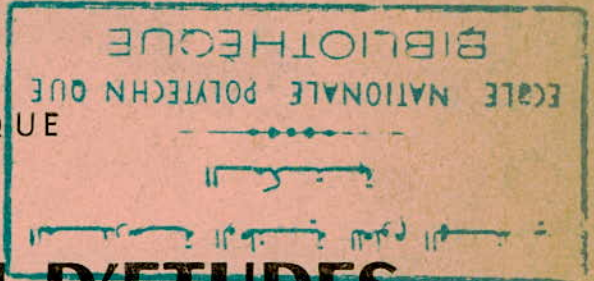


ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : ELECTRONIQUE



PROJET DE FIN D'ETUDES

SUJET

METHODES DE TEST

D'UNE FAMILLE DE MICRO - ORDINATEUR

Proposé par :

Mme B.KAMINSKA

Etudié par :

BOUTRI A Mohamed
SOUFI Messaoud

Dirigé par :

Mme B.KAMINSKA



PROMOTION :

JANVIER 1985.

أهلنا

فيه خلاصة عمل بذات فيه جهداً لا أجد
له أجراً سوى رضى كل من أبى وأمى علي. فإليهما
أهديه لعلهما يجزيانني عليه.

أبتكما البار : سويف مسعود

DÉDICACES :

je dédie ce modeste travail que je considère
un des fruits d'une vingtaine d'années d'études

à :

- Mon père
- Ma mère
- Mes frères : Boualem et Karim
- Mes Sœurs
- Et à tous ceux qui luttent pour
que règne La vérité sur le fanatisme.

M. BOUTRIA

REMERCIEMENTS

Au terme de nos études, nous voulons exprimer nos remerciements et nos sentiments de respect pour tous ceux qui ont participé à notre formation.

Nous tenons à remercier, particulièrement M^{me} B. KAMINSKA, pour le grand effort qu'elle a fourni pour nous diriger dans l'étude de ce sujet.

Nous remercions aussi M^l H. MEFTAH, pour l'aide qu'il a apporté à la préparation matérielle de ce polycopié.

SOMMAIRE

| | |
|---|----|
| Introduction | 1 |
| Chapitre I | |
| Spécification du problème | 3 |
| Chapitre II | |
| Fautes et Tests | 9 |
| II. A. Les fautes dans les circuits intégrés | 9 |
| . 1. Fautes technologiques | 9 |
| . 2. Fautes électriques | 9 |
| . 3. Fautes logiques | 10 |
| II. B. Les Modèles de Fautes | 11 |
| . 1. Fautes de collages | 11 |
| . 2. Les courts-circuits | 12 |
| . 3. Fautes dynamiques | 13 |
| II. C. Les Fautes spécifiques aux mémoires | 14 |
| II. D. Les Fautes spécifiques aux microprocesseurs | 15 |
| II. E. Classification des tests | 17 |
| . 1. Les tests paramétriques | 17 |
| . 2. Les tests fonctionnels | 17 |
| . 3. Les tests de caractérisation | 18 |
| Chapitre III | |
| Test des mémoires | 22 |
| III. 1. Présentation des mémoires à semi-conducteurs. | 22 |

| | |
|--|-----|
| . a. Introduction. | 22 |
| . b. Point mémoire statique | 24 |
| . c. Point mémoire dynamique. | 25 |
| . d. Paramètres caractéristiques des mémoires. | 26 |
| III. 2. Méthodes de test des mémoires | 29 |
| . a. Méthode de parité | 29 |
| . b. Méthode des graphes | 33. |
| . c. Méthode des séquences de mots. | 35. |

Chapitre IV.

| | |
|--|----|
| Test des microprocesseurs | 48 |
| IV. 1. Présentation des microprocesseurs | 48 |
| . a. Définition des microprocesseurs | 48 |
| . b. Classification des microprocesseurs | 48 |
| . c. Structure des microprocesseurs | 49 |
| IV. 2. Stratégies et méthodes de test | 53 |
| . a. Stratégies de test | 53 |
| . b. Méthodes de vérification | 54 |
| . c. Méthodes de test | 55 |
| . d. Techniques de test | 58 |

Conclusion 74.

Bibliographie.

INTRODUCTION

Les micro-ordinateurs ont pris la part du lion dans l'industrie des semi-conducteurs, dès leur jeune âge en 1971, et ceci est dû à leurs applications dans l'automatique, la commande industrielle, l'instrumentation scientifique et technique et bien d'autres domaines.

La structure générale d'un micro-ordinateur est composée essentiellement : d'un microprocesseur (Unité centrale), d'une RAM (mémoire vive ou mémoire à lecture et écriture), d'une ROM (mémoire morte ou à lecture seulement) et des unités de liaison d'entrées et de sorties, reliés entre eux par une structure de communication de type BUS.

Dans la recherche du maximum de fiabilité, les constructeurs ainsi que les utilisateurs, se confrontent à de nombreux nouveaux problèmes en matière de test.

On a constaté qu'il n'existe pas à l'heure actuelle de méthodes donnant satisfaction pour tous les aspects du problème. Depuis quelques années, on a assisté à de nombreux efforts visant à adapter des matériels et des techniques de test, permettant la résolution de ce nouveau problème.

Dans ce qui suit nous allons exposer les spécifications du problème de test des circuits intégrés (LSI et VLSI) entrants dans la constitution des micro-ordinateurs.

Nous donnerons par la suite quelques méthodes utilisées pour le test de ce type de circuit.

CHAPITRE

I

SPECIFICATION du PROBLEME

Avant d'étudier les méthodes utilisables pour tester les circuits constituant les micro-ordinateurs, il est nécessaire de rappeler les aspects spécifiques à ce type de circuits, qui sont responsables de difficultés nouvelles apportées au problème de test.

Dans le développement des calculateurs et des équipements de traitement de données, le nombre de circuits logiques et de mémorisation, croît exponentiellement.

Traditionnellement on distingue plusieurs degrés de complexité dans les circuits intégrés :

SSI : (Small Scale Integration) où le circuit contient moins de cent composants.

MSI : (Medium Scale Integration) où le circuit regroupe cent à mille composants.

LSI : (Large Scale Integration) où mille à dix mille composants sont intégrés dans la puce.

VLSI : (Very Large Scale Integration) plus de dix mille composants sont dans la puce.

Ainsi avec la technologie VLSI, qui a permis un maximum de miniaturisation, les systèmes sont devenus très complexes.

Avec les circuits intégrés de faibles densités, le test des assemblages était essentiellement un test des broches des circuits, où on mesurait les tensions et on observait la forme des signaux.

Avec le progrès de la miniaturisation, il devient de plus en plus difficile d'accéder individuellement

aux éléments constituant le circuit intégré, à partir des broches de ce circuit. En effet le rapport du nombre des points d'accès, où l'action du circuit peut être testée et observée, au nombre de circuits internes, est d'environ, un point d'accès pour vingt circuits, dans la technologie VLSI. C'est ce qui a rendu justement le test dépendant des méthodes automatiques de génération des vecteurs-test.

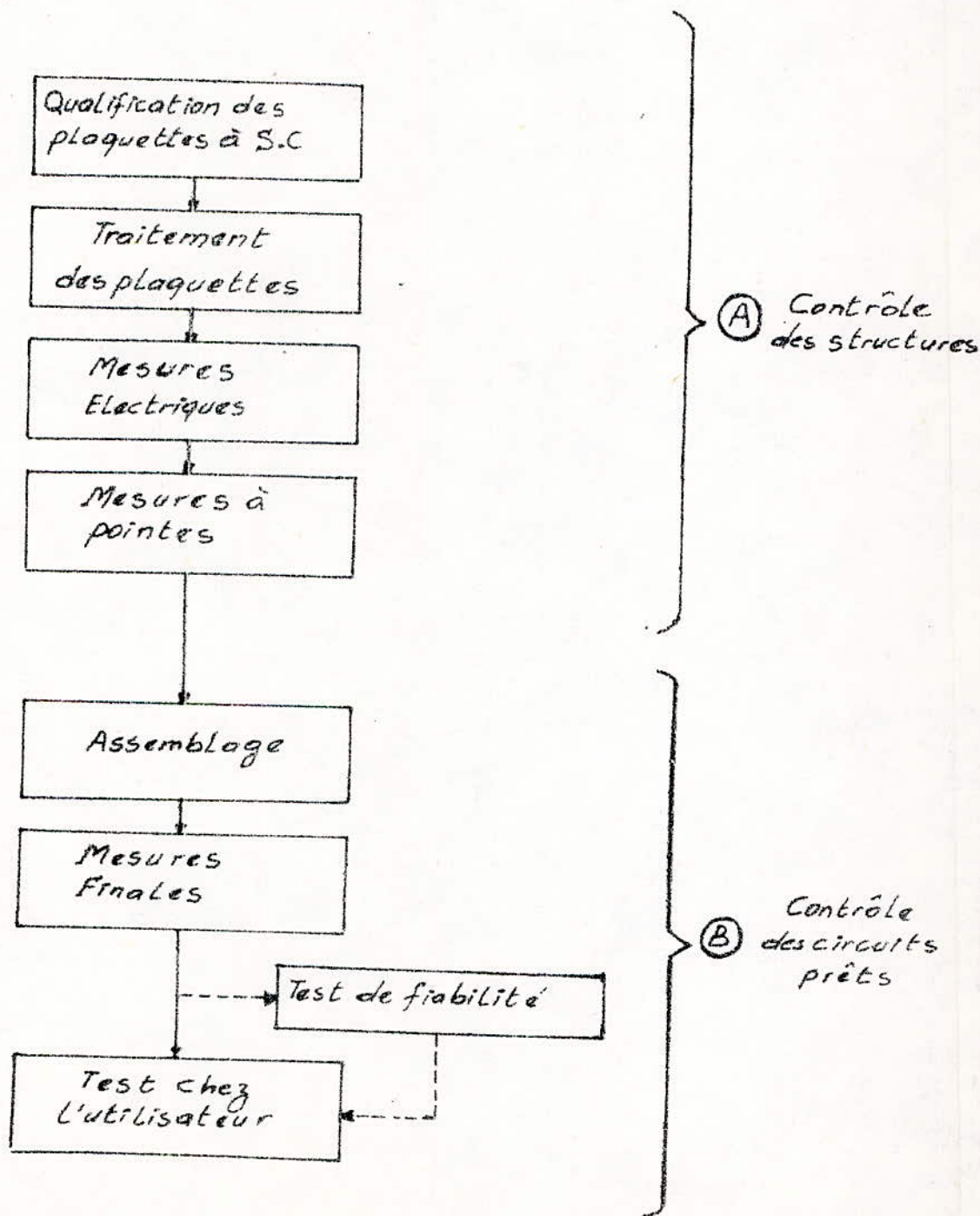
D'autre part, le test des produits de structures simples ou régulières, pouvant même contenir un nombre important d'éléments actifs (MSI), peut s'effectuer sans trop de difficultés avec des testeurs de semi-conducteurs. Ces équipements de plus en plus rapides envoient et contrôlent des tables de vérité, enregistrées dans leur mémoire, ou des "Patterns" définis par un algorithme. L'opération de test était confiée à des groupes de techniciens qui utilisaient donc, des équipements de laboratoire simples, sans l'intervention du fabricant du circuit.

Ces méthodes de test applicables aux circuits SSI et à la plus part des circuits MSI, sont insuffisantes pour les circuits complexes LSI et VLSI. La difficulté de test des circuits intégrés augmente avec l'agrandissement de l'échelle d'intégration et la complexité fonctionnelle.

De nouvelles approches sont donc nécessaires, et dans lesquelles, le fabricant est en partie responsable de la testabilité de ses circuits. Ainsi en plus de son souci de la fonction de ses circuits, il

doit se préoccuper aussi de leur structures, car il doit les concevoir de façon à ce qu'ils soient testables.

Considérons le diagramme simplifié et typique du processus de fabrication des circuits intégrés et des tests qui l'accompagnent, présenté par la figure suivante.



Dans ce processus, le test se fait en deux phases :

- Ⓐ Les tests de contrôle pendant la création des structures.
- Ⓑ Les tests des circuits prêts.

Ⓐ Le circuit intégré est fabriqué à partir d'une plaquette de silicium (de type P par exemple) appelée "substrat" de diamètre 3 cm et d'épaisseur 0,25 mm, que l'on obtient à partir de lingots cylindriques de silicium monocristalin pur.

Les paramètres des matériaux semi-conducteurs sont mesurés sur les plaquettes vierges, ainsi qu'après chaque étape du processus de fabrication, puis sur les structures finies.

Les plaquettes sont ensuite traitées en deux étapes : l'étape du projet de construction (schématisation du circuit) et l'étape technologique (Processus de fabrication).

Le projet de construction est typique pour tout les circuits intégrés numériques, seulement la complexité du schéma augmente avec la complexité de la fonction du circuit à réaliser. Les fonctions de grande complexité exigent l'utilisation de la simulation des circuits, avec l'emploi des grands ordinateurs ayant une large capacité de traitement.

Le projet technologique consiste à réaliser les masques nécessaires pour faire les couches constitutives du circuit. Puis réaliser les différentes couches par l'introduction de quantités convenables d'impuretés, créant ainsi les jonctions nécessaires à

la réalisation de la fonction logique désirée.

Après le processus technologique de fabrication, on effectue les mesures électriques préliminaires. On place sur les plaquettes de semi-conducteur, les circuits de contrôle permettant l'évaluation des paramètres caractéristiques critiques pour chaque technologie utilisée (bipolaire, MOS, ...).

Les plaquettes, pour lesquelles les valeurs des paramètres mesurés ne sont pas dans les limites admises, sont éliminées. Ceux qui restent sont soumis aux mesures de pointes.

Pendant les mesures de pointes, on teste chaque structure se trouvant sur la plaquette. Les bonnes structures sont repérées pour être retenues.

La plaquette est ensuite découpée et les structures retenues sont alors montées dans des boîtiers en plastique ou céramiques.

On passe ensuite à la deuxième phase de test.

③ C'est la phase des tests finaux, et qui sont les plus importants car ils assurent la vérification de tout les paramètres du circuit et son fonctionnement correct.

Certains circuits sont soumis à des tests de fiabilité spéciaux, qui dépendent du type de circuit, du boîtier, et des conditions d'utilisation du produit.

Les tests faits par l'utilisateur ont un caractère particulier, du fait qu'ils vérifient le bon fonctionnement du circuit dans les conditions particulières de son utilisation.

CHAPITRE

III

FAUTES et TESTS

A) Les fautes dans les circuits intégrés :

La faute est une condition physique qui cause une panne ou une erreur quand le circuit logique est utilisé pour assurer une fonction donnée.

L'histoire, de développement des semi-conducteur, montre que les pannes statiques résultent généralement d'un court-circuit, d'une absence ou une insuffisance d'alimentation.

Cependant un autre grand nombre de fautes existent et restent encore, inconnues.

Les fautes sont généralement classées en trois catégories :

1) Fautes technologiques :

Elles sont dues à la technologie choisie par le constructeur pour réaliser ses circuits. En effet les technologies de réalisation des circuits intégrés telles que la bipolaire, MOS, et leur variantes, ont chacune ses inconvénients qui causent ce genre de fautes dites : technologiques.

2) Fautes électriques :

Ce sont celles qui se manifestent par un changement des spécifications du circuit et qui sont dues soit à des erreurs de fabrication soit à un vieillissement des composants. L'exemple le plus courant de ce type de fautes est le court-circuit.

3) Fautes logiques :

Ce type de fautes, cause l'erreur dans la fonction demandée au circuit, qui par conséquent exécutera une tâche autre que celle pour laquelle il est conçu. Ces fautes sont généralement faites au niveau de la conception du circuit, à l'étude ou à la réalisation.

Le classement des fautes, que nous venons de faire, reste insuffisant pour l'élaboration de tests spéciaux pour chaque type de fautes. C'est pourquoi, une modélisation des fautes s'impose.

Rappelons que la modélisation consiste à représenter un ensemble de fautes, ayant la même cause ou le même effet, par une seule faute.

Ce qui permet de préparer un test pour chaque modèle de fautes. Ainsi, en appliquant un test sur un circuit, on peut déterminer la faute à partir du modèle relatif au test appliqué.

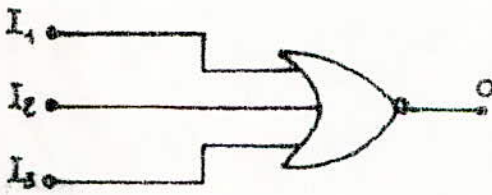
B) Les modèles de fautes :

Les fautes de mécanismes les plus souvent observées dans les réseaux logiques LSI sont : les chutes de tension, les surtensions, les coupures des chemins des signaux, la déficience des portes logiques et les excès de retards entre les signaux.

Ces fautes se présentent généralement sous une des formes données par les trois modèles suivants :

1) Faute de collage à une valeur logique :

Dans ce cas, la sortie, ou l'entrée, d'une porte est toujours collée à une valeur logique fixe : 1 ou 0. La figure suivante illustre ce cas en prenant, comme exemple la porte NOR et dont la sortie est collée à 1.

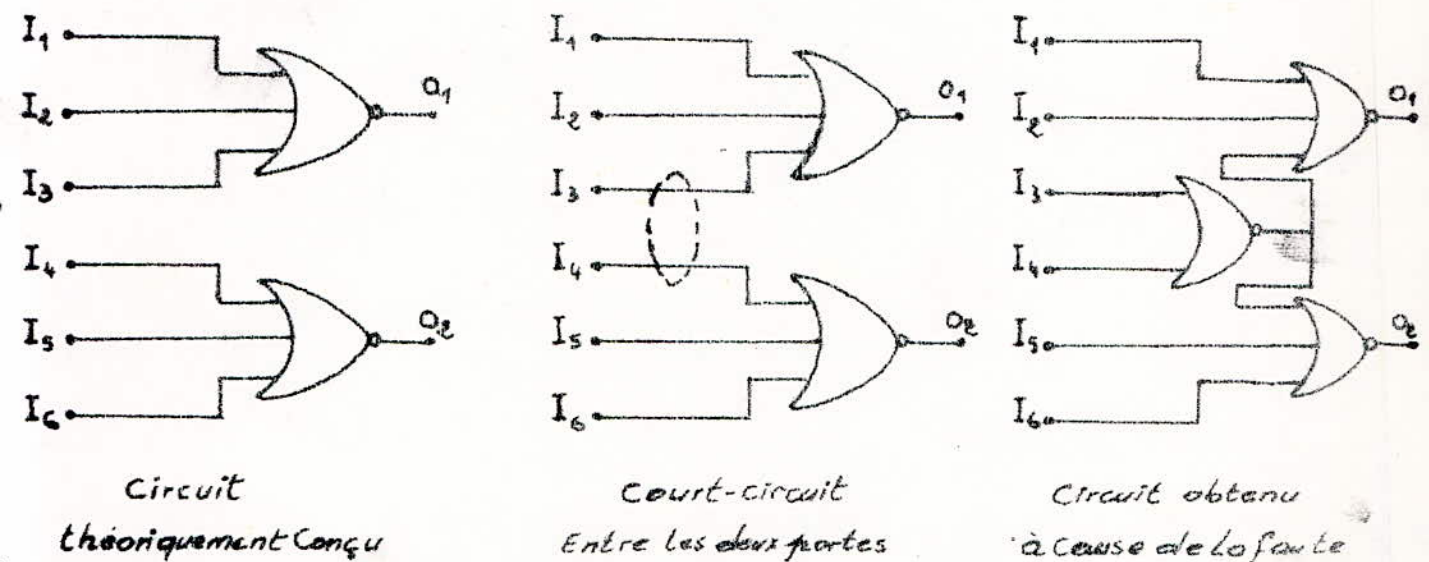


| Entrées | | | Sortie | |
|---------|-------|-------|--------|------------------|
| I_1 | I_2 | I_3 | | |
| 1 | 0 | 0 | 0 | Sortie théorique |
| 1 | 0 | 0 | 1 | Sortie pratique |

Fig. II.1 : Porte NOR collée à 1.

2) Les courts-circuits.

Se sont des courts-circuits entre les lignes des différents signaux dans un réseau logique, ce qui provoque la modification de la fonction du réseau. La figure suivante illustre un court-circuit entre les entrées de deux portes NOR.



| Entrées | | | | | | Sorties | | |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--|
| I ₁ | I ₂ | I ₃ | I ₄ | I ₅ | I ₆ | O ₁ | O ₂ | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | Etat théorique des E/S |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | Etat obtenu aux E/S à cause de la faute |

Fig. I.2 : Circuit logique, présentant un court-circuit aux entrées.

3) Fautes dynamiques.

Ce sont généralement toutes les fautes relatives aux délais des circuits qui sont en dehors des performances spécifiques. C'est par exemple le cas des portes lentes, qui causent des aléas de décalage des signaux dans les circuits logiques.

La figure suivante montre un montage réalisé avec deux portes AND et deux bascules D et dans lequel il existe un aléa de décalage d'horloge pour la deuxième bascule. Le montage ne fonctionnera pas.

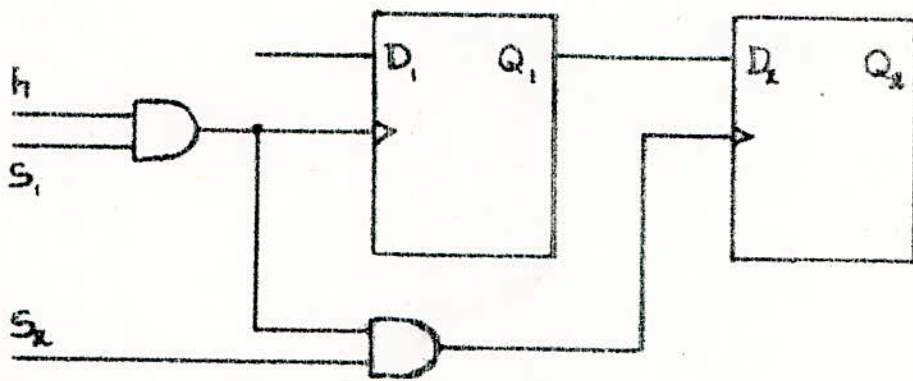


Fig II.3. Circuit logique, présentant un défaut de synchronisation.

En effet le catalogue donne les valeurs suivantes des temps de propagation :

t_p bascule 7474 : 10 ns minimum

t_p porte 7408 : 17,5 ns typique; 27 ns maximum.

Le décalage d'horloge dans le montage est égal au retard de la porte calculant s_1, h, s_2 soit 27 ns maximum. Ce retard dépasse le temps de basculement minimum de la première bascule et l'horloge de la seconde bascule arrive trop tard.

c) Les fautes spécifiques aux mémoires:

Les fautes les plus souvent rencontrées dans le test des mémoires sont les suivantes:

1. Court-circuit et circuit-ouvert:
liaison parasite ou mauvais contact
2. Claquage de jonction:
Diode, ou tout autre jonction à semi-conducteur, claquée.
3. Fuite à l'entrée ou à la sortie:
Fuite de courant, à l'entrée ou à la sortie du circuit, par rapport à sa valeur spécifique.
4. Décodeur défaillant:
Mauvais adressage dû à la faille du décodeur.
5. Ecriture multiple:
Une donnée, adressée à une seule cellule, mais se trouve mémorisée, aussi, dans d'autres cellules.
6. Sensibilité aux vecteurs de test:
Le dispositif ne répond pas convenablement aux excitations de test d'écriture ou de lecture ou d'adressage.
7. Rafraichissement défaillant (volatilité):
Perte d'information, durant le temps de rafraichissement spécifique, dans les RAM dynamiques.

B. Longueur du temps d'accès :

L'acquisition des données par la mémoire ne s'effectue pas dans les délais prévus.

9. Recouvrement de l'écriture :

Quand la lecture suit immédiatement l'écriture le temps d'accès devient long.

10. Sensibilité de l'amplificateur :

Quand une même donnée accède pour un nombre de fois, puis soudain elle change, l'amplificateur tend à rester dans le même état, il devient insensible au changement de la donnée.

D) Les fautes spécifiques aux micro-processeurs

Les micro-processeurs étant des circuits intégrés très complexes (LSI et VLSI), les défauts des mémoires, tels que les courts-circuits, les circuits-ouverts, et les défaillances de décodeurs, existent certainement dans ce type de circuits ; cependant, et à cause de leur nouveauté, ces circuits présentent d'autres modèles de fautes et qui restent encore inconnus.

Parmi les anomalies qu'on peut rencontrer lors de la vérification des micro-processeurs, on note les anomalies suivantes :-

1. Defaillance de registres:
Si par exemple le registre compteur ordinal est defaillant, des sauts imprevis seront exécutés durant le déroulement du programme.
2. Defaillance du séquenceur:
Dans l'unité de commande, si le séquenceur est le siège d'une anomalie, des micro-instructions imprevises seront générées.
3. Défaut de synchronisation:
L'arrivée, du signal de validation d'écriture, après les données, dans une opération de stockage par exemple, serait la cause de beaucoup de problèmes.
4. Sensibilité du circuit
L'expérience a démontré l'existence de séquences d'instructions et de données pour lesquelles le micro-processeur est insensible.

E) Classification des tests :

Dans les essais des circuits intégrés numériques, on distingue trois types de tests :

1. Les tests paramétriques :

Ces tests jouent un double rôle :

- a. Ils vérifient si le circuit réalise, pour les paramètres, les valeurs données par le constructeur (niveaux des tensions, courants, délais, ...).
- b. Ils permettent l'évaluation rapide de certains processus techniques au niveau du projet de construction. Par exemple les valeurs incorrectes des tensions de sorties du circuit indiquent l'irrégularité du processus d'implantation d'ions.

2. Les tests fonctionnels :

- a. Les tests fonctionnels simples (faible fréquence)
Ils ont pour rôle de détecter rapidement les défauts grossiers, l'existence et le fonctionnement des lignes de communication internes du circuit.
- b. Les tests fonctionnels plus complets
Ce sont des mesures qui se font sur les paramètres dynamiques et qui sont callées sur les spécifications du circuit.

La préparation de la séquence de test, repose sur la connaissance de l'architecture interne du circuit.

En premier lieu, on teste les lignes de communication entre modules, puis les modules un par un.

Le test fonctionnel complet est découpé en plusieurs étapes. Chaque étape est répétée plusieurs fois dans chacune d'elles on fait varier un paramètre, par exemple l'alimentation qu'on fait varier de sa valeur min à sa valeur max.

3. Les tests de caractérisation :

Ces tests sont effectués avec des séquences similaires à celles des tests fonctionnels complets mais en faisant varier les paramètres d'utilisation du circuit par groupe, de deux par exemple, ce qui permet de tracer le domaine de fonctionnement du circuit. On peut ainsi analyser l'influence des paramètres extérieurs sur le fonctionnement du circuit.

La figure II.4, suivante illustre le test de deux échantillons d'un même type de micro-processeur et pour lesquelles on a varié l'alimentation et la fréquence, traduite en période. Dans les deux figures, les rectangles définissent le domaine de fonctionnement donné par le constructeur. Le signe plus (+) indique un point de bon fonctionnement. Les points indiquent un fonctionnement incorrect.

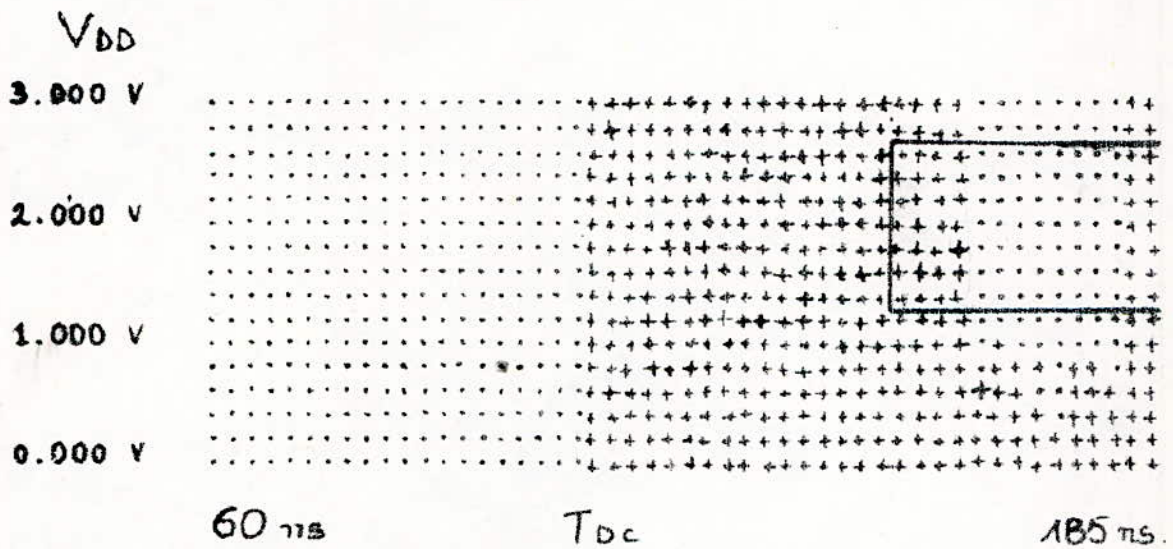
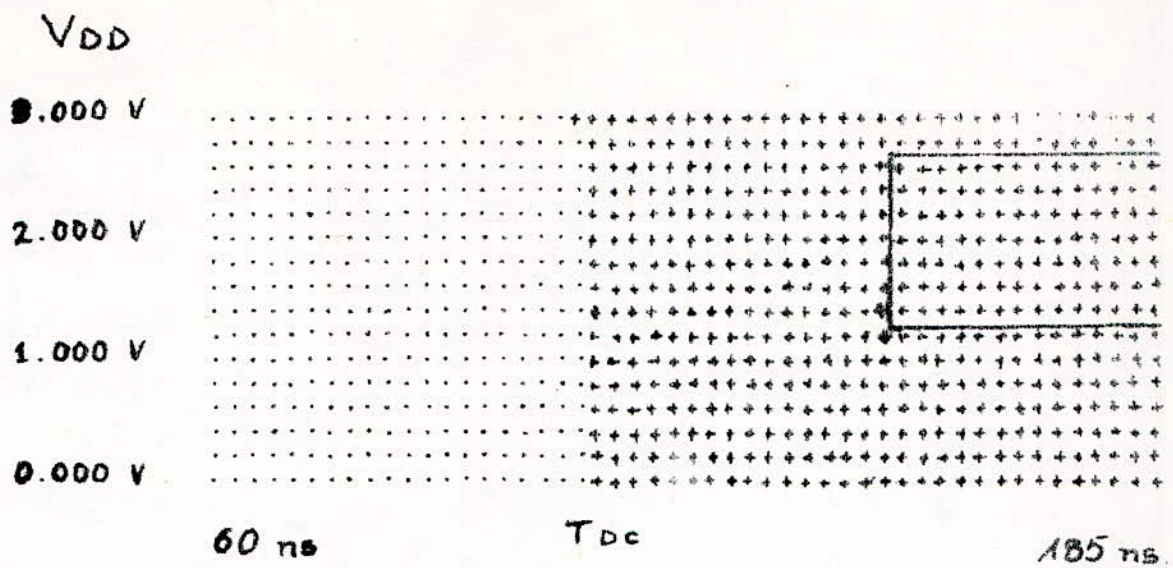


Fig II.4. Graphes de caractérisation de deux microprocesseurs du type INTEL 8080.

Bien qu'elle ne soit pas pour autant facile, l'exécution des tests paramétriques et dynamiques ne pose pas de problèmes fondamentaux. De plus, ces tests jouent un rôle insignifiant dans la détermination des performances du circuit, d'une façon satisfaisante.

Les tests fonctionnels, par contre, posent de sérieux problèmes. Les tests consistent à vérifier la table de vérité du circuit, ce qui s'effectue en appliquant aux entrées du circuit, un certain nombre de vecteurs de test, composés de "1" et de "0" que l'on désigne par les "Patterns" et l'on vérifie que les vecteurs de sortie correspondent bien à ceux prévus par la table de vérité.

Dans le cas d'un réseau logique combinatoire, la génération de tous les vecteurs d'entrées est relativement simple, ainsi que la vérification de la sortie car celle-ci est fonction seulement du Pattern présent aux entrées du circuit. Par contre lorsqu'il s'agit d'un réseau logique séquentiel, et c'est le cas de tous les circuits constituant le micro-ordinateur, le problème se complique car la sortie n'est pas seulement fonction du "Pattern" appliqué à un instant donné, mais aussi des "Patterns" appliqués auparavant. Ceci à cause des chaînes de retour existantes dans ce type de circuit.

Pour un circuit logique de vingt entrées, 2^{20} Patterns sont nécessaires pour son test, s'il est combinatoire. S'il est séquentiel 2^{20+r} sont nécessaires, avec r le nombre de lignes des chaînes de retour dans le circuit.

CHAPITRE

III

TEST des MEMOIRES

1. Présentation des mémoires à semi-conducteurs.

a. Introduction :

Avant de présenter les méthodes de test des circuits de mémorisation à semi-conducteur, nous allons rappeler les aspects spécifiques de ce type de circuits, et exposer brièvement leur architectures.

L'organisation de base d'un circuit de mémorisation est celle donnée par la figure suivante.

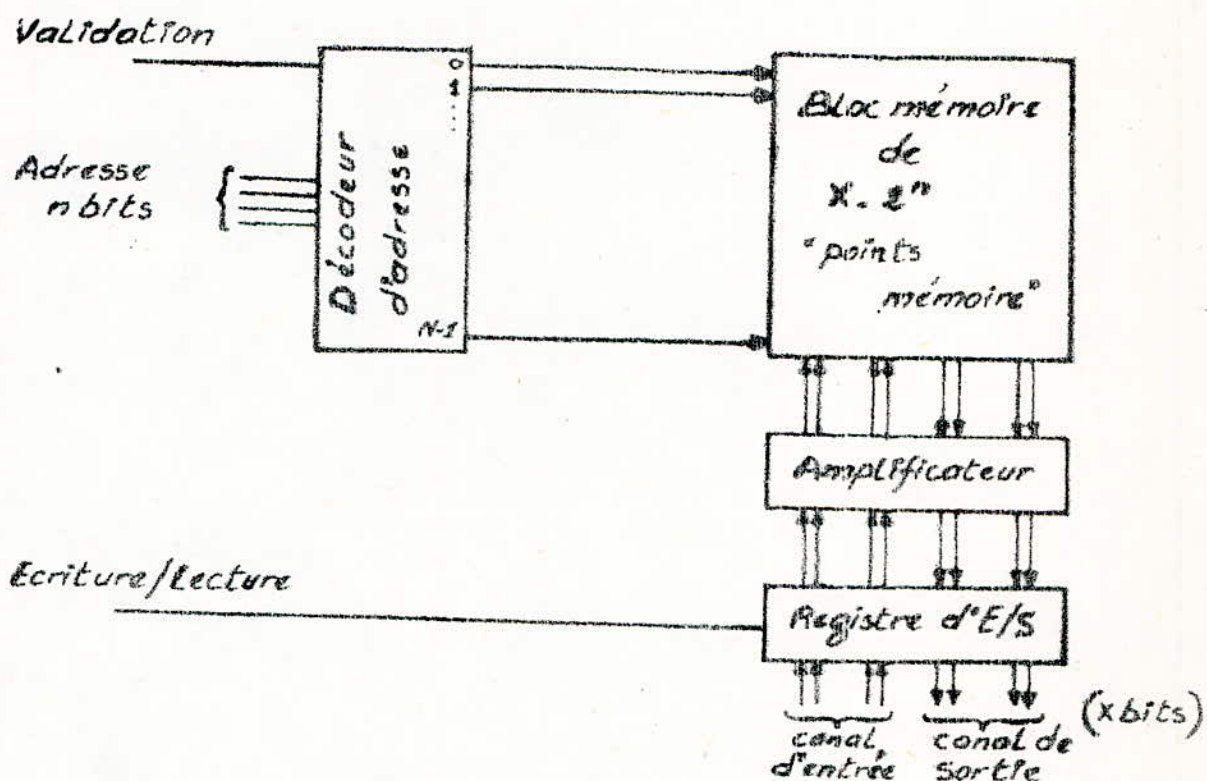


Figure III. A. Vue d'ensemble d'une mémoire à lecture/écriture de 2^n mots de x bits.

On distingue dans ce schéma les organes suivants :

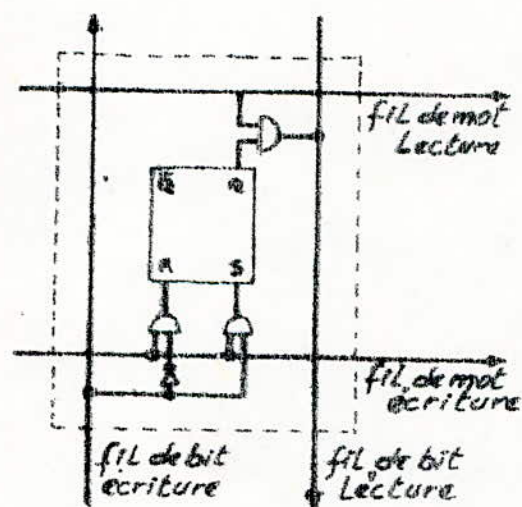
- Un decodeur d'adresse (m bits)
- Un canal d'entrée (x bits)
- Un canal de sortie (x bits)
- Une commande lecture / écriture, ou éventuellement deux commandes séparées.
- Une entrée "selection du module" ou entrée de validation.
- Le bloc mémoire proprement dit constitué de $x \cdot 2^m$ éléments de mémoire.
- Une entrée horloge pour les mémoires à rafraichissements
- Un amplificateur des signaux entrants et sortants du bloc mémoire.

b. Point mémoire statique:

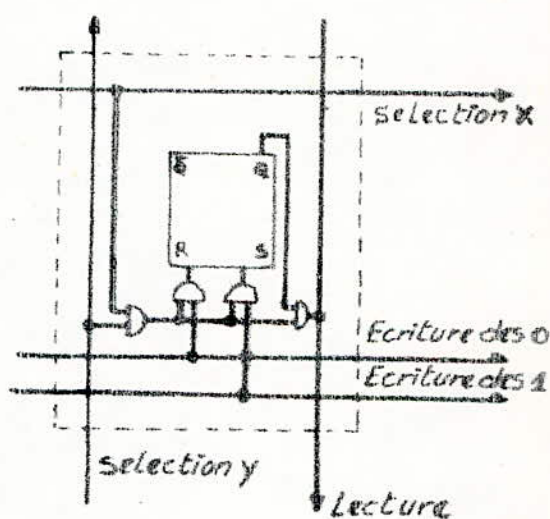
Dans les mémoires statiques, le point mémoire le plus élémentaire est le plus simple des circuits logiques séquentiels, appelé bascule ou FLIP-FLOP. Son rôle consiste à noter la présence d'une information fugitive et à conserver cet état lorsque l'information en question disparaît.

La bascule RS joue formidablement ce rôle. En effet, cette bascule dispose de deux entrées R et S, et d'une sortie Q. Lorsque l'entrée R (Reset) reçoit un signal d'effacement de la mémoire, la sortie Q se met à 0. Lorsque l'entrée S (Set) reçoit l'information fugitive à mémoriser, la sortie Q passe à 1.

La figure III.2. représente un point d'une mémoire organisée en mots et de sélection linéaire.



a Sélection linéaire 2D



b Sélection par courants coïncidents 3D

figure III.2. Exemples de point mémoire dans des mémoires cellulaires à semi-conducteurs.

c. Point mémoire dynamique.

Dans un transistor MOS, il existe naturellement une capacité, même minimale, entre sa grille isolée et son canal. C'est essentiellement ce condensateur, que l'on peut charger, qui constitue le point mémoire des mémoires dynamiques et c'est son état chargé ou non qui représente l'information binaire.

Cet état présente quand même des fuites, de telle sorte que l'information disparaît, et il faut le rafraîchir périodiquement.

La figure III.3. montre un point mémoire dynamique et indique le chemin de la donnée à stocker ainsi que la commande des opérations d'écriture et de lecture. Des structures, plus complexes, à trois transistors MOS par point mémoire, sont aussi utilisées.

La figure III.4. donne un exemple de ce type de points-mémoire et qui a l'avantage d'incorporer l'amplification des signaux d'entrée et de sortie.

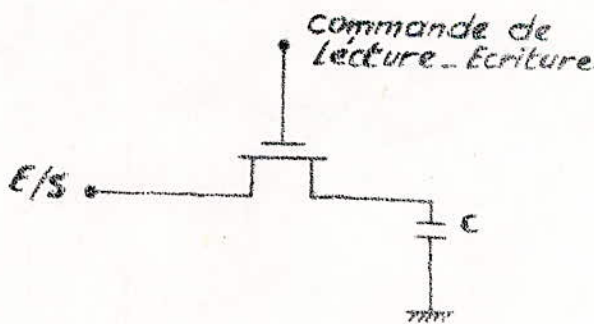


Figure III.3. Cellule à un transistor

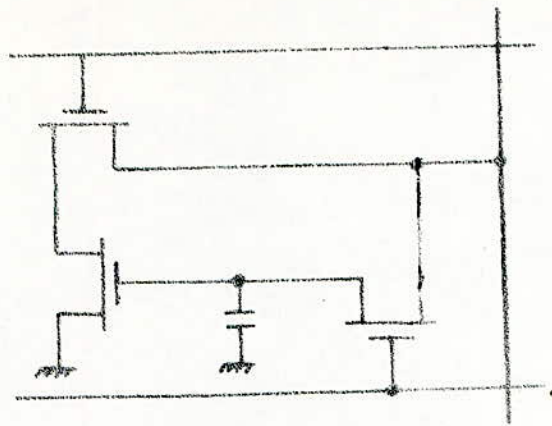


Figure III.4 Montage de base d'un point mémoire à trois transistors

d. Paramètres caractéristiques des mémoires:

Les mémoires à semi-conducteurs ont plusieurs paramètres qui les caractérisent et les différencient les uns des autres. Parmi ces paramètres, les plus importants à noter sont les suivants:

a) la capacité : qui est le nombre d'éléments binaires que peut stocker la mémoire.

En 1970 les utilisateurs disposaient d'un module circuit-intégré d'une capacité de 64 bits en technologie bipolaire et d'un module de 256 bits en technologie MOS. En début 1974 des éléments mémoire de 1024 bits (bipolaire ou MOS statique) et de 4096 bits (MOS dynamique) étaient couramment utilisés.

b/ La vitesse : elle est exprimée en termes temporels et donnée par les deux paramètres suivants :

Le temps d'accès : durée d'obtention d'une information après une demande de lecture. Il varie d'un circuit à un autre suivant la technologie utilisée et la capacité de mémorisation. Ainsi pour une RAM de 1024 bits (bipolaire) il est de l'ordre de 40 ns. Il est d'environ 450 ns pour une RAM de 4096 bits (MOS dynamique).

Le temps de cycle : durée minimum séparant l'application successive de deux adresses. Il donne la fréquence maximum de fonctionnement.

c/ La consommation : La consommation en puissance d'une mémoire dépend surtout de la technologie utilisée. Ainsi la bipolaire consomme une puissance de l'ordre de 2 mW/bit, tandis que la MOS dynamique a une consommation d'environ 0,5 mW/bit.

d/ Compatibilité avec la TTL.

La logique TTL (Transistor-Transistor-logic) est la plus utilisée et la plus diversifiée des familles logiques. Car elle présente beaucoup d'avantages pour l'utilisateur :

- Très bonnes caractéristiques dynamiques.
- Paramètres divers, très bien spécifiés.
- Gamme de circuits très étendue, (plus de deux cent fonctions).

- bons produits, faciles à utiliser
- sous-familles compatibles, avec des règles d'assemblage bien définies et simples
- Produits robustes.
- Bas prix.

Tous ces avantages font de la TTK, une référence de simplicité, et tous les constructeurs essaient de concevoir leur produits compatibles avec la TTK.

2. Methodes de test des memoires.

Tout, comme, chaque menage a ses propres recettes, chaque constructeur et utilisateur a sa propre façon de tester ses circuits de memorisation qui depend essentiellement des equipements qu'il a dans son laboratoire.

Cependant, chaque une des methodes utilisees, peut être classée dans une des trois strategies suivantes :-

a/ Methode de "parité"

Dans cette methode un bit de parité, dependant de l'information à stocker, est ajouté à cette information, dans l'operation d'écriture.

A la phase de lecture, on recalcule le bit de parité et on le compare avec le premier. Si la valeur du bit de parité n'a pas changé, on conclue que l'information stockée n'a pas été modifié et que le circuit de memorisation fonctionne convenablement. Sinon, il est defaillant.

La diversité du choix de l'information dont dependra le bit de parité, a donné naissance à plusieurs variantes de cette methode, nous en donnons quelques unes dans ce qui suit :-

1. Un bit de parité par ligne:

Dans les memoires organisées en matrice, le dernier bit de chaque ligne est réservé à la

parité. Ce bit est fonction des informations contenues dans tout les mots de cette ligne.

La figure III.5. suivante illustre cette variante avec un exemple d'une mémoire constituée de quatre modules mémoire de N mots de quatre bits. On trouve dans cette figure, le bit de parité P dans le dernier mot de chaque ligne.

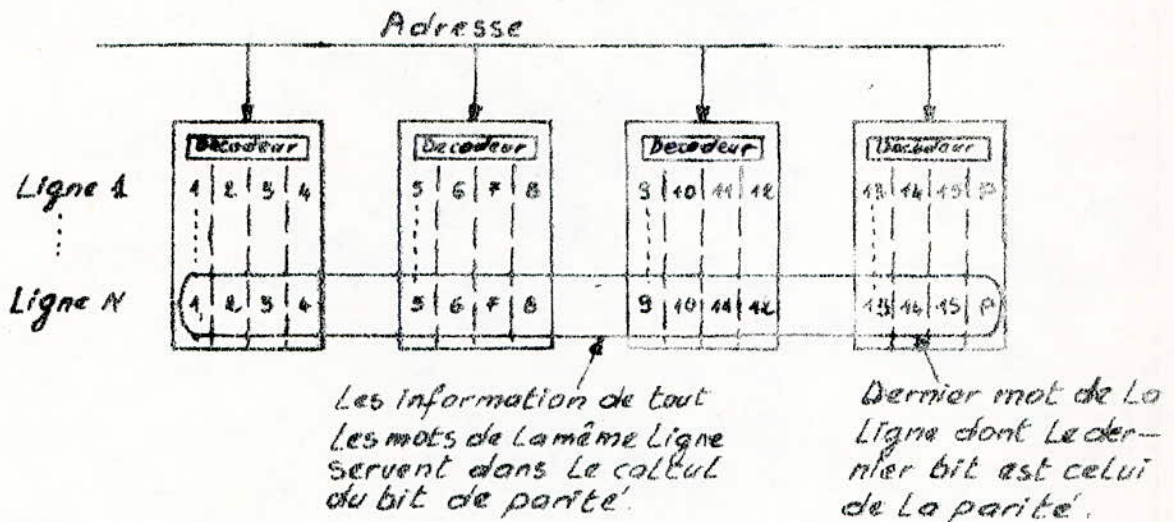


Figure III.5.
Un bit de parité par ligne

2. Un bit de parité par mot.

Dans cette variante, de la méthode de test par parité, le dernier bit de chaque mot est chargé par l'information de parité, fonction des informations contenues dans ce mot.

La figure III.6. suivante schématise cette variante sur le même exemple de la figure

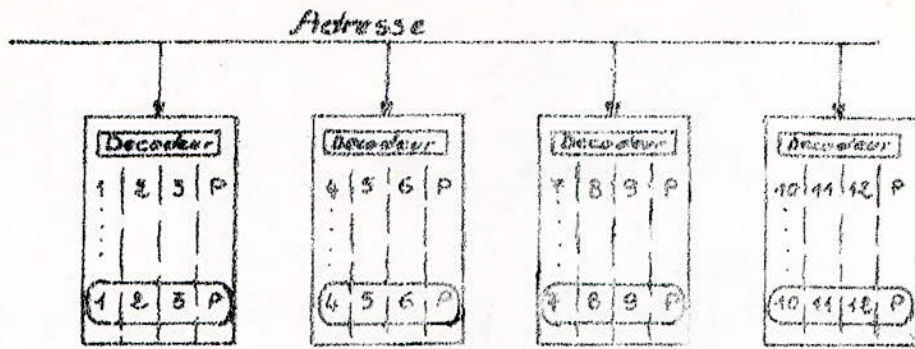


figure III.6
Un bit de parité par mot.

3. Un mot de parité par ligne:

A lieu d'utiliser un bit pour toute la ligne, cette variante variante, utilise un mot entier dont les bits sont la parité des bits de même poids dans tout les mots de la ligne. Ainsi le bit de poids i dans le mot de parité est la parité des bits de poids i dans tout les autres mots de la même ligne.

La figure III.7 suivante donne un exemple d'application de cette variante, appliqué à la mémoire des deux figures précédentes. Dans cette figure on trouve le mot de parité placé en premier et dans lequel le quatrième bit P entouré d'un cercle est la parité des bits 4, 8, 12 et 16 des quatre mots suivants de la même ligne.

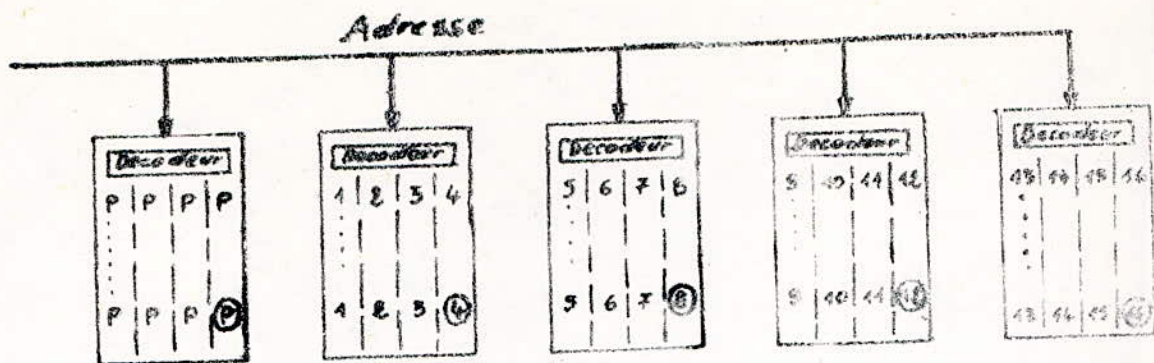


figure III.7. Un mot de parité par ligne.

b) Méthode des graphes.

Cette méthode est l'application des tests de caractérisation, car elle consiste à tracer les graphes qui lient les paramètres d'utilisation du circuit et permet de définir le domaine de fonctionnement du circuit pour des variations quasi-continues de ces paramètres.

Le nombre de tracés qu'il est possible de faire est très important, puisqu'on peut considérer un grand nombre de paramètres, tel que la tension d'alimentation, le temps d'accès, la fréquence de l'horloge, la température ambiante...

L'intérêt de ce type de tracés est multiple. En effet, ils permettent d'analyser l'influence des paramètres externes sur les performances du circuit.

Ils orientent le fabriquant dans l'amélioration de ses produits et de ses processus de fabrication.

Ce type de graphes permet aussi de déceler les composants marginaux, qui, même s'ils sont conformes aux spécifications, ce sont les premiers candidats aux défaillances futures.

La figure III.13. donne l'exemple de deux tracés pour deux échantillons d'un même type de mémoires et pour lesquelles on a fait varier la tension d'alimentation (VCC) (représentée avec une translation, pour des raisons pratiques), 5V correspond à la valeur repérée par la flèche. La tension de l'état logique "1" est représentée par E1.

Ces tracés ont été obtenus sur un matériel SENTRY II de Fairchild.

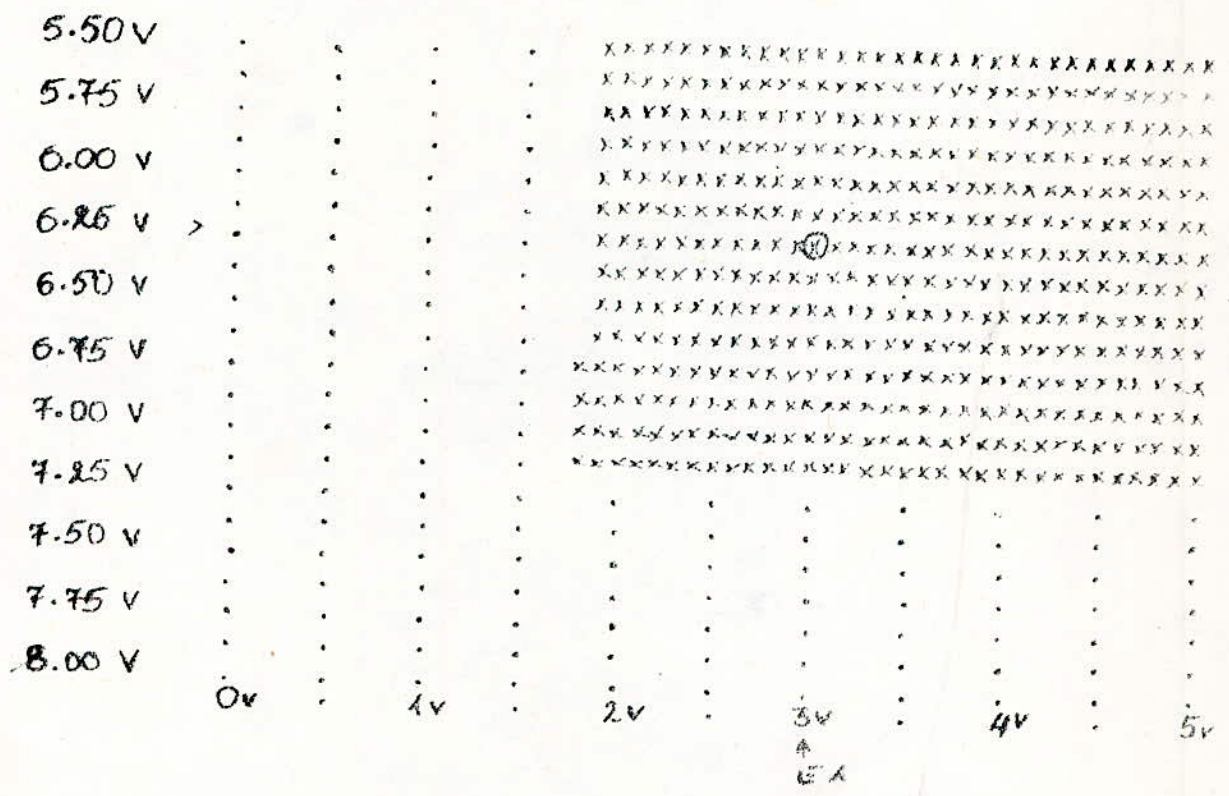
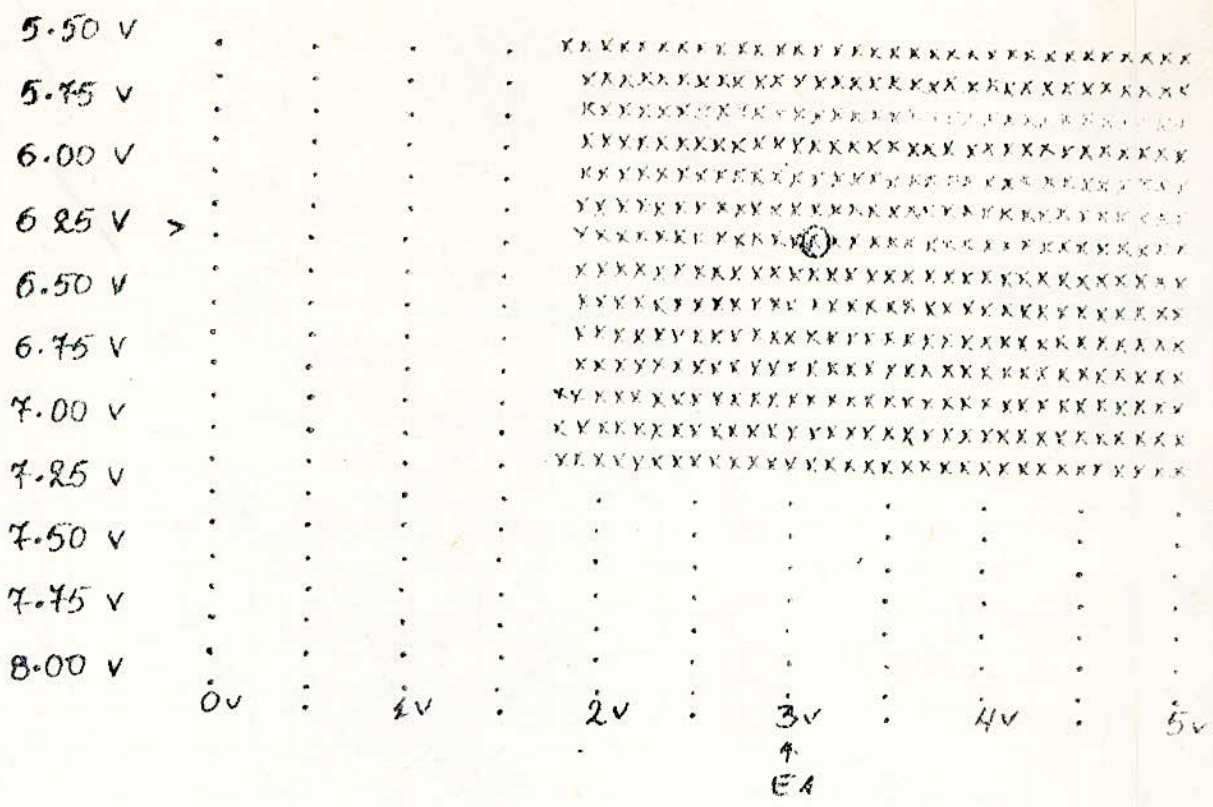


Fig. II.B. Graphes de caractérisation de deux échantillons d'un même type de mémoires.

c) Methode des sequences de mots:

Cette methode de test consiste à appliquer au circuit une sequence de mots et exercer ce circuit de toutes les facons possibles. Dans le cas des circuits de memorisation, ce sont des operations de lecture et d'écriture, qu'il faut appliquer et dont il faut controler les resultats pour juger de l'état du systeme.

Dans ce qui suit, nous allons presenter quelques techniques de test par cette methode et qui, motons le tout-de suite different les unes des autres par le temps necessaire à leur executions, l'organisation du circuit à tester et le sequencement des operations de lecture et d'écriture executees pour le test.

1. ENALTCOL:

Cette technique a été developpée à la suite de la realisation, sur une même puce, des RAM dynamiques de capacité 4K bits

Elle decèle les transitions d'adresses entre des cellules memoires dans une même ligne, sans modifications de tout les autres cellules dans cette ligne. Elle montre s'il ya des parasites dus aux voisinages des cellules les unes des autres dans une même colonne. Elle fait aussi le test de la sensibilité au rafraichissement de la RAM.

Cette technique de test est composé des instructions suivantes et qui sont illustres dans la figure III.9.

1. Initialisation de la mémoire à tester.
Chargement de tout les emplacements mémoires par des "0" (ou des "1").
2. Chargement de l'emplacement zéro par la valeur test: "1" (resp "0").
3. Lecture et exécution du test qui détecte la transition d'adresse entre l'emplacement contenant la valeur test et les autres emplacements de la même ligne.
Le test consiste à: Lire l'emp 0, l'emp 64, l'emp 0, l'emp 128, l'emp 0, l'emp 192, l'emp 0, ..., l'emp 4032, l'emp 0.
4. Effacement de la valeur test de l'emplacement 0 et reinitialisation de cet emplacement à "0" (resp "1").
5. Lecture du premier emplacement des deux lignes suivantes. (emp 1 et 2 pour la première phase).
6. Vérification de la non-écriture de la valeur test dans d'autres emplacements.
7. Si aucun emplacement ne contient la valeur test après son effacement, cette dernière est réécrite dans le premier emplacement de la séquence suivante (soit l'emplacement 3, pour la première phase).
8. On répète la suite d'instruction de 3 à 7 jusqu'à ce que la valeur test soit écrite dans tout les emplacements.
9. Le test est ensuite répété pour la valeur complémentaire de la première valeur test soit "1".

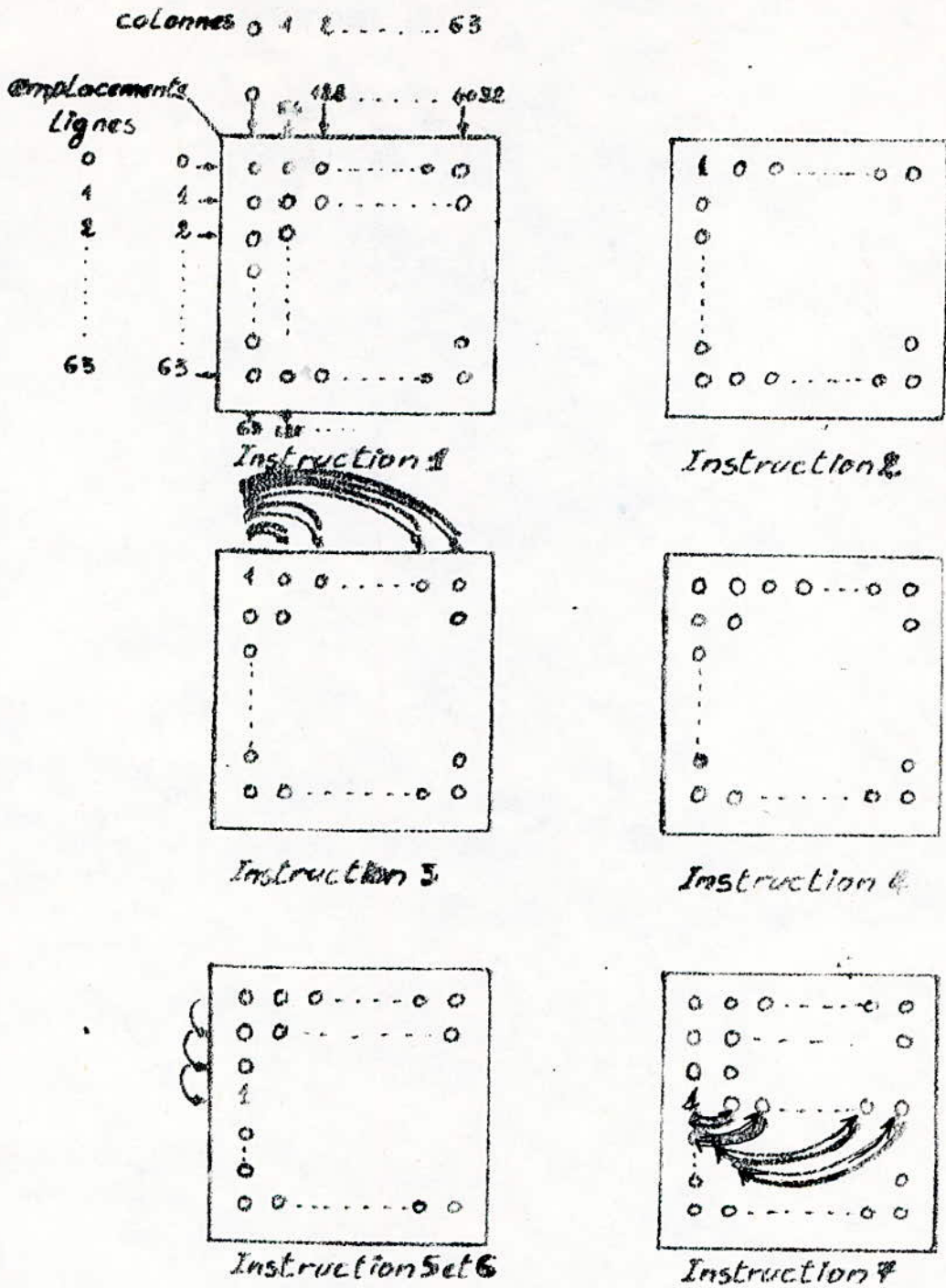


Figure III. 9
Schémas illustrants le contenu de chaque emplacement mémoire pour les instructions de ① à ⑦ pour le GALTEOL.

Pour rafraîchir la mémoire, chaque ligne doit être lue tout les 2ms et dans la technique de test GALTCOL l'instruction 3 maintient, durant son exécution, l'adresse de la ligne constante. Si le saut, de la valeur test, se fait d'une ligne à la ligne suivante, au lieu d'aller à la troisième, le temps que prendra le test de toutes les cellules, est donné par la formule:

$$3N \text{ cycles horloge}$$

où N est le nombre d'emplacements mémoire.

Pour une RAM de 4K bits, synchronisée avec une horloge de période 470ms, ce temps serait de 5,8ms. Ce qui est en dehors du temps spécifique de rafraîchissement.

Ce problème a été résolu en déplaçant la valeur test d'une ligne à la troisième ligne qui la suit et en lisant le premier emplacement des deux lignes intermédiaires. Ce qui permet de rafraîchir ces deux lignes. De cette manière le pattern scrute la mémoire, trois fois. A chaque fois vingt-et-une lignes sont testées, dans un temps donné par la formule:

$$(3N^{re} + 5) (21) \text{ cycles-horloge.}$$

Ce qui donne, pour une RAM de 4K bits, et un cycle d'horloge de 470ms, un temps égal à 1,9ms, qui est dans le domaine du temps de rafraîchissement.

Un autre paramètre important dans cette technique concerne l'analyse de l'erreur dans la donnée stockée. Les instructions trois et sept utilisent deux lectures distinctes et séparées, ce qui permet de savoir si l'erreur est due au passage d'une colonne à une autre, ou à l'influence des cellules voisines sur la même colonne.

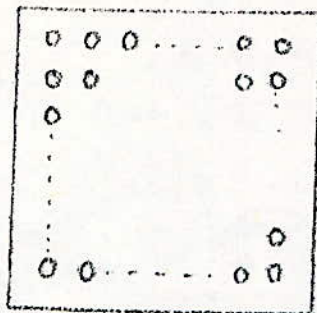
2. DIAPAT.

Ce pattern est du type "Shifting-diagonals", c'est à dire que ce test est appliqué aux cellules disposées suivant une même diagonale de la matrice de mémorisation avant de passer aux cellules de la diagonale suivante.

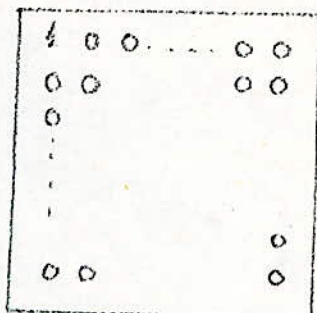
Il est utilisé pour détecter la destruction des données stockées dans les RAM de 4K, 16 bits, et qui est due soit aux opérations internes de sélection dans la matrice, soit à l'influence des cellules voisines, soit aux erreurs d'amplifications.

Les instructions composant ce test sont les suivantes :

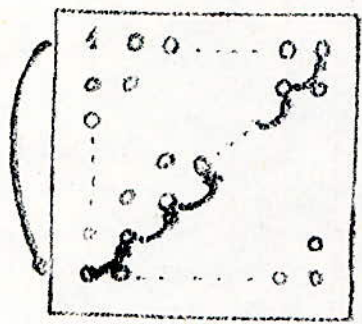
1. Initialisation de la mémoire : Ecriture de "0" ("ou 1") dans tout ses emplacements mémoire.
2. Ecriture de la valeur-test "1" (resp "0") dans l'emplacement 0.
3. On saute à l'emplacement 63.
On lit, tout les emplacements de la diagonale et on les ajoute au contenu de l'emplacement 63.
4. On efface, après lecture de toute la diagonale, la valeur-test et on réécrit la valeur initiale.
5. On incremente l'adresse. la valeur-test est écrite dans tout les emplacements de la diagonale, dont les extrémités coïncident avec la ligne et la colonne données par l'incrementation. (emplacement 1 et 64 pour la première phase).
6. On repete ③ après avoir ajouté 64 à 63.
7. Le procédé est répété, jusqu'à ce qu'on ait écrit la valeur test dans toutes les diagonales.



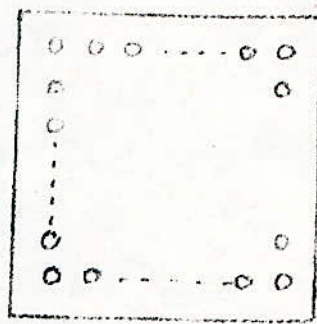
Instruction 1



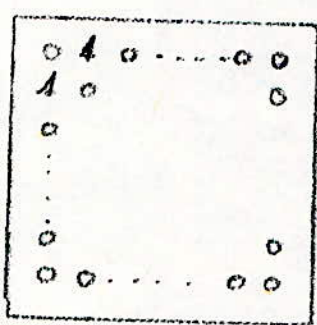
Instruction 2



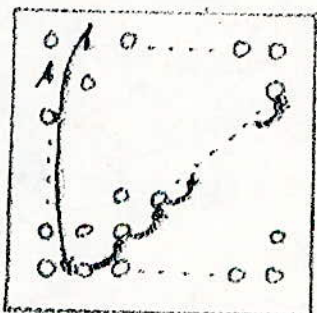
Instruction 3



Instruction 4



Instruction 5



Instruction 6

Figure III. A0.

Schémas illustrant le contenu de chaque emplacement mémoire après exécutions des instructions de ① à ⑥ du DIAPY

B. On refait les instructions de ① à ⑦ avec la valeur-test complémentaire "0".

La figure III.10. montre le contenu de la mémoire après chaque une des instructions du DIAPAT. Le rafraichissement ne pose aucun problème dans cette technique, car l'instruction ③ touche toutes les lignes durant l'intervalle de temps de rafraichissement.

D'autre part, le DIAPAT, sépare les opérations d'écriture et de test, ce qui permet de déterminer la phase dans laquelle l'erreur est commise. Soit à l'initialisation ①, ou à l'écriture de la valeur-test ou pendant le test des diagonales.

Avant le développement des techniques GAITCOX et DIAPAT, que nous venons d'exposer, d'autres patterns étaient déjà appliqués. Nous allons en décrire brièvement, les techniques MASEST, MARCH, WALKING et GALLOPING.

Les deux premières sont caractérisées par leur temps d'executions relativement courts et donnent une assez bonne assurance du bon fonctionnement du circuit.

Les deux autres techniques ont l'avantage d'exercer le circuit d'une façon rigoureuse, mais elles ont des temps d'executions trop lents.

3. MASEST

On commence dans cette technique par charger la mémoire par une suite de "1" et de "0". Voir figure III.11. Le test débute par la lecture des

contenu de l'emplacement 0 puis on lit celui du complément à 0 par rapport à $n-1$, ensuite on relit l'emplacement 0, l'emplacement 1, le complément à 1, l'emplacement 1, l'emplacement 2, l'emplacement complément à 2, l'emplacement 2, ..., l'emplacement $n-1$, l'emplacement complément à $n-1$, l'emplacement $n-1$. La mémoire est ainsi lue entièrement en débutant de l'emplacement 0.

On répète ensuite le test avec le complément de la donnée initialement utilisée.

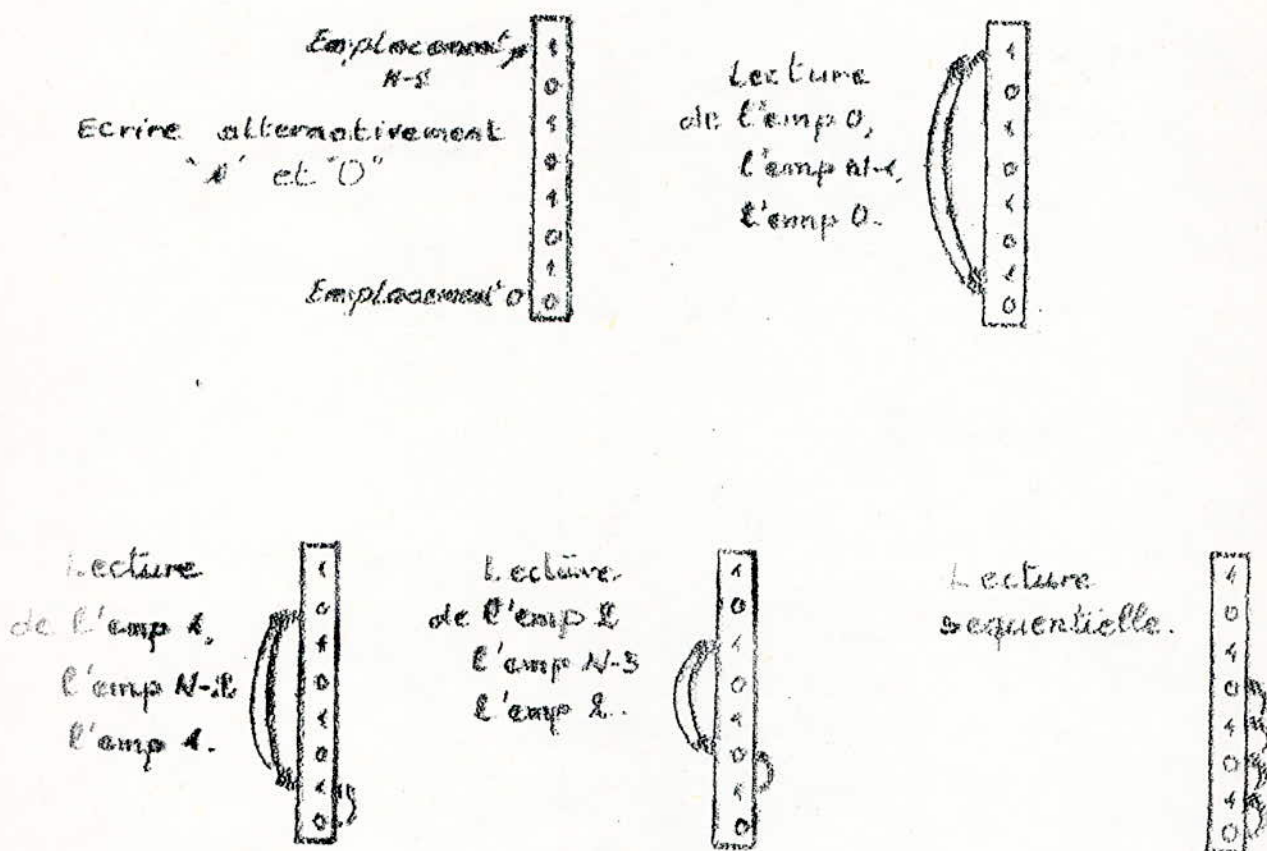


Figure III. 11. MASEST

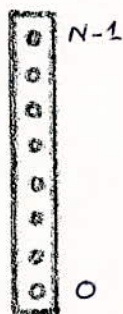
4. MARCHING 1's and 0's:

On initialise la mémoire à tester par cette technique, en la chargeant par "0". On commence le test par la lecture du zéro dans l'emplacement "0" où l'on écrit ensuite "1". Puis, on passe l'emplacement 1 où on lit le zéro qu'on change ensuite par 1. On répète le processus de lecture du "zéro" et d'écriture du "un" dans tous les emplacements mémoire suivants, jusqu'à ce que toute la mémoire soit chargée par "un". La phase suivante consiste à relire les "uns" et les remplacer par des "zéros" en commençant par le dernier emplacement mémoire soit l'emplacement $N-1$, N étant le nombre d'emplacements dans la mémoire. A la fin du test on retrouve le contenu initial de la mémoire, soit des "zéros" partout.

Tout le processus est schématisé par la figure II.12. et la figure II.13

Ce processus est ensuite répété avec l'initialisation de la mémoire à 1.

Initialisation de la mémoire par l'écriture d'un "zéro" partout



lire le "zéro" et le remplacer par le "un" à partir de l'emplacement zéro



Figure II.12 MARCHING 1's and 0's.

Fin des premiers
parcours : tous
Les emplacements
contenant un
"1"

| |
|---|
| 1 |
| 1 |
| 1 |
| 1 |
| 1 |
| 1 |
| 1 |
| 1 |

Repetition de
l'opération en
commençant
par l'empla-
cement (N-1) et
remettre "0"
au lieu de "1"

| |
|---|
| 0 |
| 0 |
| 0 |
| 1 |
| 1 |
| 1 |
| 1 |
| 1 |

Figure 11. A3. Marching 1's and 0's

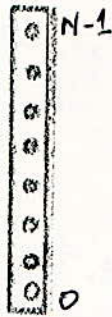
5. WAKPAT :

Dans le WAKPAT, l'initialisation de la mémoire consiste en son chargement par "zéro". Puis, dans l'emplacement "0" on écrit "1" et on vérifie le contenu de tous les autres emplacements, qui doit être "0". On relit ensuite l'emplacement "0", on efface le "1" et on passe à l'emplacement "1". On y écrit "1" et on vérifie le contenu de tous les autres emplacements, à partir de l'emplacement "2", et qui doit être "0". On relit ensuite l'emplacement "1", on efface le "1" et on passe à l'emplacement "2".

On répète le processus jusqu'à ce que le "x" soit écrit et effacé de tous les emplacements mémoire.

Le test est ensuite répété avec le complément de la valeur test déjà utilisée autrement dit le "0".

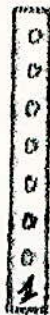
Initialisation
à "zéro"



Écriture d'un
"un" dans
l'emplacement
zéro et lecture
des autres
emplacements



Retour à l'emp-
lacement zéro
où l'on doit
retrouver le
"un"



Incrementation
de l'adresse et
répétition du
test pour tous
les emplacements

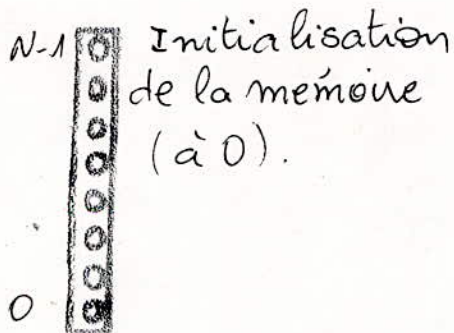


Figure III. 14. WAKPAT.

6. GALPAT.

La phase d'initialisation dans le GALPAT, consiste à mettre zéro dans tout les emplacements de la mémoire à tester. Un "1" est ensuite écrit à l'emplacement zéro. On fait après, une suite de lectures dans l'ordre suivant: emp 1, emp 0, emp 2, emp 0, emp 3, emp 0, ..., emp N-1, emp 0. (N étant le nombre d'emplacements mémoire). On efface le 1 de l'emp 0 et l'écrit dans l'emp 1 et répète le cycle de lecture précédent en revenant à chaque fois à l'emp 1. On répète le processus jusqu'à ce que le "1" soit écrit et effacé de tout les emplacement mémoire. La figure III. 15 illustre ces pas.

Le test est alors répété avec le complément de la valeur test.



- Mise à 1 de l'emp. 0.
- Lecture de tous les autres emp. en revenant à chaque fois à l'emp 0.

A la fin de la phase précédente tous les emp sont à 0 sauf l'emp 0.



Le 1 est ensuite effacé de l'emp. 0. et est écrit à l'emp 1.

On répète alors le processus de lecture

On refait cette phase jusqu'à avoir écrit le 1. dans tous les emp.



Fig. III. 15.

GALPAT.

CHAPITRE

IV

TESTS des MICROPROCESSEURS

1. Présentation des microprocesseurs.

a. Définition :

Le micro-processeur est, avant tout, un circuit intégré, mais arrivé à un certain stade d'évolution. On peut dire que c'est un composant électronique, auquel les progrès de la technologie ont permis d'atteindre un niveau de complexité tel qu'il est devenu un système informatique programmable, d'usage général.

La notion de composant est justifiée par le fait que les microprocesseurs sont compris sur une puce.

b. Classification des microprocesseurs.

On peut classer les microprocesseurs de différentes manières : chronologiquement, par technologie, par "famille". Cependant, la manière la plus adoptée à l'heure actuelle, est le classement par famille.

Chaque constructeur propose dans son catalogue un ensemble de circuits à haute intégration (LSI) associés à un type de microprocesseur. Ces circuits constituent la famille du microprocesseur.

- Généralement, une famille type comprend :
- L'unité centrale (le microprocesseur).
 - Des circuits de mémorisation (ROM, RAM, PROM...).
 - Des circuits d'E/S ou de gestion de périphériques.
 - Des circuits spéciaux (horloge, amplificateur de signaux, capteurs/cadenceur).

La famille constitue l'environnement du microprocesseur et l'ensemble forme le microordinateur.

La figure IV. 1. présente un microprocesseur dans un environnement minimum et la structure d'interconnexion du type bus, utilisée pour la communication entre les composants de cet environnement.

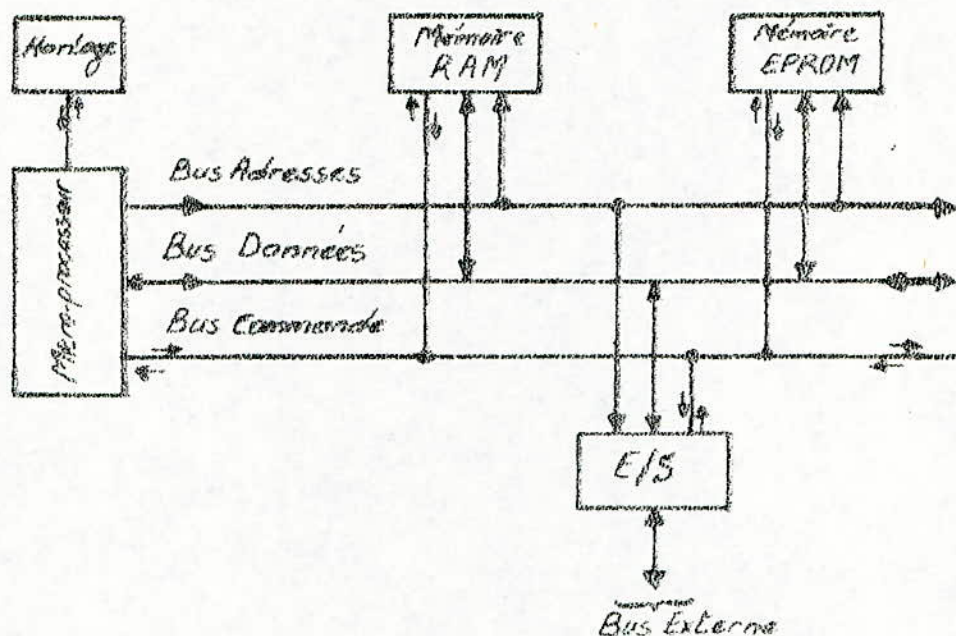


Figure IV.1. Un microordinateur composé d'un $\mu.P$ et son environnement minimum.

c. Structure du microprocesseur.

La figure IV.2. présente la structure interne d'un microprocesseur 8bits et dans laquelle on distingue l'UAL, chargée d'exécuter les opérations arithmétiques et logique, l'unité de commande où sont stockées les micro-commandes et qui est chargée du contrôle du fonctionnement du système, un ensemble de registres servant au stockage provisoire des informations, qui sont, suivant le type de registre, soit des données, soit des instructions, soit des adresses. La liaison entre ces éléments est assurée

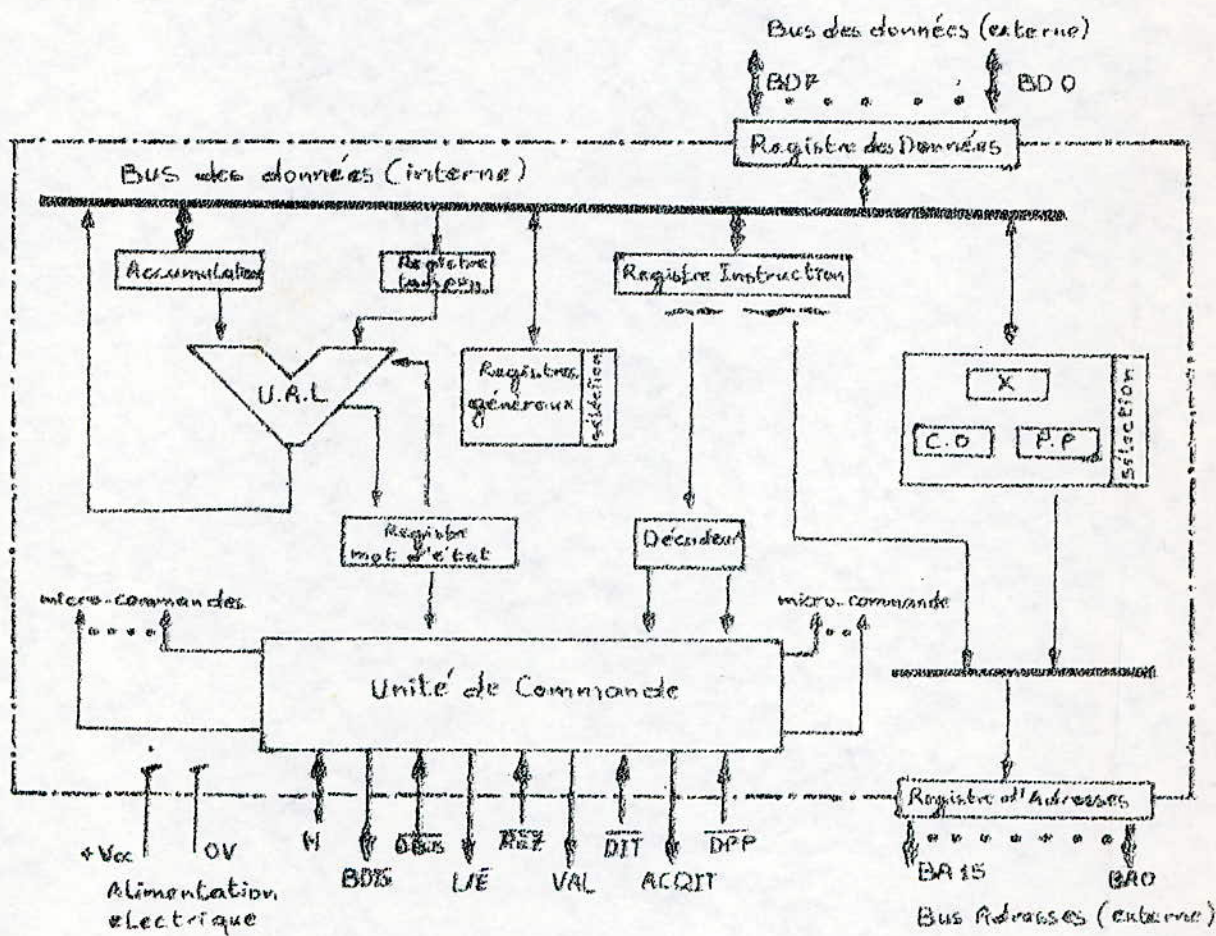


Figure III. Structure Interne du microprocesseur

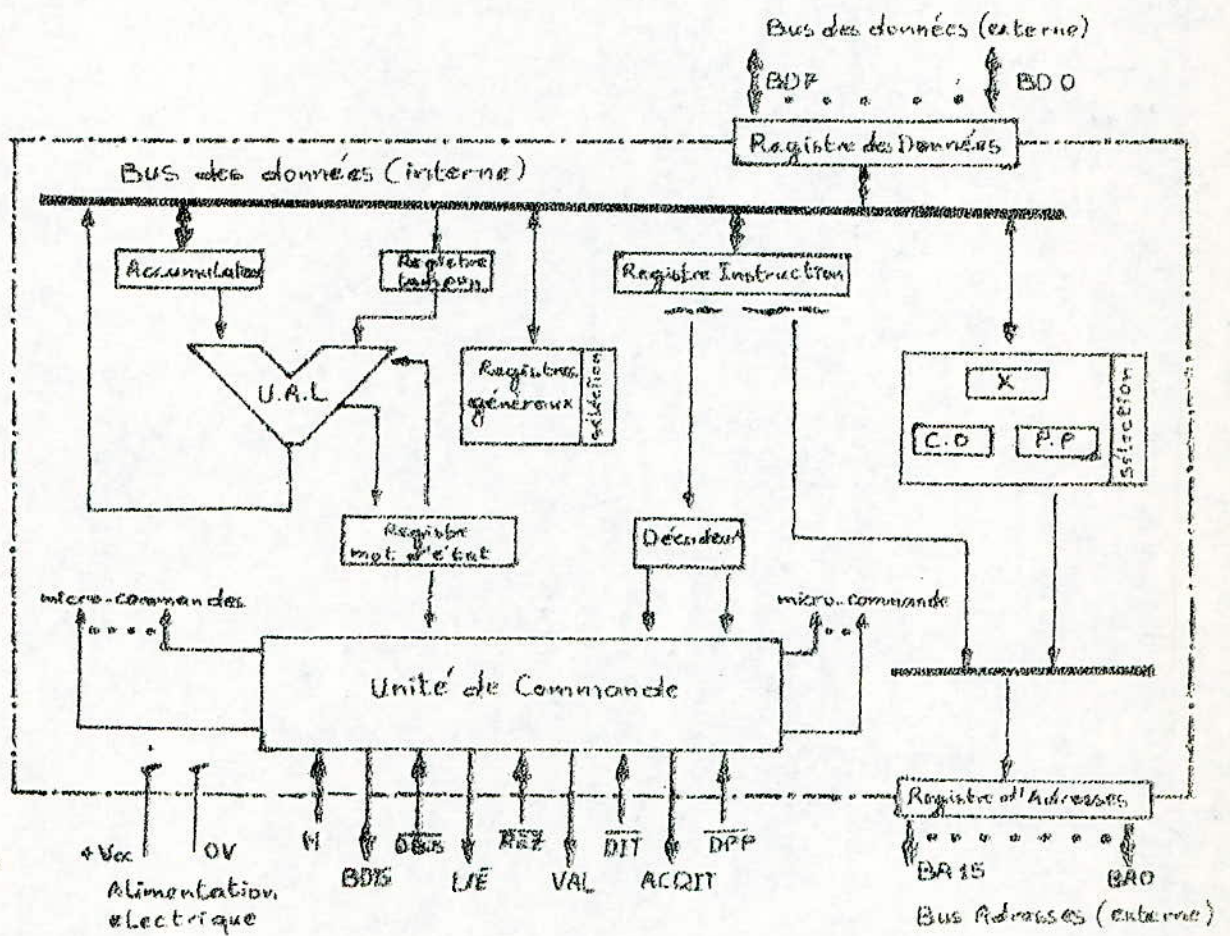


Figure III. Structure Interne du microprocesseur

par une structure de bus interne.

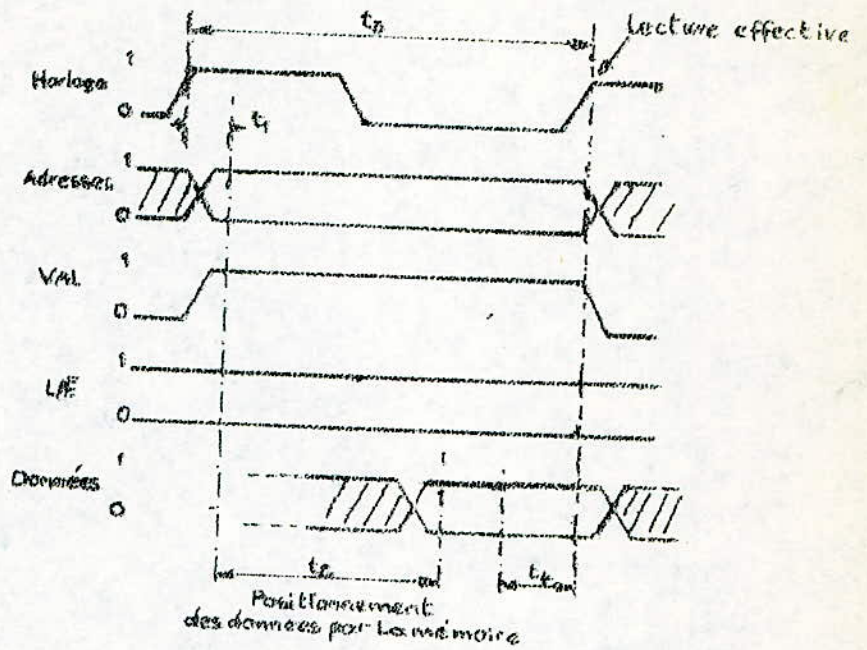
Comme il a été souligné, le microprocesseur communique avec son environnement par un ensemble de bus. Il y dirige les transferts d'informations; il est le pilote ou le maître des bus.

D'autres signaux entrent au microprocesseur et qui lui permettent soit la synchronisation soit l'initialisation soit de se rendre compte de l'état de son environnement.

L'ensemble de ces signaux évolue dans le temps pendant la réalisation d'une fonction externe au microprocesseur, suivant des chronogrammes précis donnés par le constructeur. L'examen approfondi de ces chronogrammes, est nécessaire pour la réalisation de systèmes à base de microprocesseur.

La figure II.3 présente les diagrammes temporels des opérations d'écriture et de lecture de données en mémoire.

Dans ces diagrammes on distingue: le signal d'horloge qui synchronise les opérations à exécuter; le signal d'adresse qui précise l'adresse, dans la mémoire, de la donnée concernée par l'opération; le signal V_{AK} qui sert à valider l'adresse après sa stabilité; le signal L/\bar{E} qui définit l'opération à réaliser et le signal des données à écrire ou lues dans la mémoire.



Lecture d'une donnée en mémoire

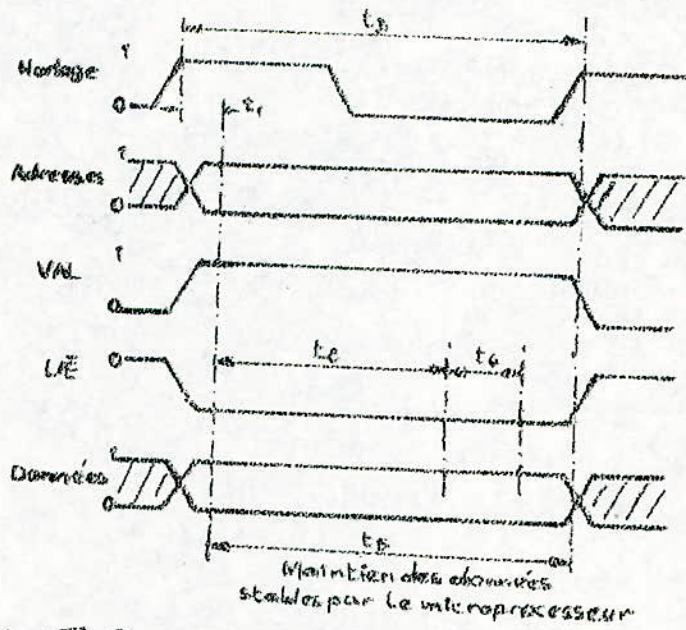


Fig. 11.3. Écriture d'une donnée en mémoire

2. Stratégies et méthodes de test :

a. Stratégies de test des microprocesseurs.

L'approche qui considère le microprocesseur comme un composant, est inspirée des techniques mises en œuvre pour le test des circuits intégrés SSI et MSI.

Son principe de base est de vérifier les spécifications électriques et dynamiques qui ont été déduites, par les concepteurs du circuit, des spécifications fonctionnelles. En effet, ces dernières ont peu de chance d'être respectées si la réalisation électrique n'est pas conforme aux objectifs.

Dans cet esprit, la première tâche à effectuer des tests paramétriques statiques, ayant pour objet de vérifier que les tensions et les courants présents sur les broches du boîtier sont conformes aux spécifications. Ces tests, qui ne posent pas de problèmes particuliers, permettent d'effectuer un premier tri, qui est, cependant, tout à fait insuffisant pour des circuits aussi complexes que les microprocesseurs.

Pour effectuer un test complet, il faudrait appliquer au circuit une séquence de combinaisons d'entrées, et vérifier que chaque variation sur les entrées est suivie par des transitions correctes sur les sorties, et ceci à des instants correctes et pour toutes les conditions d'environnement possibles (température, humidité...). La séquence d'entrées appliquée devrait, de plus, exercer le circuit de toutes les façons possibles. Pratiquement, cet objectif est irréalisable dans son intégrité. Car le nombre de cycles nécessaires à sa réalisation est de plusieurs millions.

C'est pourquoi, on doit chercher une nouvelle méthode pour approcher ce problème. Ainsi a-t-on adopté une approche qui consiste à traiter le microprocesseur, comme une "boîte-noire", chargée d'exécuter des instructions. Le nombre de ces instructions étant plus ou moins grand, rend le problème de vérification du microprocesseur très complexe.

b. Méthodes de vérification des microprocesseurs:

Pour vérifier le bon fonctionnement d'un microprocesseur, des méthodes et des techniques ont été développées et qui dépendent de l'utilisation et des moyens dont disposent soit le fabricant soit l'utilisateur pour faire cette vérification. Ainsi trois méthodes de vérification existent.

b.1. Méthode d'utilisation:

On fait exécuter au microprocesseur un programme d'utilisation, puis on fait le diagnostic. Cependant les microprocesseurs ont une utilisation universelle et pour les vérifier, avec cette méthode, le constructeur devra leur appliquer tous les programmes d'utilisation possibles ce qui est irréalisable. Mais on peut appliquer cette méthode aux microprocesseurs spécialisés, c'est à dire susceptibles de réaliser un programme très spécialisé à partir d'un jeu d'instructions réduit, mais orienté vers une application très particulière.

b.2. Méthode de comparaison:

C'est une méthode classique de vérification des circuits logiques. Elle consiste à faire une comparaison entre le circuit à vérifier et un autre circuit

réputé sans défaut appelé "référence". L'inconvénient évident de cette méthode est sa mauvaise précision car elle dépend du circuit de référence dont le choix est très difficile.

b.3. Méthode des réponses stockées.

Dans cette méthode, une séquence d'entrée est préparée au préalable, et des réponses du circuit microprocesseur sont élaborées soit par simulation logique, soit par émulation. Les informations logiques d'entrée et les réponses attendues sont stockées (ou enregistrées) dans une mémoire de masse.

Au moment du test effectif, elles transitent par une mémoire rapide, de façon à pouvoir exercer le circuit à la fréquence normale de fonctionnement.

c. Méthodes de test des microprocesseurs :-

La connaissance de la configuration logique du microprocesseur, est nécessaire pour l'application, d'une manière efficace, des méthodes de test. C'est pourquoi seul le constructeur est bien placé pour faire ce genre de travail.

Le test se fait en trois méthodes :

- c.1 tester les modules internes un par un
- c.2 faire le test de l'unité de commande
- c.3 faire le test des deux à la fois.

c.1. Test des modules:

Dans ce test le microprocesseur est partagé en modules indépendants ou mutuellement dépendants.

Chaque module est testé individuellement par l'introduction d'une séquence d'instruction par le circuit d'ETB du microprocesseur. On ne considère que le résultat globale de l'exécution, ou parfois d'une partie des instructions, sans considérer les cycles élémentaires mis en œuvre, ni les conditions dans lesquelles chaque cycle se déroule.

Par nature, ce principe permet de réaliser un test uniquement fonctionnel.

Selon cette méthode, le premier pas consistera à attaquer le compteur programme (16 bits) qu'on incrémentera de 0 à $2^{16}-1$, pour vérifier tous les bits du compteur et les lignes d'adresses. Ensuite le registre d'index, le pointeur de pile, l'accumulateur et l'UAL peuvent être vérifiés un à un individuellement.

c.2. Test de la commande:

Il est insuffisant de vérifier que chaque composant est satisfaisant pour conclure que le microprocesseur fonctionne correctement, car la vérification du bon fonctionnement de certaines instructions (comme les interruptions, les sauts...) est relativement complexe, puisqu'il faut stocker un nombre important d'informations avant de prendre une décision à partir du résultat obtenu.

Il est donc nécessaire de simuler réellement la situation du circuit à tout instant du programme en utilisant de vraies données, c'est à dire, celles fournies par le microprocesseur lui-même, et cela revient à vérifier la fonction essentielle du microprocesseur, à savoir sa capacité de prendre des décisions en fonction des données qui lui sont fournies. Ces dernières permettent de définir à quel instant, à l'intérieur d'un cycle d'horloge, les informations logiques doivent être présentées aux entrées, et à quel instant les mesures doivent être effectuées.

C.3. Test complet :

Ce type de test, combinaison des deux tests précédents, consiste à faire un test complet en un temps raisonnable, on fait de telle manière qu'on vérifie quelques modules sensibles ainsi que les commandes les plus délicates.

d. Techniques de test :

Nous allons maintenant passer en revue les différentes techniques utilisées pour appliquer les méthodes de base pour le test des circuits micro-processeurs.

En effet, en s'inspirant des méthodes de vérification de ces circuits, on peut distinguer cinq grandes méthodes de test :

- d.1 Méthode d'auto-test.
- d.2 Méthode de test par comparaison
- d.3 Méthode algorithmique en temps réel.
- d.4 Méthode des patterns enregistrés.
- d.5 Méthode LEAD.

d.1 Méthode d'auto-test :

Le contrôle ne s'effectue pas sur un matériel de test proprement dit. Le microprocesseur est testé dans un ensemble constituant l'environnement naturel (ROM, RAM, BUS, périphériques...). On charge un programme diagnostic dans la RAM et le microprocesseur l'exécute. Ce programme doit faire exécuter le maximum d'instructions dans les plus mauvaises conditions pour les registres internes. Si toutes ces instructions se déroulent correctement le programme se branche à une adresse finale "BON" sinon, il se branche à une adresse "MAUVAIS".

Parmi les avantages de cette méthode on note :

- Le circuit est dans son environnement naturel.
- Le prix de revient, de l'opération de test, est faible.

Toutefois, cette technique présente des inconvénients, parmi lesquels on note :

- Plusieurs erreurs peuvent se compenser et ne pas être détectées.
- La cause du défaut ne peut, généralement, pas être déterminée, d'où la difficulté de l'analyse du circuit.
- Il faut attendre le déroulement complet du programme pour savoir si le microprocesseur est BON ou MAUVAIS, ce qui peut entraîner un temps de test inutile si le défaut apparaît dès le début.

d.2. Méthode de test par comparaison :

Dans cette méthode, on compare la réponse fournie par un microprocesseur présumé bon, qui sert de référence, avec celle du circuit sous-test. On note tout de suite que le microprocesseur de référence et celui en essai doivent être de même type, de même modèle et provenir du même fabricant, c'est-à-dire, être en tout point identiques.

Pour appliquer cette méthode il faut disposer de deux pilotes (drivers) et de détecteur.

La figure IX.4 donne le principe de montage d'application de cette méthode.

Comme dans la méthode d'auto-test, les deux microprocesseurs peuvent travailler à partir de différents circuits (ROM, RAM, E/S, etc...). Toutes les informations sont transmises simultanément aux deux circuits. Les informations fournies par les deux CPU, sont comparées entre-elles. Si elles présentent une

différence, le circuit en essai est déclaré "MAUVAIS".
 L'avantage de cette méthode est que le premier défaut apparent peut être détecté immédiatement.

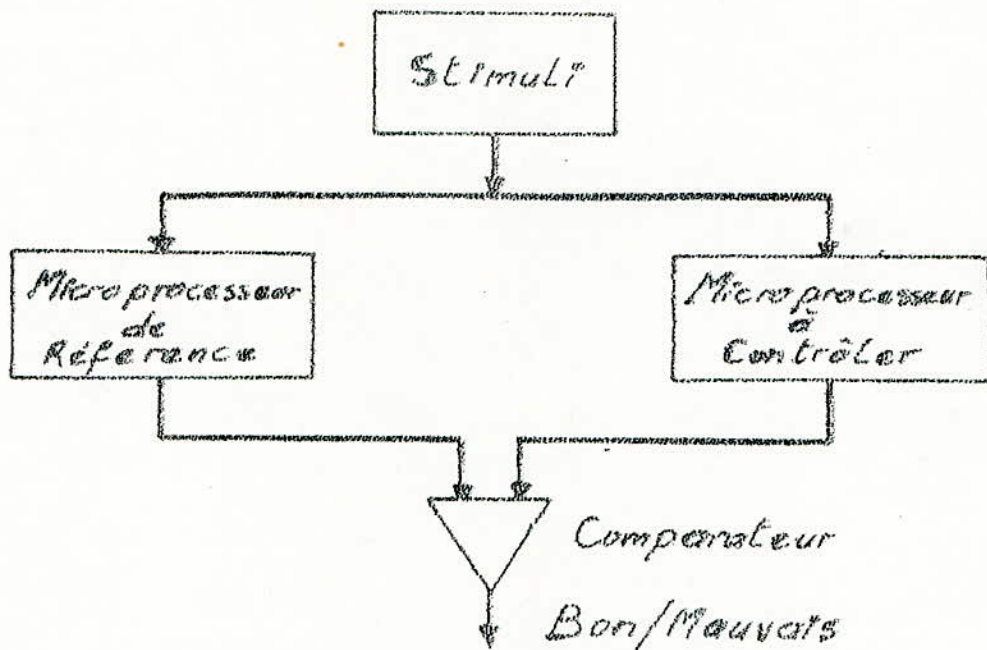


Figure III-4. Test par comparaison à un μP de référence.

Parmi les inconvénients que présente cette méthode on note :

- Les matériels de test dépendent du microprocesseur à tester et l'utilisateur doit posséder autant de matériel que de types de microprocesseur.
- Le résultat du test dépend du circuit de référence choisi.
- Si les deux circuits présentent le même défaut, celui-ci ne peut être détecté (cette évantailité n'est pas négligeable, une erreur de conception pouvant même ne pas être observée).

- Il n'y a pas de possibilité d'intervention en quoi que ce soit sur les paramètres de contrôle.

d.3. Méthode algorithmique en temps réel.

Dans cette méthode, un programme est écrit en langage machine propre au microprocesseur et est chargé dans une mémoire de masse. Chaque instruction est ensuite renvoyée au microprocesseur qui fournit une réponse et un signal d'attente. Ces échanges d'informations s'effectuent à la vitesse réelle de l'horloge. En même temps que s'accomplit ce travail, le contrôleur génère une réponse qu'il compare à celle du microprocesseur. Si les deux réponses sont identiques l'instruction suivante est envoyée au microprocesseur avec un signal "RESTART". Si les réponses sont différentes, le spécimen en essai est considéré comme défectueux.

Le principe de base de cette méthode est illustré par la figure II.5.

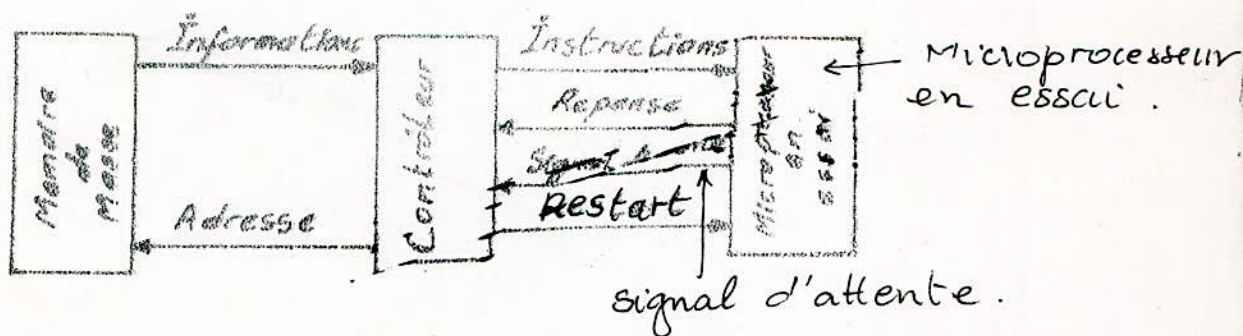


Fig. II.5 : Test par la méthode algorithmique en temps réel

On remarque que, dans cette méthode le déroulement du contrôle de fonctionnement est interrompu. Comme beaucoup d'instructions utilisées par les microprocesseurs sont complexes (appel sous-programme, gestion d'interruptions...), leur vérifications demandent plusieurs mots consécutifs, et elles sont donc difficilement contrôlables par cette méthode.

Les inconvénients de cette méthodes sont:

- le contrôle s'effectue instruction par instruction et non par cycle d'horloge.
- Il est impossible de vérifier les anomalies apparaissant à la fin d'un cycle d'horloge.
- Les phénomènes secondaires entraînant les anomalies au cours de l'exécution d'une instruction ne sont pas détectés.

d.4 Méthode des patterns enregistrés:

On considère ici le microprocesseur, comme un assemblage de circuits logiques. Par conséquent, on simule son fonctionnement par un module logique, et pour ce faire, on utilise un mini-ordinateur.

Ainsi cette méthode s'exécute en deux étapes; dans la première, le microprocesseur est simulé par un mini-ordinateur, chaque réponse simulée peut être identifiée et associée à l'instruction qui lui correspond. L'ensemble est contrôlé 16 eb par 16 eb puis envoyé dans un "buffer" à des périodes définies. Le contenu du "buffer" est ensuite sauvé sur disque ou sur une bande magnétique.

Dans la deuxième phase, les patterns sont chargés depuis le disque ou la bande magnétique dans le "buffer"

puis transférées sur le microprocesseur qui est contrôlé 16 eb par 16 eb. La figure III.6. donne le montage du matériel d'application de cette méthode.

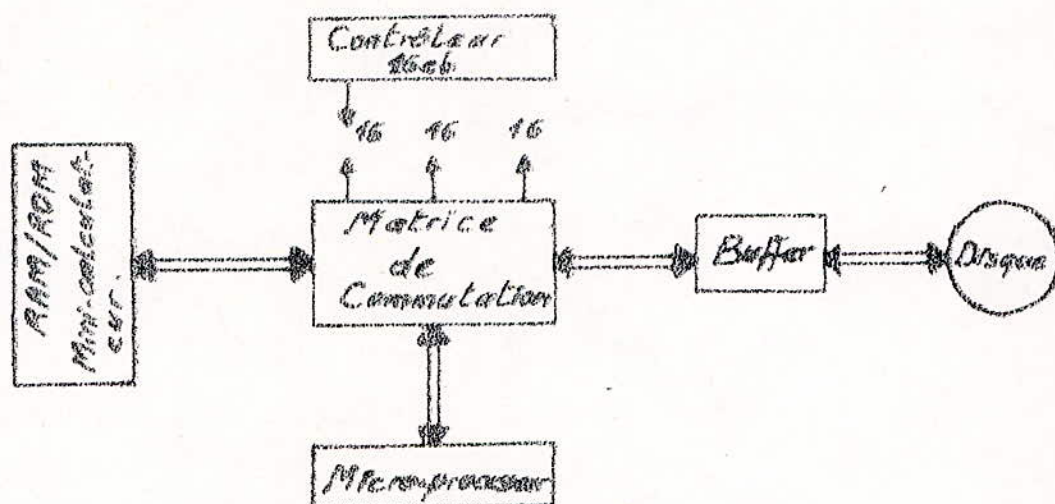


Figure III.6: Test par la méthode des "patterns" enregistrés.

Les avantages de cette méthode sont surtout, sa facilité de mise en œuvre et, une certaine souplesse dans le test.

- Elle présente cependant des défauts importants:
- Nécessité d'un grand "buffer" pour le transfert des Patterns, et dont le prix est élevé.
 - A chaque changement même minime des spécifications des échantillons à tester, la phase première est à reprendre pour une nouvelle simulation.
 - La simulation des interruptions n'est pas possible.
 - Importance du support logiciel.

d.5 Méthode LEAD: (Learn, Execut and diagnostic).

Cette méthode est aussi appelée méthode algorithmique par un procédé heuristique.

Elle permet le contrôle des circuits en appliquant des séquences d'événements possibles à partir d'un test séquentiel. La stratégie LEAD est composée par les phases suivantes:

- Création de l'environnement matériel du circuit.
- Utilisation d'un programme de génération pour charger un programme de diagnostic dans la mémoire du testeur et ceci dans le langage du microprocesseur sous test.
- Exécution du programme diagnostic pour générer la stimulation du microprocesseur et l'étude des réponses auxquelles il faut s'attendre, ceci par la génération d'une table de vérité de toutes les activités fonctionnelles.
- Exécution du programme de test pour l'évaluation technique, le test de la production ou le diagnostic, en utilisant la table de vérité préparée dans la phase précédente.
- Diagnostiquer la donnée caractéristique et la corrélier avec la donnée prévue.

Nous allons, dans ce qui suit, éclaircir ces cinq phases d'exécution de la méthode LEAD.

- Création de l'environnement naturel.

L'environnement naturel du microprocesseur est un système digital de calcul, composé du CPU, du système de mémorisation et des périphériques. (Voir figure IV. 7.)

Avec un programme introduit dans la mémoire du système le CPU met une adresse sur le bus d'adresses. Il exécute l'instruction, qu'il reçoit de la mémoire sur le bus de données, et sélectionne ensuite l'instruction suivante jusqu'à la fin du programme. Le CPU dans l'exécution des instructions, envoie et reçoit la donnée de la mémoire et émet des signaux de contrôle, pour choisir entre la mémoire et les périphériques. A leur tour, ces derniers émettent d'autres signaux pour appeler le CPU, quand ils sont disponibles ou quand l'introduction d'une donnée est souhaitée. D'autres signaux sont encore envoyés, par le CPU, pour informer le reste du système qu'il est en arrêt ou en attente ou pour répondre à une demande d'interruption.

Ces phases de recherche, d'exécution et de contrôle d'instructions sont des états courants dans le fonctionnement du CPU et peuvent être simulés dans un système automatique de test, qui sera l'environnement naturel du microprocesseur, car il fournira au microprocesseur les périphériques et le système de mémorisation.

Par exemple dans la mémoire de masse du système Sentry II, développé par Fairchild, une zone appelée LMI 000 (Local Memory Image) est utilisée comme système de mémorisation pour le microprocesseur 8080.

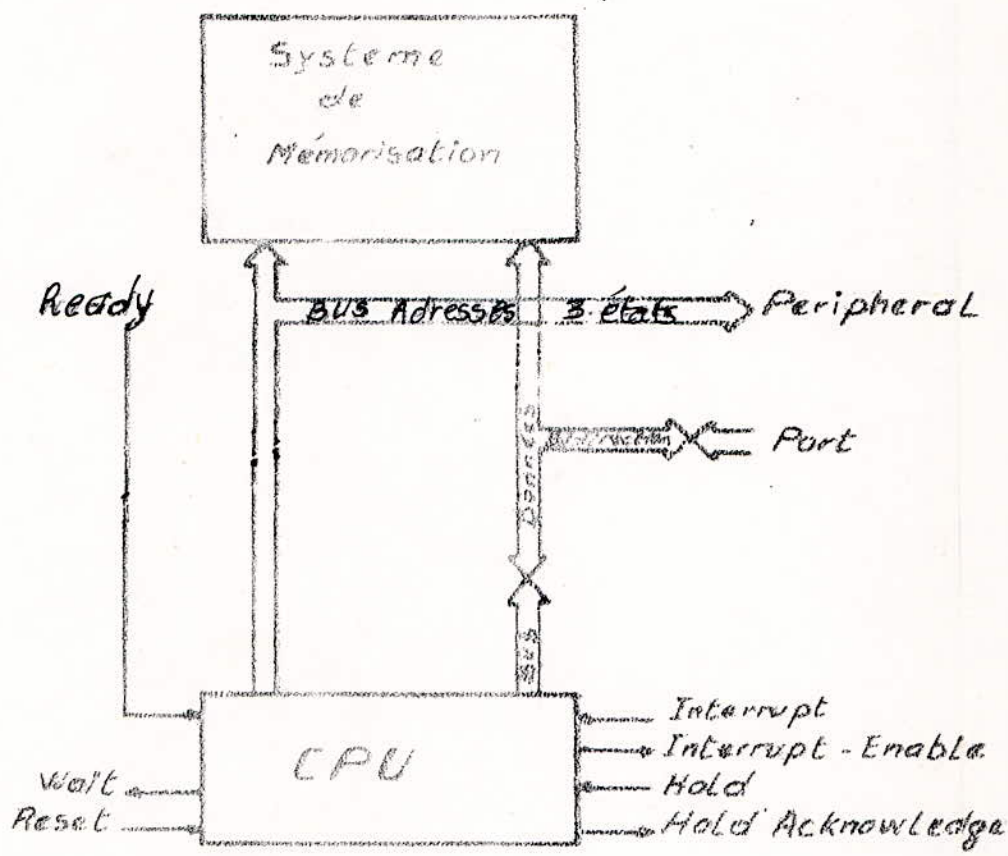


Figure IV. 4. Environnement naturel du micro-processeur.

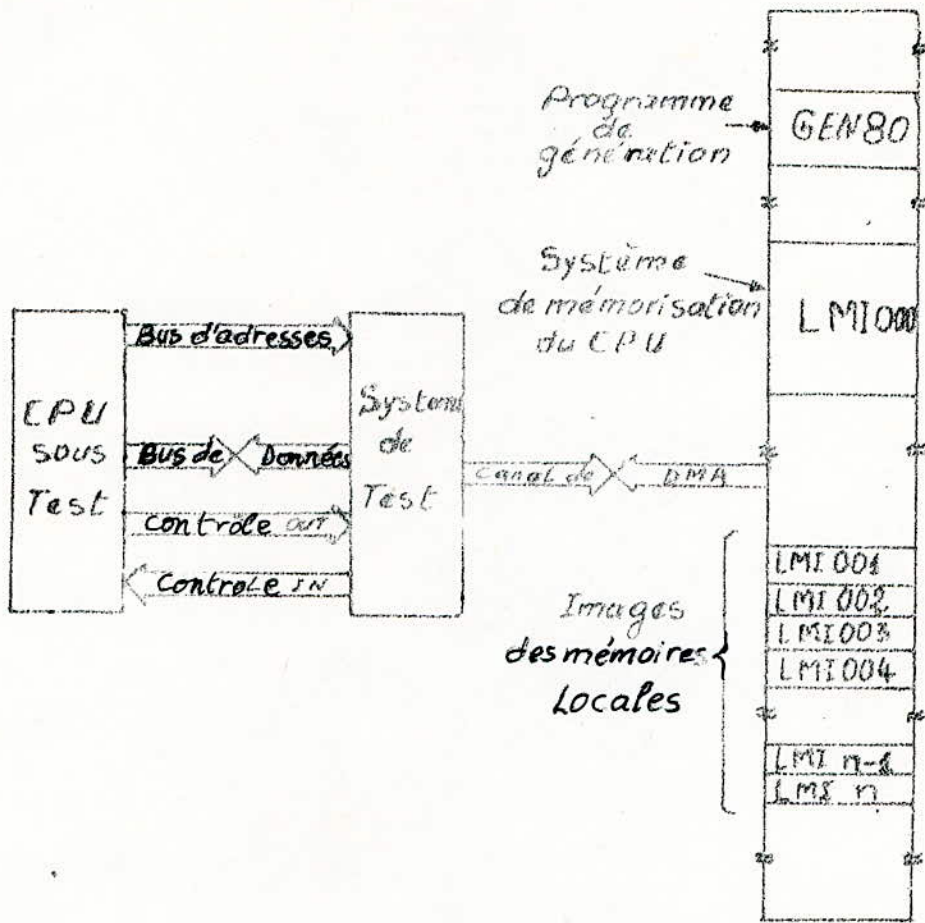


Figure IV.8. Test de l'environnement

d'Intel. La figure II.8. présente le testeur et la répartition de sa mémoire de masse.

Les signaux d'adressage et de contrôle, issus du CPU, entrent dans le testeur, qui renvoie d'autres signaux de contrôle au CPU. La communication sur le bus de données est bidirectionnelle.

Etude des réponses et des stimulus :

L'environnement étant établi, la phase suivante de cette méthode, est l'introduction du programme diagnostique dans une zone de la mémoire du testeur sous contrôle du programme de génération. Comme le programme diagnostique est écrit dans le langage propre au microprocesseur, il devient un programme d'utilisation. Avec un microprocesseur de référence, vérifié par le déroulement du test et le diagnostic introduit dans la mémoire du testeur, le système simule l'environnement naturel sous le contrôle du programme de génération. Ce dernier, commence l'initialisation, applique l'alimentation et pré-conditionne le microprocesseur de référence pour qu'il démarre l'exécution du programme diagnostique. Le microprocesseur place une adresse sur son bus d'adresses et le testeur lit cette adresse et vérifie l'état du microprocesseur pour voir s'il est dans un cycle d'écriture, de lecture ou en état de repos. Puis le testeur se branche sur le programme de génération, à la suite d'opérations qui simulent ce qui se passe dans l'environnement, quand le microprocesseur exécute le type de cycle sélectionné.

Par exemple dans le cycle d'écriture, les données sont transmises par le microprocesseur à la mémoire du testeur ou aux périphériques. Le testeur lit les données à partir du bus de données et les range, sous le contrôle du programme de génération, dans le programme diagnostique à l'adresse indiquée précédemment par le microprocesseur. Le programme de génération se branche ensuite pour lire l'adresse suivante. Pendant ce temps, et tandis que le dispositif de référence exécute le diagnostic constitué par des instructions et des données rangées dans la LMI 000, le testeur enregistre tous les stimulus et les réponses dans une table de vérité fonctionnelle du type "réponses-stockées" que la figure III.3 donne sa représentation avec les différents composants de cette phase et leur connexions.

Dans un cycle de repos le microprocesseur n'échange aucune donnée et le testeur n'écrit pas de données dans le programme diagnostique ou dans le bus de données. Cependant, le programme de génération enregistre le stimulus du microprocesseur et les réponses. Quand ce dernier exécute son diagnostic et le testeur prépare la table de vérité, le programme de génération imprime le programme de diagnostic.

Le processus d'étude de l'exécution du programme diagnostique et l'enregistrement des stimulus et des réponses du microprocesseur se poursuit jusqu'à la fin de ce programme, ce qui achève le programme de génération. A ce moment, la table de vérité à réponses-stockées et qui occupe plusieurs pages ou zones de la mémoire de masse du testeur se trouve

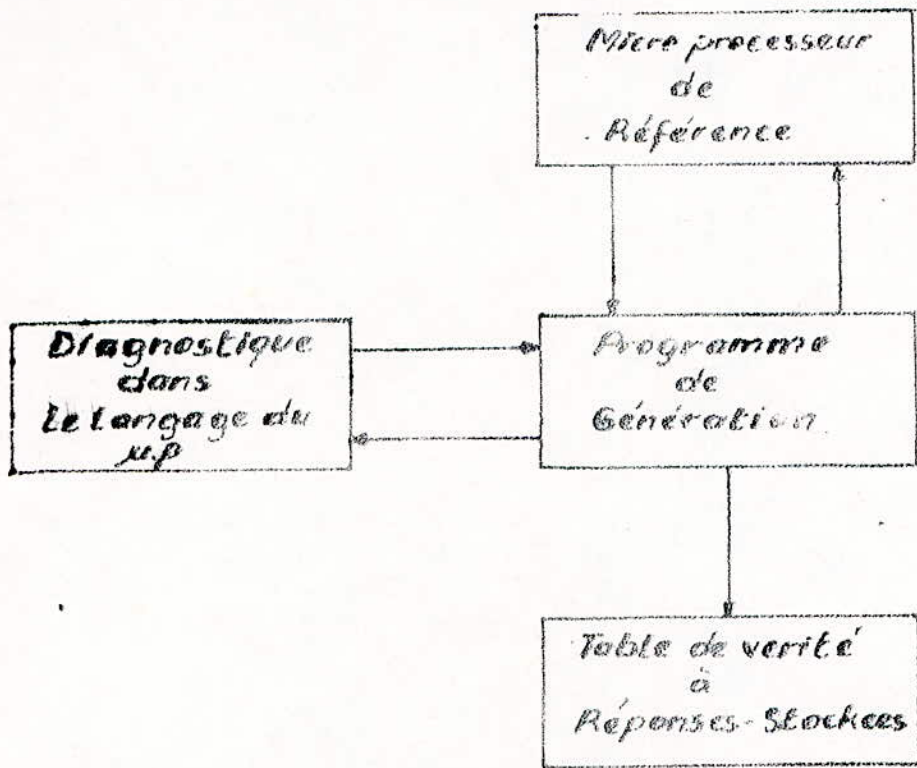


Figure IV.9. Chargement de la table de vérité par le programme de génération.

placée dans cette mémoire sous forme d'images de mémoires locales LMI 001, LMI 002, LMI 003 ..., comme le montre la figure II. 10.

Chaque table est maintenant une série d'instructions, composée de stimulus et de réponses préalablement calculés, à la base desquelles, les tests de production ou les évaluations techniques vont être faites.

- Exécuter et diagnostiquer les résultats :

La mise au point du programme de test est achevée. Le testeur exécute alors, les patterns fonctionnels à réponses stockées pour le diagnostic dans les ~~plus~~ cas de chronologie, comme un programme de production. L'opérateur charge simplement le programme dans le poste de test et commence le test du dispositif. Pour l'évaluation technique, le choix du programme diagnostique adéquat est primordial. Plusieurs programmes diagnostique peuvent être stockés dans la mémoire de masse du testeur, comme des images de mémoires locales, et le programme de test peut alors en choisir, celui qui convient au besoin de l'utilisateur.

Pour une évaluation technique plus flexible, toutes les tensions, les temps et tous les paramètres mesurables sont enregistrés dans les formats qu'on change très simplement. Le programme test, comprend un générateur de données dynamiques, qui permet l'identification de toutes les pannes fonctionnelles. Avec chaque panne, le système montre la broche, l'adresse de la

Mémoire de masse

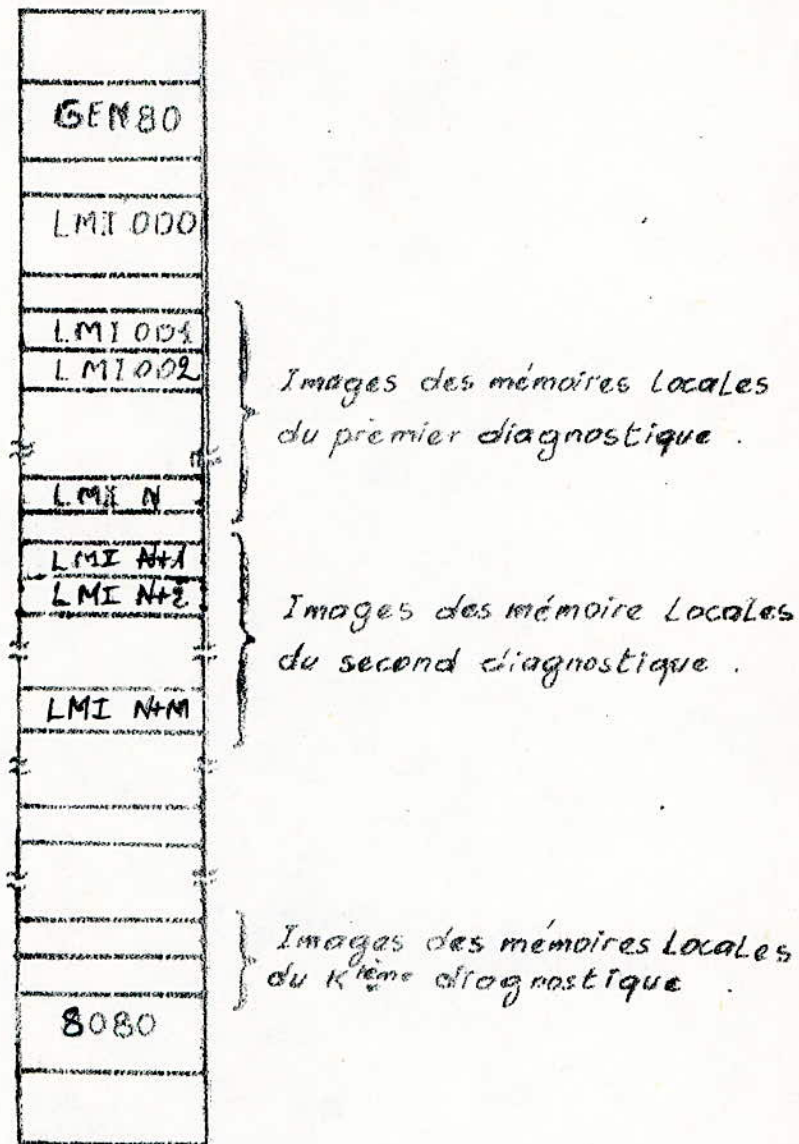


Figure III. 10. Repartition de la mémoire de masse du testeur en zones occupées par des "tables de vérités".

mémoire locale et sa page. L'utilisateur peut alors comparer cette information avec l'information prévue par le programme et identifier chaque faille par broche ou par adresse du CPU ou par la donnée. Il peut aussi examiner tous les modes de pannes du dispositif sous test.

Comme la panne emprisonne le programme dans une boucle où il tournera au hasard, d'autres données erronées peuvent apparaître. C'est pourquoi une analyse manuelle est prévue en option dans le programme de test et avec laquelle l'utilisateur peut contrôler le maximum de pannes enregistrées.

Le programme d'évaluation technique peut utiliser des interruptions programmées, pour par exemple arrêter le test dès l'apparition de la première panne, ou pour imprimer des données sommaires, ou pour tracer des courbes qui montrent la relation entre deux paramètres.

Parmi les avantages de cette méthode particulière de test des microprocesseurs on note :

- la facilité de mise en œuvre et de programmation.
- la connaissance intrinsèque du microprocesseur n'est pas nécessaire.
- Le testeur simule lui-même l'environnement du microprocesseur.
- Test de l'ensemble des instructions sans interruption.

CONCLUSION

En conclusion, nous pouvons dire qu'avec le progrès de la miniaturisation et la complexité de certains circuits constituant les micro-ordinateurs, les tests de ces derniers se compliquent de plus en plus.

Plusieurs techniques de test ont été développées, des matériels de contrôle disposants des mémoires rapides importantes et de générateurs algorithmiques de séquences contrôlés par programme, ont été mis en œuvre, afin d'effectuer des mesures paramétriques dynamiques précises, et des tests fonctionnels complets.

Mais jusqu'à l'heure actuelle, aucune méthode n'a donné satisfaction pour tous les aspects du problème.

BIBLIOGRAPHIE

- R. DELSOL
Circuits intégrés et techniques numériques
Ed. Cepadue, 1974.
- D. GIROD et R. DUBOIS
Au cœur des microprocesseurs
Ed. Eyrolles, 1980.
- E. HORLAÏT et R. JOLY
Les microprocesseurs 8 bits
Ed. Eyrolles, 1982.
- J. P. MENADIER
Structure et fonctionnement des ordinateurs.
Ed. Larousse, 1975.
- W. BARRACLOUGH, A. C. L. CHIANG and W. SOHL
Techniques for Testing the μ -computer family
proceedings of the IEEE Vol. 64 No 6 June 76.
- G. DEMARS et J. C. RAULT
Rapport final d'une étude des méthodes de test des μP
et des équipements composants des μP . Août 1977.
- C. LECLERC et P. PRODHOMME
Méthodes de contrôle des microprocesseurs
et de leurs circuits annexes.
Minsat micros. NR 146 du 24 Avril 1980.
- E. I. MUEHLDORF and A. D. SAVKAR
LSI Logic Testing - An Overview
IEEE Transactions on computers
Vol. C-30 No 1 January 1981.
- B. SCHUSHEIM
A flexible Approach to microprocessor Testing
Computer Design, March 1976.