX. Configuration de la mémoire.....	135
IX.1. Introduction.....	135
IX.2. Génération des groupes.....	137
IX.2.1. Critères d'évaluation.....	138
IX.2.2. Formulation mathématique du problème de la configuration.....	141
IX.2.3. Algorithmes heuristiques de reconfiguration.....	142
IX.2.4. Simulation.....	149
IX.3. Configuration physique de la mémoire WSI.....	151

IX.3.1. Configuration du chemin de données.....	151
IX.3.2. Configuration des décodeurs de chaque groupe.....	152
IX.3.3. Programmation des portes de transfert.....	153
X. Test de la mémoire 4.5Mbits à basse fréquence.....	153
XI. Reconfiguration et test de la mémoire 4.5Mbits.....	156
XII. Epaisseur en cuivre.....	156
XIII. Capacité de découplage.....	156
XIV. Test final sur la tranche de la mémoire WSI	157
XV. Encapsulation.....	158
XVI. Test final en boîtier de la mémoire WSI.....	158
CONCLUSION	159
ANNEXES	163
Annexe 2.1 : Différentes étapes de réalisation d'un circuit en "HCMOS3".....	165
Annexe 2.2 : Chronogramme de lecture de la mémoire SRAM	166
Annexe 2.3 : Chronogramme d'écriture de la mémoire SRAM 64K.....	167
Annexe 2.4 : Programme de calcul du rendement.....	168

Annexe 2.5 : Table des valeurs des figures 2.5, 2.6a et 2.6b.....	172
Annexe 3.1 : Dispositifs situés en haut d'une mémoire 64k.....	175
Annexe 3.2 : Dispositifs situés en bas d'une mémoire 64k.....	177
Annexe 3.3 : Résultats de simulation.....	179
Annexe 3.4 : Décodeur de rangée.....	180
Annexe 3.5 : Résultats de la simulation.....	181
Annexe 3.6 : Programme de placement des réticules.....	182
Annexe 3.7 : Réseaux d'adresses.....	184
Annexe 3.8 : Résultats de la simulation en phase de test.....	187
Annexe 3.9 : Résultats de la simulation en phase finale.....	188
Annexe 3.10 : Une ligne de données en lecture pendant la phase de test...	189
Annexe 3.11 : Résultats de la simulation en phase de test.....	191
Annexe 3.12 : Résultats de la simulation en phase finale.....	192
Annexe 3.13 : Une ligne de données amplifiée en lecture pendant la phase de test.....	193
Annexe 3.14 : Résultats de la simulation en phase de test.....	195

Annexe 3.15 : Réseaux d'alimentation.....	196
Annexe 3.16 : Résultats de la simulation en phase de test.....	200
Annexe 3.17 : Résultats de la simulation en phase finale.....	201
Annexe 3.18 : dessin au μ d'un réticule.....	202
Annexe 3.19 : Implantation sur la tranche de la mémoire 4.5Mbits.....	203
REFERENCES BIBLIOGRAPHIQUES.....	205
TABLE DES MATIERES.....	215



A U T O R I S A T I O N de S O U T E N A N C E

VU les dispositions de l'article 15 Titre III de l'arrêté du 5 juillet 1984 relatif aux études doctorales

VU les rapports de présentation de Messieurs

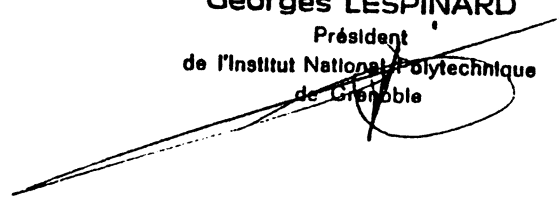
.C. LANDRAULT, Directeur de recherche
. F. JUTAND, Professeur

Monsieur NASREDDINE Bassam

est autorisé(e) à présenter une thèse en soutenance en vue de l'obtention du diplôme de DOCTEUR de L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE, spécialité "Microélectronique"

Fait à Grenoble, le 21 juin 1988

Georges LESPINARD
Président
de l'Institut National Polytechnique
de Grenoble





Résumé

L'objet de cette étude est la conception d'une mémoire statique intégrée sur tranche (4.5 Mbits). Cette mémoire réalisée à partir de cellules de 64 kbits est tolérante aux défauts de fin de fabrication. Des éléments en réserve remplacent les éléments défectueux à l'aide de connexions du type grille flottante FET ou fusible et antifusible laser.

Le travail de recherche a consisté :

- à étudier la faisabilité de cette mémoire
- à définir l'architecture d'une telle mémoire en tenant compte du rendement
- à définir une stratégie de test pour chaque circuit mémoire 64 Kbits et pour leur ensemble, de façon à déterminer une cartographie des éléments non défectueux et des interconnexions utilisables permettant de réaliser la mémoire cible de 4.5 Mbits
- à étudier les dispositifs de connexion/déconnexion (fusibles et antifusibles, grille flottante FET) qui permettront de réaliser physiquement la mémoire finale
- à développer des algorithmes de configuration qui détermineront, partant de la cartographie précédente, les groupements de cellules en paquets de 256 kbits.

Ce travail a été réalisé dans le cadre du projet ESPRIT-824. Un premier essai de fabrication a permis de caractériser les dispositifs de connexion. La mémoire de 4.5 Mbits a été envoyée en fabrication fin 1987.

Mots clés : Mémoire intégrée sur tranche, WSI, tolérance aux défauts de fabrication, test, haut rendement.