



HAL
open science

Test en ligne du microprocesseur MC 68000 : modélisation et programmes de test

Pierre Marchal

► **To cite this version:**

Pierre Marchal. Test en ligne du microprocesseur MC 68000 : modélisation et programmes de test. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1983. Français. NNT: . tel-00306726

HAL Id: tel-00306726

<https://theses.hal.science/tel-00306726>

Submitted on 28 Jul 2008

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

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

THESE

présentée à

l'Institut National Polytechnique de Grenoble

pour obtenir le grade de

**DOCTEUR DE 3^{ème} CYCLE
INFORMATIQUE**

par

Pierre MARCHAL



**TEST EN LIGNE DU MICROPROCESSEUR MC68000.
- MODELISATION ET PROGRAMMES DE TEST -**



Thèse soutenue le 5 juillet 1983 devant la Commission d'Examen:

Monsieur L. BOLLIET : Président

Messieurs J.A. ABRAHAM

J. BOREL Examineurs

A. COSTES

Monsieur B. COURTOIS Rapporteur

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1982-1983

Président de l'Université : D. BLOCH

**Vice-Président : René CARRE
Hervé CHERADAME
Marcel IVANES**

PROFESSEURS DES UNIVERSITES :

ANCEAU François	E.N.S.I.M.A.G.
BARRAUD Alain	E.N.S.I.E.G.
BAUDELET Bernard	E.N.S.I.E.G.
BESSON Jean	E.N.S.E.E.G.
BLIMAN Samuel	E.N.S.E.R.G.
BLOCH Daniel	E.N.S.I.E.G.
BOIS Philippe	E.N.S.H.G.
BONNETAIN Lucien	E.N.S.E.E.G.
BONNIER Etienne	E.N.S.E.E.G.
BOUVARD Maurice	E.N.S.H.G.
BRISSONNEAU Pierre	E.N.S.I.E.G.
BUYLE BODIN Maurice	E.N.S.E.R.G.
CAVAIGNAC Jean-François	E.N.S.I.E.G.
CHARTIER Germain	E.N.S.I.E.G.
CHENEVIER Pierre	E.N.S.E.R.G.
CHERADAME Hervé	U.E.R.M.C.P.P.
CHERUY Arlette	E.N.S.I.E.G.
CHIAVERINA Jean	U.E.R.M.C.P.P.
COHEN Joseph	E.N.S.E.R.G.
COUMES André	E.N.S.E.R.G.
DURAND Francis	E.N.S.E.E.G.
DURAND Jean-Louis	E.N.S.I.E.G.
FELICI Noël	E.N.S.I.E.G.
FOULARD Claude	E.N.S.I.E.G.
GENTIL Pierre	E.N.S.E.R.G.
GUERIN Bernard	E.N.S.E.R.G.
GUYOT Pierre	E.N.S.E.E.G.
IVANES Marcel	E.N.S.I.E.G.
JAUSSAUD Pierre	E.N.S.I.E.G.
JOUBERT Jean-Claude	E.N.S.I.E.G.
JOURDAIN Geneviève	E.N.S.I.E.G.
LACOUME Jean-Louis	E.N.S.I.E.G.
LATOMBE Jean-Claude	E.N.S.I.M.A.G.

.../...

LESSIEUR Marcel	E.N.S.H.G.
LESPINARD Georges	E.N.S.H.G.
LONGUEQUEUE Jean-Pierre	E.N.S.I.E.G.
MAZARE Guy	E.N.S.I.M.A.G.
MOREAU René	E.N.S.H.G.
MORET Roger	E.N.S.I.E.G.
MOSSIERE Jacques	E.N.S.I.M.A.G.
PARIAUD Jean-Charles	E.N.S.E.E.G.
PAUTHENET René	E.N.S.I.E.G.
PERRET René	E.N.S.I.E.G.
PERRET Robert	E.N.S.I.E.G.
PIAU Jean-Michel	E.N.S.H.G.
POLOJADOFF Michel	E.N.S.I.E.G.
POUPOT Christian	E.N.S.E.R.G.
RAMEAU Jean-Jacques	E.N.S.E.E.G.
RENAUD Maurice	U.E.R.M.C.P.P.
ROBERT André	U.E.R.M.C.P.P.
ROBERT François	E.N.S.I.M.A.G.
SABONNADIÈRE Jean-Claude	E.N.S.I.E.G.
SAUCIER Gabrielle	E.N.S.I.M.A.G.
SCHLENKER Claire	E.N.S.I.E.G.
SCHLENKER Michel	E.N.S.I.E.G.
SERMET Pierre	E.N.S.E.R.G.
SILVY Jacques	U.E.R.M.C.P.P.
SOHM Jean-Claude	E.N.S.E.E.G.
SOUQUET Jean-Louis	E.N.S.E.E.G.
VEILLON Gérard	E.N.S.I.M.A.G.
ZADWORNY François	E.N.S.E.R.G.

PROFESSEURS ASSOCIES

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

PROFESSEURS UNIVERSITE DES SCIENCES SOCIALES (Grenoble II)

BOLLIET Louis
Chatelin Françoise

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

RIEU Jean
SOUSTELLE Michel

CHERCHEURS DU C.N.R.S.

FRUCHART Robert
VACHAUD Georges

Directeur de Recherche
Directeur de Recherche

.../...

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

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

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

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

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

.../...

GALERIE Alain	E.N.S.E.E.G.
HAMMOU Abdelkader	E.N.S.E.E.G.
MALMEJAC Yves	E.N.S.E.E.G. (CENG)
MARTIN GARIN Régina	E.N.S.E.E.G.
NGUYEN TRUONG Bernadette	E.N.S.E.E.G.
RAVAINE Denis	E.N.S.E.E.G.
SAINFORT	E.N.S.E.E.G. (CENG)
SARRAZIN Pierre	E.N.S.E.E.G.
SIMON Jean-Paul	E.N.S.E.E.G.
TOUZAIN Philippe	E.N.S.E.E.G.
URBAIN Georges	E.N.S.E.E.G. (Laboratoire des ultra-réfractaires ODEILLON)
GUILHOT Bernard	E.N.S. Mines Saint Etienne
THOMAS Gérard	E.N.S. Mines Saint Etienne
DRIVER Julien	E.N.S. Mines Saint Etienne
BARIBAUD Michel	E.N.S.E.R.G.
BOREL Joseph	E.N.S.E.R.G.
CHOVET Alain	E.N.S.E.R.G.
CHEHIKIAN Alain	E.N.S.E.R.G.
DOLMAZON Jean-Marc	E.N.S.E.R.G.
HERAULT Jeanny	E.N.S.E.R.G.
MONLLOR Christian	E.N.S.E.R.G.
BORNARD Guy	E.N.S.I.E.G.
DESCHIZEAU Pierre	E.N.S.I.E.G.
GLANGEAUD François	E.N.S.I.E.G.
KOFMAN Walter	E.N.S.I.E.G.
LEJEUNE Gérard	E.N.S.I.E.G.
MAZUER Jean	E.N.S.I.E.G.
PERARD Jacques	E.N.S.I.E.G.
REINISCH Raymond	E.N.S.I.E.G.
ALEMANY Antoine	E.N.S.H.G.
BOIS Daniel	E.N.S.H.G.
DARVE Félix	E.N.S.H.G.
MICHEL Jean-Marie	E.N.S.H.G.
OBLÉD Charles	E.N.S.H.G.
ROWE Alain	E.N.S.H.G.
VAUCLIN Michel	E.N.S.H.G.
WACK Bernard	E.N.S.H.G.
BERT Didier	E.N.S.I.M.A.G.
CALMET Jacques	E.N.S.I.M.A.G.
COURTIN Jacques	E.N.S.I.M.A.G.
COURTOIS Bernard	E.N.S.I.M.A.G.
DELLA DORA Jean	E.N.S.I.M.A.G.
FONLUPT Jean	E.N.S.I.M.A.G.
SIFAKIS Joseph	E.N.S.I.M.A.G.
CHARUEL Robert	U.E.R.M.C.P.P.
CADET Jean	C.E.N.G.
COEURE Philippe	C.E.N.G. (LETI)

.../...

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

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

LABORATOIRES EXTERIEURS

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

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

~~Ordinatum
Errare Humanum est.~~

... ainsi, au lieu de ce grand nombre de préceptes dont la logique est composée, je crus que j'aurais assez des quatre suivants pourvu que je prisse une ferme et constante résolution de ne manquer pas une seule fois à les observer. Le premier était de ne recevoir jamais aucune chose pour vraie, que je ne la connusse évidemment être telle : c'est-à-dire, d'éviter soigneusement la précipitation et la prévention; et de ne comprendre rien de plus en mes jugements, que ce qui se présenterait si clairement et si distinctement à mon esprit, que je n'eusse aucune occasion de le mettre en doute.

Le second, de diviser chacune des difficultés que j'examinerais, en autant de parcelles qu'il se pourrait, et qu'il serait requis pour les mieux résoudre.

Le troisième, de conduire par ordre mes pensées, en commençant par les objets les plus simples et les plus aisés à connaître, pour monter peu à peu, comme par degrés, jusqu'à la connaissance des plus composés; et supposant même de l'ordre entre ceux qui ne se précèdent point naturellement les uns les autres.

Et le dernier, de faire partout des dénombrements si entiers, et des revues si générales, que je fusse assuré de ne rien omettre.

AVANT-PROPOS

Je tiens à remercier,

- Monsieur Louis BOLLINET, Professeur à l'Université de Grenoble, pour l'honneur qu'il me fait en acceptant de présider le jury de cette thèse.
- Monsieur Bernard COURTOIS, Chargé de Recherche au CNRS et rapporteur de cette thèse, pour ses conseils, ses encouragements et sa confiance qui m'ont permis de mener à bien la présente étude.
- Monsieur François ANCEAU, Professeur à l'Institut National Polytechnique de Grenoble, pour m'avoir accueilli dans son Equipe de Recherche.
- Messieurs Jacob A. ABRAHAM, Professeur au Coordinated Science Laboratory de l'Université d'Illinois, Urbana (U.S.A.), Joseph BOREL, Directeur Technique à la Société EFCIS, Alain COSTES, Professeur à l'Institut National Polytechnique de Toulouse, qui ont bien voulu participer au jury de cette thèse, malgré leurs nombreuses occupations.
- tous les chercheurs de l'Equipe de Recherche en Architecture d'Ordinateurs, sans les travaux desquels ce document n'aurait pu voir le jour; et particulièrement Jean-Pierre SCHOELLKOPF Michaelis NICOLAIDIS et Christian ANTONINO, pour leur aide et leurs précieux conseils.
- les personnes qui ont contribué à la dactylographie de ce document : Brigitte VUADENS et plus particulièrement Evelyne RIOU qui a assuré presque la totalité de la dactylographie, ainsi que Clothilde CHALAND et Hélène DIAZ pour leur gentillesse et leur compétence tout au long de ce travail.
- le Service de Reprographie de l'IMAG : D. IGLESIAS ainsi que C. ANGUILLÉ, C. LABORIE, J.M. MOLLIER, P. MOUNET. La présentation de cette thèse est le reflet de leur compétence.

Cette thèse a été réalisée dans le cadre
d'un contrat de recherche soutenu par l'Agence de
l'Informatique (Action Permanente SURF , contrat
ADI: SURF n° 80/067).

AVERTISSEMENTS

Dans ce document, chacun des chapitres possède ses propres références, ses propres numéros de figures, de tables et éventuellement d'équations.

C'est délibérément, pour augmenter la clarté, que des résultats et étapes intermédiaires de certains calculs ont été supprimés. Mais, la totalité des résultats et les différentes étapes intermédiaires nécessaires, pourront être trouvées dans les rapports suivants, dans l'ordre des chapitres :

"Présentation du MC 68000"

Note technique, NT1-82, Octobre 1982.

"Modélisation du temps de détection des pannes de la périphérie et de pannes du coeur de la pastille du MC 68000"

Rapport de Recherche IMAG, RR n° 283, Janvier 1982.

"Test analytique de la partie opérative du microprocesseur MC 68000"

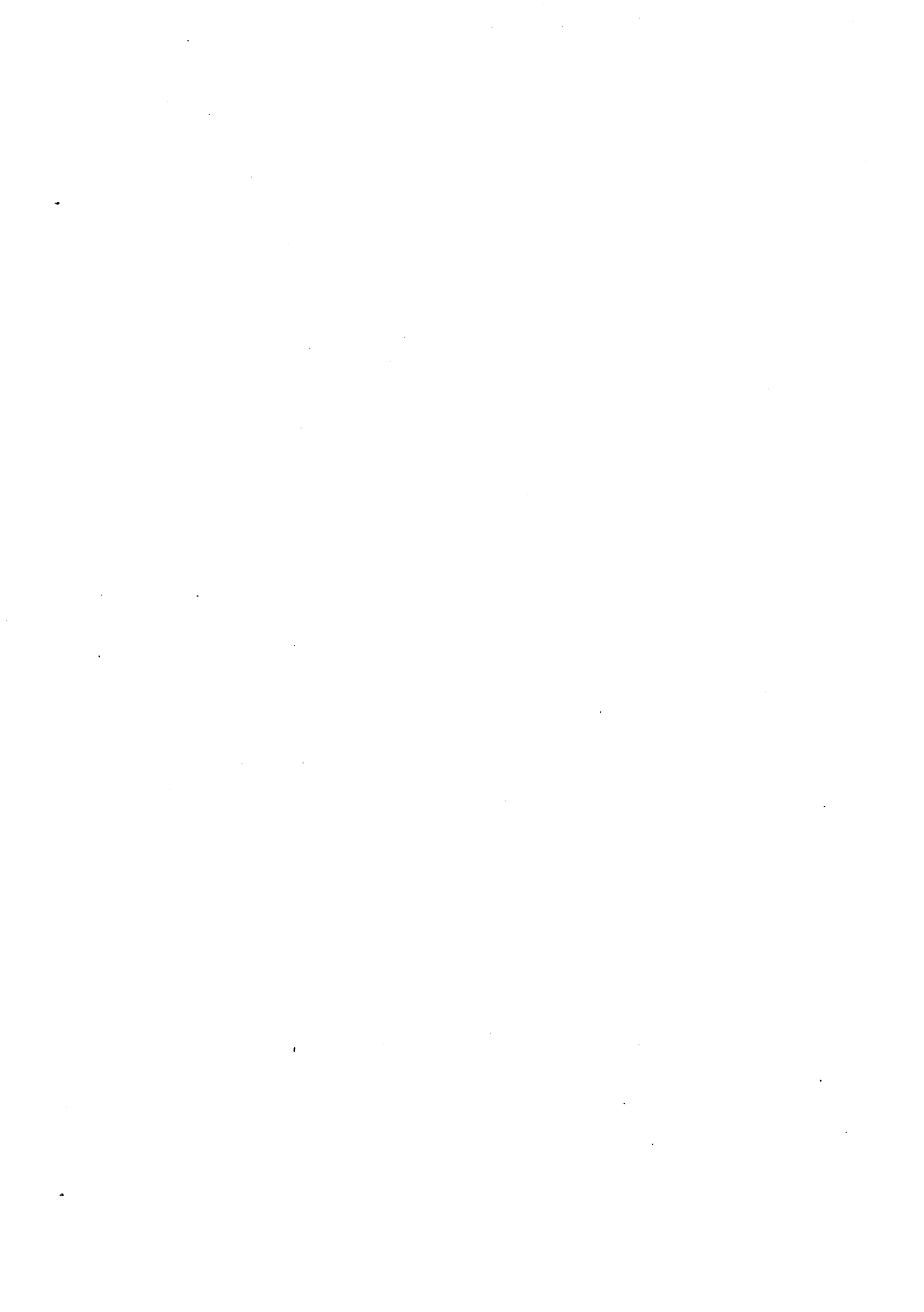
Rapport de Recherche IMAG, RR n° 361, Mars 1983.

"Le circuit MP82AS1 : Analyseur de signatures pour le test hors ligne de microprocesseurs"

Rapport de Recherche IMAG, RR n° 300, Avril 1982.

Un sommaire général figure en page 15, chaque chapitre étant précédé de son propre sommaire.

CHAPITRE I
INTRODUCTION



CHAPITRE I - INTRODUCTION

I	- CLASSIFICATION DES DIFFERENTS TYPES DE TEST	8
	I.1. Critère en ligne/hors ligne	9
	I.2. Critère continy/discontiny	9
	I.3. Critère in-sity/ex-sity	9
II	- METHODOLOGIE	10
	SOMMAIRE	15
	REFERENCES	17



La motivation de ce travail est la détection des pannes matérielles pouvant se produire à l'intérieur du microprocesseur MC 68000. Le but recherché est de détecter les pannes de ce microprocesseur sans utiliser de duplication ni de redondance massive. En effet, l'ajout de circuits annexes diminue la fiabilité du système, néanmoins cet inconvénient est contrebalancé par une détection immédiate.

L'idée de base des mécanismes de détection considérés ici, est que les fonctions réalisées par un microprocesseur sont si complexes et tellement rapidement exécutées que, si une panne se produit, les manifestations de désordre croîtront rapidement dans le système et quelque chose d'anormal apparaîtra. C'est en 1976 que F. ANCEAU exprima cette idée dans [ANC 76], et appela cet effet "catastrophe algorithmique". Les mécanismes suggérés par F. ANCEAU susceptibles de détecter l'apparition de désordres sont l'implémentation d'un chien de garde, la détection de l'apparition de codes illégaux et une protection mémoire fonction de zones à accès interdit ou uniquement destinées à être lues.

Une méthodologie permettant d'étudier l'efficacité de tels mécanismes a été développée par B. COURTOIS et appliquée au M 6800, tout en étant conscient que le mécanisme de protection mémoire est dépendant d'une application spécifique. Des résultats ont été ainsi donnés pour une application quelconque ne considérant que les mécanismes de chien de garde et de codes opérations invalides.

Un inconvénient du mécanisme de codes opérations invalides était pour le M 6800, la nécessité de réaliser à l'extérieur du microprocesseur cette détection, soit par des circuits standards [MON 79] soit par un circuit spécial [GUY 80]. Cet inconvénient n'existe pas pour le microprocesseur MC 68000 puisque ce détecteur est interne au microprocesseur. De plus, un quatrième mécanisme, également intégré

à l'intérieur de la pastille du MC 68000, permet de détecter les "adresses-erreurs", ces erreurs sont dues à l'implémentation particulière du MC 68000, qui a la possibilité d'adresser des octets, des mots ou des doubles mots, mais ne peut adresser ces deux derniers types que sur une adresse paire. L'efficacité de ces quatre mécanismes est étudiée dans ce document, bien que, de manière similaire au M 6800, des résultats soient donnés en ne considérant que les mécanismes de chien de garde, de codes opérations invalides, d'adresses-erreurs, de manière à pouvoir être utilisés pour toute application (l'efficacité étant alors meilleure que les bornes indiquées, et l'implémentation facilitée par le fait que le MC 68000 génère des signaux spécifiant que l'adressage se fait dans la zone programmes ou dans la zone données, et que l'accès est demandé par un utilisateur ou par le superviseur.

Le gain en circuits annexes apporté par la non-utilisation d'une redondance massive est contrebalancé par une détection non immédiate. C'est l'étude de ce temps de détection, et des moyens à mettre en oeuvre pour cette détection qui fait l'objet de cette étude.

Avant d'aborder la méthodologie appliquée pour cette étude, un rappel sur les différents types de test est nécessaire.

I. Classification des différents types de test :

La classification donnée par [COU 82] est reprise ici, 4 types de test peuvent être envisagés afin de détecter les pannes matérielles pouvant affecter l'intérieur d'un microprocesseur.

Ces quatre types de test se distinguent par l'environnement du microprocesseur d'une part, et par la façon d'exécuter les programmes de test d'autre part.

On distingue donc :

- le test en-ligne continu
- le test en-ligne discontinu
- le test hors-ligne in-situ
- le test hors-ligne ex-situ.

1.1. Critère en ligne/hors-ligne :

Un test est dit en-ligne lorsque la détection des pannes s'effectue pendant l'exécution du programme d'application.

Un test est dit hors-ligne lorsque le test s'exécute en dehors de tout programme d'application. Le microprocesseur n'exécute alors que le programme de test.

1.2. Critère continu/discontinu :

Ce critère ne s'explique que pour un test en-ligne puisqu'il définit les rapports existants entre les séquences de test et les séquences d'application.

Un test en ligne est continu lorsqu'il y a simultanéité du test et de l'exécution du programme d'application.

Un test en-ligne est discontinu lorsqu'il y a arrêt du programme d'application pour lancer le programme de test.

1.3. Critère in-situ/ex-situ :

Ce critère ne s'explique que pour un test hors-ligne puisqu'il définit l'observabilité des signaux sur les broches du circuit pendant l'exécution

du programme de test.

Un test hors-ligne est in-situ lorsque le circuit est dans son environnement normal de travail. Les signaux transitant sur les bus adresses et données ou les signaux de contrôle ne sont donc pas observables.

Un test hors-ligne est ex-situ lorsque tous les signaux sur les broches sont observables, le microprocesseur n'est pas dans son environnement de travail.

On remarque ainsi, d'après les critères définis, qu'un test en-ligne est de facto in-situ, et qu'un test hors-ligne est de facto continu (le programme d'application étant dans ce cas particulier : le programme de test).

Bien que les quatre types de test puissent être obtenus à partir des résultats de la présente étude, c'est un test en-ligne continu qui est le but premier recherché, bien qu'au chapitre IX soit abordé le principe d'une machine automatique de test basée sur les résultats de cette étude.

II. METHODOLOGIE :

Cette méthodologie destinée à permettre la quantification du temps de détection pour le test en-ligne continu, a été développée par B. COURTOIS et appliquée au microprocesseur M6800 [COU 81].

Cette étude va donc permettre de vérifier l'adéquation des modèles développés, et permettre la comparaison des résultats obtenus pour ces deux microprocesseurs.

En effet, bien que fabriqués tous deux par la firme MOTOROLA, ils ont six ans d'écart ce qui est considérable dans le domaine de la micro-informatique. La complexité du MC 68000, en nombre de transistors est 10 fois plus grande que celle du M 6800, du point de vue fonctionnel il possède 5 fois plus de registres, d'opérateurs, 2 fois plus de bus que son prédécesseur, et fonctionne 4 fois plus rapidement.

La méthodologie globale se décompose en six étapes ;

- i) déterminer, par simulation fonctionnelle le temps de détection consécutif à l'apparition de pannes dans la périphérie de la pastille (transistors de sorties, "bonds",....) ;
- ii) déterminer, par des études "papier", le temps de latence au niveau du microprocesseur (le temps pendant lequel il n'y a aucune conséquence de la panne), et la fréquence de l'apparition de l'évènement E ; "il y a une erreur sur le compteur ordinal", supposant qu'il n'y a pas de détection, pour les pannes de coeur de la pastille ;
- iii) déterminer, par simulation fonctionnelle, les caractéristiques de la détection consécutive à l'apparition d'un évènement E ;
- iv) extrapoler les résultats précédents, par modélisation, de façon à obtenir des résultats en fonction de la fréquence de l'évènement E ;
- v) déterminer, pour les pannes qui ne créent pas d'évènements E, des "programmes de test" qui seront exécutés périodiquement ;
- vi) déterminer les fréquences optimales de tous les phénomènes périodiques impliqués dans la détection, et/ou l'ordonnement ;

des procédures de test périodiques.

La première étape permet de vérifier qu'aucune panne n'affecte l'interface entre la machine et le monde extérieur, moyennant un certain temps de détection. Cette interface est constituée des amplificateurs d'entrée/sortie, des plots, des fils reliant les plots aux broches, et des broches elles-mêmes.

Les étapes 2 à 6 concernent le coeur de la pastille.

La seconde étape permet de déterminer la latence de la machine, c'est-à-dire le temps séparant l'occurrence d'une panne de la génération d'une erreur dans un registre ou la mémoire. Un registre, le compteur ordinal, sera particulièrement étudié du fait des conséquences que provoque une erreur dans ce registre.

La troisième étape permet de connaître le temps de détection consécutive à l'apparition d'une erreur sur le compteur ordinal, correspondant ainsi à une "latence de l'erreur".

La quatrième étape généralise les résultats de l'étape précédente afin de prendre en compte la fréquence de génération d'erreurs sur le compteur ordinal.

La cinquième étape détermine, pour les pannes ne générant pas de déséquence, les vecteurs de test nécessaires. Ces vecteurs peuvent être obtenus de deux façons différentes selon le fait que :

- le fonctionnement est considéré (test fonctionnel)
- la structure est considérée (test analytique).

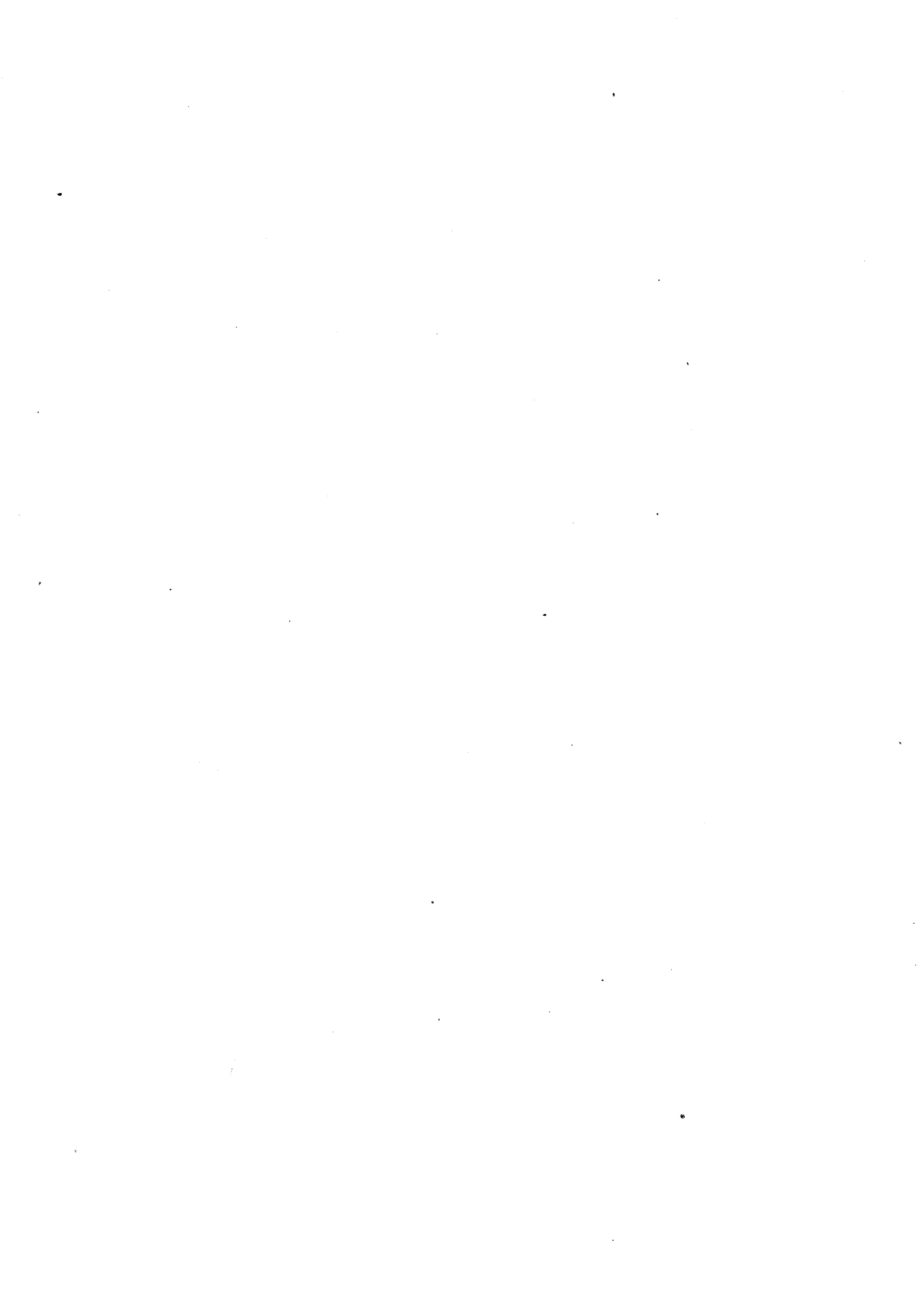
La partie opérative de la machine fera l'objet d'un test analytique, la

partie contrôle d'un test fonctionnel.

La dernière étape permet d'exprimer les fréquences, optimales en fonction des coûts et temps d'exécution et/ou l'ordonnement des différentes procédures de test périodiques.

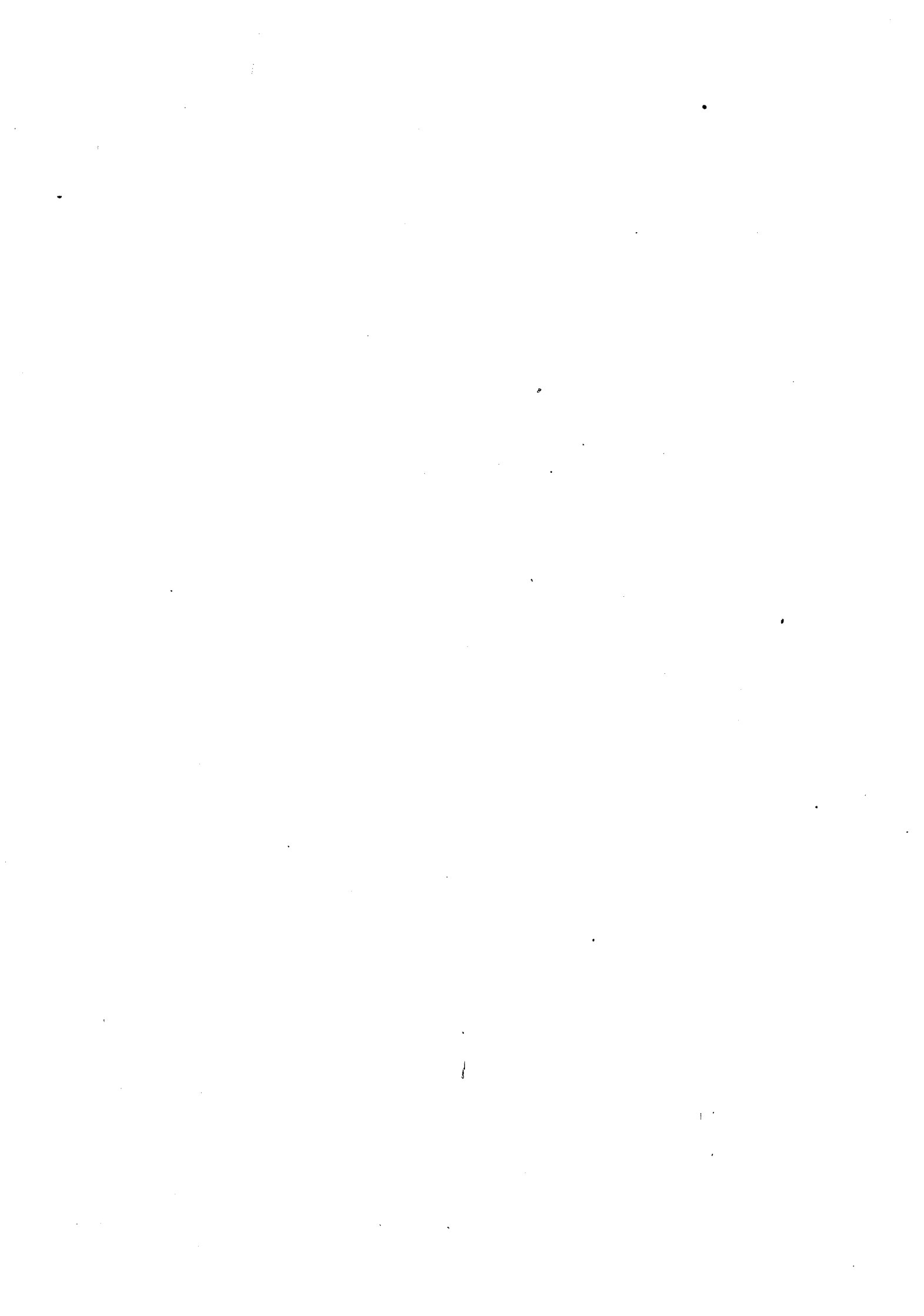
Enfin, un test hors-ligne obtenu simplement par concaténation de toutes les procédures développées (certaines n'étant pas utiles pour le test en-ligne) sera abordé par le biais d'une machine automatique de test ; les circuits conçus et développés pour cette application seront alors présentés.

Les différentes étapes constituent autant de chapitres dans cette étude. Le sommaire de ce document s'établit donc comme il suit.



S O M M A I R E

- CHAPITRE I. - INTRODUCTION
- CHAPITRE II. - LE MC 68000
Présentation du microprocesseur.
- CHAPITRE III. - PERIPHERIE DE LA PASTILLE
Etape (i).
- CHAPITRE IV. - LATENCE D'ERREUR - LATENCE ET FREQUENCE DE
L'EVENEMENT E.
Etape (ii).
- CHAPITRE V. - COEUR DE LA PASTILLE CREANT E.
Etape (iii)
Etape (iv).
- CHAPITRE VI. - COEUR NE CREANT PAS E. PARTIE OPERATIVE
Etape (v).
- CHAPITRE VII. - COEUR NE CREANT PAS E. PARTIE CONTROLE.
Etape (v).
- CHAPITRE VIII. - FREQUENCES OPTIMALES.
ORDONNANCEMENT DES PROCEDURES.
Etape (vi).
- CHAPITRE IX. - CIRCUITS D'ANALYSE DE SIGNATURE MACHINE AUTOMATIQUE
DE TEST
(hors méthodologie).
- CHAPITRE X. - CONCLUSION.

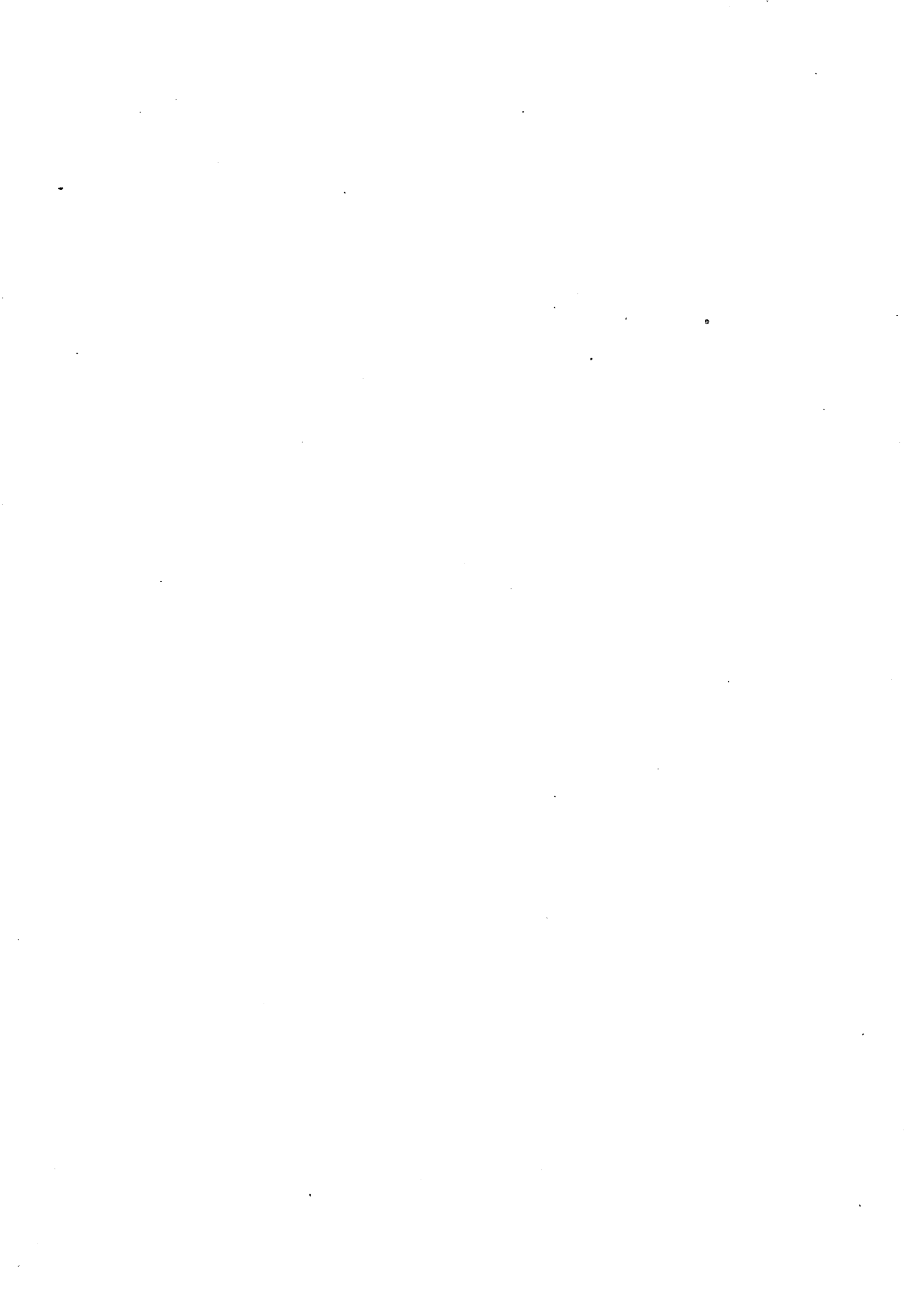


REFERENCES

- [ANC 76] ANCEAU F.
"Sécurité dans les systèmes complexes : application aux autocommutateurs téléphoniques".
Note Interne IMAG, Décembre 1976.
- [COU 81] COURTOIS B.
"Test et LSI".
Thèse de doctorat d'Etat, Grenoble 1981.
- [COU 82] COURTOIS B., EYNARD J.P., GEAY F., MARCHAL P.,
POSTIGO C.
"Test en-ligne et hors-ligne : classification et résultats récents".
3ème colloque international sur la Fiabilité et la Maintainabilité, Toulouse, Octobre 1982.
- [GUY 80] GUYOT A. & HARRIS N.
"Motorola M 6800 coprocessor (COP)".
Computer Systems Laboratory, Department of Electrical Engineering, Stanford University, 1980.
- [MON 79] MONNIER J., DAMIEN M., CHEVREAU B.
"Carte de détection d'erreurs pour microprocesseurs".
Electronique Applications Industrielles, n° 265 du 15.3.79.



CHAPITRE II
LE MC 68000
PRÉSENTATION DU MICROPROCESSEUR



CHAPITRE II - LE MC 68000

PRESENTATION DU MICROPROCESSEUR

I	- INTRODUCTION	23
II	- LES RESSOURCES	23
	1. Modèle de programmation	23
	2. Organisation des registres	25
	2.1. Registres données	25
	2.2. Registres adresses	26
	2.3. Le pointeur de pile système	26
III	- LES INSTRUCTIONS	27
	1. Format de l'instruction	27
	2. Jeu d'instructions	29
	2.1. Opérations arithmétiques entières	29
	2.2. Opérations logiques	30
	2.3. Décalages	31
	2.4. Manipulation de bits	31
	2.5. Mouvement de données	32
	2.6. Opération BCD	32
	2.7. Contrôle de programmes	32
	2.8. Contrôle système	33
IV	- L'ADRESSAGE	34
	1. Le bus adresse	34
	2. Modes d'adressage	34
	3. Référence programme/Référence donnée	35
	4. Adresse effective	36
	4.1. Mode registre direct	36
	4.2. Mode d'adressage mémoire	37
	4.3. Mode d'adressage spécial	40
	4.4. Références implicites	43
	4.5. Codage de l'adresse effective	43

5. Les catégories d'adressage	44
5.1. La catégorie données.....	44
5.2. La catégorie mémoire	44
5.3. La catégorie contrôle	44
5.4. La catégorie altérable	44
V - LES DONNEES	45
1. Types de données supportés	46
2. Organisation des données en mémoire	46
VI - LES SIGNAUX ET BUS	48
1. Le bus adresse	49
2. Le bus donnée	49
3. Le bus contrôle	49
3.1. Contrôle du bus asynchrone	49
3.2. Contrôle de l'arbitrage du bus	51
3.3. Contrôle des interruptions	51
3.4. Contrôle système	52
3.5. Contrôle de la périphérie M 6800	53
3.6. Etat du processeur	53
VII - CONCLUSION	54
NOTE BIBLIOGRAPHIQUES	55
REFERENCES	57

I - INTRODUCTION

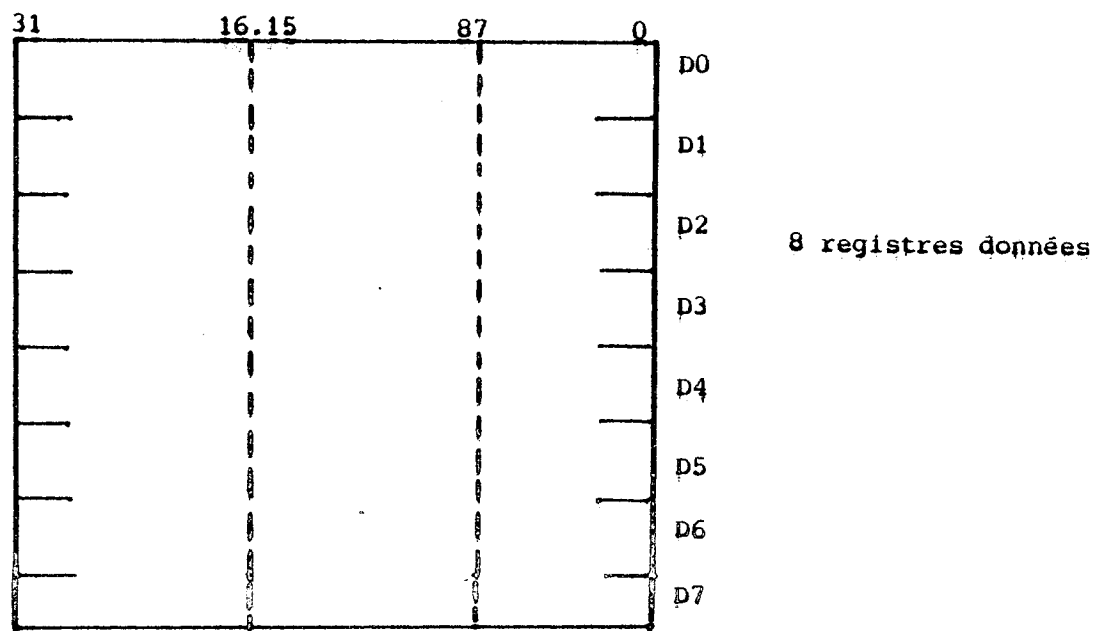
Ce chapitre présente le microprocesseur MC 68000 fabriqué par la firme américaine MOTOROLA. Il fait partie de la "nouvelle vague" des processeurs 16 bits en un seul circuit. En fait, il est prévu pour une largeur de travail de 32 bits.

La technologie employée est la technologie H-MOS (high density MOS) qui permet de réduire la surface du circuit à moins de la moitié et la consommation à moins du quart de ce qu'elle aurait été en technologie N-MOS.

Le circuit se présente sous la forme d'un boîtier unique de 82 mm de longueur, pour 23 mm de largeur, il est pourvu de 64 broches réalisant l'interface avec le monde extérieur. La surface de la puce est de 44,40 mm² (7,19 mm x 6,25 mm), la plus petite porte à une surface de 27 μm² (3,8 μm x 7 μm). Une seule alimentation de 5 volts lui est nécessaire, et il ne consomme que 1,2 W. Pour une horloge de 8 MHz.

II - LES RESSOURCES

1. Modèle de programmation



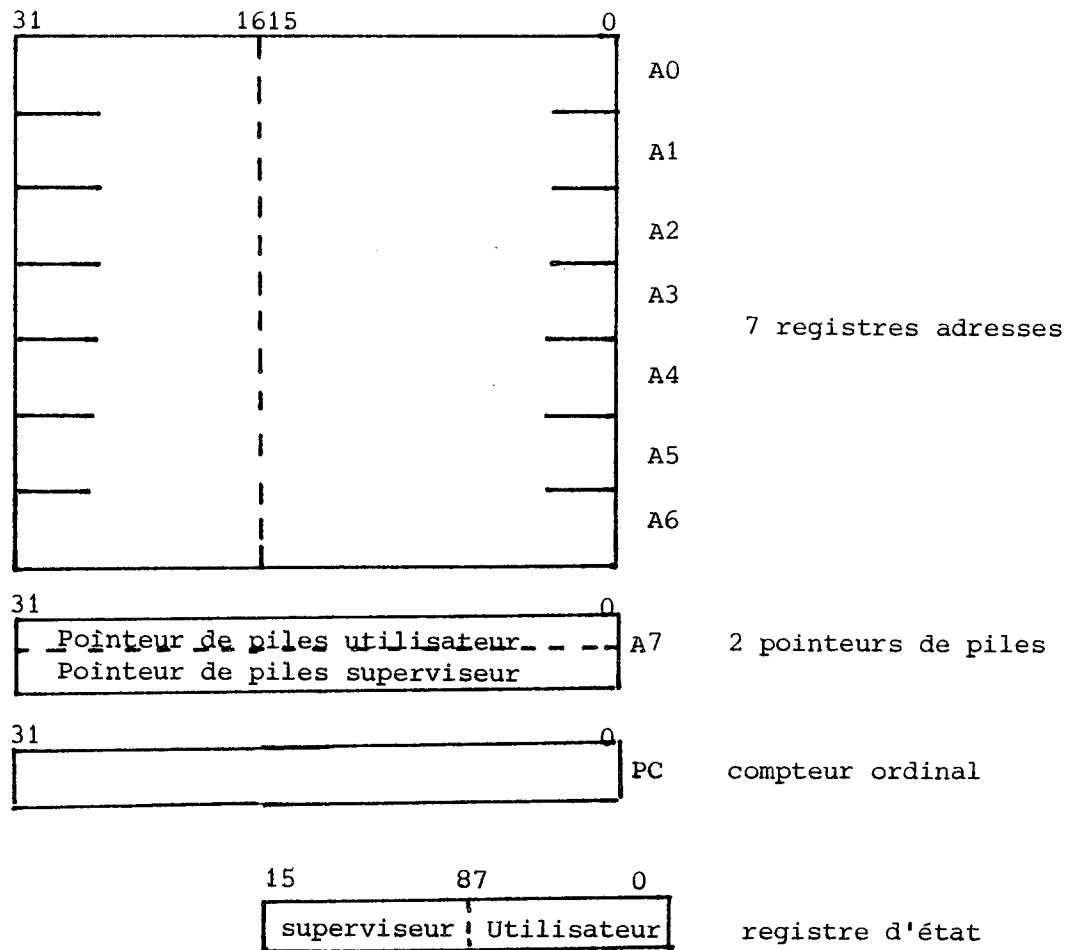


Fig. 1 Modèle du M 68000

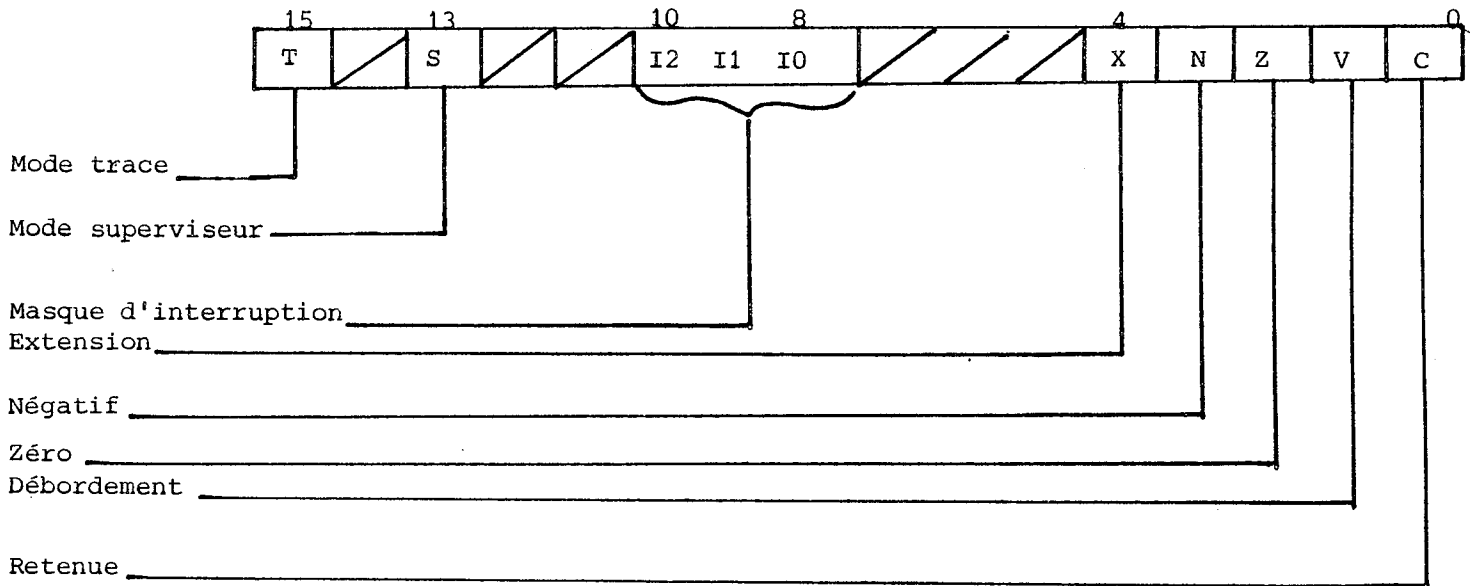


Fig. 2 Structure interne du registre d'état

Le M68000 dispose de 17 registres de 32 bits qui sont répartis entre :

- 8 registres de données (D0 à D7)
- 7 registres d'adresses (A0 à A6)
- 2 registres pointeurs de piles mutuellement exclusifs (A7).

Ces 17 registres peuvent servir d'index.

Ce processus dispose en outre d'un compteur ordinal (PC) de 32 bits dont 24 seulement sont utilisés dans la première version du circuit étudié, ainsi qu'un registre d'état ou mot d'état de 16 bits, composé lui-même de :

- un bit indiquant le mode trace : T
- un bit indiquant le mode superviseur : S
- un masque d'interruption de 3 bits (8 niveaux)
- le code condition :
 - . extension : X
 - . négatif : N
 - . zéro : Z
 - . débordement : V
 - . retenue : C

2. Organisation des registres

2.1. Registres données

Les huit registres données permettent de travailler sur des données de 1, 8, 16 ou 32 bits. Chaque registre possédant 32 bits, les données se trouvent :

- pour un octet dans les 8 bits poids faibles du registre
- pour un mot dans les 16 bits poids faibles du registre
- pour un double mot dans le registre complet.

Pour une donnée de 1 bit : celui-ci est précisé par son numéro de 0 à 31 sachant que le bit 0 est le moins significatif et le bit 31 le plus significatif.

On peut remarquer que lorsqu'un registre **données** est utilisé en source ou en destination d'une opération, seule la "portion" poids faible correspondant à la taille de la donnée, est modifiée, le reste n'est ni utilisé ni changé.

2.2. Registres adresses

Chaque registre adresse, ainsi que les pointeurs de piles ont 32 bits et supportent 32 bits d'adresses. Aucun ne supporte des opérandes d'un octet. Toutefois lorsque le registre est utilisé comme source, un mot ou un double mot est accessible, selon la taille de l'opérande. Lorsqu'un registre adresse est utilisé comme destination, le registre complet (32 bits) est affecté sans s'occuper de la taille de l'opérande. Si la taille de ce dernier, était un mot, il y a extension préalable sur 32 bits avant de ranger le résultat.

2.3. Le pointeur de piles système

Le registre adresse n° 7 (A7) est le pointeur de piles système (SP). Il est utilisé par de nombreuses instructions, mais n'a pas en fait d'existence propre. Il permet la création et utilisation de piles et de files pour l'utilisateur grâce aux différents modes d'adressage.

Le registre A7 en fait qualifie deux pointeurs :

- le pointeur de piles superviseur (SSP)
- le pointeur de piles utilisateur (USP).

Si le bit S du registre d'état indique que l'on est dans le mode superviseur ($S = 1$), le pointeur de piles superviseur (SSP) est le pointeur de piles actif et le pointeur de piles utilisateur ne peut être référencé. Si le bit S indique que l'on est dans le mode utilisateur ($S = 0$) le pointeur de piles utilisateur (USP) est le pointeur de piles actif, et le pointeur de piles superviseur (SSP) ne peut être référencé.

Chaque pointeur de piles remplit ses piles depuis les adresses les plus hautes jusqu'aux adresses les plus basses.

III - LES INSTRUCTIONS

1. Format de l'instruction

Les instructions sont composées de 1 à 5 mots de 16 bits (cf. fig. 3). La longueur de l'instruction et les opérations à exécuter sont précisées dans le premier mot appelé mot opération. Les mots suivants spécifient les opérandes ou leurs adresses.

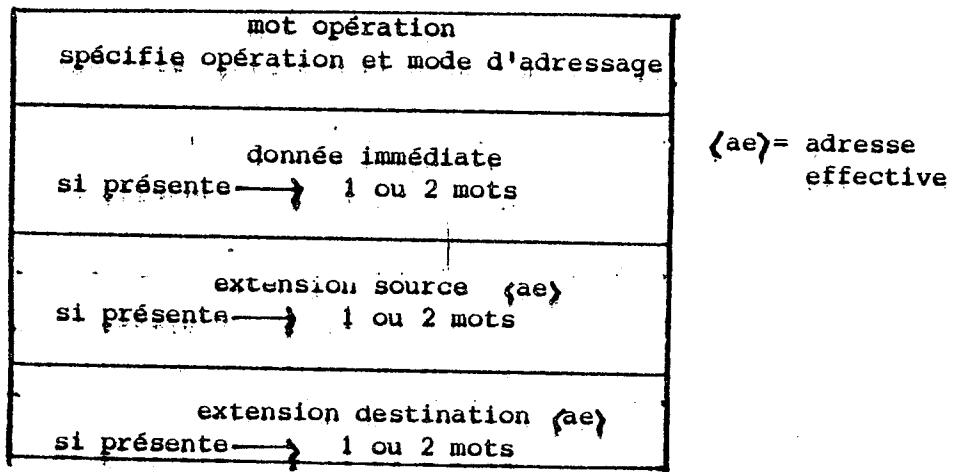


Fig. 3 Format de l'instruction

Le mot opération du M68000 contient deux types d'information :

- le type de fonction à exécuter
- l'emplacement de (s) l'opérande(s) sur lequel exécuter la fonction.

L'instruction précise l'emplacement de l'opérande de trois façons :

- spécification des registres : le champ registre de l'instruction spécifie le numéro du registre à utiliser ; d'autres champs dans l'instruction précisent s'il s'agit d'un registre donnée ou d'un registre adresse et comment l'utiliser ;
- adresse effective : le champ adresse effective donne la définition de la donnée ou de son adresse, au moyen des modes d'adressage ;

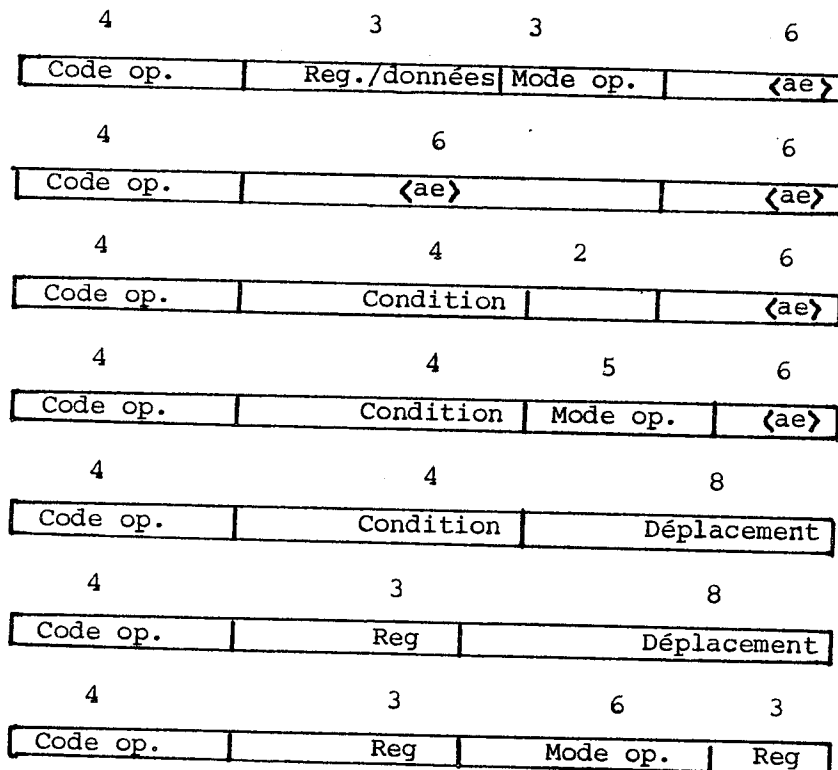
- référence implicite : certaines instructions impliquent l'utilisation de registres particuliers ou de données immédiates.

La taille des opérandes peut être :

- d'un octet (8 bits)
- d'un mot (16 bits)
- d'un double mot (32 bits).

Elle peut être soit précisée explicitement au moyen du champ taille, soit implicite pour l'instruction considérée. Les différents champs utilisés sont présentés à la figure 4.

Code opération sur 4 bits :

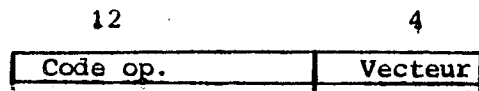


<ae> = adresse effective

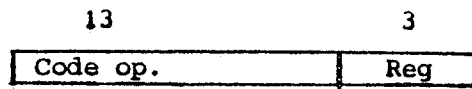
Code opération sur 8 bits :



Code opération sur 12 bits :



Code opération sur 13 bits ;



Code opération sur 16 bits :

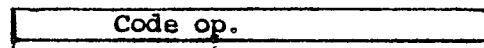


Fig. 4 Différents champs utilisés

2. Jeu d'instructions

Ce jeu d'instructions qui peut paraître mince à première vue avec ses 56 types d'instructions, permet avec l'utilisation des différents modes d'adressage, d'obtenir plus de 1 000 instructions différentes (192 pour le M6800).

2.1. Opérations arithmétiques entières

	<u>Nombre de codes valides</u>
. ADD	
- add (addition binaire)	2 408
- adda (addition d'adresses)	976
- addq (addition rapide)	1 328
- addi (addition immédiate)	150
- addx (addition avec extension)	384
. CLR (remise à zéro)	150

	<u>Nombre de codes valides</u>
. CMP	
- cmp (comparaison)	1 400
- cmpa (comparaison d'adresses)	976
- cmpm (comparaison de mots mémoire)	192
- cmpi (comparaison immédiate)	150
. DIVS (division signée)	424
. DIVU (division sans signe)	424
. EXT (extension du signe)	16
. MULS (multiplication signée)	424
. MULU (multiplication sans signe)	424
. NEG	
- neg (négation)	150
- negx (négation avec extension)	150
. SUB	
- sub (soustraction)	2 408
- suba (soustraction d'adresses)	976
- subq (soustraction rapide)	1 328
- subi (soustraction immédiate)	150
- subx (soustraction avec extension)	384
. TST (test d'un opérande)	<u>150</u>
Total	15 522

2.2. Opérations logiques

. AND	
- and (et logique)	2 280
- andi (et logique immédiat)	150
. EOR	
- eor (ou exclusif)	1 200
- eori (ou exclusif immédiat)	150

	<u>Nombre de codes valides</u>
. OR	
- or (ou logique)	2 280
- ori (ou logique immédiat)	150
. NOT (complément logique)	150
. SCC (mise à faux selon condition : 16)	<u>800</u>
Total	7 160

2.3. Décalages

. AS	
- asl (arithmétique à gauche)	426
- asr (arithmétique à droite)	426
. LS	
- lsl (logique à gauche)	426
- lsr (logique à droite)	426
.. RO	
- rol (rotation à gauche)	426
- ror (rotation à droite)	426
. ROX	
- roxl (rotation à gauche avec extension)	426
- roxr (rotation à droite avec extension)	<u>426</u>
Total	3 408

2.4. Manipulation de bits

. BCHG (changement d'un bit)	450
. BCLR (effacement d'un bit)	450
. BSET (mise à un d'un bit)	450
. BTST (test d'un bit)	<u>476</u>
Total	1 826

Nombre de
codes valides

2.5. Mouvement de données

. EXG (échange de registres)	192
. LEA (chargement d'adresse effective)	224
. LINK (sauvegarde d'une pile)	8
. MOVE	
- move (transfert de données)	8 750
- movea (transfert d'adresses)	976
- moveq (transfert rapide)	2 048
. MOVEM (transfert multiple de r \grave{e} gistres)	140
. MOVEP (transfert alterné)	256
. PEA (rangement d'une adresse effective)	28
. SWAP (permutation P. faible p. fort d'un registre)	8
. UNLK (retour de pile)	<u>8</u>
Total	12 638

2.6. Opérations BCD

. ABCD (addition DCB)	128
. SBCD (soustraction DCB)	128
. NBCD (complément DCB)	<u>50</u>
Total	306

2.7. Contrôle de programmes

- conditionnel :

. Bcc (branchement sur condition : 14)	3 584
. DBcc (décrément test et branchement sur condition : 16)	128

- inconditionnel :

. BRA (branchement)	256
. BSR (branchement à un sous-programme)	256
. JMP (saut)	28

Nombre de
codes valides

. JSR (saut à un sous-programme)	28
. TAS (test et mise à faux)	50
. RTR (retour de sous-programme et restauration)	1
. RTS (retour de sous-programme)	1
. NOP (pas d'opération)	<u>1</u>
Total	4 333

2.8. Contrôle système

- privilégié :

. RESET (réinitialisation extérieur)	1
. RTE (retour d'une exception)	1
. STOP (arrêt)	1
. ORI to SR (ou logique avec registre d'état)	1
. MOVE USP (transfert avec pointeur de pile utilisateur)	16
. ANDI to SR (et logique avec registre d'état)	1
. EORI to SR (ou exclusif avec registre d'état)	1
. MOVE to SR (transfert vers registre d'état)	53
. MOVE from SR (transfert depuis registre d'état)	50

- Trap :

. TRAP (branchement d'exception)	16
. TRAPV (branchement d'exception si débordement)	1
. CHK (contrôle de bornes)	424

- Code condition :

. ANDI to CCR (et logique)	1
. EORI to CCR (ou exclusif)	1
. MOVE to CCR (transfert vers CCR)	53
. OR to CCR (ou logique)	<u>1</u>

Total 622

On obtient donc :

45 815 codes valides et donc (70 %)	
19 721 codes invalides " (30 %)	
<hr/>	
65 536 codes possibles	

On peut déjà remarquer que la proportion de codes invalides est supérieure à celle de 6800 (59 parmi 256) soit environ 23 %.

IV - L'ADRESSAGE

Le but d'un processus étant de traiter des données, la façon d'accéder à ces données tient une grande part dans la puissance des instructions et les facilités de programmation. Le M6800 possédait 5 modes d'adressage. Le M68000 en possède quant à lui 14. La largeur du bus adresse est de 23 bits au lieu de 16 pour le M6800.

1. Le bus adresse

Le bus adresse dans la première implémentation du microprocesseur ne possède donc que 23 bits, alors que les registres adresses, le compteur ordinal et les opérations qui leur correspondent sont effectuées sur 32 bits. Malgré tout, la puissance d'adressage atteint 16 Mega-octets (16 777 216 octets).

De plus on peut séparer l'espace : des programmes utilisateurs, des données utilisateurs, des programmes superviseurs, des données superviseurs ; ce qui multiplie par 4 l'espace total adressable qui est donc de 64 Mega-octets (en ne disant que 64 Mega octets, on ne mentionne pas les chiffres restant représentant 48 fois l'espace adressable du 6800).

2. Modes d'adressage

Nous allons juste énumérer les différents modes d'adressage qui sont explicités dans le paragraphe suivant (§ 4).

- Registre direct :
 - . registre donnée direct
 - . registre adresse direct
- Registre (adresse) indirect :
 - . registre adresse indirect
 - . registre adresse indirect postincrémenté
 - . registre adresse indirect prédécémenté
 - . registre adresse indirect plus déplacement (16 bits)
 - . registre adresse indirect indexé plus déplacement (8 bits)
- Adressage absolu :
 - . absolu court (0 - 64 k)
 - . absolu long (0 - 16 Mega)
- Immédiat :
 - . immédiat
 - . immédiat rapide
- Relatif (au compteur ordinal) :
 - . pc relatif plus déplacement
 - .. pc relatif indexé plus déplacement
- Implicite :
 - . registre implicite (SR, USP, SSP, PC).

3. Référence programme / Référence donnée

Les références mémoire sont séparées en deux classes :

- les références programmes : référencent la partie de mémoire contenant le programme ;
- les références données ; référencent la partie de mémoire contenant les données.

Les opérandes lus se trouvent en général dans la mémoire donnée et donc de référence donnée. Tous les opérandes écrits sont dans la référence donnée.

4. Adresse effective (notée "ae")

La plupart des instructions spécifient l'emplacement de l'opérande par l'utilisation du champ adresse effective dans le mot opération.

L'adresse effective est composée de deux champs de 3 bits (cf. fig. 5) :

- le champ mode
- le champ registre

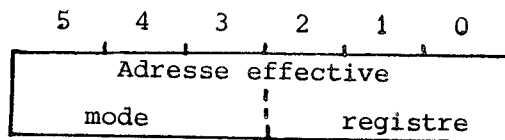


Fig. 5 Le champ adresse effective

Les valeurs du champ mode permettent de sélectionner parmi les différents modes. Le champ registre contenant le numéro du registre à utiliser.

L'adresse effective peut nécessiter des informations supplémentaires pour définir complètement l'opérande. Ces informations supplémentaires sont appelées extensions de l'adresse effective (cf. fig. 3) et occupent le (ou les) mot(s) suivant(s) l'instruction et sont considérés comme partie intégrante de l'instruction.

Les modes d'adresse effective sont groupés en trois catégories ;

- registres directs
- adressage mémoire
- adressage spécial.

4.1. Mode registre direct

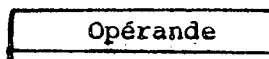
Ce mode d'adresse effective spécifie que l'opérande est l'un des 16 registres adresses ou données.

- Registre donnée direct (noté Dn) :

L'opérande est dans le registre donnée dont le numéro est spécifié par le champ registre de l'adresse effective.

- . génération: EA = Dn
- . mode = 000
- . registre ; n

. Registre donnée Dn :

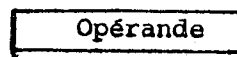


- Registre adresse direct (noté An) :

L'opérande est dans le registre adresse dont le numéro est spécifié par le champ registre de l'adresse effective.

- . génération: EA = An
- . mode = 001
- . registre = n

. Registre adresse An :



4.2. Mode d'adressage mémoire

Ce mode d'adresse effective précise que l'opérande est en mémoire et fournit donc l'adresse spécifique de l'opérande.

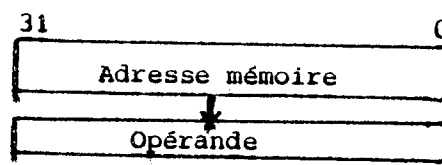
- Registre adresse indirect (noté An) :

L'adresse de l'opérande se trouve dans le registre adresse dont le numéro est spécifié par le champ registre de l'adresse effective. La référence est classée comme référence donnée, à l'exception des instructions "jump" et "jump to subroutine", pour lesquelles elle est classée comme référence programme.

- . génération ; EA = (An)
- . mode = 010
- . registre = n

. Registre adresse An :

Adresse mémoire

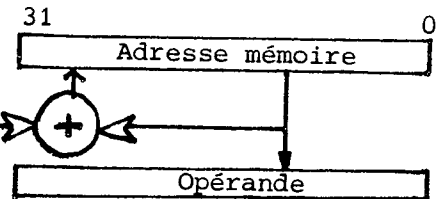


- Registre adresse indirect postincrémenté (noté (An)+) :

L'adresse de l'opérande se trouve dans le registre adresse dont le numéro est spécifié par le champ registre de l'adresse effective. Après utilisation de l'adresse, le registre est incrémenté de 1, 2 ou 4 selon que la taille de l'opérande est l'octet, le mot ou le double mot. Si le registre est le pointeur de pile système (A7) et que la taille soit d'un octet, le registre est incrémenté de 2 au lieu de 1 pour continuer à pointer sur la borne d'un mot (adresse paire). Cette référence est classée comme référence donnée.

- . génération : $EA = (An)$
 $An = An + N$
- . mode = 011
- . registre = n

- . Registre adresse An
- longueur de l'opérande (1, 2 ou 4)
- Adresse mémoire

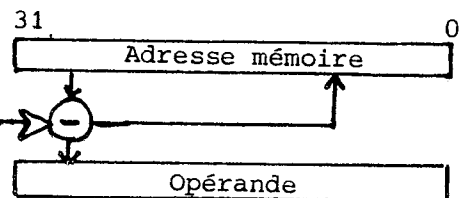


- Registre adresse indirect prédécémenté (noté -(An)) :

L'adresse de l'opérande se trouve dans le registre adresse dont le numéro est spécifié par le champ registre de l'adresse effective. Avant utilisation de cette adresse, le registre est décrétementé de 1, 2 ou 4 selon que la taille de l'opérande est l'octet, le mot ou le doublé mot. Si le registre est le pointeur de pile système (A7) et que la taille soit d'un octet, le registre est décrétementé de 2 au lieu de 1 pour continuer à pointer sur la borne d'un mot (adresse paire). Cette référence est classée comme référence donnée.

- . génération : $An = An - N$
 $Ea = (An)$
- . mode = 100
- . registre = n

- . Registre adresse Au
- longueur de l'opérande (1, 2, ou 4)
- Adresse mémoire



- Registre adresse indirect avec déplacement (noté d(An)) :

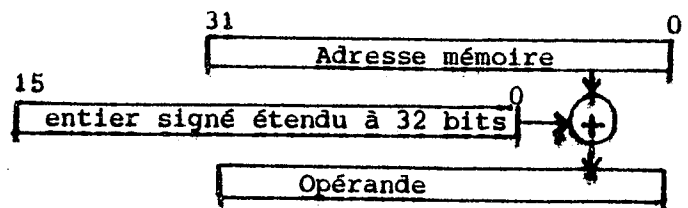
Ce mode d'adressage nécessite un mot d'extension. L'adresse de l'opérande est la somme de l'adresse contenue dans le registre adresse dont le numéro est spécifié dans le champ registre de l'adresse effective et de l'extension à 32 bits du déplacement contenu dans les 16 bits du mot d'extension. Cette référence est classée comme référence donnée à l'exception des instructions "jump" et "jump to subroutine" pour lesquelles elle est classée comme référence programme.

- . génération : $EA = (An) + d$
- . mode = 101
- . registre = n

. Registre adresse An

Déplacement

Adresse mémoire



- Registre adresse indirect indexé plus déplacement (noté d(An, Ri)) :

Ce mode d'adressage nécessite un mot d'extension dont le format est le suivant :

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
D/A	Reg				W/L	0	0	0	Déplacement						

D/A = $\begin{cases} 0 & \text{registre donnée} \\ 1 & \text{registre adresse} \end{cases}$

Reg = numéro du registre d'index

W/L = $\begin{cases} 0 & = \text{extension de 16 bits poids faible de l'index} \\ 1 & = \text{32 bits de l'index} \end{cases}$

Déplacement = signé sur 8 bits.

Fig. 6 Extension de l'index

L'adresse de l'opérande est la somme de l'adresse contenue dans le registre adresse dont le numéro est spécifié dans le champ registre de l'adresse effective, de l'extension à 32 bits du déplacement signé occupant les 8 bits poids faible du mot d'extension et du contenu du registre d'index. La référence est classée

comme référence donnée à l'exception des instructions "jump" et "jump to subroutine" pour lesquelles elle est classée comme référence programme.

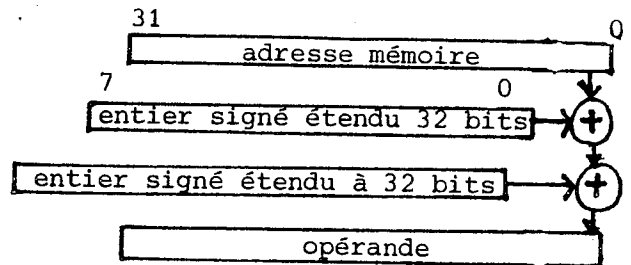
- . génération : $EA = (An) + (Ri) + d$
- . mode = 110
- . registre = n

. Registre adresse An

Déplacement

Registre d'index

Adresse mémoire



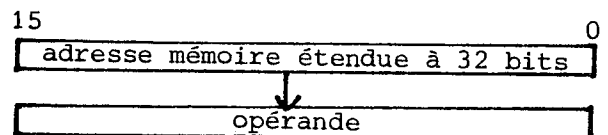
4.3. Mode d'adressage spécial

Ce mode d'adressage utilise le champ registre non plus pour préciser un numéro de registre mais pour spécifier un mode spécial d'adressage.

- Adresse absolue courte (noté xxx.w) :

Ce mode d'adressage nécessite un mot d'extension. L'adresse de l'opérande est contenue dans le mot extension. L'adresse de 16 bits est étendue à 32 bits avant d'être utilisée. Cette référence est classée comme référence donnée à l'exception des instructions "jump" et "jump to subroutine" pour lesquelles elle est classée comme référence programme.

- . génération : EA donnée
- . mode = 111
- . registre = 000
- . mot d'extension
- . adresse mémoire

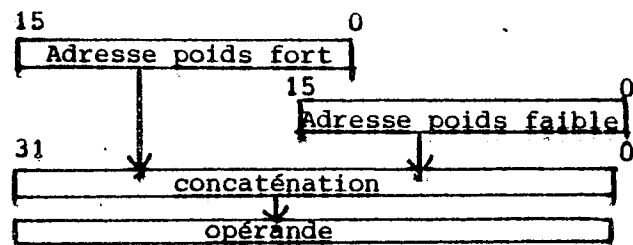


- Adresse absolue longue (notée : xxx.L) :

Ce mode d'adressage nécessite deux mots d'extension. L'adresse de l'opérande est développée par concaténation des mots d'extension. La partie poids fort de l'adresse est contenue dans le

mot d'extension. La partie poids faible est contenue dans le second mot d'extension. Cette référence est classée comme référence donnée à l'exception des instructions "jump" et "jump to subroutine" pour lesquelles elle est classée comme référence programme.

- . génération ; EA donnée
- . mode = 111
- . registre = 001
- . premier mot d'extension
- . second mot d'extension
- . adresse mémoire

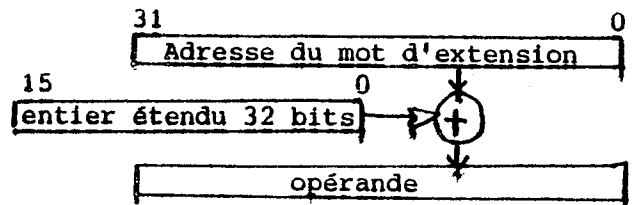


- Compteur ordinal plus déplacement (noté d(PC)) :

Ce mode d'adressage nécessite un mot d'extension. L'adresse de l'opérande est la somme du compteur ordinal (PC) et de l'extension sur 32 bits du déplacement sur 16 bits contenu dans le mot d'extension. La valeur du compteur ordinal est l'adresse du mot d'extension. Cette référence est classée comme référence programme.

- . génération ; EA = (PC) + d
- . mode = 111
- . registre = 010

- . compteur ordinal
- . mot d'extension
- . adresse mémoire

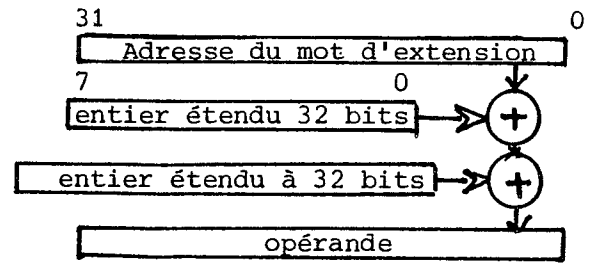


- Compteur ordinal indexé plus déplacement (note d (PC, Ri)) :

Ce mode d'adressage nécessite un mot d'extension (cf. fig. 6), L'adresse de l'opérande est la somme du compteur ordinal (PC), de l'extension sur 32 bits du déplacement contenu dans les 8 bits poids faible du mot d'extension et du contenu du registre d'index.

La valeur du compteur ordinal est l'adresse du mot d'extension.
 Cette référence est classée comme référence programme.

- . génération : $EA = (PC) + (Ri) + d$
- . mode = 111
- . registre = 011
- . compteur ordinal
- . déplacement
- . registre d'index
- . adresse mémoire

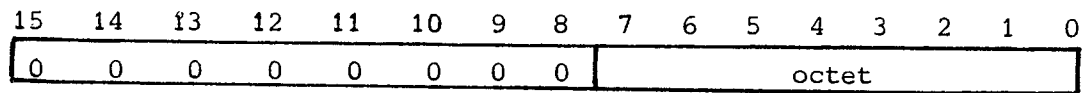


- donnée immédiate (notée #xxx) :

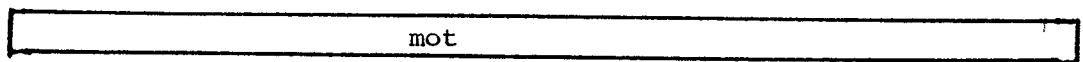
Ce mode d'adressage nécessite 1 ou 2 mots d'extension selon la taille de l'opération.

- ° opération sur des octets = l'opérande est l'octet poids faible du mot d'extension ;
- ° opération sur des mots = l'opérande est le mot d'extension ;
- ° opération sur des doubles mots = l'opérande est dans les deux mots d'extension. Les poids forts sont dans le premier mot d'extension, les poids faibles dans le second mot d'extension.

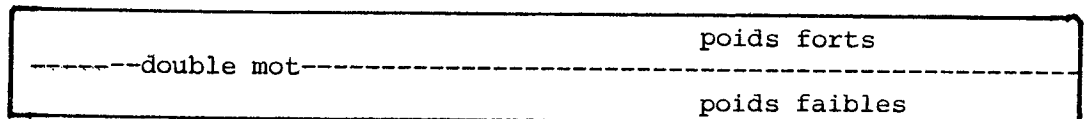
- . génération : opérande donné
- . mode = 111
- . registre = 100
- . formats du(des) mot(s) d'extension :



ou



ou



- Registre d'état ou de code condition (noté SR ou CCR) :

Un certain sous-ensemble d'instructions peut faire référence au registre d'état du processeur que ce soit complètement (SR = 16 bits) ou en partie (CCR = 8 bits) par le biais du champ d'adresse effective. Ce sont les instructions :

- ANDI to SR ; EORI to SR ; ORI to SR
- ANDI to CCR ; EORI to CCR ; ORI to CCR
- . génération ; SR ou CCR
- . mode = 111
- . registre = 100

4.4. Références implicites

Certaines instructions font des références implicites au compteur ordinal (PC), au pointeur de pile système (SP), au pointeur de pile superviseur (SSP), au pointeur de pile utilisateur (USP) au registre d'état (SR ou CCR).

4.5. Codage de l'adresse effective

Le tableau de la figure 7 montre le codage utilisé pour les différents modes de l'adresse effective.

Adressage	Mode	Registre	Adressage	Mode	Registre
Dn	000	n° reg.	d(An, ix)	110	n° reg.
An	001	n° reg.	xxx.W	111	000
(An)	010	n° reg.	xxx.L	111	001
(An)+	011	n° reg.	d(PC)	111	010
-(An)	100	n° reg.	d(Pc, ix)	111	011
d(An)	101	n° reg.	# xxx	111	100

Fig. 7 Codage de l'adresse effective

5. Les catégories d'adressage

Le M68000 possède 14 types d'adressage, les différentes instructions ne peuvent utiliser tous les types d'adressage. C'est pourquoi ces types ont été groupés en catégories d'adressage. Une catégorie d'adressage est donc un sous-ensemble de l'ensemble des types d'adressage. Il y a quatre catégories principales.

5.1. La catégorie données

Cette catégorie est la plus grande, elle regroupe tous les types d'adressage fournissant une donnée, à l'exception donc de l'adressage direct de registre adresse qui fournit une adresse.

5.2. La catégorie mémoire

Cette catégorie regroupe les types d'adressage fournissant une adresse mémoire pour accéder à la donnée, plus l'adressage immédiat.

5.3. La catégorie contrôle

Cette catégorie regroupe les types d'adressage fournissant une adresse mémoire à l'exception de l'adressage indirect postincrémenté et de l'adressage indirect précédécrementé.

5.4. La catégorie altérable

Cette catégorie regroupe les données 'modifiables', c'est-à-dire tous les types d'adressage à l'exception de ceux référençant le programme (relatif au PC et relatif au PC indexé) et de l'adressage immédiat.

Modes	Catégories			
	Donnée	Mémoire	Contrôle	Altérable
Dn	X			X
An				X
(An)	X	X	X	X
(An) +	X	X		X
- (An)	X	X		X
d(An)	X	X	X	X
d(An, ix)	X	X	X	X
xxx.W	X	X	X	X
xxx.L	X	X	X	X
d(PC)	X	X	X	
d(PC, ix)	X	X	X	
# xxx	X	X		

Fig. 8 Les catégories d'adressage

La figure 8 montre les différentes catégories d'adressage. En fait les instructions utilisent aussi bien ces catégories que des intersections de ces catégories ; par exemple donnée-altérable, contrôle-altérable, etc...

V - LES DONNEES

Les données quelles qu'elles soient transitent par le bus données qui est un bus bidirectionnel de 16 bits de largeur. On s'aperçoit donc que pour une donnée de 32 bits il faudra faire deux lectures, ou deux écritures, en mémoire.

1. Type de données supportés

- le bit	(1 bit)
- la position décimale	(4 bits)
- l'octet	(8 bits)
- le mot	(16 bits)
- le double mot	(32 bits)

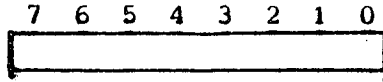
Pour le bit, la position décimale, l'octet, les entrées sorties se feront sur l'octet complet contenant la donnée. Pour le mot sur le mot complet ; pour le double mot, sur les deux mots successifs qui seront concaténés le premier mot étant le poids fort, le second le poids faible.

2. Organisation des données en mémoire

Les octets sont adressables individuellement. L'octet de poids fort ayant une adresse paire (la même que le mot correspondant). L'octet de poids faible ayant une adresse impaire (celle du mot correspondant plus un). On ne peut accéder aux données couvrant plusieurs octets que sur la borne d'un mot (octet paire). Si un double mot est situé à l'adresse n (n pair) alors le second mot de cette donnée se trouvera à l'adresse $(n + 2)$.

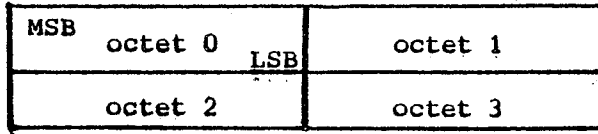
Les types de données et leurs représentations se trouvent à la figure 9. La représentation en mémoire est à la figure 10.

bit



numéro du bit
1 octet = 8 bits

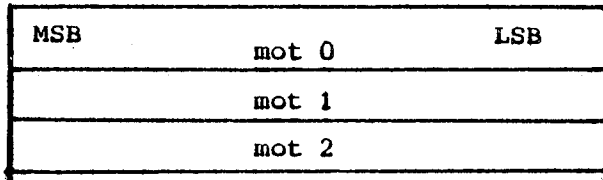
octet



1 octet = 8 bits

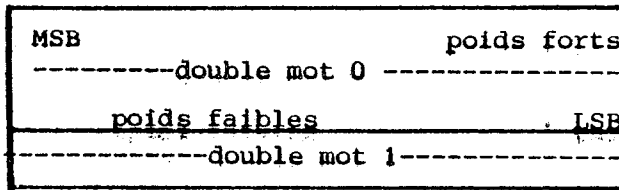
MSB = bit le plus significatif
LSB = bit le moins significatif

mot



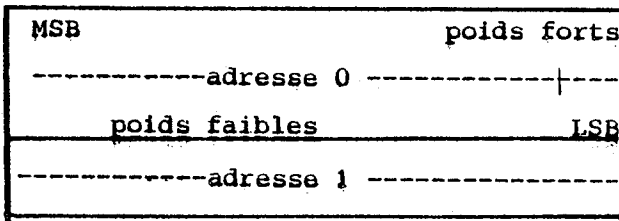
1 mot = 16 bits

double mot



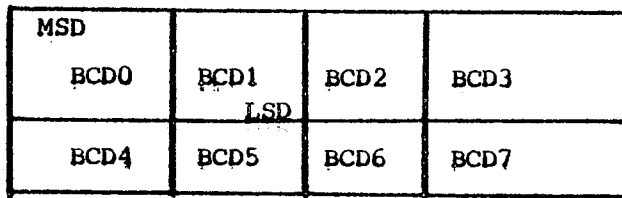
1 double mot = 32 bits

adresse



1 adresse = 32 bits

BCD



1 octet = 2 BCD

MSD = chiffre le plus significatif
LSD = chiffre le moins significatif

Fig. 9 Représentation des données

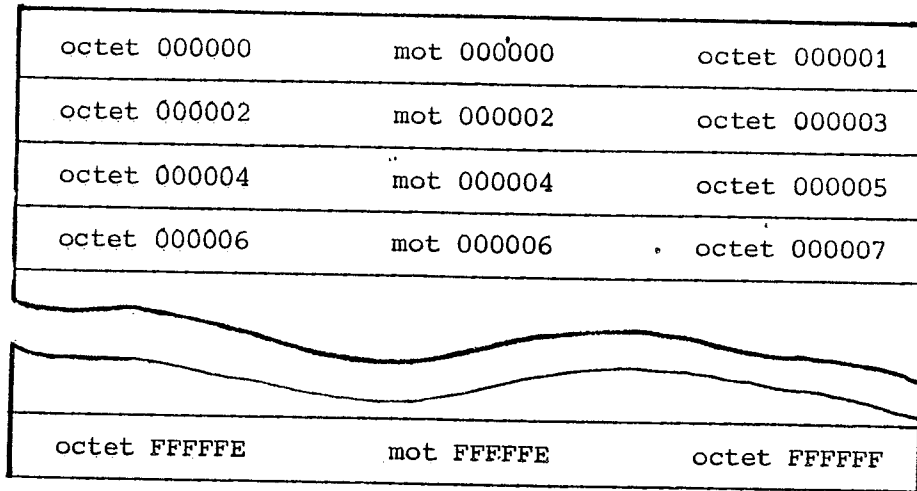


Fig. 10 Représentation mémoire

VI - LES SIGNAUX ET BUS

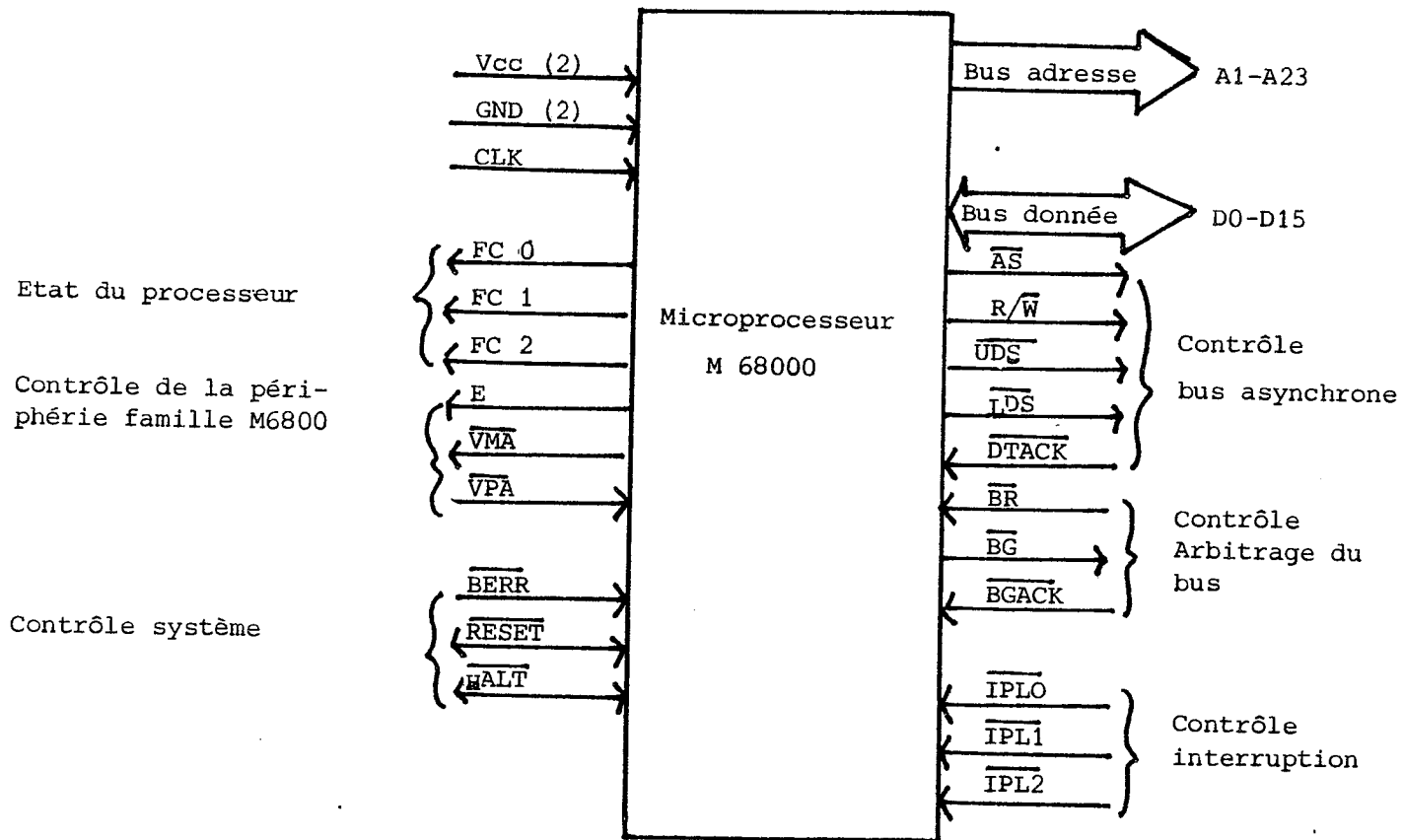


Fig. 11 Signaux et bus

Le MC68000 est donc construit sur un boîtier de 64 broches (cf. fig. 11). Il est composé de trois bus principaux.

1. Le bus adresse

C'est un bus trois-états unidirectionnel d'une largeur de 23 bits, donnant la possibilité d'adresser 8 Méga-mots. Il fournit l'adresse pour les opérations des bus pendant tous les cycles sauf pendant les cycles d'interruption. Durant les cycles d'interruption les bits A1, A2 et A3 donnent le niveau de l'interruption traitée tandis que les bits A4 à A23 sont mis à l'état logique 1.

2. Le bus donnée

C'est un bus trois états bidirectionnel d'une largeur de 16 bits. Il supporte des données d'un octet ou d'un mot. Pendant les cycles d'interruption, le circuit extérieur fournit le numéro du vecteur d'interruption sur les lignes D0 à D7.

3. Le bus contrôle

Ce bus a une largeur de 20 bits, il est composé de différents sous contrôles :

3.1. Contrôle du bus asynchrone

C'est la partie contrôlant les entrées-sorties avec la mémoire et les différents périphériques (adresses projetées en mémoire). Les transferts de données asynchrones sont prises en compte par l'utilisation des signaux suivants :

- Adresse strobe (\overline{AS}) :

Ce signal indique que l'adresse se trouvant sur le bus adresse, est valide pour une opération d'entrée/sortie.

- Read/Write (R/W) :

Ce signal indique : à l'état haut que le processeur effectue une lecture ; inversement, à l'état bas le processeur effectue une écriture.

- Upper Data Strobe et Lower Data Strobe (UDS, LDS) :

Ces deux lignes permettent d'accéder soit au niveau d'un mot ou au niveau d'un octet. La figure 12 donne les différentes utilisations en conjonction avec le signal de lecture et d'écriture (R/W).

<u>UDS</u>	<u>LDS</u>	<u>R/W</u>	D8 D15	D0 D7
H	H	-	Pas de donnée valide	Pas de donnée valide
B	B	H	Donnée valide bits 8-15	Donnée valide bits 0-7
H	B	H	Pas de donnée valide	Donnée valide bits 0-7
B	H	H	Donnée valide bits 8-15	Pas de donnée valide
B	B	B	Donnée valide bits 8-15	Donnée valide bits 0-7
H	B	B	Donnée valide bits 0-7	Donnée valide bits 0-7
B	H	B	Donnée valide bits 8-15	Donnée valide bits 8-15

H = état haut ; B = état bas ; - = état haut/état bas.

Fig. 12 Contrôle des transferts de données

- Data transfert acknowledge (DTACK) :

Cette entrée indique que le transfert est terminé. Quand le processeur reconnaît le signal DTACK pendant un cycle de lecture, les données sont mémorisées (latch) et le cycle est terminé. Quand le signal DTACK est reconnu en écriture, le cycle de bus est terminé.

3.2. Contrôle de l'arbitrage du bus

Ces signaux permettent de déterminer qui est maître du bus. Ils sont au nombre de trois :

- Bus request (\overline{BR}) ;

A cette entrée sont connectés tous les circuits périphériques qui peuvent manipuler le bus. A l'état bas, elle indique au processeur qu'un circuit désire être maître du bus.

- Bus grant (\overline{BG}) ;

Cette sortie permet d'indiquer, à tous les circuits périphériques ayant fait une demande pour être maître du bus, que le processeur abandonne le contrôle de ce dernier à la fin du cycle en cours.

- Bus grant acknowledge (\overline{BGACK}) :

Cette entrée indique au processeur qu'un autre circuit périphérique est devenu maître du bus. Ce signal ne peut être positionné, que si les quatre conditions suivantes sont vérifiées :

1. le signal \overline{BG} a été reçu ;
2. le signal \overline{AS} est inactif, indiquant que le processeur n'utilise plus le bus ;
3. le signal \overline{DTACK} est inactif, indiquant qu'aucune mémoire ou périphérique n'utilise le bus ;
4. le signal \overline{BGACK} est inactif, indiquant qu'aucun autre circuit périphérique continue à demander la commande du bus.

3.3. Contrôle des interruptions : (IPL0, IPL1, IPL2) :

Ces trois entrées indiquent le niveau de priorité du circuit périphérique demandant une interruption. Le niveau 7 est le plus haut niveau de priorité. Le niveau zéro indique qu'aucune interruption n'est demandée. Le bit le moins significatif est IPL0 et le plus significatif est IPL2.

3.4. Contrôle système

C'est le bus qui gère, initialisations, erreurs de bus, arrêts. Il est composé de 3 signaux :

- Bus error (BERR) :

Cette entrée informe le processeur qu'il y a un problème pendant l'exécution du cycle en cours. Le problème peut être le résultat de :

- ° Un circuit périphérique ne répondant pas ;
- ° Une erreur sur l'acquisition du numéro de vecteur d'interruption ;
- ° Une demande d'accès illégale donnée par le MMU (Memory management unit) ;
- ° Des erreurs dépendant de l'application.

- Reset (RESET) :

Ce signal bidirectionnel est utilisé pour réinitialiser le processeur (exécution de la séquence d'initialisation du système) en réponse à un signal reset externe. Un reset interne (instruction reset) provoque la réinitialisation de tous les circuits périphériques externes et l'état interne du processeur n'est pas affecté. Un reset total du système (processeur et circuits périphériques externes) est le résultat de l'application au même instant des signaux HALT et RESET.

- Halt (HALT) :

Cette ligne est bidirectionnelle. Quand elle est dirigée par un circuit externe, elle provoque l'arrêt du processeur à la fin du cycle en cours. Quand le processeur a été arrêté par cette entrée, tous les signaux de contrôle sont inactifs et toutes les lignes 3-états sont mises à haute impédance (3e état).

Quand le processeur arrête l'exécution d'instruction (après une double erreur sur le bus par exemple) la ligne HALT est dirigée par le processeur pour indiquer aux circuits externes que le processeur s'est arrêté.

3.5. Contrôle de la périphérie M6800

Ces signaux sont utilisés pour interfacer les périphériques synchrones de la famille M6800 avec le processeur MC68000 asynchrone.

- Enable (\overline{E}) :

Ce signal est le signal enable (E) "standard" de la famille M6800. La période de cette sortie est de 10 cycles 68000.

- Valid peripheral address (\overline{VPA}) :

Cette entrée indique que le périphérique adressé appartient à la famille M6800 et que le transfert des données doit être synchrone avec le signal enable (E). Cette entrée indique aussi que le processeur doit utiliser l'adressage automatique par vecteur pour une interruption.

- Valid memory address (\overline{VMA}) :

Cette sortie est utilisée pour indiquer aux périphériques de la famille M6800 que l'adresse se trouvant sur le bus adresse est valide et que le processeur est synchronisé avec enable. Ce signal est une réponse au signal VPA.

3.6. Etat du processeur (FC0, FC1, FC2) :

Ces trois lignes sont des sorties indiquant l'état dans lequel se trouve le processeur. La figure 13 donne les différentes possibilités.

Code de fonction			Classe de référence
FC2	FC1	FC0	
0	0	0	(non utilisé)
0	0	1	Données utilisateur
0	1	0	Programmes utilisateur
0	1	1	(non utilisée)
1	0	0	(non utilisée)
1	0	1	Données superviseur
1	1	0	Programmes superviseur
1	1	1	Acquittement d'interruption

Fig. 13 Code de l'état du processeur

Remarque : L'utilisation de ce code en complément des lignes du bus adresse permet au M68000 d'accéder à 4 * 8 Méga-mots = 32 Mega-mots.

A ces signaux se rajoutent :

- les signaux d'alimentation (2)
- les signaux de masse (2)
- le signal d'horloge (1)

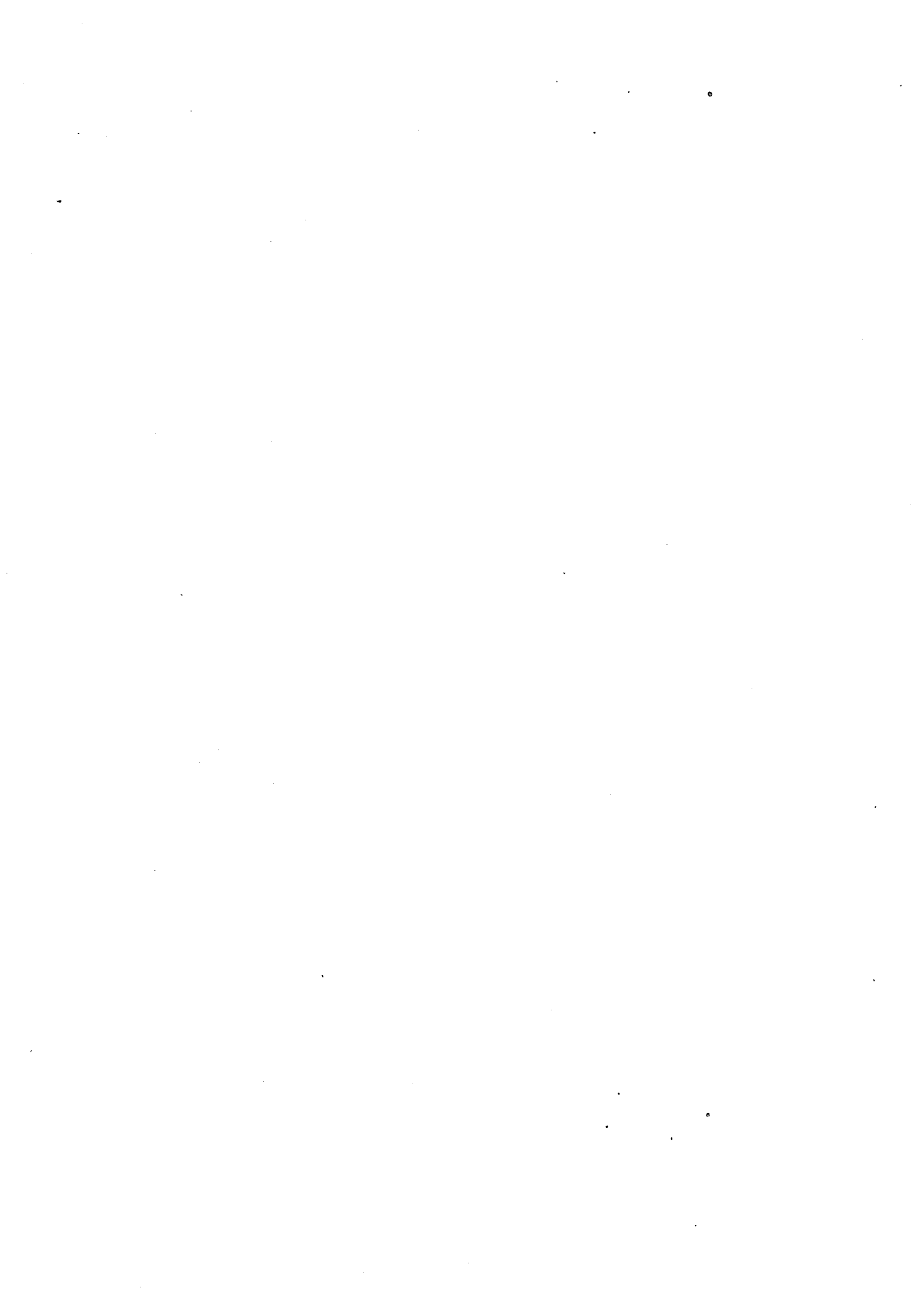
VII - CONCLUSION

Le MC68000 est un microprocesseur VLSI (very large scale of integration). Il a une faible surface d'intégration 44 mm² pour ses 68 000 transistors. Ceci a nécessité 100 hommes-mois pour sa conception auxquels s'ajoutent 70 hommes-mois pour le dessin du circuit. On peut déjà remarquer que les détecteurs qu'il avait fallu "fabriquer" pour le MC6800 sont maintenant internes au circuit: détecteur de codes invalides, détecteur d'adresse erreur, auxquels s'ajoute le détecteur de protection mémoire si le MC68000 est utilisé avec le circuit MMU (memory management unit).

On peut par contre regretter le manque d'informations, concernant certaines opérations, dans la documentation fournie par MOTOROLA avec le microprocesseur. L'utilisation de la maquette MOTOROLA n° 1463BA02488 muni du microprocesseur XC68000L - R9M7948 S/N 1258, a permis de compléter la documentation fournie.

NOTES BIBLIOGRAPHIQUES

Une documentation globale sur le sujet est donnée dans [MOT] .
Des explications concernant certains points précis se trouvent dans [GIR] . Les efforts fournis par Motorola pour cette nouvelle technologie (H-MOS) pourront être trouvés dans [STR-79] . Pour plus de précision sur le fonctionnement de la partie interne on pourra consulter [ICE-79] , [BOS-80] , [DED-81] .



REFERENCES

- [MOT] MOTOROLA - Semiconductors
 "MC 68000 - Advance information"
 "MC 68000 - User's Manual"
 "MC 68000 - Reference guide"
 "MC 68000 - MACSbyg Guide"
 "MC 68000 - User's Guide"
 "MC 68000 - Design module"
- [GIR] GIROD D.
 "Nouveaux concepts logiciels sur le microprocesseur 16 bits
68000", Minis et Micros n° 90.
 "Micros 16 bits : les modes de fonctionnement du 68000",
Minis et Micros n° 91.
 "Jeu d'instructions microprogrammées sur le microprocesseur
16 bits 68000", Minis et Micros n° 95.
- [STR-79] STRITTER E., GUNTER T. "A Microprocessor architecture for a
changing world : The Motorola 68000". Computer Feb. 1979.
- [ICE-79] ICE-RADC "Product evaluation of the Motorola MC 68000 16 bits
microprocessor", n° 10379EE, 1979.
- [BOS-80] BOSSEBOEUF A. "Analyse du fonctionnement interne du MC 68000",
Rapport de stage du DEA de Microélectronique, IMAG, Juin 1980.
- [DED-81] DEDEURWAERDER D., MOUGEAT P.
 "Présentation du fonctionnement interne du microprocesseur
68000, 1ère partie", note interne EFCIS/DRD n° 81-24. Janvier
1981.
 "Microprocesseur 68000. Etude détaillée du fonctionnement, 2e
partie", note interne EFCIS/DRD n° 81-32, Avril 1981.

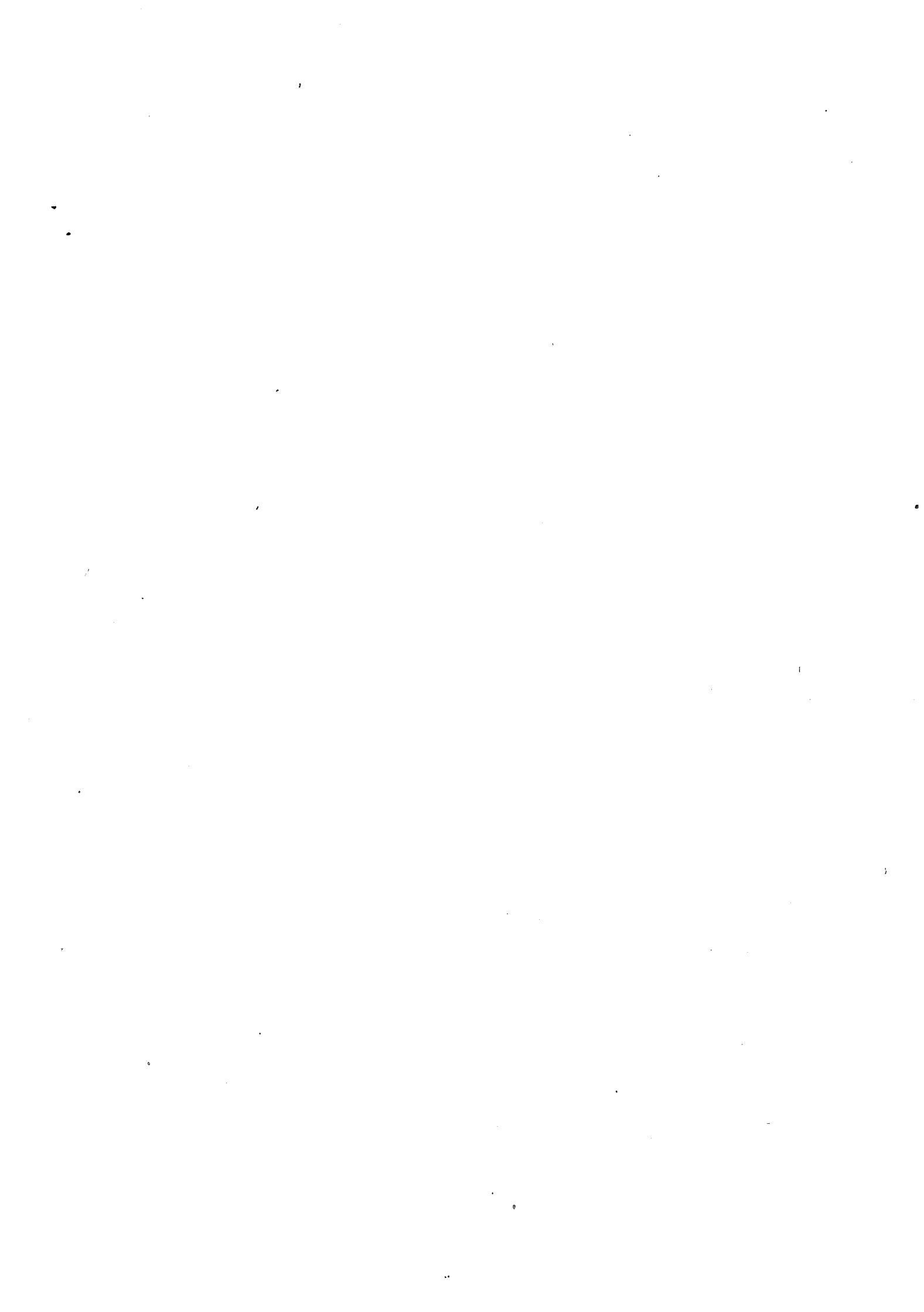


CHAPITRE III
PÉRIPHÉRIE DE LA PASTILLE
Etape (i)

CHAPITRE III - PERIPHERIE DE LA PASTILLE

Etape (i)

I	- INTRODUCTION	63
II	- DEFINITION DU MODELE	63
	1. Le modèle utilisé	63
	2. Hypothèses et définition	64
	3. Grandeurs caractéristiques du temps de détection.....	65
	3.1. Densité du temps de détection	66
	3.2. Grandeurs propres à chacun des détecteurs	66
	3.3. Probabilité de détection	67
	3.4. Résultats approchés	68
	3.5. Expressions limites	69
III	- RESULTATS EXPERIMENTAUX	70
	1. Pannes affectant le bus données	70
	2. Pannes affectant le bus adresses	75
	3. Généralisation à une application quelconque	78
IV	- CONCLUSION	80
ANNEXE	81
NOTES BIBLIOGRAPHIQUES	85
REFERENCES	87



I - INTRODUCTION

La première étape de cette étude était donc de mesurer les paramètres de la détection des pannes de la périphérie du microprocesseur MC 68000, et ce sans redondance massive du type duplication et comparaison/vote. L'inconvénient de ce genre de détecteur est qu'il augmente considérablement la quantité de matériel et d'autre part que la fiabilité des comparateurs/voteurs selon leur complexité est aussi à prendre en compte. Cet inconvénient est contrebalancé par une détection immédiate de l'erreur. Le modèle utilisé implique par contre un temps de latence de détection de l'erreur et c'est ce temps de latence, accompagné de la probabilité de détection qui est le but de cette étude. Ce chapitre présente donc les résultats pour le MC 68000. Ces résultats ne sont pas moins bons ni meilleurs que ceux obtenus pour le M 6800, ils sont une caractéristique propre du circuit considéré. Le modèle est d'abord présenté, puis les définitions des grandeurs caractéristiques sont données. Enfin les résultats expérimentaux sont comparés aux valeurs théoriques données par le modèle.

II - DEFINITION DU MODELE

1. Le modèle utilisé

Le modèle utilisé correspond aussi bien pour diverses applications sur un même CPU que pour différents CPUs. Il a été développé pour l'étude du M 6800, et cette expérience acquise a été d'un grand bénéfice pour l'application au MC 68000. Les détecteurs employés pour le M6800 ont été réutilisés à savoir : un détecteur de codes invalides, un détecteur de protection mémoire (tentative d'accès anormale en mémoire), un chien de garde périodiquement réexcité par logiciel. A ces détecteurs est venu s'ajouter un détecteur d'adresse erreur, le MC68000 adressant en effet des opérandes de plusieurs octets sur la borne d'un mot (adresse paire). Ce détecteur est comme le détecteur de code invalide et le chien de garde, indépendant de l'application. La protection mémoire dépend, quant à elle, de l'application.

Ces types de détecteurs sont les seuls envisageables puisqu'un programme ne peut s'exécuter correctement, les pannes perturbant l'exécution de celui-ci.

Si, pour le 6800, il était nécessaire de prévoir des circuits annexes exécutant la détection (circuit COP), pour le 68000 tout se trouve déjà à l'intérieur :

- détecteur de codes opérations invalides,
- détecteur d'adresse erreur.

Le détecteur de protection mémoire se trouvant sur un circuit annexe (le MMU = Memory Management Unit). Seul le chien de garde reste à implémenter.

2. Hypothèses et définition

Les hypothèses sont :

a) Au cours de l'exécution d'instructions, la CPU va exécuter un code instruction :

- cas 1 : Soit ce code est transformé par la panne en un code invalide et il s'en suit une détection par COI.
- cas 2 : Soit ce code va générer un calcul d'adresse, pour un mot ou un double mot, de résultat impair et il s'en suit une détection par AE.
- cas 3 : Soit ce code va générer un branchement à l'extérieur de la zone de mémoire du programme ou tenter une entrée/sortie anormale (écriture en ROM), il s'en suit une détection par PM.
- cas 4 : Soit faire boucler le programme indéfiniment en transformant le code en un code de bouclage, et on va répéter indéfiniment la même séquence déjà exécutée donc sans possibilité de détection ni par COI, ni par PM, ni par AE.
- cas 5 : Soit ne rien produire des 4 cas précédents.

- b) Ces cinq cas sont exclusifs les uns des autres. Chacun d'entre eux ayant une probabilité précise pour chaque panne étudiée. Ces probabilités sont constantes dans le temps jusqu'à occurrence de l'un des quatre autres cas.
- c) Le temps maximum de détection est égal à la période T_0 du chien de garde. En effet, celui-ci ne peut être réexcité si il se produit une erreur sur la périphérie.
- d) Le taux de panne est supposé constant (circuit dans sa période de maturité). Sachant qu'une panne intervient dans l'intervalle $[0, T_0]$, elle a la probabilité $1/T_0$ de survenir pendant les T intervalles de une unité (cf. fig. 1).

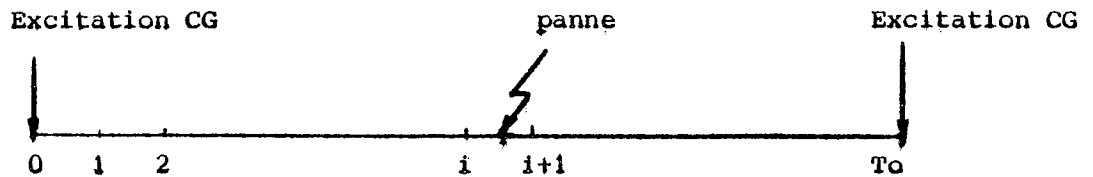


Fig. 1 Période du chien de garde

- e) La durée t_0 de la routine d'excitation est petite devant la période T_0 . On suppose ainsi qu'aucune panne ne survient pendant la durée d'excitation du chien de garde (1 instruction).
- f) L'exécution des instructions est supposée ponctuelle dans le temps, l'occurrence des pannes survenant entre deux instructions. L'unité de temps est la durée d'une instruction, supposée égale pour toutes les instructions (durée moyenne d'exécution déterminée par les probabilités d'utilisation des instructions).

3. Grandeurs caractéristiques du temps de détection

Soit K la variable aléatoire représentant le nombre d'exécutions d'instructions avant détection (c'est-à-dire dès qu'un détecteur a fonctionné).

Soient P1, P2, P3, P4 et P5, les probabilités respectives des cinq cas précédents :

P1 = Probabilité de détection par COI

P2 = Probabilité de détection par AE

P3 = Probabilité de détection par PM

P4 = Probabilité de bouclage, plus de détection par COI, ni par AE, ni par PM

P5 = 1 - P1 - P2 - P3 - P4.

3.1. Densité du temps de détection

$$\text{Pr } [K = 0] = P1 + P2 + P3$$

$$\text{Pr } [K = 1] = (T_0 - 1) \frac{1}{T_0} T_5 (P1 + P2 + P3) + \frac{1}{T_0} (1 - P1 - P2 - P3)$$

$$\begin{aligned} \text{Pr } [K = k] = & (T_0 - k) \frac{1}{T_0} [P5^k (P1 + P2 + P3)] \\ & + \frac{1}{T_0} [P5^k + P4 \left(\frac{1 - P5^k}{1 - P5} \right)] \end{aligned}$$

Par la suite, on notera Pk : Pr [K = k] .

3.2. Grandeurs propres à chacun des détecteurs

- Détecteur de code opération invalide (COI) :

$$\text{Pr } [K_{\text{COI}} = 0] = P1$$

$$\text{Pr } [K_{\text{COI}} = k] = (T_0 - k) \frac{1}{T_0} P5^k P1$$

- Détecteur d'adresse erreur (AE) :

$$\text{Pr } [K_{\text{AE}} = 0] = P2$$

$$\text{Pr } [K_{\text{AE}} = k] = (T_0 - k) \frac{1}{T_0} P5^k P2$$

- Détecteur de protection mémoire (PM) :

$$\text{Pr} [K_{PM} = 0] = P3$$

$$\text{Pr} [K_{PM} = k] = (T_0 - k) \frac{1}{T_0} P5^k P3$$

- Chien de garde (CG) :

$$\text{Pr} [K_{CG} = 0] = 0$$

$$\text{Pr} [K_{CG} = 1] = \frac{1}{T_0} (P4 + P5)$$

$$\text{Pr} [K_{CG} = k] = \frac{1}{T_0} (P5^k + P5^{k-1} P4 + P5^{k-2} P4 + \dots + P4)$$

Il est facile de voir que :

$$Pk = \text{Pr} [K_{COI} = k] + \text{Pr} [K_{AE} = k] + \text{Pr} [K_{PM} = k] + \text{Pr} [K_{CG} = k]$$

3.3. Probabilité de détection

$$\text{Pr} [\text{détection par COI}] = \sum_{k=0}^{T_0} \text{Pr} [K_{COI} = k]$$

$$= \sum_{k=0}^{T_0} (T_0 - k) \frac{1}{T_0} P5^k P1$$

$$= P1 [1 + A - \frac{1}{T_0} B]$$

De même, pour les autres détecteurs :

$$\text{Pr} [\text{détection par AE}] = P2 [1 + A - \frac{1}{T_0} B]$$

$$\text{Pr} [\text{détection par PM}] = P3 [1 + A - \frac{1}{T_0} B]$$

$$\text{Pr} [\text{détection par CG}] = \frac{P4}{1-P5} + \frac{1}{T_0} A [\frac{P1 + P2 + P3}{1-P5}]$$

$$\text{où } A = \sum_{k=1}^{T_0} P_5^k = \frac{P_5 - P_5^{T_0+1}}{1 - P_5}$$

$$\text{et } B = \sum_{k=1}^{T_0} k P_5^k = \frac{1}{T_0} \left[\frac{P_5 - (T_0+1) P_5^{T_0+1} + T_0 P_5^{T_0+2}}{(1 - P_5)^2} \right]$$

On peut remarquer que l'on a bien :

$$\text{Pr [détection par COI]} + \text{Pr [détection par AE]} + \text{Pr [détection par PM]} + \text{Pr [détection par CG]} = 1$$

On trouvera en annexe toutes les caractéristiques du temps de détection des pannes à la périphérie d'un CPU.

3.4. Résultats approchés

Les résultats des paragraphes précédents sont difficilement exploitables dans leur forme générale. Une valeur approchée, en raison de la valeur de T_0 dans une application pratique, est donnée maintenant.

$$\text{Pr [détection par COI]} \approx \frac{P_1}{1-P_5} \left[1 - \frac{1}{T_0} \frac{P_5}{1-P_5} \right]$$

$$\text{Pr [détection par AE]} \approx \frac{P_2}{1-P_5} \left[1 - \frac{1}{T_0} \frac{P_5}{1-P_5} \right]$$

$$\text{Pr [détection par PM]} \approx \frac{P_3}{1-P_5} \left[1 - \frac{1}{T_0} \frac{P_5}{1-P_5} \right]$$

$$\text{Pr [détection par CG]} \approx \frac{P_4}{1-P_5} + \frac{1}{T_0} \frac{P_5 (P_1 + P_2 + P_3)}{(1 - P_5)^2}$$

$$E[K_{\text{COI}}] = E[K_{\text{AE}}] = E[K_{\text{PM}}] \approx \frac{P_5 - \frac{1}{T_0} P_5 (1 - P_5)^2}{1 - P_5 - \frac{1}{T_0} P_5}$$

$$E[K_{\text{CG}}] \approx \left[\frac{1}{T_0} P_5 (P_1 + P_2 + P_3) \frac{1}{(1-P_5)^2} + P_4 \frac{T_0+1}{2} \right]$$

$$\frac{1}{P_4 + \frac{1}{T_0} \frac{P_5 (P_1 + P_2 + P_3)}{1 - P_5}}$$

$$\sigma^2 [K_{COI}] = \sigma^2 [K_{AE}] = \sigma^2 [K_{PM}] =$$

$$\left[\frac{P5 + P5^2}{1 - P5} - \frac{1}{T_0} \frac{P5 + 4P5^2 + P5^3}{(1 - P5)^2} \right] \frac{1}{1 - P5 - \frac{1}{T_0} P5} - E[K_{COI}]^2$$

$$\sigma^2 [K_{CG}] = \frac{1}{T_0} \left[\frac{P5 + P5^2}{(1 - P5)^2} + P4 \frac{T_0 (T_0+1) 2T_0+1}{6} - \frac{P4}{(1 - P5)^3} \right]$$

$$\frac{1}{P4 + \frac{1}{T_0} \frac{P5 (P1 + P2 + P3)}{1 - P5}} - E[K_{CG}]^2$$

3.5. Expressions limites

Dans les expressions approchées, le développement limité a été arrêté aux termes en $\frac{1}{T_0}$. Nous obtenons une valeur limite pour T_0 tendant vers

l'infini en s'arrêtant aux termes en T_0 . On obtient alors :

$$\text{Pr [détection par COI]} = \frac{P1}{1-P5}$$

$$\text{Pr [détection par AE]} = \frac{P2}{1-P5}$$

$$\text{Pr [détection par PM]} = \frac{P3}{1-P5}$$

$$\text{Pr [détection par CG]} = \frac{P4}{1-P5}$$

$$E [K] = \frac{P4}{1-P5} \frac{T_0}{2}$$

$$E [K_{COI}] = E [K_{AE}] = E [K_{PM}] = \frac{P5}{1-P5}$$

$$E [K_{CG}] = \frac{T_0}{2}$$

$$\sigma^2 [K] = \frac{P4}{1-P5} \left(\frac{1}{3} - \frac{P4^2}{4(1-P5)^2} \right) T_0^2$$

$$\sigma^2 [K_{COI}] = \sigma^2 [K_{AE}] = \sigma^2 [K_{PM}] = \frac{P5}{(1-P5)^2}$$

$$\sigma^2 [K_{CG}] = \frac{T_0^2}{12}$$

III - RESULTATS EXPERIMENTAUX

Les expériences ont été faites sur le simulateur pour déterminer le temps de détection des pannes. Le temps pris en compte est celui du premier détecteur ; le premier détecteur est pris en compte, lorsqu'une panne est détectée par plusieurs détecteurs dans la même instruction. L'instant d'occurrence des pannes est tiré aléatoirement (générateur aléatoire uniforme disponible sur Multics). Tous les collages simples représentatifs ont été étudiés, ainsi que quelques collages multiples. Le nombre d'instructions 68000 simulées est d'environ 200 000.

1. Pannes affectant le bus données

Les pannes étudiées sont tous les collages simples affectant le bus données en lecture, ainsi que des collages multiples.

Les résultats obtenus sont donnés à la table I. Le temps est mesuré en nombre d'instructions. Chaque panne est tirée aléatoirement 20 fois et les moyennes et écart-types du temps de détection sont calculés (quand elles sont significatives). La période T_0 est de 340 instructions.

On peut remarquer :

- Code opération invalide :

Les espérances et variances qui doivent être sensiblement égales le sont bien.

- Adresse erreur :

Les espérances et variances qui doivent être sensiblement égales le sont bien.

L'introduction au modèle, de ce nouveau détecteur, ne perturbe pas les résultats attendus. Il s'intègre donc parfaitement au modèle comme le montrent les résultats expérimentaux.

- Protection mémoire :

Les espérances et variances qui doivent être sensiblement égales le sont bien.

Pour ces trois premiers détecteurs, les espérances (respectivement variances) qui doivent être égales (fonction de P4 seul), le sont approximativement (quand la taille des échantillons permet la comparaison). Les quelques écarts provenant des effets de bords de l'intervalle [0, To] ,

- Chien de garde :

Les valeurs obtenues ne convergent pas pour chaque panne vers les valeurs attendues soient $\frac{340}{2}$ et $\frac{340}{\sqrt{12}}$, (170 et 98), ceci est dû bien sûr à la faible taille de chaque échantillon, mais si l'on effectue les cumuls pour toutes les détections des collages simples, on arrive aux nombres 163 et 103, ce qui reflète une bonne répartition des tirages aléatoires d'occurrences de pannes.

Le modèle adopté semble bien, une fois de plus refléter quantitativement et qualitativement les résultats obtenus expérimentalement.

Il faut maintenant déterminer les valeurs des paramètres à inclure dans le modèle de façon à prévoir le temps de détection. Avec les valeurs acquises par les expériences on est à même de déterminer les valeurs de

n°bit	collage	# E[K _{COI}] σ[K _{COI}]	# E[K _{AE}] σ[K _{AE}]	# E[K _{PM}] σ[K _{PM}]	#E[K _{COI+AE+PM}] σ[K _{COI+AE+PM}]	# E[K _{CG}] σ[K _{CG}]	# E[K] σ[K]
0	0	E=5 11 σ=4	E=13 3 σ=1	E=44 4 σ=69	E=15 18 σ=33	E=155 2 σ=12	E=29 σ=54
	1	E=5 3 σ=5	E=3 14 σ=3		E=3 17 σ=3	E=5 3 σ=4	E=3 σ=3
1	0		E=17 3 σ=12	E=43 8 σ=23	E=36 11 σ=23	E=109 9 σ=52	E=69 σ=53
	1	E=4 5 σ=1		E=4 13 σ=3	E=4 18 σ=2	E=152 2 σ=219	E=19 σ=69
2	0	E=13 10 σ=6	E=17 4 σ=14	E=14 3 σ=9	E=13 17 σ=9	E=219 3 σ=22	E=45 σ=76
	1	E=10 8 σ=10	(E=4) 1	E=7 9 σ=4	E=8 18 σ=7	E=6 2 σ=6	E=8 σ=7
3	0	E=5 8 σ=4	E=7 6 σ=3	E=15 3 σ=5	E=8 17 σ=5	E=219 3 σ=22	E=40 σ=77
	1	E=3 16 σ=2		E=4 4 σ=4	E=3 20 σ=2		E=3 σ=2
4	0	E=14 15 σ=8	(E=4) 1	(E=1) 1	E=13 17 σ=9	E=219 3 σ=22	E=44 σ=76
	1	E=3 2 σ=1	E=8 3 σ=4	E=19 11 σ=30	E=15 16 σ=26	E=243 4 σ=154	E=61 σ=114
5	0	E=6 10 σ=4	E=5 6 σ=2		E=6 16 σ=4	E=241 4 σ=12	E=53 σ=97
	1	E=4 7 σ=4	(E=3) 1	E=3 12 σ=2	E=3 20 σ=3		E=3 σ=3
6	0	E=8 17 σ=8	E=5 2 σ=4	(E=11) 1	E=8 20 σ=8		E=8 σ=8
	1	E=5 11 σ=3	(E=12) 1	E=6 8 σ=2	E=6 20 σ=3		E=6 σ=3
7	0	E=39 11 σ=36		E=9 9 σ=7	E=26 20 σ=31		E=26 σ=31
	1	E=6 11 σ=4	(E=3) 1	E=6 8 σ=4	E=5 20 σ=4		E=5 σ=4

Table I. Temps de détection expérimentaux des pannes de la périphérie
du bus donnée (1ère partie)

(T₀ = 340)

n° bit	collage	# E[K _{COI}] σ[K _{COI}]	# E[K _{AE}] σ[K _{AE}]	# E[K _{PM}] σ[K _{PM}]	# E[K _{COI+AE+PM}] σ[K _{COI+AE+PM}]	# E[K _{CD}] σ[K _{CG}]	E[K] σ[K]
8	0	E=16 17 σ=17			E=16 17 σ=17	E=141 3 σ=11	E=35 σ=49
	1	E=5 19 σ=4		(E=3) 1	E=5 20 σ=4		E=5 σ=4
9	0	E=9 9 σ=5		E=5 7 σ=5	E=8 16 σ=6	E=300 4 σ=30	E=66 σ=121
	1	E=31 15 σ=25		E=8 2 σ=1	E=29 17 σ=24	E=89 3 σ=69	E=38 σ=38
10	0	E=4 17 σ=3	(E=6) 1	E=7 2 σ=8	E=5 20 σ=3		E=5 σ=3
	1	E=4 18 σ=3		E=99 2 σ=2	E=13 20 σ=30		E=13 σ=30
11	0	E=6 16 σ=5	(E=5) 1	E=13 3 σ=11	E=7 20 σ=6		E=7 σ=6
	1	E=17 6 σ=8	E=10 8 σ=6	E=38 6 σ=16	E=21 20 σ=16		E=21 σ=16
12	0	E=6 12 σ=6	(E=21) 1	E=11 7 σ=7	E=8 20 σ=7		E=8 σ=7
	1	E=6 17 σ=4	E=3 2 σ=0	(E=1) 1	E=5 20 σ=4		E=5 σ=4
13	0	E=4 16 σ=2		E=8 4 σ=8	E=5 20 σ=4		E=5 σ=4
	1	E=5 8 σ=5	E=7 7 σ=7	(E=6) 1	E=7 16 σ=6	E=146 4 σ=170	E=34 σ=89
14	0	E=3 16 σ=2		E=1 4 σ=1	E=2 20 σ=2		E=2 σ=2
	1	E=6 10 σ=5		E=2 10 σ=3	E=4 20 σ=5		E=4 σ=5
15	0	E=5 11 σ=3	E=2 2 σ=3	E=8 2 σ=1	E=5 15 σ=3	E=153 5 σ=14	E=42 σ=66
	1	E=3 16 σ=2		E=4 4 σ=8	E=3 20 σ=3		E=3 σ=3

Table I. Temps de détection expérimentaux des pannes de la périphérie du bus donnée (2ème partie)

$$(T_0 = 340)$$

paramètres P1, P2, P3, P4. Ces valeurs seront établies à partir de collages simples, en effet, le nombre des collages simples se monte à 32, les collages doubles s'élevant à 480 pour le MC 68000 (112 pour le 6800). D'autre part on peut remarquer que certains collages multiples se ramènent à des collages simples (ceux-ci "masquant" le collage multiple). Quelques expériences ont été réalisées sur des collages multiples. Elles montrent que l'on peut conserver toute la généralité du modèle en se restreignant aux collages simples.

Donc pour l'application utilisée, et celle-ci seulement car les résultats dépendent de l'existence de la protection mémoire, elle-même dépendant entièrement de l'application, on peut déterminer les valeurs des paramètres P1, P2, P3 et P4. La valeur de ces paramètres s'obtient aisément à partir des expressions limites (cf §3.5). Les espérances de détection données par COI, AE, PM donnent P5 puis les probabilités de détection des différents détecteurs donnent respectivement P1, P2, P3 et P4 (qui ne sont pas indépendantes).

L'établissement de ces valeurs pour chaque panne individuelle n'est pas mentionné ici, car l'inconvénient est la faible précision de ces résultats due aux faibles tailles des échantillons. Une pondération selon les probabilités respectives de chaque panne est plus intéressante car faisant intervenir l'échantillon au complet.

En reprenant le modèle déjà évoqué, et en pondérant chaque panne considérée par sa probabilité d'occurrence, on obtient ainsi une pondération selon l'occurrence des pannes. Aucune panne n'étant privilégiée par rapport aux autres, on utilisera donc une équiprobabilité d'occurrence de chaque panne. Ceci nous conduit aux résultats de la Table II.

$\text{Pr} [\text{détecton par COI}] = 0.58$
$\text{Pr} [\text{détecton par AE}] = 0.11$
$\text{Pr} [\text{détecton par PM}] = 0.23$
$\text{Pr} [\text{détecton par CG}] = 0.08$
$E[K] = 0.08 \frac{T_0}{2}$
$E[K_{\text{COI}} + \text{AE} + \text{PM}] = 9.430$
$E[K_{\text{CG}}] = \frac{T_0}{2}$
$\sigma^2 [K] = 0.027 T_0^2 \quad (\sigma [K] = 0.163 T_0)$
$\sigma^2 [K_{\text{COI}} + \text{AE} + \text{PM}] = 91 \quad (\sigma [K_{\text{COI}} + \text{AE} + \text{PM}] = 9.5)$
$\sigma^2 [K_{\text{CG}}] = \frac{T_0^2}{12}$

Table II. Valeurs limites des caractéristiques du temps de détection (MC 68000) pour l'application et une pondération uniforme sur le bus donnée.

On peut remarquer l'importance de l'écart-type par rapport à la moyenne : le rapport $\sigma [K] / E [K]$ tend vers la valeur 4 (pour le 6800 cette valeur était déjà supérieure à 3). Enfin, il ne faut pas perdre de vue que ces valeurs sont des valeurs limites, pour T_0 très grand, et ne coïncident pas avec les données expérimentales.

2. Pannes affectant le bus adresse

Pour le bus adresse la totalité des pannes simples n'a pas été abordée, seules les pannes dont les conséquences ne sont pas triviales (détection immédiate par protection mémoire) ont été étudiées. La table III donne les résultats expérimentaux obtenus. On a donc testé $2 \times 8 = 16$ pannes simples au lieu des $2 \times 23 = 46$ pannes simples. Chacune de ces pannes a été simulée 20 fois, l'instant d'apparition étant aléatoire. La période du chien de garde est $T_0 = 340$ instructions.

n° bit collage	# E[K _{COI}] σ[K _{COI}]	# E[K _{AE}] σ[K _{AE}]	#E[K _{PM}] σ[K _{PM}]	#E[K _{COI+AE+PM}] σ[K _{COI+AE+PM}]	# E[K _{CG}] σ[K _{CG}]	E[K] σ[K]	
0	0				E=198 20 σ=90	E=198 σ=90	
	1		E=1 20 σ=0		E=1 20 σ=0	E=1 σ=0	
1	0	E=14 10 σ=11	E=2 5 σ=4	E=4 4 σ=7	E=9 19 σ=10	(E=3) 1 σ=10	E=8 σ=10
	1	E=12 17 σ=8		E=2 3 σ=0	E=11 20 σ=9		E=11 σ=9
2	0	E=37 10 σ=23	E=10 7 σ=7	(E=8) 1	E=25 18 σ=22	E=30 2 σ=36	E=25 σ=23
	1	E=10 16 σ=7		E=7 4 σ=9	E=10 20 σ=7		E=10 σ=7
3	0	E=16 11 σ=17	E=10 2 σ=8	E=4 5 σ=3	E=12 18 σ=14	E=242 2 σ=25	E=25 σ=72
	1	E=10 13 σ=9	(E=1) 1	(E=1) 1	E=9 15 σ=8	E=115 5 σ=114	E=35 σ=71
4	0	E=12 17 σ=11		E=1 2 σ=1	E=11 19 σ=11	(E=2) 1	E=11 σ=11
	1	E=12 7 σ=20	E=7 4 σ=2	E=10 9 σ=12	E=10 20 σ=13		E=10 σ=13
5	0	E=26 15 σ=31		E=17 5 σ=12	E=24 20 σ=27		E=24 σ=27
	1	E=32 10 σ=26	E=3 3 σ=1	E=24 4 σ=6	E=25 17 σ=23	E=201 3 σ=57	E=52 σ=70
6	0	E=92 3 σ=137	E=8 4 σ=4	E=14 10 σ=11	E=26 17 σ=58	E=331 3 σ=3	E=72 σ=124
	1	E=34 6 σ=52	(E=0) 1	E=11 3 σ=7	E=24 10 σ=41	E=185 10 σ=154	E=105 σ=137
7	0	E=2 8 σ=3	(E=5) 1	E=8 5 σ=7	E=4 14 σ=5	E=189 6 σ=108	E=60 σ=103
	1	E=25 9 σ=35	(E=0) 1	E=28 8 σ=18	E=25 18 σ=27	E=59 2 σ=6	E=29 σ=28
8	0	E=1 6 σ=0			E=1 6 σ=0	E=147 14 σ=103	E=103 σ=110
	1	E=59 7 σ=70		E=6 11 σ=4	E=27 18 σ=50	E=116 2 σ=114	E=36 σ=60

Table III : Temps de détection expérimentaux de la périphérie
du bus adresse
(T₀ = 340)

Toutes les remarques concernant les interactions entre le temps de latence et le temps de détection faites pour le 6800, se retrouvent ici aussi, c'est-à-dire, les valeurs des divers paramètres varient avec les pannes (théoriquement le type de code-machine ne dépend pas de l'emplacement physique de la mémoire où se trouve ce code). En effet, si les espérances et variances respectives de chaque détecteur sont sensiblement voisines comme elles devraient l'être, il n'en est pas de même si l'on considère $E[K_{COI}]$, $E[K_{AE}]$, $E[K_{PM}]$ d'une part et $\sigma[K_{COI}]$, $\sigma[K_{AE}]$, $\sigma[K_{PM}]$ d'autre part, qui devraient avoir les mêmes valeurs.

Comme il l'a été fait pour le bus donnée, une expression limite pour une équiprobabilité des 16 pannes simples étudiées est donnée à la Table IV.

Pour les raisons évoquées ci-dessus les valeurs respectives de $E[K_{COI} + AE + PM]$ et $\sigma^2[K_{COI} + AE + PM]$ bien que similaires aux résultats obtenus pour le bus données, sont sûrement faussées par le temps de latence. On remarquera aussi que le rapport $\sigma[K] / E[K]$ qui tendait vers 4 pour le bus donnée tend maintenant vers 2,5 pour le bus adresse.

$$\begin{aligned} \text{Pr [détection par COI]} &= 0.46 \\ \text{Pr [détection par AE]} &= 0.14 \\ \text{Pr [détection par PM]} &= 0.20 \\ \text{Pr [détection par CG]} &= 0.20 \\ E[K] &= 0.20 \frac{T_0}{2} \\ E[K_{COI} + AE + PM] &= 15.916 \\ E[K_{CG}] &= \frac{T_0}{2} \\ \sigma^2[K] &= 0.065 T_0^2 \quad (\sigma[K] = 0.254 T_0) \\ \sigma^2[K_{COI} + AE + PM] &= 363 \quad (\sigma[K_{COI} + AE + PM] = 19.1) \\ \sigma^2[K_{CG}] &= \frac{T_0^2}{12} \end{aligned}$$

Table . IV. Valeurs limites des caractéristiques du temps de détection (MC 68000) pour l'application envisagée et une pondération uniforme du bus adresses.

3. Généralisation à une application quelconque

La généralisation à une application quelconque des résultats obtenus ici modifie les hypothèses du modèle de la façon suivante :

- la protection mémoire, dépendant entièrement du programme d'application n'est donc plus valable, en effet les emplacements de programmes de données, de ROM, de RAM seront différents ;
- les caractéristiques de localité d'un programme lui sont spécifiques et donc les 16 pannes simples étudiées au lieu de 46 ne seront peut-être plus suffisantes.

Les caractéristiques d'un programme, telles que l'utilisation globale des instructions, varient peu d'un programme à un autre, seul leur emploi respectif varie.

Pour le premier point, une bonne généralisation est de considérer que l'espace mémoire est adressé dans sa totalité par le programme. Donc plus de détection possible par protection mémoire. On pourrait alors reporter la détection par PM sur les autres détecteurs proportionnellement à leur probabilité de détection. Mais il est préférable de prendre le cas le plus pessimiste, c'est-à-dire une détection par chien de garde. On obtient ainsi une borne maximum du temps de détection sans protection mémoire. Ainsi les coefficients $(P_1, P_2, P_3, P_4, P_5)_i$ deviennent pour une application sans protection mémoire $(P_1, P_2, 0, P_3+P_4, P_5)_i$.

Pour le second point, les résultats sur le bus adresse ne sont pas le seul fait du temps de détection mais de l'interaction latence-détection. Une borne pessimiste est donc prise si on ne considère le résultat que comme temps de détection et non comme somme d'un temps de détection et d'un temps de latence.

Compte-tenu de ces remarques, les caractéristiques (pessimistes) pour une application quelconque, utilisant tout l'espace mémoire et donc sans PM, sont données à la Table V.

Bus données	Bus adresses
Pr [détection par COI] = 0.58	Pr [détection par COI] = 0.46
Pr [détection par AE] = 0.11	Pr [détection par AE] = 0.14
Pr [détection par CG] = 0.31	Pr [détection par CG] = 0.40
$E [K] = 0.31 \frac{T_0}{2}$	$E [K] = 0.40 \frac{T_0}{2}$
$E [K_{COI} + AE] = 9.430$	$E [K_{COI} + AE] = 15.916$
$E [K_{CG}] = \frac{T_0}{2}$	$E [K_{CG}] = \frac{T_0}{2}$
$\sigma^2 [K] = 0.096 T_0^2$	$\sigma^2 [K] = 0.117 T_0^2$
$\sigma^2 [K_{COI} + AE] = 91$	$\sigma^2 [K_{COI} + AE] = 366$
$\sigma^2 [K_{CG}] = \frac{T_0^2}{12}$	$\sigma^2 [K_{CG}] = \frac{T_0^2}{12}$

Bus données et adresses
Pr [détection par COI] = 0.54
Pr [détection par AE] = 0.12
Pr [détection par CG] = 0.34
$E [K] = 0.34 \frac{T_0}{2}$
$E [K_{COI} + AE] = 11.592$
$E [K_{CG}] = \frac{T_0}{2}$
$\sigma^2 [K] = 0.103 T_0^2$
$\sigma^2 [K_{COI} + AE] = 183$
$\sigma^2 [K_{CG}] = \frac{T_0^2}{12}$

Table V. Caractéristiques (pessimistes) pour une application quelconque utilisant l'espace total RAM (pas de PM).

IV - CONCLUSION

Le gain apporté par COI, AE et PM est important par rapport à une détection par CG seul. L'inconvénient de PM est de dépendre de l'application, contrairement à COI et AE.

Contrairement à ce qui était dit pour le 6800 : "les bonnes performances de rapidité de la détection par COI proviennent du nombre de codes invalides", en fait le 68000 en a proportionnellement plus, mais les probabilités de détection et temps de détection sont pourtant plus longs.

En fait, cette bonne rapidité doit provenir plus du profil des codes invalides et donc de leur répartition au sein des codes valides, que de leur nombre. Il est sûr que pour un même microprocesseur, augmenter le nombre de codes invalides augmentera la détection par COI. Mais comme on vient de le voir ceci n'est pas valable en passant d'un composant à un autre.

Il est bon de voir que les fabricants sont touchés par les problèmes de fiabilité des composants puisque MOTOROLA intègre les détecteurs COI et AE dans le boîtier du microprocesseur, et que le détecteur PM est disponible en utilisant le boîtier MMU.

Il ne faut pas perdre de vue que ces résultats sont valables pour le microprocesseur MC 68000 dans sa version actuelle, et que l'augmentation ou la diminution du nombre d'instructions (possibles car microprogrammées) changerait les caractéristiques obtenues dans ce rapport.

ANNEXE - CARACTERISTIQUES DU TEMPS DE DETECTION DES PANNES DE LA PERIPHERIE D'UN CPU

$$\Pr [K = k] = (T_0 - k) \frac{1}{T_0} [P_5^k (P_1 + P_2 + P_3)] + \frac{1}{T_0} [P_5^k + P_4 \frac{1-P_5^k}{1-P_5}] \quad k \neq 0$$

$$\Pr [K = 0] = P_1 + P_2 + P_3$$

$$\Pr [K < k] = P_1 + P_2 + P_3 + A' (P_1 + P_2 + P_3) \left[1 + \frac{1}{T_0 (1-P_5)} \right] - \frac{P_1 + P_2 + P_3}{T_0} B' + k \frac{1}{T_0} \frac{P_4}{1-P_5}$$

$$\Pr [K_{COI} = k] = (T_0 - k) \frac{1}{T_0} [P_5^k P_1]$$

$$\Pr [K_{AE} = k] = (T_0 - k) \frac{1}{T_0} [P_5^k P_2]$$

$$\Pr [K_{PM} = k] = (T_0 - k) \frac{1}{T_0} [P_5^k P_3]$$

$$\Pr [K_{CG} = k] = \frac{1}{T_0} [P_5^k + P_4 \frac{1-P_5^k}{1-P_5}] \quad k \neq 0$$

$$\Pr [K_{CG} = 0] = 0$$

$$\Pr [\text{détection par COI}] = P_1 \left[A + 1 - \frac{1}{T_0} B \right]$$

$$\Pr [\text{détection par AE}] = P_2 \left[A + 1 - \frac{1}{T_0} B \right]$$

$$\Pr [\text{détection par PM}] = P_3 \left[A + 1 - \frac{1}{T_0} B \right]$$

$$\Pr [\text{détection par CG}] = \frac{P_4}{1-P_5} + \frac{1}{T_0} A \frac{P_1 + P_2 + P_3}{1-P_5}$$

$$E [K] = B (P_1 + P_2 + P_3) \frac{1 + \frac{1}{T_0} - P_5}{1 - P_5} - \frac{P_1 + P_2 + P_3}{T_0} \quad (C - 3B - 2A)$$

$$+ \left[1 - \frac{P_1 + P_2 + P_3}{1 - P_5} \right] \frac{T_0 + 1}{2}$$

$$E [K_{COI}] = [B - \frac{1}{T_0} (C - 3B - 2A)] \frac{1}{1 + A - \frac{1}{T_0} B}$$

$$E [K_{AE}] = E [K_{PM}] = E [K_{COI}]$$

$$E [K_{CG}] = [\frac{1}{T_0} B (P_1 + P_2 + P_3) + \frac{T_0 + 1}{2} P_4] \frac{1}{\frac{1}{T_0} A (P_1 + P_2 + P_3) + P_4}$$

$$\sigma^2 [K] = (P_1 + P_2 + P_3) \frac{1 + \frac{1}{T_0} - P_5}{1 - P_5} (C - 3B - 2A)$$

$$- \frac{1}{T_0} (P_1 + P_2 + P_3) (D - 6C + 7B + 6A) + \frac{P_4}{1-P_5} \frac{(T+1)(2T+1)}{6} - E [K]^2$$

$$\sigma^2 [K_{COI}] = [C - 3B - 2A - \frac{1}{T_0} (D - 6C + 7B + 6A)] \frac{1}{A + 1 - \frac{1}{T_0} B} - E [K_{COI}]^2$$

$$\sigma^2 [K_{AE}] = \sigma^2 [K_{PM}] = \sigma^2 [K_{COI}]$$

$$\sigma^2 [K_{CG}] = \frac{1}{T_0} [C - 3B - 2A + \frac{P_4}{1-P_5} \frac{T(T+1)(2T+1)}{6} - \frac{P_4}{1-P_5} (C - 3B - 2A)] \frac{1}{Pr \text{ détection par CG}} - E [K_{CG}]^2$$

Avec les valeurs :

$$A = \frac{P_5 - P_5^{T_0+1}}{1 - P_5}$$

$$A' = \frac{P_5 - P_5^{k+1}}{1 - P_5}$$

$$B = \frac{P_5 - (T_0+1) P_5^{T_0+1} + T_0 P_5^{T_0+2}}{(1 - P_5)^2}$$

$$B' = \frac{P_5 - (k+1) P_5^{k+1} + k P_5^{k+2}}{(1 - P_5)^2}$$

$$C = \frac{1}{(1 - P5)^3} [6P5 + 6P5^2 + 2P5^3 - (T_0+3) (T_0+2) P5^{T_0+1} \\ + 2 (T_0+2) (T_0+1) P5^{T_0+2} - (T_0+2) (T_0+1) P5^{T_0+3}]$$

$$D = \frac{1}{(1 - P5)^4} [24P5 - 36P5^2 + 24P5^3 - 6P5^4 \\ - (T_0+4) (T_0+3) (T_0+2) P5^{T_0+1} + 3 (T_0+4) (T_0+3) (T_0+1) P5^{T_0+1} \\ - 3 (T_0+4) (T_0+2) (T_0+1) P5^{T_0+3} + (T_0+3) (T_0+2) (T_0+1) P5^{T_0+4}]$$

NOTES BIBLIOGRAPHIQUES

Une étude approfondie du modèle pourra être trouvée dans [COU-79] et [COU-81] . Les études particulières du coeur se trouvent dans [COU-80.1] , celles de la périphérie dans [COU-80.2] .

Le circuit COP a été mentionné on trouvera une description de ce circuit dans [GUY-80] .

Pour les problèmes de localité de programmes on pourra se reporter à [MAD 76] et [SHE-75] .



REFERENCES

- [COU 79] COURTOIS B.
"Modélisation du temps de détection".
Rapport final de contrat CNET n° 769 B 378 : "Etude d'une
commande répartie à microprocesseurs pour un autocommutateur
téléphonique". Mars 1979.
- [COU 80.1] COURTOIS B.
"Modélisation du temps de détection de pannes du coeur d'un
CPU".
Rapport IMAG RR 213. Octobre 1980.
- [COU 80.2] COURTOIS B.
"Modélisation du temps de détection des pannes de la périphérie
de la pastille d'un CPU".
Rapport IMAG RR 209. Juillet 1980.
- [COU 81] COURTOIS B.
"Test et LSI".
Thèse de docteur d'Etat. Juin 1981.
- [GUY 80] GUYOT A., MARRIS N.
"Motorola M6800 Coprocessor (COP)".
Computer Systems Laboratory, Department of Electrical
Engineering, Stanford University 1980.
- [MAD 76] MADISON A.W., BATSON A.P.
"Characteristics of program localities".
CACM, vol. 19, n° 5, May 1976.
- [SHE 75] SHEDLETSKY J., Mc CLUSKEY E.J.
"The error latency of a fault in a combinational digital circuit".
Proc. of the 5th Fault-Tolerant Computing Symposium. Paris June
1975.



CHAPITRE IV
LATENCE D'ERREUR
LATENCE ET FREQUENCE
DE L'EVENEMENT E
Etape (ii)



CHAPITRE IV - LATENCE D'ERREUR. LATENCE

ET FREQUENCE DE L'EVENEMENT E

Étape (iii)

I	- INTRODUCTION	93
II	- DEFINITION DES PROBABILITES DE TRANSFERT D'OBSERVABILITE	94
	II.1. Divergence et reconvergence	97
	II.2. Séquentialité	97
	II.3. Injection de pannes	97
III	- APPLICATION DYNAMIQUE AU MC 68000	99
	III.1. Evénements considérés	99
	III.2. Partie opérative.	100
	III.2.1. Modèle utilisé	100
	III.2.2. Propriétés dynamiques.	100
	III.3. Partie contrôle	103
IV	- RESULTATS OBTENUS POUR LE MC 68000	106
	IV.1. Partie opérative	106
	IV.1.1. Opérateurs	106
	IV.1.1.1. Opérateurs arithmétique et logique (ALU)....	106
	IV.1.1.2. Opérateur décimal (DCB)	107
	IV.1.1.3. Extension de l'opérateur de calcul (ALUE)...	107
	IV.1.1.4. Opérateur de multiplexage (MUX I/O)	107
	IV.1.1.5. Opérateur de manipulation de bits (DCR)	108
	IV.1.1.6. Opérateur Fool-it	108
	IV.1.1.7. Opérateur encodeur de priorité (PREN)	108
	IV.1.1.8. Opérateur d'extension de signe (SIGNEX)	109
	IV.1.1.9. Opérateur de calcul d'adresses (AUII-AUL) ...	109

IV.1.2. Registres	134
IV.1.2.1. Registres données (Dn)	134
IV.1.2.2. Registres adresses (An)	135
IV.1.2.3. Registres temporaires (DT et AT)	136
IV.1.2.4. Buffer d'adresse (AOB)	136
IV.1.2.5. Compteur ordinal (PC)	137
IV.1.2.6. Latch d'AU (AUR)	137
IV.1.2.7. Registre de FTU	137
IV.1.2.8. Registre de PREN	137
IV.1.2.9. Registre de DCR	138
IV.1.2.10. Registre DBIN	138
IV.1.2.11. Registre DOB	138
IV.1.2.12. Registre d'ALUE	139
IV.1.2.13. Latch d'ALU (ALUR)	139
IV.1.2.14. Registre ALUB.	139
IV.1.3. Bus	140
IV.1.3.1. Bus ABD, DBD	140
IV.1.3.2. Bus ABL, DBL	140
IV.1.3.3. Bus ABH, DBH	140
IV.1.3.4. Bus EAB	141
IV.1.3.5. Bus EDB	141
IV.1.3.6. Bus BC	142
IV.2. Partie contrôle	145
IV.2.1. Classe 0	146
IV.2.2. Classe I	146
IV.2.3. Classe II	147
IV.2.3.1. Sous-classe 1	147
IV.2.3.2. Sous-classe 2	148
IV.2.4. Classe III	150
IV.2.5. Classe IV	150
IV.2.6. Classe V	150
IV.2.7. Pannes à conséquences hors-classes.	150
V. - CONCLUSION	154
REFERENCES	155

I. INTRODUCTION

Ce chapitre va permettre de déterminer, le temps écoulé entre l'occurrence d'une panne dans le circuit et l'apparition d'une erreur dans un des registres ou dans la mémoire ; c'est le temps de latence de la machine. Un second temps est calculé, c'est le temps qui sépare l'occurrence de la panne de l'apparition d'une erreur sur le compteur ordinal.

Un rappel sur les notions utilisées en sûreté de fonctionnement, et les probabilités de transferts d'observabilité décrites dans [COU 81] est présenté succinctement.

L'application au MC 68000 est faite ensuite, les paramètres obtenus sont donnés pour la partie contrôle et la partie opérative de ce microprocesseur. Des explications sur les différents opérateurs, registres bus de la partie opérative, pourront être trouvées au Chapitre VI (test de la partie opérative). Des explications sur le principe du test fonctionnel de la partie contrôle, ainsi que sur les classes de pannes envisagées pour cette partie contrôle pourront être trouvées au Chapitre VII. (test de la partie contrôle).

II. DEFINITION DES PROBABILITES DE TRANSFERT D'OBSERVABILITE [COU 81]

Définitions :

Un module est une portion de circuit possédant des liens d'entrée et des liens de sortie, tel que présenté à la figure 1. Un circuit est composé d'un ensemble de modules interconnectés par leurs liens.

Les liens d'entrée de chaque module correspondent soit à des entrées primaires (entrées du circuit) soit à des sorties d'autres modules.

Les liens de sortie de chaque module correspondent soit à des sorties primaires (sortie du circuit) soit à des entrées d'autres modules.

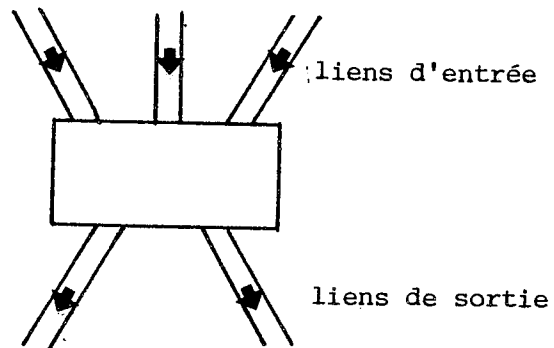


Figure 1 - Un module.

Il est possible de définir pour chaque lien, la probabilité que ce lien porte une erreur, cette erreur étant provoquée par une panne quelque part ailleurs dans le circuit.

Cette probabilité est précisément la défiabilité du signal = SU (Signal Unreliability) définie par OGUS dans [OGU 75], simplement étendu à plusieurs lignes. Elle correspond à la "fiabilité fonctionnelle" définie par AMAREL et BRZOZOWSKI dans [AMA 62], tandis que la "fiabilité fonctionnelle" de [OGU 75] correspond à la "fiabilité d'entrée/sortie" de [AMA 62]. Il n'y a

pas d'incompatibilité avec la fiabilité du signal utiliser dans [COU 76].

La probabilité de transfert d'observabilité va donc caractériser la SU de chacun des liens de sortie en fonction des SUs des liens d'entrée.

Dans un premier temps, les modules sont supposés être des modules purement combinatoires, les entrées sont supposées être indépendantes et le module particulier étudié est supposé sans panne. Ces conditions seront levées dans les paragraphes suivants.

Soient D et D' les distributions des valeurs sur un lien d'entrée, telles que :

- D s'il n'y a pas d'erreur sur le lien.
- D' s'il y a une erreur sur le lien.

La distribution D n'est pas nécessairement une distribution uniforme parmi les valeurs valides. La distribution D' peut par contre être approximée à une distribution uniforme sur l'ensemble des valeurs possibles différentes de la valeur juste. Les distributions D et D' sont supposées être telles, que la probabilité d'une valeur ne dépend pas de la valeur apparue précédemment.

Soit E_i un ensemble d'évènements disjoints tels que $\bigcup_i E_i = 1$, représentant les états possibles (erreur/pas d'erreur) de tous les liens d'entrées.

Il est clair que, pour chaque lien de sortie, on a :

$$\begin{aligned} \text{SU (sortie)} &= \text{Pr [erreur sur la sortie|E}_0\text{]} \cdot \text{Pr [E}_0\text{]} \\ &+ \text{Pr [erreur sur la sortie|E}_1\text{]} \cdot \text{Pr [E}_1\text{]} \\ &+ \text{-----} \\ &+ \text{Pr [erreur sur la sortie|E}_n\text{]} \cdot \text{Pr [E}_n\text{]} \end{aligned}$$

Les liens d'entrée étant indépendants, l'ensemble des E_i est évident.
Si \tilde{I}_j désigne l'état du j ème groupe d'entrées, I_j vaut :

- \tilde{I}_j si le j ème groupe d'entrées est sans erreur.
- $\overline{\tilde{I}_j}$ si le j ème groupe d'entrées a une erreur.

On obtient :

$$SU(\text{sortie}) = \sum \Pr [\text{erreur sur la sortie} | \tilde{I}_0 \tilde{I}_1 \dots \tilde{I}_k] \cdot \Pr [\tilde{I}_0 \tilde{I}_1 \dots \tilde{I}_k]$$

$\tilde{I}_0 \tilde{I}_1 \dots \tilde{I}_k$ décrivant tous les états possibles des liens d'entrées.

Les probabilités sont définies pour les distributions D ou D' selon l'état de chaque \tilde{I}_j .

Les OTP_s sont les probabilités conditionnelles correspondant aux différents états possibles des liens d'entrée. Les OTP_s pour un module possédant plusieurs liens de sortie. Les probabilités $\Pr[\tilde{I}_0 \tilde{I}_1 \dots \tilde{I}_k]$ s'expriment directement en fonction des SU_s des liens d'entrées.

Les entrées étant indépendantes, on a par exemple :

$$\Pr[\overline{\tilde{I}_0} \overline{\tilde{I}_1} \dots \overline{\tilde{I}_k}] = [(1-SU(I_0)) \cdot SU(I_1) \dots SU(I_k)].$$

Les OTP_s peuvent être désignées comme étant les probabilités conditionnelles transférant l'observabilité (défiabilité du signal) des entrées vers les sorties.

La formule précédente peut donc s'écrire :

$$SU(\text{Sortie}) = \sum_{\tilde{I}_0 \tilde{I}_1 \dots \tilde{I}_k} OTP_{\tilde{I}_0 \tilde{I}_1 \dots \tilde{I}_k} \cdot SU(\tilde{I}_0 \tilde{I}_1 \dots \tilde{I}_k)$$

$\tilde{I}_0 \tilde{I}_1 \dots \tilde{I}_k$ décrivant tous les états possibles des liens d'entrée.

II.1. Divergence et reconvergence :

Ce paragraphe permet de lever les hypothèses d'indépendance des entrées. En effet, la dépendance des liens d'entrée entre eux doit être envisagée lorsque les liens reconvergent après avoir divergé. Deux types de divergence doivent être considérés : la divergence aux sorties d'un module, et la divergence d'un lien en sous-liens. Des relations de dépendance sont alors déterminées dans chaque cas, ce sont ces relations qui sont utilisées lors de la reconvergence. On distingue de la même manière reconvergence à l'entrée d'un module et reconvergence de sous-liens. Tous les résultats peuvent être trouvés dans [COU 80].

II.2. Séquentialité :

De la même manière que l'hypothèse d'indépendance a été levée, l'hypothèse faite que le circuit est purement combinatoire doit être levée pour refléter la réalité physique. Ce problème sera résolu en remplaçant la séquentialité dans le temps par une multiplicité des sorties.

II.3. Injection de pannes :

Ce paragraphe permet de lever la dernière hypothèse faite selon laquelle un module était considéré sans panne.

L'injection de pannes dans un module se fait par le biais de valeurs particulières assignées aux OTPs. Seules, les pannes permanentes seront considérées. Les OTPs définies en cas de pannes seront alors notées $OTPs_f$, si f est la panne considérée, interne au module.

Des bornes inférieures des $OTPs_f$ ont été évaluées, et pourront être trouvées

dans [COU 81]. Seule, la solution générale de la borne recherchée est donnée :

Si E_i représente chacun des événements disjoints

D_1, \dots, D_p correspondent aux distributions des valeurs des P liens d'entrée.

$X_j, X_k \dots X_p$ représentent les valeurs valides des liens correspondants.

On obtient alors :

$$B_{Ei} = \min_j \Pr_{D_1} [X_j] \cdot \min_k \Pr_{D_2} [X_k] \dots \min_l \Pr_{D_p} [X_l]$$

III. APPLICATION DYNAMIQUE AU MC 68000 :

Le modèle précédent est maintenant utilisé pour une étude dynamique de l'occurrence de certains événements dans une machine, le cas du microprocesseur MC 68000 est abordé.

III.1. Evènements considérés :

Deux événements sont particulièrement considérés ;

- Evènement E = l'adresse de la prochaine instruction est fautive, elle contient une erreur (un mauvais calcul de la condition de branchement d'un branchement conditionnel ne crée pas un évènement E).
- Evènement E_1 = l'état de la machine doit être considéré comme faux. Cet évènement apparaît dès qu'un registre, ou la mémoire, contient une erreur (de manière plus précise, un registre directement affecté d'une panne, ne peut produire une erreur générant E_1 que si ce registre est utilisé, que ce soit en lecture ou en écriture).

Les temps moyens jusqu'à l'occurrence de ces événements sont notés :

- M.T.T.E. : Temps moyen jusqu'à l'occurrence de l'évènement E, à compter de l'instant d'occurrence de la panne considérée.
- M.T.T.E₁ : Temps moyen jusqu'à l'occurrence de l'évènement E_1 . Ce temps correspond au temps de latence au niveau de la machine.

Ces temps sont mesurés en nombre d'instructions exécutées. Supposant qu'il n'y ait pas de détection de la panne après la première occurrence de E, un autre paramètre sera utile :

M.T.B.-E : Temps moyen entre occurrences de l'évènement E (fréquence de génération de l'évènement E).

Enfin, on peut remarquer que $E \subset E_1$ et par suite :

$$MTTE_1 \leq MTTE$$

III.2. Partie opérative :

III.2.1. Modèle utilisé :

Les entrées primaires du système sont l'ensemble des variables internes du système (les registres de la machine) et la mémoire. Ces registres sont des modules.

Ils peuvent être affectés par des pannes, au moyen de la détermination de la SU des valeurs prises à la sortie de ces registres (lecture). Le niveau du modèle est celui des instructions donc plus élevé qu'un modèle de transfert de registres. Les probabilités calculées le sont pour une instruction (chaque instruction du répertoire), les probabilités finales étant pondérées par un mix d'instructions. Enfin, la modélisation du système peut dépendre de l'instruction considérée.

III.2.2. Propriétés dynamiques :

Considérons les deux probabilités :

Pr[Occurrence de E]

Pr[Occurrence de E₁]

Elles sont dépendantes des OTPs f des modules ayant des pannes (dépendant des pannes particulières étudiées), des OTPs des autres modules, et finalement des SU g des groupes d'entrée des modules directement connectés aux entrées primaires.

La SU de ces liens devrait en toute rigueur dépendre de la fiabilité (confiance) des données contenues dans les registres et la mémoire. L'évènement E, correspondant précisément à l'occurrence d'erreurs dans les registres et la mémoire, il n'y a pas de difficulté à ouvrir les boucles de retour vers les entrées primaires. La "fiabilité" des données contenues dans les registres est ainsi considérée égale à 1 (mais l'injection de pannes dans ces registres est faite, sur les sorties de ces registres). La probabilité Pr[occurrence de E_1] donne le $MTPPE_1$, la distribution étant une distribution géométrique.

Mais ainsi, l'évènement E peut avoir une occurrence après E_1 , due à la mémorisation d'une erreur dans un registre par une instruction précédente. Un calcul exact de la distribution de E devrait être fait d'une manière récursive, analogue à la fiabilité du signal, calculée pour un circuit séquentiel [KOR 77].

En raison de la complexité d'un tel calcul, un évènement E' est considéré, au lieu de E. Cet évènement E' correspond au même système, mais les boucles de retour aux entrées primaires (registres) étant ouvertes. La "fiabilité" des données contenues dans les registres est égale à 1. En d'autres termes, si la machine était considérée comme un automate, les états étant mémorisés par les registres, l'état suivant de la table d'état considérée pour le calcul de la latence [SME 76-1] serait toujours l'état juste. Si l'on suppose que les modules sont tels que les SU g des sorties sont des fonctions non décroissantes des SU s des entrées (par analogie avec [ESA 62]), de tels modules

pourraient être appelés cohérents) nous aurons :

$$MTTE \leq MTTE'$$

Le $MTTE'$ est calculé à partir de $Pr[\text{occurrence de } E']$ par une distribution géométrique. De plus, et pour les mêmes raisons, nous aurons :

$$MTB-E \leq MTTE'$$

Par suite une borne inférieure de la fréquence de E est obtenue. Le temps de détection pouvant être calculé à partir de E_1 puisqu'avant cet événement la machine ne produit pas d'erreurs, les différents temps sont représentés à la figure 2. On peut noter que la "latency difference" définie dans [SHE 76-2] correspond, pour un module au temps écoulé entre E_1 et la première détection.

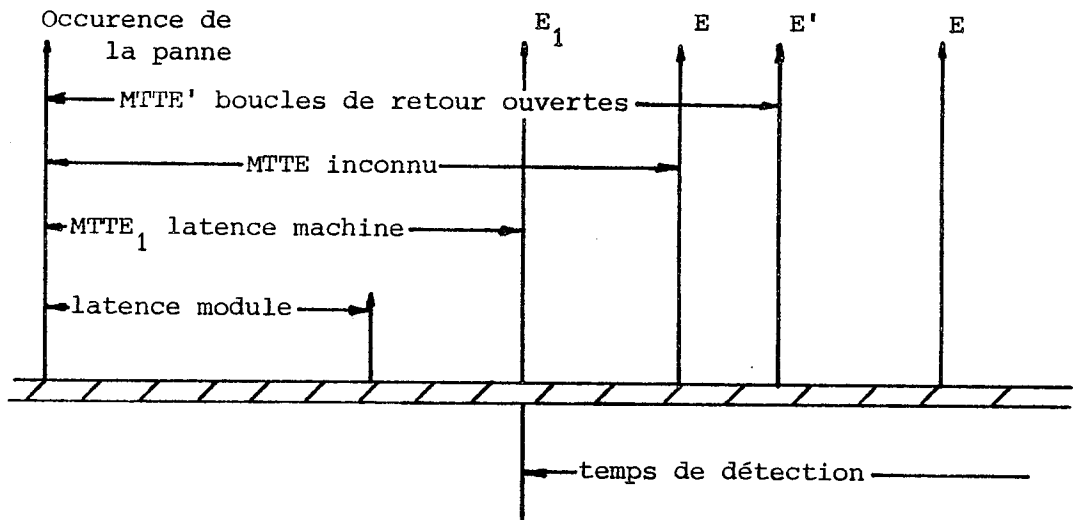


Figure 2 - Evènements, temps de latence et temps de détection.

Le $MTTE'$ est utilisé par la suite pour déterminer le temps de détection, qui dépend de cette valeur. Ainsi, si le nombre moyen d'occurrences de E

est égal à m pour qu'il y ait détection, on aura :

$$\text{Temps de détection} \leq m \text{ MTE}' - \text{MTE}_1$$

III.3. Partie contrôle :

Une utilisation directe du modèle peut être facilement faite pour la partie opérative, mais l'utilisation pour la partie contrôle demande à être examinée. La partie contrôle, qu'elle soit réalisée par PLAs, séquenceur cablé, etc... doit générer de manière séquentielle des commandes activant la partie opérative. Ces commandes sont des entrées-sorties de registres sur des bus interne et des lignes de sélection d'opérations pour les opérateurs. Ce sont ces commandes, plutôt que la manière de les générer, qui sont considérées. La partie contrôle est ainsi modélisée à un niveau plus élevé que la partie opérative, de la même manière que THATTE et ABRAHAM modélisent la partie contrôle dans [THA 78] pour le test fonctionnel.

Ainsi, pour le MOTOROLA MC 68000, on considère environ 120 lignes de commandes d'opérateurs et de sélection de registres plutôt que cinq PLAs dont 4 (A_0, A_1, A_2, A_3) fournissent une adresse de 10 bits correspondant à une micro-instruction stockée dans une ROM de 31 K Bits ; le dernier PLA, IRD est un PLA à paramètres, générant à partir des micro commandes et du mot opération, les commandes directes de la partie opérative.

En ce qui concerne les lignes de sélection des opérations sur les opérateurs, il est aisé de déterminer des OTPs_f particuliers pour ces modules, prenant en compte des erreurs (activation de ligne ne devant pas être activées, non-activation de lignes devant être activées) concernant ces lignes. Tout ce qui est déterminé au chapitre VII pour les pannes des commandes d'opérateurs s'applique alors. Il en va différemment des commandes

connectant les registres aux bus. Pour des erreurs sur ces commandes telles, que l'évènement E est susceptible d'être généré lors de l'exécution même de l'instruction à mauvais séquençement, il est possible d'insérer des modules représentant les portes autorisant les transferts registres \rightarrow bus et bus \rightarrow registres, dans les modèles précédents. Ces modules sont situés sur les liens appropriés du modèle : les liens connectant les entrées primaires et les liens de retour aux entrées primaires. Les OTPs_f pour ces modules sont déterminées à partir de la connaissance des conséquences en cas de malfonction : par exemple pour le MC 68000, source du cycle précédent en cas de panne de précharge, mélange des sources ou des destinations comme défini au chapitre VII .

Les OTPs_f dépendent des distributions D des valeurs, et de chaque instruction ou type d'instructions. Mais, des erreurs sur ces commandes qui ne résultent qu'en l'altération intempestive du contenu des registres (Classe 0, I, II et III du Chapitre VII l'illustrent), contenu non utilisé dans l'instruction mal séquençée, ne seraient pas prises en compte, puisque la SU des entrées primaires serait redevenue égale à 0 ("fiabilité" 1) pour les instructions suivantes. On conviendra alors, pour ces pannes, qu'une erreur puisse subsister de l'exécution d'une instruction à un autre (sauf si elle est "effacée" par un chargement du registre, etc...). On cherche alors le temps moyen nécessaire pour que cette erreur produise l'évènement E. Ce temps peut être calculé de manière analogue à DAVID et THEVENOD-FOSSE [DAV 79], par l'utilisation de séquences de détection pour le test aléatoire de parties opératives, sous l'hypothèse que chaque instruction ait une probabilité de générer l'évènement E, analogue à celle faite dans [DAV 79].

Les boucles de retour étant ouvertes, seul le contenu d'un registre peut ne pas avoir une SU égale à 0 au cours du temps, si l'on suppose qu'un

mauvais séquençement n'altère qu'un registre. On ne considère pas ainsi la possibilité d'altération d'un registre, de transfert de cette erreur dans un autre registre, puis de production de E. A partir des probabilités correspondant aux expressions régulières des instructions nécessaires à la production de E (une instruction étant supposée indépendante de la précédente, ce qui n'est pas réellement vrai), de la SU du contenu de l'entrée primaire altérée par l'instruction à mauvais séquençement et des valeurs des OTPs, il est possible de déterminer la valeur d'un $MTTE'$, encore une fois tel que :

$$MTTE \leq MTTE'$$

Les expressions régulières sont celles des séquences de transitions (instructions) minimales produisant E. Le calcul des probabilités correspondant aux expressions régulières se trouve également dans [PAR 78]. Pour ce qui concerne E_1 en particulier, il faut bien noter que le $MTTE_1$ est différent selon les conséquences :

Si un résultat est entaché d'erreurs lors de l'exécution d'une instruction, E_1 est produit, alors que si l'erreur est uniquement une modification intempestive du contenu d'un registre, E_1 n'est susceptible d'être produit que lorsque ce registre sera ultérieurement utilisé, en lecture.

IV. RESULTATS OBTENUS POUR LE MC 68000 :

Toutes les parties du circuit vont être étudiées ici. Les pannes créant des événements E sont étudiées, et la latence et la fréquence de E sont données. Dans le cas où la fréquence serait trop faible, le calcul n'est pas conduit, un test périodique sera alors nécessaire. Les pannes considérées pour les différentes parties seront en général des pannes conduisant à des bornes de la latence et de la fréquence de E (borne supérieure pour la latence et borne inférieure pour la fréquence).

IV.1. Partie opérative :

On examinera successivement, les opérateurs, les registres et les bus internes.

IV.1.1. Opérateurs :

IV.1.1.1. Opérateur arithmétique et logique (ALU)

L'opérateur arithmétique et logique (ALU) est parfois utilisé pour calculer les adresses d'instructions, et plus précisément lors des opérations de saut (JMP, JSR) quand l'adressage indirect indexé plus déplacement est utilisé. En effet, dans ce cas, le déplacement de 8 bits traverse l'ALU pour une extension sur 16 bits. Des événements E pourraient donc être générés par suite de pannes affectant cet opérateur. Mais, il serait néanmoins nécessaire de tester périodiquement l'opérateur, en vue d'opération de décalage, d'addition, d'opérations logiques, etc... De plus, l'opérateur ALU est moins utilisé pour calculer des adresses instructions que l'opérateur AU pour lequel nous verrons que les valeurs de la fréquence sont infinitésimales.

On préférera utiliser un test périodique complet de l'opérateur. Les valeurs à retenir sont pour cet opérateur :

$$MTTE' \rightarrow \infty \qquad MTTE_1 = 77\ 500.$$

IV.1.1.2. Opérateur décimal (DCB) :

Cet opérateur n'est jamais utilisé pour l'évaluation d'une adresse instruction. Il ne peut donc pas générer E. Il fera l'objet d'un test périodique conjointement à l'ALU. Les valeurs à retenir sont pour cet opérateur :

$$MTTE' \rightarrow \infty \qquad MTTE_1 = 79 \times 10^6.$$

IV.1.1.3. Extension de l'opérateur de calcul (ALUE) :

Cet opérateur n'est jamais utilisé pour l'évaluation d'une adresse instruction. Il ne génère donc pas E. Il fera l'objet d'un test périodique. Les valeurs à retenir sont pour cet opérateur :

$$MTTE' \rightarrow \infty \qquad MTTE_1 = 458\ 300$$

IV.1.1.4. Opérateur de multiplexage (MUX I/O) :

Pour cet opérateur, la même remarque que pour l'ALU peut-être faite. En effet, en cas d'opération de sauts (JMP et JSR), pour tous les modes d'adressage sauf par registre indirect, le mot d'extension d'adresse transite par le MUX I/O. Mais, néanmoins, il serait nécessaire de tester périodiquement cet opérateur utilisé pour toutes les entrées/sorties de données effectuées entre la partie opérative et la mémoire externe.

On préférera donc utiliser un test périodique pour cet opérateur. Les valeurs à retenir sont pour cet opérateur :

$$MTTE' \rightarrow \infty \qquad MTTE_1 = 52\ 100$$

IV.1.1.5. Opérateur de manipulation de bits (DCR) :

Cet opérateur est utilisé uniquement pour les instructions de manipulation de bits, il ne peut pas créer d'évènement E. Un test périodique de cet opérateur est donc envisagé. Les valeurs à retenir pour cet opérateur sont :

$$MTTE' \rightarrow \infty \qquad MTTE_1 = 79 \times 10^6.$$

IV.1.1.6. Opérateur Fool-it :

Cet opérateur sert exclusivement aux opérations de taille octet sur les données, il n'est jamais utilisé pour évaluer une adresse instruction et ne peut donc pas générer E. Un test périodique de cet opérateur est donc envisagé. Les valeurs à retenir pour cet opérateur sont :

$$MTTE' \rightarrow \infty \qquad MTTE_1 = 210\ 000$$

IV.1.1.7. Opérateur encodeur de priorité (PREN) :

Cet opérateur est utilisé uniquement pour les instructions de transferts multiples de registres adresses ($A0_2$ à $A7$ et $A'7$) et données ($D0$ à $D7$) (instruction MOVEM). Il ne peut pas générer d'évènement E. Un test

périodique est donc envisagé pour cet opérateur. Les valeurs à retenir sont pour cet opérateur :

$$MTE' \rightarrow \infty \qquad MTE_1 = 1 \times 10^6.$$

IV.1.1.8. Opérateur d'extension de signe (SIGNEX) :

Cet opérateur est utilisé conjointement à l'opérateur arithmétique (AU) pour les calculs d'adresses. Il rentre en compte pour les adressages indirect plus déplacement ; d(An), indirect indexé plus déplacement : d(An,Ri), absolu court : xxx.w, relatif : d(PC), relatif indexé : d(Pc,Ri), et ceci uniquement pour la première évaluation de l'adresse. Le résultat étant ensuite soit stocké dans le registre temporaire adresse (AT) soit conservé dans l'opérateur arithmétique AU pour les adressages suivants. Cet opérateur est donc nettement moins utilisé que l'AU pour le calcul d'adresses instructions et, vu les très faibles fréquences obtenues pour l'AU, un test périodique sera envisagé pour cet opérateur. Les valeurs à retenir pour cet opérateur sont :

$$MTE' \rightarrow \infty \qquad MTE_1 = 152\ 500$$

IV.1.1.9. Opérateur de calcul d'adresses (AUH-AUL) :

L'opérateur de calcul d'adresse est utilisé à chaque fois qu'une adresse est envoyée sur les plots adresses. C'est cet opérateur qui évalue le nouveau compteur ordinal. Nous allons considérer une panne sur cet opérateur AU, et plus précisément une panne qui ne serait manifestée que par un vecteur d'entrée, le moins probable. Ce cas correspond aux bornes inférieures de $OTPs_f$ étudiées au paragraphe II.3. ; appelons p cette borne telle que :

$$B_{EO} = B_{E1} = \min_D \Pr[X_j] = p.$$

L'hypothèse selon laquelle il n'ya pas dépendance des valeurs d'une utilisation à l'autre (d'une instruction à la suivante) n'est pas très réaliste, mais sera quand même retenue. Cet opérateur est utilisé pour le calcul de l'adresse de l'instruction suivante et de manière opérative pour le calcul de l'adresse des données, ainsi que pour décompter le nombre de cycles des instructions de décalage dynamique, de multiplications et de divisions. Comme il l'a été dit au paragraphe II.2. , la séquentialité dans le temps est remplacée par une multiplicité des sorties si nécessaire.

Les probabilités d'occurrence des différentes instructions (types et modes d'adressage) sont supposées connues. Les calculs menés ici, on été fait à partir du moniteur résidant de la carte MOTOROLA.

Ce calcul est donné à titre d'exemple, et ainsi toutes les valeurs nécessaires sont données. Les résultats obtenus pour chaque instruction ne sont pas explicités, mais on pourra se reporter à [COU 81] où un calcul similaire a été développé pour l'incrémenteur du M 6800.

Les cinq tableaux suivants vont être nécessaires aux calculs des probabilités $Pr[E']$ et $Pr[E_1]$ pour toutes les instructions 'normales' du jeu d'instruction du MC 68000. Le cas des exceptions est traité après.

Le tableau I présente, pour l'opérateur AU, les probabilités $Pr[E']$ et $Pr[E_1]$ selon le mode d'adressage, pendant la phase d'adressage uniquement.

Le tableau II présente, pour l'opérateur AU, la probabilité $Pr[E']$ pour toutes les instructions (sauf exceptions) selon le mode d'adressage de la source et de la destination.

Le tableau III présente, pour l'opérateur AU, la probabilité $Pr[E_1]$ pour toutes les instructions (sauf exceptions) nécessitant 0 ou 1 accès mémoire pour les opérations de calcul et 2 accès mémoire pour les transferts (MOVE), la taille de l'opération étant l'octet ou le mot. Pour les 16 cases centrales, 2 résultats sont présentés : le résultat supérieur correspond à l'utilisation du même registre adresse pour la source et la destination, le résultat inférieur correspond à l'utilisation de 2 registres différents.

Le tableau IV présente, pour l'opérateur AU, la probabilité $Pr[E_1]$ pour toutes les instructions (sauf exceptions) nécessitant 0 ou 1 accès mémoire pour les opérations de calcul et 2 accès mémoire pour les transferts (MOVE), la taille de l'opération étant le long mot. Même remarque pour les 25 cases centrales du tableau IV que pour les 16 cases centrales du tableau III.

ADRESSAGE	B.W		L	
	Pr[E']	Pr[E ₁]	Pr[E']	Pr[E ₁]
-	p	1 - (1-p)	p	1 - (1-p)
Dn	p	1 - (1-p)	p	1 - (1-p)
Ah	p	1 - (1-p)	p	1 - (1-p)
(An)	p	1 - (1-p)	p	1 - (1-p) (1-p)
(An) +	p	1 - (1-p) (1-p)	p	1 - (1-p) (1-2p)
- (An)	p	1 - (1-p) (1-p)	p	1 - ((1-p) (1-2p))
d (An)	2p	1 - (1-2p) (1-p)	2p	1 - (1-2p) (1-2p)
d (An, Ri)	2p	1 - (1-2p) (1-2p)	2p	1 - (1-2p) (1-3p)
XXX.W	2p	1 - (1-2p)	2p	1 - (1-2p) (1-2p)
XXXX.L	3p	1 - (1-3p)	3p	1 - (1-3p) (1-2p)

Tableau I : Pr[E'] et Pr[E₁] durant la phase d'adressage (B = octet, W = mot, L = long mot).
(début)

ADRESSAGE	B.W		L
	Pr[E']	Pr[E ₁]	
(Pc)	2p	1 - (1-2p) (1-p)	1 - (1-2p) (1-2p)
(Pc, R1)	2p	1 - (1-2p) (1-2p)	1 - (1-2p) (1-3p)
XXX	2p	1 - (1-2p)	1 - (1-3p)

Tableau I. Pr[E'] et Pr[E₁] durant la phase d'adressage (B = octet, W = mot, L = long mot).
(fin)

DESTINATION												
SOURCE	-	Dn	An	(An)	(An) +	- (An)	d (An)	d (An, Ri)	XXX.W	XXX.L	d (Pc)	d (Pc, Ri)
-	p	p	p	p	p	p	2p	2p	2p	3p	2p	2p
Dn	p	p	p	p	p	p	2p	2p	2p	3p	2p	2p
An	p	p	p	p	p	p	2p	2p	2p	3p	2p	2p
(An)	p	p	p	p	p	p	2p	2p	2p	3p	2p	2p
(An) +	p	p	p	p	p	p	2p	2p	2p	3p	2p	2p
- (An)	p	p	p	p	p	p	2p	2p	2p	3p	2p	2p
d (An)	2p	2p	2p	2p	2p	2p	3p	3p	3p	4p	3p	3p
d (An, Ri)	2p	2p	2p	2p	2p	2p	3p	3p	3p	4p	3p	3p
XXX.W	2p	2p	2p	2p	2p	2p	3p	3p	3p	4p	3p	3p
XXX.L	3p	3p	3p	3p	3p	3p	4p	4p	4p	5p	4p	4p
d (Pc)	2p	2p	2p	2p	2p	2p	3p	3p	3p	4p	3p	3p

Tableau II. Pr[E'] pour toutes les instructions (sauf exceptions) selon le mode d'adressage source et destination (début)

(B = octet ; W = mot ; L = long mot).

		DESTINATION										
SOURCE	-	Dn	An	(An)	(An) +	- (An)	d (An)	d (An, Ri)	XXX.W	XXX.L	d (Pc)	d (Pc, Ri)
d (Pc, Ri)	2p	2p	2p	2p	2p	2p	3p	3p	3p	4p	3p	3p
# XXX	B	2p	2p	2p	2p	2p	3p	3p	3p	4p	3p	3p
	L	3p	3p	3p	3p	3p	4p	4p	4p	5p	4p	4p

Tableau II. Pr[E'] pour toutes les instructions (sauf exceptions) selon le mode d'adressage source et destination (fin).

(B = octet, W = mot, L = long mot).

DESTINATION												
SOURCE	-	Dn	An	(An)	(An) +	- (An)	d̄ (An)	d̄ (An, Ri)	XXX.W	XXX.L	d̄ (Pc)	d̄ (Pc, Ri)
-	p	p	p	p	2p-p ²	2p-p ²	3p-2p ²	4p-4p ²	2p	3p	3p-2p ²	4p-4p ²
Dn	p	p	p	p	2p-p ²	2p-p ²	3p-2p ²	4p-4p ²	2p	3p	3p-2p ²	4p-4p ²
An	p	p	p	p	2p-p ²	2p-p ²	3p-2p ²	4p-4p ²	2p	3p	3p-2p ²	4p-4p ²
(An)	p	p	p	p	2p-p ²	2p-p ²	3p-2p ²	4p-4p ²	2p	3p	3p-2p ²	4p-4p ²
(An) +	2p-p ²	2p-p ²	2p-p ²	2p-p ²	3p-2p ²	3p-2p ²	4p-4p ²	5p-6p ²	3p-2p ²	4p-3p ²	4p-5p ² + 2p ³	5p-8p ² + 5p ³
- (An)	2p-p ²	2p-p ²	2p-p ²	2p-p ²	3p-2p ²	3p-2p ²	4p-4p ²	5p-6p ²	3p-2p ²	4p-3p ²	4p-5p ² + 2p ³	5p-8p ² + 5p ³
d̄ (An)	3p-2p ²	3p-2p ²	3p-2p ²	3p-2p ²	4p-4p ²	4p-4p ²	5p-6p ²	6p-9p ²	4p-3p ²	5p-4p ²	5p-7p ² + 3p ³	7p-11p ² + 6p ³
d̄ (An, Ri)	4p-4p ²	4p-4p ²	4p-4p ²	4p-4p ²	5p-6p ²	5p-6p ²	6p-9p ²	7p-12p ²	5p-5p ²	6p-8p ²	7p-11p ² + 6p ³	7p-16p ² + 12p ³

Tableau III. Pr[E₁] pour instructions à 0 et 1 accès mémoire, et transferts (MOVE) à 2 accès mémoire pour la taille octet et mot. (début)

XXX.W	2p	2p	2p	2p	3p-2p ²	3p-2p ²	4p-3p ²	5p-6p ²	3p	4p	4p-3p ²	5p-6p ²
XXX.L	2p	2p	2p	2p	4p-3p ²	4p-3p ²	5p-4p ²	6p-8p ²	4p	5p	5p-4p ²	6p-8p ²
d (Pc)	3p-2p ²	3p-2p ²	3p-2p ²	3p-2p ²	4p-5p ² +2 3p	4p-5p ² + 2p ³	5p-7p ² + 3p ³	7p-11p ² + 6p ³	4p-3p ²	5p-4p ²	5p-6p ²	6p-9p ²
d (Pc, Ri)	4p-4p ²	4p-4p ²	4p-4p ²	4p-4p ²	5p-8p ² +5 3p	5p-8p ² + 5p ³	7p-11p ² + +6p ³	7p-16p ² + 12p ³	5p-6p ²	6p-8p ²	6p-9p ²	7p-12p ²
* XXX	2p	2p	2p	2p	3p-2p ²	3p-2p ²	4p-4p ²	5p-6p ²	4p	5p	4p-4p ²	5p-6p ²

Tableau III. Pr[E_i] pour instructions à 0 et 1 accès mémoire, et transferts (MOVE) à 2 accès mémoire pour la taille octet et mot.

(fin)

DESTINATION												
SOURCE	-	Dn	An	(An)	(An) +	- (An)	d (An)	d (An, Ri)	XXX.W	XXX.L	d (Pc)	d (Pc, Ri)
-	P	P	P	2p-2p ²	3p-2p ²	3p-2p ²	4p-4p ²	5p-6p ²	4p-4p ²	5p-6p ²	4p-4p ²	5p-6p ²
Dn	P	P	P	2p-p ²	3p-2p ²	3p-2p ²	4p-4p ²	5p-6p ²	4p-4p ²	5p-6p ²	4p-4p ²	5p-6p ²
An	P	P	P	2p-p ²	3p-2p ²	3p-2p ²	4p-4p ²	5p-6p ²	4p-4p ²	5p-6p ²	4p-4p ²	5p-6p ²
(An)	2p-p ²	2p-p ²	2p-p ²	3p-2p ²	4p-3p ²	4p-3p ²	5p-6p ²	6p-8p ²	5p-8p ² + 4p ³	6p-11p ² + 6p ³	5p-8p ² + 4p ³	6p-11p ² + 6p ³
(An) +	3p-2p ²	3p-2p ²	3p-2p ²	4p-3p ²	5p-4p ²	5p-4p ²	6p-8p ²	7p-10p ²	6p-12p ² + 8p ³	7p-16p ² + 12p ³	6p-12p ² + 8p ³	7p-16p ² + 15p ³
- (An)	3p-2p ²	3p-2p ²	3p-2p ²	4p-3p ²	5p-4p ²	5p-4p ²	6p-8p ²	7p-10p ²	6p-12p ² + 8p ³	7p-16p ² + 12p ³	6p-12p ² + 8p ³	7p-16p ² + 15p ³
d (An)	4p-4p ²	4p-4p ²	4p-4p ²	5p-6p ²	6p-8p ²	6p-8p ²	7p-12p ²	8p-15p ²	7p-16p ² + 12p ³	8p-20p ² + 16p ³	7p-16p ² + 12p ³	8p-21p ² + 18p ³

Tableau IV. Pr[E₁] pour instructions à 0 et 1 accès mémoire et transferts (MOVE) à 2 accès mémoire pour la taille long mot. (début)

(An, Ri)	$5p-6p^2$	$5p-6p^2$	$5p-6p^2$	$6p-8p^2$	$7p-10p^2$	$7p-10p^2$	$8p-15p^2$	$9p-18p^2$	$6p-21p^2$ $+18p^3$	$10p-33p^2$ $+36p^3$	$8p-21p^2$ $+18p^3$	$9p-27p^2$ $+27p^3$
XX.W	$4p-4p^2$	$4p-4p^2$	$4p-4p^2$	$5p-8p^2+4p^3$	$6p-12p^2+8p^3$	$7p-16p^2+12p^3$	$7p-16p^2+12p^3$	$6p-21p^2+18p^3$	$7p-16p^2+12p^3$	$8p-20p^2+16p^3$	$7p-16p^2+12p^3$	$6p-21p^2+18p^3$
XX.L	$5p-6p^2$	$5p-6p^2$	$5p-6p^2$	$6p-11p^2+6p^3$	$7p-16p^2+15p^3$	$7p-16p^2+15p^3$	$8p-20p^2+16p^3$	$10p-33p^2+36p^3$	$8p-20p^2+16p^3$	$9p-24p^2+20p^3$	$8p-20p^2+16p^3$	$10p-33p^2+36p^3$
(Pc)	$4p-4p^2$	$4p-4p^2$	$4p-4p^2$	$5p-8p^2+4p^3$	$6p-12p^2+8p^3$	$6p-12p^2+8p^3$	$7p-16p^2+12p^3$	$3p-21p^2+18p^3$	$7p-16p^2+12p^3$	$8p-20p^2+16p^3$	$7p-12p^2$	$8p-15p^2$
(Pc, Ri)	$5p-6p^2$	$5p-6p^2$	$5p-6p^2$	$6p-11p^2+6p^3$	$7p-16p^2+15p^3$	$7p-16p^2+15p^3$	$8p-21p^2+18p^3$	$9p-27p^2+27p^3$	$6p-21p^2+18p^3$	$10p-33p^2+36p^3$	$8p-15p^2$	$9p-18p^2$
XXX	$3p$	$3p$	$3p$	$4p-3p^2$	$5p-6p^2$	$5p-6p^2$	$6p-8p^2$	$7p-12p^2$	$6p-8p^2$	$7p-10p^2$	$6p-8p^2$	$7p-12p^2$

bleau IV. Pr[E_i] pour instructions à 0 et 1 accès mémoire et transferts (MOVE) à 2 accès mémoire pour la taille long mot.

(fin).

Destination	Pr[E ₁]	
	B.W	L
(An)	p	3p - 3p ² + p ³
(An)+	3p - 3p ² + p ³	5p - 8p ² + 4p ³
-(An)	3p - 3p ² + p ³	5p - 8p ² + 4p ³
d(An)	4p - 5p ² + 2p ³	6p - 12p ² + 8p ³
d(An, Ri)	6p - 12p ² + 8p ³	8p - 21p ² + 18p ³
XXX.W	4p - 4p ²	6p - 12p ² + 8p ³
XXX.L	5p - 6p ²	7p - 16p ² + 12p ³
d(P _c)	4p - 5p ² + 2p ³	6p - 12p ² + 8p ³
d(P _c , R _i)	6p - 12p ² + 8p ³	8p - 21p ² + 18p ³
(Source = - , D _n , A _n)		

Tableau V - Pr[E₁] taille octet (B), mot (W) ou long mot (L)
opérations à 2 opérandes, à résultats en mémoire.

Les exceptions sont les suivantes :

* Instructions de décalage :

+ Décalage DY par DX :

$$\Pr[E'] = p$$

$$\Pr[E_1] = (m+1)p - mp^2 \text{ avec } m = [DX] \bmod 64$$

+ Décalage DY par # donnée :

$$\Pr[E'] = p$$

$$\Pr[E_1] = (m+1)p - mp^2 \text{ avec } \underline{\text{Si donnée}} = 0$$

$$\underline{\text{Alors}} m = 8$$

$$\underline{\text{Sinon}} m = \text{donnée}$$

+ Décalage d'un mot mémoire :

Résultats de la table V.

* Instructions Bcc et BRA :

+ Déplacement sur octet :

$$\Pr[E'] = p$$

$$\Pr[E_1] = p$$

+ Déplacement sur mot :

$$\Pr[E'] = 2p$$

$$\Pr[E_1] = 2p$$

* Instructions BSR :

+ Déplacement sur octet :

$$\Pr[E'] = p$$

$$\Pr[E_1] = 4p - 5p^2 + 2p^3$$

+ Déplacement sur mot :

$$\Pr[E^A] = 2p$$

$$\Pr[E_1] = 5p - 9p^2 + 6p^3$$

* Instructions CHK :

Le tableau VI résume les différentes valeurs obtenues en fonction du mode d'adressage.

Mode d'adressage <ea>	Dn < 0 ou Dn > (<ea>)		0 < Dn < (<ea>)	
	Pr[E']	Pr[E ₁]	Pr[E']	Pr[E ₁]
CHK Dn, Dm	$8p-13p^2+6p^3$	$9p-20p^2+18p^3$	p	2p
CHK Dn, (An)	$8p-13p^2+6p^3$	$9p-20p^2+18p^3$	p	2p
CHK Dn, (An) +	$8p-13p^2+6p^3$	$9p-20p^2+18p^3$	p	2p
CHK D, -(An)	$8p-13p^3+6p^3$	$9p-20p^2+18p^3$	p	2p
CHK Dn, d(An)	$9p-20p^2+18p^3$	$10p-27p^2+18p^3$	2p	3p
CHK Dn, d(An, Ri)	$9p-20p^2+18p^3$	$10p-27p^2+18p^3$	2p	3p
CHK Dn, XXX.W	$9p-20p^2+18p^3$	$10p-27p^2+18p^3$	2p	3p
CHK Dn, XXX.L	$10p-27p^2+18p^3$	$11p-34p^2+30p^3$	3p	4p
CHK Dn, d(Pc)	$9p-20p^2+18p^3$	$10p-27p^2+18p^3$	2p	3p
CHK D, d(Pc, Ri)	$9p-20p^2+18p^3$	$10p-27p^2+18p^3$	2p	3p
CHK Dn, # XXX	$9p-20p^2+18p^3$	$10p-27p^2+18p^3$	2p	3p

Tableau VI. Pr[E'] et Pr[E₁] pour l'instruction CHK en fonction du mode d'adressage.

* Instructions DBcc :

Pr[E'] : Si Dn = 1
Alors Pr[E'] = 2p
Sinon Pr[E'] = 2p-p²

Pr[E₁] : Si cc
Alors Pr[E₁] = 2p
Sinon Pr[E₁] = 4p-4p²

* Instructions DIVU, DIVS :

Le tableau VII résume les différentes valeurs des probabilités de E' et E_1 , pour les instructions DIVU, DIVS, selon le mode d'adressage et le fait qu'il y ait ou non division par zéro (ou débordement en division).

Mode d'adressage	Division par zéro (ou débordement)	Division correcte	
	$Pr[E'] = Pr[E_1]$	$Pr[E']$	$Pr[E_1]$
Dn	$8p-13p^2+6p^3$	p	$16p$
(An)	$8p-13p^2+6p^3$	p	$17p-16p^2$
(An) +	$9p-21p^2+19p^3-6p^4$	p	$18p-33p^2+16p^3$
- (An)	$9p-21p^2+19p^3-6p^4$	p	$18p-33p^2+16p^3$
d(An)	$10p-29p^2+32p^3-12p^4$	2p	$19p-50p^2+32p^3$
d(An, Ri)	$11p-38p^2+52p^3-24p^4$	2p	$20p-68p^2+64p^3$
XXX.W	$10p-29p^2+32p^3-12p^4$	2p	$19p-50p^2+16p^3$
XXX.L	$11p-37p^2+45p^3-18p^4$	3p	$20p-67p^2+48p^3$
d(Pc)	$10p-29p^2+32p^3-12p^4$	2p	$19p-50p^2+32p^3$
d(Pc, Ri)	$11p-38p^2+52p^3-24p^4$	2p	$20p-68p^2+64p^3$
# XXX	$9p-20p^2+12p^3$	2p	$18p-32p^2$

Tableau VII. $Pr[E']$ et $Pr[E_1]$ pour les instructions DIVU et DIVS en fonction du mode d'adressage.

* Instructions JMP

. JMP (An)

$$\Pr[E'] = 0$$

$$\Pr[E_1] = 0$$

. JMP d(An)

$$\Pr[E'] = 2p-2p^2$$

$$\Pr[E_1] = 2p-2p^2$$

. JMP d(An,Ri)

$$\Pr[E'] = 3p-2p^2$$

$$\Pr[E_1] = 3p-2p^2$$

. JMP XXX.W

$$\Pr[E'] = p$$

$$\Pr[E_1] = p$$

. JMP XXX.L

$$\Pr[E'] = 2p$$

$$\Pr[E_1] = 2p$$

* Instructions JSR

. JSR (An)

$$\Pr[E'] = 0$$

$$\Pr[E_1] = 2p$$

. JSR d(An)

$$\Pr[E'] = 3p-2p^2$$

$$\Pr[E_1] = 5p-8p^2+4p^3$$

. JSR d(An,Ri)

$$\Pr[E'] = 3p-2p^2$$

$$\Pr[E_1] = 6p-12p^2+8p^3$$

. JSR XXX.W

$$\text{Pr } E' = p$$

$$\text{Pr } E_1 = 4p - 4p^2$$

. JSR XXX.L

$$\text{Pr } E' = 2p$$

$$\text{Pr } E_1 = 5p - 6p^2$$

* Instructions LINK

On obtient :

$$\text{Pr } E' = 2p$$

$$\text{Pr } E_1 = 5p - 8p^2 + 4p^3$$

* Instructions MOVEM

Le tableau VIII résume les différentes valeurs des probabilités de E' et E_1 selon le mode d'adressage et la taille de l'opération.

Adressage	$\text{Pr}[E']$	$\text{Pr}[E_1]_{.W}$	$\text{Pr}[E_1]_L$
(An)	$2p$	$(n+1)p - n p^2$	$(2n+1)p - 2p^2$
- (An) ou (An) +	$2p$	$(n+1)p - n p^2$	$(2n+2)p - (2n+1)p^2$
d (An)	$3p$	$(n+3)p - 2(n+1)p^2$	$(2n+4)p - (4n+4)p^2$
d (An, Ri)	$3p$	$(n+4)p - 2(n+2)p^2$	$(2n+5)p - (4n+6)p^2$
XXX.W	$3p$	$(n+2)p - 2n p^2$	$(2n+2)p - 4n p^2$
XXX.L	$4p$	$(n+3)p - 3n p^2$	$(2n+3)p - 6n p^2$

Tableau VIII. $\text{Pr}[E']$. $\text{Pr}[E_1]$ pour les instructions MOVEM selon le mode d'adressage et la taille de l'opération.

* Instruction MOVEP

. MOVEP.W d(An), Rx

$$\text{Pr}[E'] = 2p$$

$$\text{Pr}[E_1] = 4p - 4p^2$$

. MOVEP.W Rx, d(An)

$$\text{Pr}[E'] = 2p$$

$$\text{Pr}[E_1] = 4p - 4p^2$$

. MOVEP.L d(An), Rx

$$\text{Pr}[E'] = 2p$$

$$\text{Pr}[E_1] = 6p - 8p^2$$

. MOVEP.L Rx, d(An)

$$\text{Pr}[E'] = 2p$$

$$\text{Pr}[E_1] = 6p - 8p^2$$

* Instructions MULS - MULU

Le tableau IX résume les différentes valeurs de E' et E_1 pour les instructions MULS et MULU selon le mode d'adressage.

ADRESSAGE	Pr[E']	Pr[E ₁]
Dn	p	16p
(An)	p	17p-16p ²
(An) +	p	18p-33p ² +16p ³
- (An)	p	18p-33p ² +16p ³
d(An)	2p	19p-50p ² +32p ³
d(An, Ri)	2p	20p-68p ² +64p ³
XXX.W	2p	19p-50p ² +16p ³
XXX.L	3p	20p-67p ² +48p ³
d(Pc)	2p	29p-50p ² +32p ³
d(Pc, Ri).	2p	20p-68p ² +64p ³
# XXX	2p	18p-32p ²

Tableau IX. Pr[E'] et Pr[E₁] pour les instructions MULS, MULU selon le mode d'adressage.

* Instruction PEA

Le tableau X résume les différentes valeurs de E' et E₁ pour l'instruction PEA.

ADRESSAGE	Pr[E']	Pr[E ₁]
(An)	p	3p-2p ²
d(An)	2p	5p-8p ² +4p ³
d(An, Ri)	2p	6p-12p ² +8p ³
XXX.W	2p	4p-4p ²
XXX.L	3p	5p-6p ²
D(Pc)	2p	5p-8p ² +4p ³
D(Pc, Ri)	2p	6p-12p ² +8p ³

Tableau X. Pr[E'] et Pr[E₁] pour l'instruction PEA selon le mode d'adressage.

* Instruction RESET

$$\text{Pr}[E'] = p$$

$$\text{Pr}[E_1] = 33p-32p^2$$

* Instructions RTE, RTR

$$\text{Pr}[E'] = 4p-3p^2$$

$$\text{Pr}[E_1] = 4p-3p^2$$

* Instruction RTS

$$\text{Pr}[E'] = 3p-2p^2$$

$$\text{Pr}[E_1] = 3p-2p^2$$

* Instruction STOP

Si TRACE

Alors $Pr[E'] = Pr[E_1] = 9p - 20p^2 + 12p^3$

Sinon

Si RESET

Alors $Pr[E'] = Pr[E_1] = 8p - 17p^2 - 10p^3$

* Instruction TAS

Le tableau XI résume les différentes valeurs de E' et E₁ pour l'instruction TAS selon les différents modes d'adressage.

ADRESSAGE	Pr[E']	Pr[E ₁]
Dn	p	p
(An)	p	$2p - p^2$
(An) +	p	$2p - p^2$
- (An)	p	$2p - p^2$
d(An)	2p	$3p - 2p^2$
d(An, Ri)	2p	$4p - 4p^2$
XXX.W	2p	$3p - 2p^2$
XXX.L	3p	$4p - 3p^2$

Tableau XI. Pr[E'], Pr[E₁] pour l'instruction TAS selon le mode d'adressage.

* Instruction TRAP

$$\text{Pr}[E'] = 9p - 13p^2 + 6p^3$$

$$\text{Pr}[E_1] = 9p - 13p^2 + 6p^3$$

* Instruction TRAPV

Si V

Alors $\text{Pr}[E'] = \text{Pr}[E_1] = 9p - 13p^2 + 6p^3$

Sinon $\text{Pr}[E'] = \text{Pr}[E_1] = p$

* Instruction UNLK

$$\text{Pr}[E'] = p$$

$$\text{Pr}[E_1] = 3p - 2p^3$$

* Interruption

$$\text{Pr}[E'] = 8p - 13p^2 + 6p^3$$

$$\text{Pr}[E_1] = 1p - 13p^2 + 6p^3$$

Le mix d'instruction, permettant d'évaluer $\text{Pr}[E']$ et $\text{Pr}[E_1]$ pour l'opérateur AU, a été calculé sur le moniteur résidant de la carte MOTOROLA. C'est un programme d'un peu plus de 1200 instructions, la table XII résume les probabilités d'occurrence des différents types d'instructions, le nombre de modes d'adressage utilisé pour chaque type, la probabilité $\text{Pr}[E']$ et $\text{Pr}[E_1]$ moyenne pour le type d'instruction considéré selon le mode d'adressage.

Instructions	Pr[occurrence]	Nombre de modes d'adressage	Pr[E']	Pr[E ₁]
Opérations arithmétiques	0.128	12	1.77p	$2.33p - 1.59p^2 + 0.98p^3$
Décalages	0.009	2	1p	$8p - 7p^2$
Divisions	0.002	1	2p	$18p - 32p^2$
Multiplications	0.003	2	2p	$18.5p - 41p^2 + 16p^3$
SWAP	0.009	1	1p	1p
Comparaison	0.103	10	1.82p	1.82p
Test	0.034	4	1.61p	1.61p
MOVE	0.313	25	2.09p ³	$3.32p - 2.65p^2 + 0.52p^3$
MOVEM	0.010	2	2p	$18.66p - 11.41p^2$
BCC	0.233	2	1p	1p
BSR	0.119	1	2p	$5p - 9p^2 + 6p^3$
JMP	0.001	1	0p	2p
JSR	0.001	1	0p	2p
RTS	0.034	1	$3p - 2p^2$	$3p - 2p^2$
RTE	0.001	1	$4p - 3p^2$	$4p - 3p^2$

Tableau XII. Probabilité d'occurrence, modes d'adressage, moyenne de Pr[E'] et Pr[E₁] pour chaque type d'instruction du mix considéré.

Les résultats obtenus à partir du tableau XII sont les suivants :

$$\text{Pr}[E'] = 1.75p - 0.07p^2$$

$$\text{Pr}[E_1] = 2.88p - 2.54p^2 + 1.05p^3$$

Si l'on considère une distribution uniforme des valeurs à l'entrée de l'opérateur AU, on a $p = \frac{1}{2^{32}}$ et par suite :

$$\text{MTE}'_1 = \text{latence de la machine.}$$

$$= \frac{1}{2.88p - 2.54p^2 + 1.05p^3}$$

$$= 1.49 * 10^9 \text{ instructions.}$$

$$\text{MTE}' = 2.45 * 10^9 \text{ instructions.}$$

En fait, contrairement au M6800, sur le MC 68000, seuls 24 des 32 bits du compteur ordinal sont utilisés pour adresser une instruction, et donc la valeur de p à prendre en compte est donnée par la formule de divergence d'un lien en sous lien. En fait nous obtenons ici :

$$\text{SU}(24 \text{ bits}) = \text{OTP}_{24\text{bits}} \cdot \text{SU}(32\text{bits}).$$

$$\text{Avec } \text{OTP}_{24 \text{ bits}} = \frac{2^{32} (2^{24} - 1) 2^{32-24}}{2^{32} (2^{32} - 1)}$$

$$\approx 1 - \frac{1}{2^8} = 0.996$$

Donc finalement il y a une légère différence entre l'erreur sur le compteur ordinal et la création d'un déséquencement.

$$\text{MTE}'_{\text{pc}} = 2.45 * 10^9 \text{ instructions}$$

$$\text{MTE}'_{\text{déseq.}} = 2.46 * 10^9 \text{ instructions.}$$

De toute façon, cette valeur conduit à une fréquence trop faible de génération de E et par suite un test périodique devra être effectué sur cet opérateur..

En effet, à une fréquence d'horloge de 8MHz, l'évènement E ne serait créé par cet opérateur que toutes les 2h30 de calcul. (ceci en comptant qu'une instruction dure environ 3.75×10^{-6} sec. soit 30 cycles en moyenne par instruction).

On peut remarquer l'écart énorme en comparaison des résultats obtenus pour le M 6800 pour lequel le MTE' était borné par 32 600 instructions représentant un temps de calcul de 0.06 seconde à 2MHz.

IV.1.2. Registres :

Les registres de la machine, correspondant aux entrées primaires des paragraphes précédents, vont être passés en revue.

On suppose que, pour une panne affectant un registre, compte-tenu des distributions D, la SU des sorties de ce registre soit égale à 0,5 vis à vis de E_1 et 0.375 vis à vis de E' lorsque ce registre est utilisé. Les circuits opératifs ont cette fois une OTP = 1.

IV.1.2.1. Registres donnés (Dn) :

Ces registres ne peuvent créer d'évènements E que lorsqu'ils sont index dans les instructions JMP et JSR. Ainsi on obtient par extrapolation sur le mix utilisé précédemment :

+ instruction JMP d(An,Dn)

$$\text{Pr}[E'] = 0.375$$

$$\text{Pr}[E_1] = 0.5$$

+ instruction JSR d(An, Dn)

$$\text{Pr}[E'] = 0.375$$

$$\text{Pr}[E_1] = 0.5$$

+ autres instructions impliquant un registre donnée

$$\text{Pr}[E'] = 0$$

$$\text{Pr}[E_1] = 0.5$$

Compte tenu des valeurs relatives, on obtient :

$$\text{MTTE}' = \frac{1}{0.375 * 0.4 * 10^{-4}} = 66\ 000$$

$$\text{MTTE}'_1 = \frac{1}{0.5 * 0.049} = 40$$

IV.1.2.2. Registres adresses (An) :

Pour les registres adresses on distinguera les registres simples des pointeurs de piles.

Pour les registres simples on obtient les résultats suivants :

$$\text{MTTE}' = \frac{1}{0.375 * 1.4 * 10^{-4}} = 19\ 000$$

$$\text{MTTE}'_1 = \frac{1}{0.5 * 0.034} = 50$$

Pour les pointeurs de piles on obtient :

$$\text{MTTE}' = \frac{1}{0.375 * 0.035} = 76$$

$$MTTE_1 = \frac{1}{0.5 \times 0.041} = 48$$

IV.1.2.3. Registres temporaires (DT et AT) :

Le registre DT est utilisé principalement pour stocker des opérandes immédiats. On obtient :

$$MTTE' \rightarrow \infty$$

$$MTTE_1 = \frac{1}{0.5 \times 0.296} = 7$$

Le registre AT est utilisé principalement pour stocker les adresses des destinations lorsqu'une instruction à 2 opérandes à son résultat en mémoire. On obtient :

$$MTTE' = \frac{1}{0.375 \times 1.64 \times 10^{-3}} = 1626$$

$$MTTE_1 = \frac{1}{0.5 \times 0.165} = 12$$

IV.1.2.4. Buffer d'adresse (AOB) :

Il est évident qu'il est utilisé à chaque instruction et immédiatement il vient :

$$MTTE' = 3$$

$$MTTE_1 \rightarrow 3$$

IV.1.2.5. Compteur ordinal (PC) :

Les calculs nous donnent :

$$MTTE' = 3$$

$$MTTE_1 = 2$$

IV.1.2.6. Latch d'AU (AU-R) :

On obtient aisément :

$$MTTE' = 3$$

$$MTTE_1 = 2$$

IV.1.2.7. Registre de FTU :

Ce registre est utilisé par des instructions très diverses pour lesquelles une donnée se trouve dans le mot instruction : Bcc Moveq, adda, etc...

On obtient :

$$MTTE' = \frac{1}{0.375 \times 0.233} = 11$$

$$MTTE_1 = \frac{1}{0.5 \times 0.249} = 8$$

IV.1.2.8. Registre de PREN :

Il n'est jamais utilisé pour évaluer une adresse instruction et ne peut donc pas créer E. On obtient pour ce registre.

$$MTTE' \rightarrow \infty$$

$$MTTE_1 = \frac{1}{0,5 \times 9.85 \times 10^{-3}} = 203$$

IV.1.2.9. Registre de DCR :

Il n'est jamais utilisé pour calculer une adresse instruction et ne peut donc pas créer E. On obtient donc pour ce registre :

$$MTTE' \rightarrow \infty$$

$$MTTE_1 = \frac{1}{0.5 \times 1.64 \times 10^{-4}} = 12\ 180$$

IV.1.2.10. Registre DBIN :

Il est utilisé pour entrer la valeur de l'adresse ou une partie pour les adressages : indirect plus déplacement, indirect indexé plus déplacement, absolu court, absolu long, relatif et relatif indexé. On obtient donc :

$$MTTE' = \frac{1}{0.375 \times 6.56 \times 10^{-4}} = 4065$$

$$MTTE_1 = \frac{1}{0.5 \times 0.822} = 2$$

IV.1.2.11. Registre DOB :

Ce registre n'est jamais utilisé pour calculer une adresse instruction. Il ne sert que pour les sorties mémoire. On obtient donc :

$$MTTE' \rightarrow \infty$$

$$MTTE_1 = \frac{1}{0.5 \times 0.214} = 9$$

IV.1.2.12. Registre d'ALUE :

Ce registre n'est jamais utilisé pour calculer une adresse instruction. Il ne sert que pour les décalages de registres. On obtient donc :

$$MTTE' \rightarrow \infty$$

$$MTTE_1 = \frac{1}{0.5 \times 9.03 \times 10^{-3}} = 221.$$

IV.1.2.13. Latch d'ALU (ALU-R) :

Ce registre est utilisé pour générer un complément d'adresse instruction, uniquement en cas d'adressage indirect indexé plus déplacement, indirect plus déplacement, relatif et relatif indexé. On obtient ainsi :

$$MTTE' = \frac{1}{0.375 * 4.7 \times 10^{-4}} = 5\ 684$$

$$MTTE_1 = \frac{1}{0.5 * 0.592} = 3$$

IV.1.2.14. Registre ALUB :

Ce registre n'est jamais utilisé pour générer une adresse d'instruction, il sert de tampon à l'ALU on obtient :

$$MTTE' \rightarrow \infty$$

$$MTTE_1 = \frac{1}{0.5 \times 0.014} = 143$$

IV.1.3. Bus :

Successivement, les 3 paires de bus traversant la partie opérative vont être passées en revue. A ces bus il convient d'ajouter les bus EAB, EDB et BC.

IV.1.3.1. Bus ADB, DBD :

En ce qui concerne la latence de ces pannes, on obtient une latence presque nulle, ceci en raison du taux d'utilisation des bus D' pour l'entrée/sortie de données. On obtient ainsi :

$$MTTE' = \frac{1}{0.5 \times 6.56 \times 10^{-4}} = 3050$$

$$MTTE_1 = \frac{1}{0.5 * 0.822} = 2$$

IV.1.3.2. Bus ABL, DBL :

Pour ce bus on obtient :

$$MTTE' = \frac{1}{0.5 * 6.80 \times 10^{-4}} = 2940$$

$$MTTE_1 = \frac{1}{0.5 * 0.820} = 2$$

IV.1.3.3. Bus ABH, DBH :

Ces bus sont légèrement moins utilisés pour les générer E_1 , en effet les opérations sur mot ou octet ne "touchent" pas à ces bus et donc on obtient :

$$MTTE' = \frac{1}{0.5 \times 6.80 \times 10^{-4}} = 2940$$

$$MTTE_1 = \frac{1}{0.5 * 0.25} = 7$$

IV.1.3.4. Bus EAB :

Ce bus relie le buffer d'adresses (AOB) aux plots d'adresses. Il est interne à la partie, bien que son nom soit external adresse bus, et ne correspond pas au bus d'adresse externe décrit dans le test de la périphérie de la pastille (Chapitre III).

Ce bus sert uniquement à transférer les adresses vers les plots adresses, Ainsi on obtient :

$$MTTE' = 2$$

$$MTTE_1 = 2$$

IV.1.3.5. Bus EDB :

Ce bus relie l'opérateur MUX I/O-aux plots données. Il est interne à la pastille, et ne correspond pas au bus externe de données décrit dans le test de la périphérie de la pastille (Chapitre III). Ce bus sert aussi bien à l'acquisition des données qu'au rangement en mémoire. On obtient ainsi :

$$MTTE' = \frac{1}{0.5 * 6.56 \times 10^{-4}} = 3050$$

$$MTTE_1 = \frac{1}{0.5 * 0.99} = 2.$$

IV.1.3.6. Bus BC :

Le bus BC est utilisé pour transférer les valeurs du déplacement sur octet pour les instructions Bcc et BSR. On obtient ainsi :

$$MPTE' = \frac{1}{0.5 \times 0.233} = 9$$

$$MPTE_1 = \frac{1}{0.5 \times 0.249} = 8.$$

Le tableau XIII regroupe tous les résultats obtenus pour la partie opérative du MC 68000.

Partie opérative		MITE'	MITE ₁
OPERATEURS	ALU	∞	77 500
	DCB	∞	79 * 10 ⁶
	ALUE	∞	458 300
	MUX I/O	∞	52 100
	DCR	∞	79 * 10 ⁶
	Fool it	∞	210 000
	PREN	∞	1 * 10 ⁶
	SIGNEX	∞	105 700
	AUH AUL	∞ (2.46 * 10 ⁹)	1.49 * 10 ⁹
REGISTRES	Dn	66 000	40
	An	19 000	59
	SP	76	48
	DT	∞	7
	AT	1626	12
	AOB	3	3
	PC	3	2
	AU - R	3	2

Tableau XIII. Valeurs de MITE' et MITE₁ pour la partie opérative du MC 68000 (début).

		MTTE'	MTTE ₁
REGISTRES	FTU	11	8
	PREN	∞	203
	DCR	∞	12 180
	DBIN	4065	2
	DOB	∞	9
	ALUE	∞	221
	ALUR	5684	3
	ALUB	∞	143
BUS	ABD, DBD	3050	2
	ABL, DBL	2940	2
	ABH, DBH	2940	7
	EAB	2	2
	EDB	3050	2
	BC	9	8

Tableau XIII. Valeurs de MTTE' et MTTE₁ pour la partie opérative du MC 68000 (fin).

IV.2. Partie contrôle :

Des pannes affectant la partie contrôle peuvent rentrer dans les hypothèses utilisées pour le test fonctionnel de la partie contrôle, ces hypothèses seront expliquées au chapitre VII.

Les cinq classes d'hypothèses de pannes sont rappelées dans le tableau XIV.

D'autres pannes ne rentreront pas dans ces classes. Successivement, les pannes entrant dans les classes 0 à V, puis les pannes à conséquences hors classes sont examinées.

Classe 0	Une instruction opérative ne "calcule" pas correctement : le résultat n'est pas le résultat juste. Elle modifie la valeur de registres opératifs qu'elle ne devrait pas modifier.
Classe I	Une instruction non-opérative modifie de manière intempestive des registres opératifs.
Classe II	Une instruction non-opérative ne "calcule" pas correctement si elle doit calculer. Elle modifie de manière intempestive des registres non opératifs.
Classe III	Une instructive opérative modifie de manière intempestive des registres non opératifs.
Classe IV	Une instruction, opérative ou non, ne gère pas correctement le code condition.
Classe V	Une instruction (non opérative) de branchement ne "branche" pas à bon escient.

Tableau XIV. Classes d'hypothèses de pannes du test fonctionnel de la partie contrôle.

IV.2.1. Classe 0 :

A cette classe 0, on inclut les pannes propres des opérateurs ALU, BCD, ALUE, MUXE/Q DCR, FOOL-IT, PREN, SIGNEX. L'évènement E n'est pas créé, en général, par cette classe 0 ou par ces opérateurs. Le temps de latence dépend évidemment de la panne particulière considérée pour ces opérateurs, de l'instruction opérative et des distributions D.

La seule borne supérieure de la probabilité de générer l'évènement E_1 peut être donnée par :

- . La somme des probabilités d'exécution des instructions utilisant chaque opérateur en cas de panne de l'opérateur considéré.
- . La probabilité d'exécution de l'instruction opérative incriminée en cas de panne de la partie contrôle,

puisque les OTP_g ne peuvent que diminuer ces probabilités pour donner la probabilité de générer E_1 . La première des deux implique une latence presque nulle la seconde pouvant être un peu plus importante. On admettra donc :

$$MTTE' \rightarrow \infty \qquad MTTE_1 \rightarrow 2$$

IV.2.2. Classe I :

La classe I ne fait que modifier de manière intempestive des registres opératifs. L'évènement E n'est en général pas créé. La latence est presque nulle, car bien qu'une instruction répondant à la définition de la classe I puisse avoir une faible fréquence, le registre opératif sera lui utilisé très rapidement par une autre instruction. On admettra donc :

$$MTTE' \rightarrow \infty \qquad MTTE_1 \rightarrow 2$$

IV.2.3. Classe II :

La classe II est intéressante, du point de vue de la détection des pannes, car si la latence est faible, grande probabilité de générer E_1 , la fréquence de E est importante pour cette classe. Il y a corrélation entre MTE' et MTE_1 . Deux cas peuvent être distingués selon le fait que la mauvaise valeur du registre non opératif résulte d'un mauvais calcul ou résulte d'une altération intempestive.

Pour chacun de ces 2 cas, une hypothèse est faite : un seul des registres non opératifs est affecté.

IV.2.3.1. Sous-classe 1 : une instruction non opérative ne calcule pas correctement :

On a alors, pour le registre Z par exemple :

$$\Pr[E'] = SU(Z), \Pr[If], \Pr[A*], \Pr[B]$$

où

SU(Z) est la SU du registre Z considéré, après exécution de If, If est l'instruction non opérative à mauvais séquençement, altérant la valeur de Z par un calcul sur Z.

A* est une suite, qui peut être vide, d'instructions autres que celles effaçant la panne (par chargement de Z par exemple) et celles créant l'évènement E_1 .

B est une instruction non opérative utilisant le registre Z considéré pour évaluer une adresse instruction (et par suite créant l'évènement E) ;

De même, on obtient :

$$\Pr[E_1] = SU(Z) \Pr[If]$$

où $SU(Z)$ et If ou la même définition que précédemment.

En effet, dans ce cas, If effectuant un calcul sur Z , l'altération de Z provoque immédiatement l'évènement E_1 .

IV.2.3.2. Sous-classe 2 : altération intempestive d'un registre non-opératif :

Pour l'évaluation de E' , la formule est de même nature, ainsi on obtient :

$$Pr[E'] = SU(Z) . Pr[Jf] . Pr[A*] Pr[B]$$

où

$SU(Z)$ est la SU du registre Z considéré, après exécution de Jf est l'instruction non-opérative à mauvais séquençement altérant de manière intempestive le registre Z .

A^* est une suite, qui peut être vide, d'instructions autres que celles effaçant la panne (par chargement de Z par exemple) et celles créant E .

B est une instruction non opérative utilisant le registre Z pour évaluer une adresse instruction (et par suite créant E).

De la même manière ; on obtient :

$$Pr[E_1] = SU(Z) . Pr[Jf] . Pr [C*] . Pr[D]$$

où

$SU(Z)$ et $Pr[Jf]$ ont la même définition que précédemment.

C^* est une suite, qui peut-être vide, d'instructions autres que celles effaçant la panne (par chargement de Z par exemple) et celles créant E_1 .

D est une instruction non opérative utilisant le registre Z considéré (et donc créant E_1)

A partir de ces formules, les valeurs obtenues pour les différents registres non opératifs sont présentées dans le tableau XV.

Registres non-opératifs	Sous-classe 1		Sous-classe 2	
	MTE'	MTE ₁	MTE'	MTE ₁
An (Sans SP)	$455 \frac{1}{Pr\{If\}}$	$2 \frac{1}{Pr\{If\}}$	$455 \frac{1}{Pr\{Jf\}}$	$18 \frac{1}{Pr\{Jf\}}$
SP	$2 \frac{1}{Pr\{If\}}$	$2 \frac{1}{Pr\{If\}}$	$124 \frac{1}{Pr\{Jf\}}$	$6 \frac{1}{Pr\{Jf\}}$
Index (An, Dn)	$459 \frac{1}{Pr\{If\}}$	$2 \frac{1}{Pr\{If\}}$	$459 \frac{1}{Pr\{Jf\}}$	$46 \frac{1}{Pr\{Jf\}}$
PC	$2 \frac{1}{Pr\{If\}}$	$2 \frac{1}{Pr\{If\}}$	$2 \frac{1}{Pr\{Jf\}}$	$2 \frac{1}{Pr\{Jf\}}$
AT-AOB-AUR	∞	$2 \frac{1}{Pr\{If\}}$	∞	∞

If = instruction non-opérative à mauvais séquençement altérant le registre considéré en effectuant un calcul sur ce registre.

Jf = instruction non-opérative à mauvais séquençement altérant intempestivement le registre considéré.

Tableau XV. Valeur de MTE' et MTE₁ pour les 2 sous-classes de la classe II.

IV.2.4. Classe III

Les estimations pour des pannes affectant la partie contrôle dans le cadre de la classe III, sont strictement semblables à celles obtenues pour la sous-classe 2 de la classe II, Jf caractérisant cette fois une instruction opérative (cf. Tableau XV).

IV.2.5. Classe IV

Une latence presque nulle et un MTTE' infini seront considérés pour cette classe. En effet, les instructions modifiant le code condition constitue la plus grande part du jeu d'instruction et créeront donc immédiatement E_1 . ^{aa} D'autre part d'après les hypothèses, une mauvaise évaluation de la condition de branchement ne crée pas E. On obtient donc :

$$\text{MTTE}' \rightarrow \infty \qquad \text{MTTE}'_1 \rightarrow 2$$

IV.2.6. Classe V

La latence pour cette classe de pannes pourrait être donnée par les branchements. Mais cette latence serait de toutes manières très courte. En ce qui concerne E_1 et conformément aux hypothèses, i.e. un mauvais calcul de la condition de branchement d'un branchement conditionnel ne produisant pas E, le MTTE' est considéré comme infini. Donc on obtient :

$$\text{MTTE}' \rightarrow \infty \qquad \text{MTTE}'_1 \rightarrow 2$$

IV.2.7. Pannes à conséquences hors classes

Il serait possible d'envisager des pannes ayant des conséquences courant plusieurs classes. Malheureusement, on peut noter que l'estimation des conséquences possibles réelles des pannes dans la partie contrôle est malaisée.

Ainsi un outil permettant une simulation fine du comportement de la partie contrôle en cas de panne permettrait d'appréhender certaines conséquences à priori imprévisibles. On verra ainsi que pour le test fonctionnel de la partie contrôle (Chapitre VII) des simulations électriques ont été nécessaires afin de déterminer les conséquences de certaines pannes durant les transferts.

Le tableau XVI regroupe les valeurs obtenues pour les différentes classes envisagées pour le test de la partie contrôle.

Partie Contrôle			MTTE'	MTTE ₁
Classe 0			∞	2
Classe I			∞	2
Classe II	Sous-classe 1	An	$455 \frac{1}{\text{Pr}[\text{If}]}$	$2 \frac{1}{\text{Pr}[\text{If}]}$
		SP	$2 \frac{1}{\text{Pr}[\text{If}]}$	$2 \frac{1}{\text{Pr}[\text{If}]}$
		Index	$459 \frac{1}{\text{Pr}[\text{If}]}$	$2 \frac{1}{\text{Pr}[\text{If}]}$
		PC	$2 \frac{1}{\text{Pr}[\text{If}]}$	$2 \frac{1}{\text{Pr}[\text{If}]}$
		AT-AOB-AUR	∞	$2 \frac{1}{\text{Pr}[\text{If}]}$
	Sous-classe 2	An	$455 \frac{1}{\text{Pr}[\text{Jf}]}$	$18 \frac{1}{\text{Pr}[\text{Jf}]}$
		SP	$124 \frac{1}{\text{Pr}[\text{Jf}]}$	$6 \frac{1}{\text{Pr}[\text{Jf}]}$
		Index	$459 \frac{1}{\text{Pr}[\text{Jf}]}$	$46 \frac{1}{\text{Pr}[\text{Jf}]}$
		PC	$2 \frac{1}{\text{Pr}[\text{Jf}]}$	$2 \frac{1}{\text{Pr}[\text{Jf}]}$
		AT-AOB-AUR	∞	∞

Tableau XVI. Valeurs de MTTE' et MTTE₁ pour les différents classes de pannes de la partie contrôle du MC 68000 (début).

		MTE°	MTE ₁
Classe III	An	455 $\frac{1}{Pr[kf]}$	18 $\frac{1}{Pr[kf]}$
	SP	124 $\frac{1}{Pr[kf]}$	6 $\frac{1}{Pr[kf]}$
	Index	459 $\frac{1}{Pr[kf]}$	46 $\frac{1}{Pr[kf]}$
	Pc	2 $\frac{1}{Pr[kf]}$	2 $\frac{1}{Pr[kf]}$
	AT-AOB AUR	∞	∞
Classe IV		∞	∞
Classe V		∞	∞

Tableau XVI. Valeurs de MTE° et MTE₁ pour les différentes classes de pannes de la partie contrôle du MC 68000 (fin).

V - CONCLUSION

Les résultats obtenus pour le MC 68000 sont donc récapitulés dans le tableau XIII pour la partie opérative et XVI pour la partie contrôle. MTTE caractérise le temps de latence au niveau de la machine, MTTE' est la borne supérieure moyenne du temps de génération de E, borne supérieure également de l'intervalle entre deux occurrences successives de E. Ces calculs ont été effectués sur un mix d'instructions constitué du moniteur de la carte MOTOROLA. Les pannes entraînant la génération de E, avec une fréquence suffisamment élevée, pourront être détectées en-ligne par les détecteurs internes ou externes (chapitres III et IV), pour les autres, un test périodique devra être envisagé.

On peut remarquer, en raison de la grande largeur du bus adresse, que l'opérateur de calcul d'adresses devra être testé périodiquement, ce qui n'était pas le cas pour le M6800 pour lequel le MTTE (32 600 instructions) correspondait à un temps réel d'environ 0,06 sec. (2 MHz), ici bien que le MC 68000 soit 4 fois plus rapide que le M 6800, le MTTE obtenu correspond à environ 2H30 de temps réel (8 MHz).

REFERENCES

- [AMA 62] AMAREL S., BRZOZOWSKI J.A.
"Theoretical considerations on reliability properties of recursive triangular switching networks"
Redundancy techniques for Computing systems, Wilcox & Mann, 1962.
- [COU 76] COURTOIS B.
"Fiabilité et sécurité de quelques systèmes logiques"
Congrès AFCET, Automatismes logiques: recherches et applications industrielles, Paris, décembre 1976.
- [COU 80] COURTOIS B.
"Observability, signal reliability and error latency in digital networks"
Rapport de recherche IMAG, RR 188, février 1980.
Soumis à publication.
- [COU 81] COURTOIS B.
"Test et LSI"
Thèse d'Etat, Grenoble, 1981.
- [DAV 79] DAVID R., THEVENOD-FOSSE P.
"Un outil pour l'étude du test aléatoire de la partie opérative des microprocesseurs"
2ème Colloque international sur la Fiabilité et la maintenabilité, Perros-Guirec, France, 8/12 septembre 1980.
- [ESA 62] ESARY J.D., PROSHAN F.
"The reliability of coherent systems"
Redundancy techniques for computing systems,
Wilcox & Mann , 1962.

- [KOR 77] KOREN I.
"Signal reliability of combinatorial and sequential circuits"
7th Fault-tolerant computing Symposium, Los Angeles, USA,
June 1977.
- [OGU 75] OGUS R.
"The probability of a correct output from a combinatorial
circuit"
IEEE Transactions on computers, vol. C 24, n°5, May 1975.
- [PAR 78] PARKER K.P., McCLUSKEY E.J.
"Sequential circuit output probabilities from regular expressions"
IEEE Transactions on computers, vol. C 27, n°3, March 1978.
- [SHE 75] SHEDLETSKY J., McCLUSKEY E.J.
"The error latency of a fault in a combinatorial digital circuit"
5th Fault-tolerant computing Symposium, Paris, France, June 1975.
- [SHE 76-1] SHEDLETSKY J., McCLUSKEY E.J.
"The error latency of a fault in a sequential digital circuit"
IEEE Transactions on computers, vol. C 25, n°6, June 1976.
- [SHE 76-2] SHEDLETSKY J.
"A rollback interval for networks with an imperfect
self-checking property"
6th Fault-tolerant computing Symposium, Pittsburgh, USA, June 1976
(IEEE Transactions on computers, vol. C 27, n°6, June 1978).
- [THA 78] THATTE S.M., ABRAHAM J.A.
"A methodology for functional level testing of microprocessors"
8th Fault-tolerant computing Symposium, Toulouse, France,
June 1978.

CHAPITRE V

COEUR DE LA PASTILLE CRÉANT E

Etape (iii)

Etape (iv)



CHAPITRE V - COEUR DE LA PASTILLE CREANT E

Etape (iii) - Etape (iv)

I.	INTRODUCTION	161
II.	ANALYSE STATIQUE MODELISATION	163
1.	Analyse Qualitative	163
2.	Définitions des modèles statiques	164
2.1.	Premier Modèle	161
2.1.1.	Accès matriciel	165
	. Probabilités d'états	168
	. Distributions, densités, probabilités limites	169
	. Moments de détection	172
2.1.2.	Accès direct	175
	. Probabilités d'état, Probabilités de détection	175
	. Moments de détection	177
2.2.	Second Modèle	178
2.2.1.	Accès matriciel	178
	. Probabilités d'états	178
	. Distributions, densités, probabilités limites	180
	. Moments de détection	182
2.2.2.	Accès direct	184
	. Probabilités d'états Probabilités de détection	184
	. Moments de détection	185
2.3.	Récapitulation	186
III.	RESULTATS EXPERIMENTAUX	191
IV.	ANALYSE DYNAMIQUE MODELISATION	202
1.	Evénements E de faibles fréquences	202
2.	Evénements E de grandes fréquences. Définitions des modèles dynamiques.	203
2.1.	Premier modèle	203
2.1.1.	Accès matriciel	205

. Probabilités d'états	205
2.1.2. Accès direct :	205
. Probabilités d'états, probabilités de détection.....	207
. Moments de détection	209
2.2. Second modèle	211
2.2.1. Accès matriciel	211
. Probabilités d'états	211
. Distributions, densités et probabilités limites	218
. Moments de détection	221
2.2.2. Accès direct	224
. Probabilités d'états et probabilités de détection	224
. Moments de détection	225
2.3. Récapitulation et discussion - Caractéristique de la fréquence de E.	226
2.3.1. Récapitulation	226
2.3.2. Valeur caractéristique de la fréquence de E.	229
V. DYNAMISATION DES RESULTATS EXPERIMENTAUX	231
VI. GENERALISATION A UNE APPLICATION QUELCONQUE	235
1. Distribution et densité du temps de détection	236
1.1. Valeurs caractéristiques pour $f = 1$	237
1.2. Valeurs caractéristiques pour $f \rightarrow 0$	238
2. Moments du temps de détection en fonction de la fréquence	242
VII. CONCLUSION	245
NOTES BIBLIOGRAPHIQUES	247
REFERENCES	249

I. INTRODUCTION

On parlera des pannes du coeur, par opposition aux pannes de la périphérie. Les pannes considérées ici sont toutes les pannes du coeur, produisant un déséquence- ment du programme, c'est-à-dire les pannes produisant un événement particulier appelé événement E.

Événement E : La panne vient de générer une erreur sur le calcul de l'adresse de l'instruction suivante.

Il va de soi que de telles pannes ne peuvent être détectées que par l'utilisation de détecteurs "externes" ou par duplication de CPU's. En effet, un programme de test ne peut s'exécuter puisqu'il serait lui même sous l'effet du déséquence- ment produit par l'occurrence de l'événement E.

Ne désirant pas utiliser de redondance massive, on utilisera les mêmes détecteurs que ceux employés pour les pannes de la périphérie : code opération invalide, adresse erreur, protection mémoire et chien de garde.

Comme pour les pannes de la périphérie, le temps de détection pris en compte sera le temps séparant l'apparition de l'événement E, de l'instant de détection (où le premier détecteur a été déclenché).

Mais, si pour les pannes de la périphérie, le temps de détection était borné par la période du chien de garde, - il ne pouvait y avoir de relance du chien de garde en cas de panne de la périphérie - ici la période du chien de garde ne peut être prise en compte car il est possible de ne pas détecter la panne.

L'événement E va survenir un certain temps après qu'une erreur se soit produite dans le coeur même du CPU, et il va se produire avec une certaine fréquence. On supposera connu ce temps de latence de l'événement E ainsi que sa fréquence, (cf chapitre IV).

Il est clair que les pannes du coeur ne créant pas l'événement E, ou le créant avec une très faible fréquence seront testées par des programmes spécifiques de test.

L'étude va se décomposer en deux temps. Tout d'abord, un modèle statique va être développé : c'est le modèle correspondant à l'occurrence d'un et un seul événement E. Les résultats de simulations fonctionnelles pour une application particulière donneront les valeurs des paramètres du modèle, ceux-ci seront ensuite extrapolés pour une application quelconque. Ces résultats fourniront donc la réponse à la question : Quel est le temps écoulé entre l'occurrence d'un événement E et l'instant où le premier détecteur est déclenché ?.

Ensuite, un modèle dynamique sera développé, en ne considérant plus un événement E mais une suite d'événements E se produisant avec une certaine fréquence. Les paramètres obtenus précédemment seront repris en fonction cette fois de la fréquence de l'événement E. Ces résultats fourniront la réponse à la question : Sachant la fréquence d'apparition d'un événement E, quel est le temps écoulé entre la première occurrence de l'événement E, et l'instant où le premier détecteur est déclenché ? .

Les modèles seront développés sous la forme de chaînes de Markov à nombre fini d'états, les résultats généraux et formules sur les chaînes de Markov ne seront pas entièrement explicités.

II. ANALYSE STATIQUE MODELISATION

On ne va s'intéresser dans un premier temps qu'au comportement du CPU après un seul événement E : c'est l'analyse statique du modèle. On suppose dans ce premier temps qu'après l'occurrence de E, tout se passe normalement.

1. Analyse Qualitative

A la suite d'un événement E :

L'adresse de la prochaine instruction est quelconque : cette adresse correspond ou ne correspond pas à une adresse d'instruction du programme :

- a) si cette adresse ne correspond pas à une adresse d'instruction du programme :
 - +) le code opération obtenu est tel qu'il génère immédiatement une détection par COI, AE, PM ou se produit à un moment tel qu'il génère immédiatement une détection par CG.

 - +) Le code opération obtenu est tel qu'il est exécutable sans détection d'erreur, mais, il va provoquer des perturbations pour la suite de l'exécution. A la suite de son exécution, on pointe de nouveau sur une adresse quelconque.

- b) si cette adresse correspond à une adresse d'instruction :
 - +) Le code opération obtenu est correct, mais comme il n'est pas exécuté dans son contexte et qu'il peut ne pas correspondre au contexte en cours, il génère immédiatement une détection par COI, AE, PM ou CG.

 - +) Le code opération obtenu est correct et s'exécute sans détection immédiate, mais il génère des perturbations pour la suite de l'exécution du programme. Après avoir exécuté cette instruction le CPU peut de nouveau pointer sur une adresse quelconque.

c) Ne rien faire de (a) et de (b) c'est-à-dire, que le code rencontré, sur une adresse d'instruction ou non, est tel que son exécution va se dérouler sans perturbation et il n'y aura jamais détection.

2. Définitions des Modèles Statiques

Si le modèle théorique développé ici est différent de celui utilisé pour le M 6800, en fait le modèle utilisé, tiré du modèle théorique, est équivalent à celui du M 6800. Comme pour le M 6800, deux modèles seront développés selon le fait que l'on différencie ou non l'exécution de la première instruction.

2.1. Premier Modèle

Pour ce modèle, l'exécution de la première instruction est différenciée. On calculera ainsi séparément les probabilités de détection à la première instruction ou aux instructions suivantes. D'après ces considérations et l'analyse qualitative du modèle, on obtient le schéma de la figure 1 .

En fait, ce modèle ne sera pas retenu, en effet, les résultats expérimentaux ne permettraient pas de connaître les probabilités de passage d'une adresse à une "non-adresse" d'instruction et réciproquement et ceci que ce soit pour l'état transitoire ou pour les états de non détection. En fait, ce résultat n'est pas nécessaire à la connaissance du temps de détection. C'est pourquoi, le modèle de la figure 2, qui regroupe les états adresse et "non adresse", sera développé.

2.1.1. Accès Matriciel

Soit K le nombre d'instructions exécutées avant détection, et K_{COI} , K_{AE} , K_{PM} , K_{CG} , les variables propres à chacun des détecteurs. Une distribution de K ne sera pas cherchée, puisqu'il peut ne pas y avoir détection. De même K_{CG} ne sera pas étudié puisque ce nombre dépend de la période T_0 du chien de garde. Ces calculs seront effectués pour K_{COI} , K_{AE} , et K_{PM} de même que les probabilités de détection pour chaque détecteur.

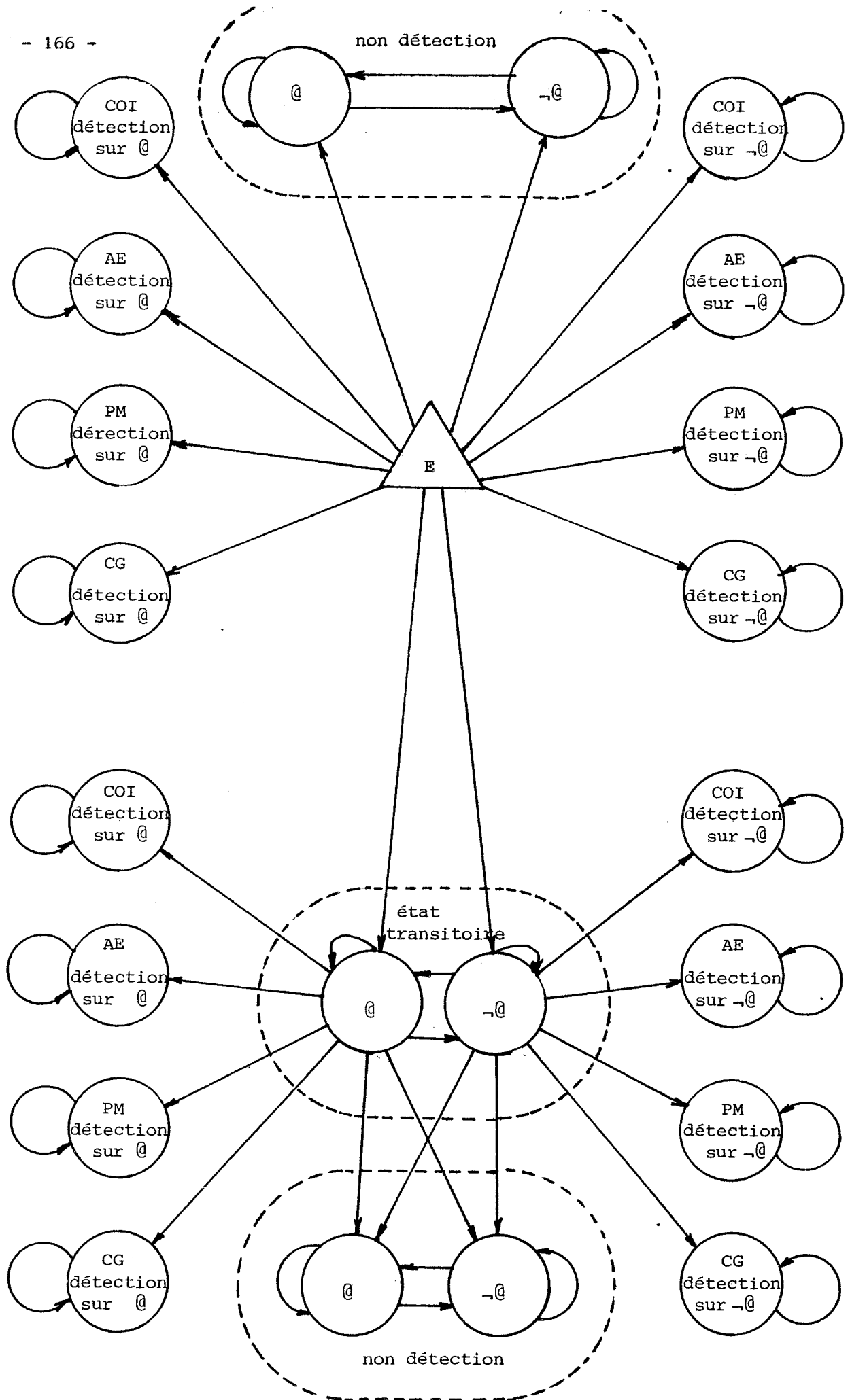
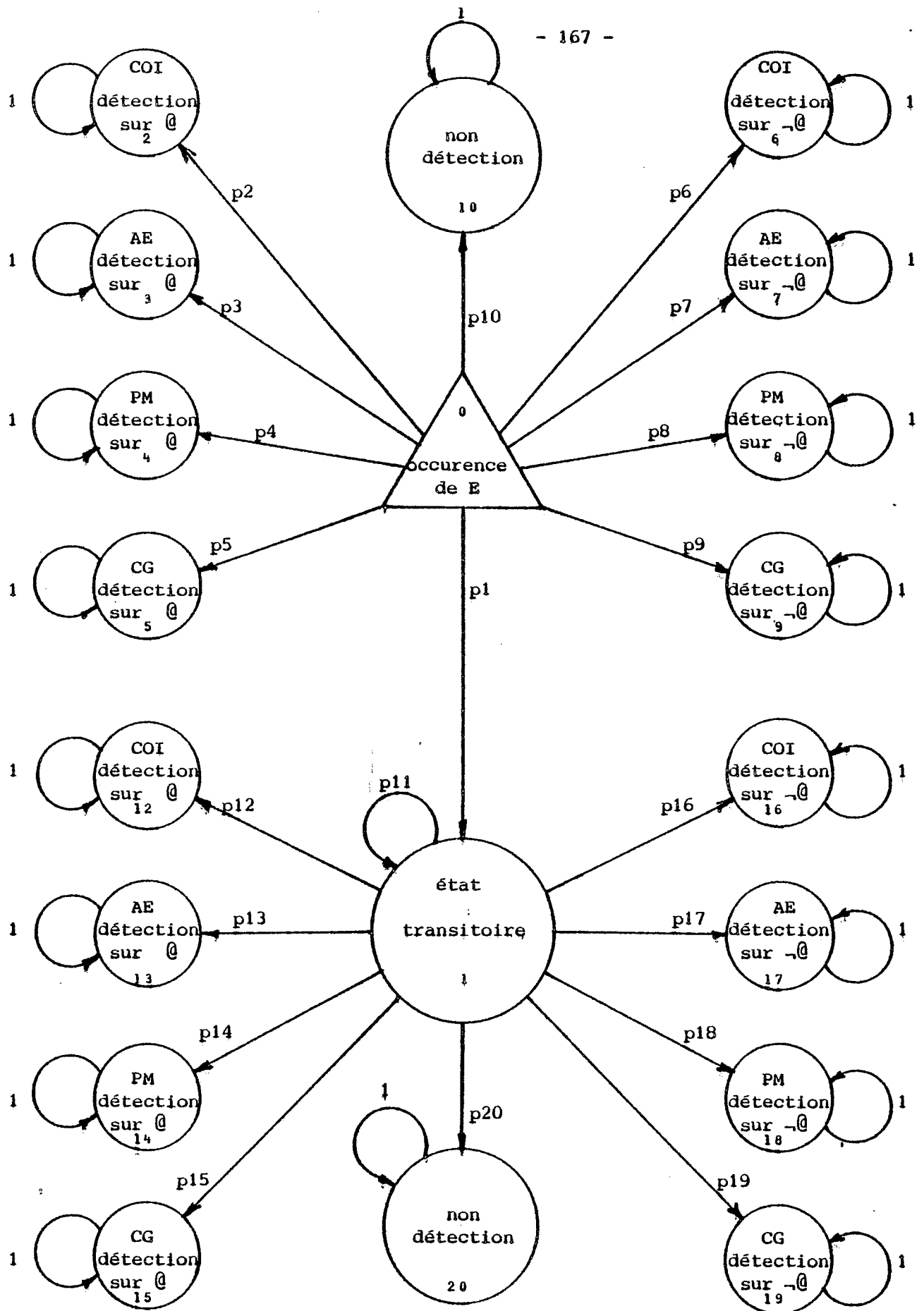


Figure 1 - Modèle théorique du comportement statique avec différenciation de la première instruction.



$$p1 + p2 + p3 + p4 + p5 + p6 + p7 + p9 + p10 = 1$$

$$p11 + p12 + p13 + p14 + p15 + p16 + p17 + p18 + p19 + p20 = 1$$

Figure 2 - Modèle statique adopté avec première instruction différenciée.

. Probabilités: d'Etat

On a, pour $i \geq 1$.

$$\begin{aligned} \text{Pr (1)}^{(i+1)} &= p_{11} \text{Pr (1)}^{(i)} \\ \text{Pr (12)}^{(i+1)} &= p_{12} \text{Pr (1)}^{(i)} + \text{Pr (12)}^{(i)} \\ \text{Pr (13)}^{(i+1)} &= p_{13} \text{Pr (1)}^{(i)} + \text{Pr (13)}^{(i)} \\ \text{Pr (14)}^{(i+1)} &= p_{14} \text{Pr (1)}^{(i)} + \text{Pr (14)}^{(i)} \\ \text{Pr (15)}^{(i+1)} &= p_{15} \text{Pr (1)}^{(i)} + \text{Pr (15)}^{(i)} \\ \text{Pr (16)}^{(i+1)} &= p_{16} \text{Pr (1)}^{(i)} + \text{Pr (16)}^{(i)} \\ \text{Pr (17)}^{(i+1)} &= p_{17} \text{Pr (1)}^{(i)} + \text{Pr (17)}^{(i)} \\ \text{Pr (18)}^{(i+1)} &= p_{18} \text{Pr (1)}^{(i)} + \text{Pr (18)}^{(i)} \\ \text{Pr (19)}^{(i+1)} &= p_{19} \text{Pr (1)}^{(i)} + \text{Pr (19)}^{(i)} \\ \text{Pr (20)}^{(i+1)} &= p_{20} \text{Pr (1)}^{(i)} + \text{Pr (20)}^{(i)} \end{aligned}$$

Avec les conditions initiales :

$$\text{Pr (1)}^1 = p_1 \text{ et}$$

$$\text{Pr (12)}^1 = \text{Pr (13)}^1 = \text{Pr (14)}^1 = \text{Pr (15)}^1 = \text{Pr (16)}^1 = \text{Pr (17)}^1 = \text{Pr (18)}^1 = \text{Pr (19)}^1 = \text{Pr (20)}^1 =$$

On obtient facilement :

$$\text{Pr (1)}^{(i)} = p_1 p_{11}^{i-1}$$

$$\text{Pr (12)}^{(i)} = p_{12} \sum_{k=1}^{i-1} \text{Pr (1)}^{(k)} \quad \text{soit}$$

$$\text{Pr (12)}^{(i)} = p_1 p_{12} \frac{1 - p_{11}^{i-1}}{1 - p_{11}}$$

$$\text{Pr (13)}^{(i)} = p_1 p_{13} \frac{1 - p_{11}^{i-1}}{1 - p_{11}}$$

$$\text{Pr (14)}^{(i)} = p_1 p_{14} \frac{1 - p_{11}^{i-1}}{1 - p_{11}}$$

$$\text{Pr (15)}^{(i)} = p_1 p_{15} \frac{1 - p_{11}^{i-1}}{1 - p_{11}}$$

$$\text{Pr (16)}^{(i)} = p_1 p_{16} \frac{1 - p_{11}}{1 - p_{11}}^{i-1}$$

$$\text{Pr (17)}^{(i)} = p_1 p_{17} \frac{1 - p_{11}}{1 - p_{11}}^{i-1}$$

$$\text{Pr (18)}^{(i)} = p_1 p_{18} \frac{1 - p_{11}}{1 - p_{11}}^{i-1}$$

$$\text{Pr (19)}^{(i)} = p_1 p_{19} \frac{1 - p_{11}}{1 - p_{11}}^{i-1}$$

$$\text{Pr (20)}^{(i)} = p_1 p_{20} \frac{1 - p_{11}}{1 - p_{11}}^{i-1}$$

Distributions, densités, probabilités limites :

$$\begin{aligned} \text{Pr} [K_{\text{COI}} \leq k, @] &= \text{Pr (2)} + \text{Pr (12)}^{k+1} \\ &= p_2 + p_1 p_{12} \frac{1 - p_{11}}{1 - p_{11}}^k \end{aligned}$$

$$\begin{aligned} \text{Pr} [K_{\text{COI}} \leq k, -@] &= \text{Pr (6)} + \text{Pr (16)}^{k+1} \\ &= p_6 + p_1 p_{16} \frac{1 - p_{11}}{1 - p_{11}}^k \end{aligned}$$

$$\begin{aligned} \text{Pr} [K_{\text{AE}} \leq k, @] &= \text{Pr (3)} + \text{Pr (13)}^{k+1} \\ &= p_3 + p_1 p_{13} \frac{1 - p_{11}}{1 - p_{11}}^k \end{aligned}$$

$$\begin{aligned} \text{Pr} [K_{\text{AE}} \leq k, -@] &= \text{Pr (7)} + \text{Pr (17)}^{k+1} \\ &= p_7 + p_1 p_{17} \frac{1 - p_{11}}{1 - p_{11}}^k \end{aligned}$$

$$\begin{aligned} \text{Pr} [K_{\text{PM}} \leq k, @] &= \text{Pr (4)} + \text{Pr (14)}^{k+1} \\ &= p_4 + p_1 p_{14} \frac{1 - p_{11}}{1 - p_{11}}^k \end{aligned}$$

$$\begin{aligned} \text{Pr} [K_{\text{PM}} \leq k, -@] &= \text{Pr (8)} + \text{Pr (18)}^{k+1} \\ &= p_8 + p_1 p_{18} \frac{1 - p_{11}}{1 - p_{11}}^k \end{aligned}$$

En considérant l'équation, pour $k > 0$:

$\Pr [K = k] = \Pr [K \leq k] - \Pr [K \leq k-1]$ on obtient ; pour $k > 0$ sur une borne d'adresse.

$$\Pr [K_{COI} = 0, @] = p_2$$

$$\Pr [K_{COI} = k, @] = p_1 p_{12} p_{11}^{k-1}$$

$$\Pr [K_{AE} = 0, @] = p_3$$

$$\Pr [K_{AE} = k, @] = p_1 p_{13} p_{11}^{k-1}$$

$$\Pr [K_{PM} = 0, @] = p_4$$

$$\Pr [K_{PM} = k, @] = p_1 p_{14} p_{11}^{k-1}$$

pour $k > 0$ sur une borne de "non-adresse" :

$$\Pr [K_{COI} = 0, \neg @] = p_6$$

$$\Pr [K_{COI} = k, \neg @] = p_1 p_{16} p_{11}^{k-1}$$

$$\Pr [K_{AE} = 0, \neg @] = p_7$$

$$\Pr [K_{AE} = k, \neg @] = p_1 p_{17} p_{11}^{k-1}$$

$$\Pr [K_{PM} = 0, \neg @] = p_8$$

$$\Pr [K_{PM} = k, \neg @] = p_1 p_{18} p_{11}^{k-1}$$

On peut alors obtenir les probabilités de détection, en supposant la période du chien de garde T_0 grande par rapport aux temps de détection.

Pour les bornes d'adresse :

$$\text{Pr [détection COI, @]} = p_2 + \frac{p_1 p_{12}}{1 - p_{11}}$$

$$\text{Pr [détection AE, @]} = p_3 + \frac{p_1 p_{13}}{1 - p_{11}}$$

$$\text{Pr [détection PM, @]} = p_4 + \frac{p_1 p_{14}}{1 - p_{11}}$$

$$\text{Pr [détection CG, @]} = p_5 + \frac{p_1 p_{15}}{1 - p_{11}}$$

Pour les bornes de "non adresse" :

$$\text{Pr [détection COI, -@]} = p_6 + \frac{p_1 p_{16}}{1 - p_{11}}$$

$$\text{Pr [détection AE, -@]} = p_7 + \frac{p_1 p_{17}}{1 - p_{11}}$$

$$\text{Pr [détection PM, -@]} = p_8 + \frac{p_1 p_{18}}{1 - p_{11}}$$

$$\text{Pr [détection CG, -@]} = p_9 + \frac{p_1 p_{19}}{1 - p_{11}}$$

D'ou l'on tire ; les probabilités globales :

$$\text{Pr [détection COI]} = p_2 + p_6 + \frac{p_1 (p_{12} + p_{16})}{1 - p_{11}}$$

$$\text{Pr [détection AE]} = p_3 + p_7 + \frac{p_1 (p_{13} + p_{17})}{1 - p_{11}}$$

$$\text{Pr [détection PM]} = p_4 + p_8 + \frac{p_1 (p_{14} + p_{18})}{1 - p_{11}}$$

$$\text{Pr [détection CG]} = p_5 + p_9 + \frac{p_1 (p_{15} + p_{19})}{1 - p_{11}}$$

$$\text{Pr [non détection]} = \text{Pr (10)} + \text{Pr (20)}$$

$$= p_{10} + \frac{p_1 p_{20}}{1 - p_{11}}$$

- Moments de détection :

Les moments de détection peuvent être calculés également à partir des densités :

$$\begin{aligned}
 E [K_{COI}, @] &= \sum_k \Pr [K_{COI} = k, @] \cdot \frac{1}{\Pr [\text{détection COI}, @]} \\
 &= \sum_k p_1 p_2 p_{11}^{k-1} \cdot \frac{1}{p_2 + \frac{p_1 p_2}{1-p_{11}}} \\
 &= \frac{p_1 p_2}{1-p_{11}} \cdot \frac{1}{p_2 - p_2 p_{11} + p_1 p_2}
 \end{aligned}$$

De manière identique :

$$E [K_{COI}, -@] = \frac{p_1 p_{16}}{1-p_{11}} \cdot \frac{1}{p_6 - p_6 p_{11} + p_1 p_{16}}$$

$$E [K_{AE}, @] = \frac{p_1 p_{13}}{1-p_{11}} \cdot \frac{1}{p_3 - p_3 p_{11} + p_1 p_{13}}$$

$$E [K_{AE}, -@] = \frac{p_1 p_{17}}{1-p_{11}} \cdot \frac{1}{p_7 - p_7 p_{11} + p_1 p_{17}}$$

$$E [K_{PM}, @] = \frac{p_1 p_{14}}{1-p_{11}} \cdot \frac{1}{p_4 - p_4 p_{11} + p_1 p_{14}}$$

$$E [K_{PM}, -@] = \frac{p_1 p_{18}}{1-p_{11}} \cdot \frac{1}{p_8 - p_8 p_{11} + p_1 p_{18}}$$

On obtient aussi :

$$E [K_{COI}, K_{COI} > 0, @] = E [K_{AE}, K_{AE} > 0, @] = E [K_{PM}, K_{PM} > 0, @] = a$$

$$E [K_{COI}, K_{COI} > 0, -@] = E [K_{AE}, K_{AE} > 0, -@] = E [K_{PM}, K_{PM} > 0, -@] = b$$

$$\text{et } E [K_{COI}, K_{COI} > 0] = E [K_{AE}, K_{AE} > 0] = E [K_{PM}, K_{PM} > 0] = c$$

$$\text{avec } a = b = c = \frac{1}{1 - p_{11}}$$

Enfin :

$$\begin{aligned} \sigma^2 [K_{COI}, @] &= \sum_k k^2 \Pr [K_{COI} = k, @] \cdot \frac{1}{\Pr [\text{détection COI}, @]} \cdot E [K_{COI}, @]^2 \\ &= \sum_k k^2 p_1 p_{12} p_{11}^{k-1} \cdot \frac{1}{p_2 + \frac{p_1 p_{12}}{1 - p_{11}}} - \left[\frac{p_1 p_{12}}{(1 - p_{11}) (p_2 - p_2 p_{11} + p_1 p_{12})} \right]^2 \\ &= p_1 p_{12} \sum_k k^2 p_{11}^{k-1} \cdot \frac{1 - p_{11}}{p_2 (1 - p_{11}) + p_1 p_{12}} - \frac{(p_1 p_{12})^2}{[p_2 (1 - p_{11})^2 + p_1 p_{12} (1 - p_{11})]^2} \\ &= \frac{p_1 p_{12} (1 + p_{11})}{(1 - p_{11})^3} \cdot \frac{1 - p_{11}}{p_2 (1 - p_{11}) + p_1 p_{12}} - \frac{(p_1 p_{12})^2}{[p_2 (1 - p_{11})^2 + p_1 p_{12} (1 - p_{11})]^2} \\ &= \frac{p_1 p_{12} (1 + p_{11})}{p_2 (1 - p_{11})^3 + p_1 p_{12} (1 - p_{11})^2} - \frac{(p_1 p_{12})^2}{[p_2 (1 - p_{11})^2 + p_1 p_{12} (1 - p_{11})]^2} \end{aligned}$$

$$\sigma^2 [K_{COI}, @] = \frac{p_1 p_{12} [p_2 (1-p_{11}^2) + p_1 p_{11} p_{12}]}{(1-p_{11})^2 [p_2 (1-p_{11}) + p_1 p_{12}]^2}$$

$$\sigma^2 [K_{COI}, -@] = \frac{p_1 p_{16} [p_6 (1-p_{11})^2 + p_1 p_{11} p_{16}]}{(1-p_{11})^2 [p_6 (1-p_{11}) + p_1 p_{16}]^2}$$

$$\sigma^2 [K_{AE}, @] = \frac{p_1 p_{13} [p_3 (1-p_{11}^2) + p_1 p_{11} p_{13}]}{(1-p_{11})^2 [p_3 (1-p_{11}) + p_1 p_{13}]^2}$$

$$\sigma^2 [K_{AE}, -@] = \frac{p_1 p_{17} [p_7 (1-p_{11}^2) + p_1 p_{11} p_{17}]}{(1-p_{11})^2 [p_7 (1-p_{11}) + p_1 p_{17}]^2}$$

$$\sigma^2 [K_{PM}, @] = \frac{p_1 p_{14} [p_4 (1-p_{11}^2) + p_1 p_{11} p_{14}]}{(1-p_{11})^2 [p_4 (1-p_{11}) + p_1 p_{14}]^2}$$

$$\sigma^2 [K_{PM}, -@] = \frac{p_1 p_{18} [p_8 (1-p_{11}^2) + p_1 p_{11} p_{18}]}{(1-p_{11})^2 [p_8 (1-p_{11}) + p_1 p_{18}]^2}$$

On obtient aussi :

$$\sigma^2 [K_{COI}, K_{COI} > 0, @] = \sigma^2 [K_{AE}, K_{AE} > 0, @] = \sigma^2 [K_{PM}, K_{PM} > 0, @] = a$$

$$\sigma^2 [K_{COI}, K_{COI} > 0, -@] = \sigma^2 [K_{AE}, K_{AE} > 0, -@] = \sigma^2 [K_{PM}, K_{PM} > 0, -@] = b$$

$$\text{et } \sigma^2 [K_{COI}, K_{COI} > 0] = \sigma^2 [K_{AE}, K_{AE} > 0] = \sigma^2 [K_{PM}, K_{PM} > 0] = c$$

$$\text{avec } a = b = c = \frac{p_{11}}{(1-p_{11})^2}$$

2.1.2. Accès Direct :

La matrice de transition P, organisée en fonction des états transitoires et absorbants, s'écrit (l'ordre des états sur les lignes et les colonnes étant : 2,3,4,5,6,7,8,9,10,12,13,14,15,16,17,18,19,20,0,1).

$$P = \begin{bmatrix} I_{18} & 0 \\ R & Q \end{bmatrix}$$

$$\text{où } Q = \begin{bmatrix} 0 & p_1 \\ 0 & p_{11} \end{bmatrix}$$

$$\text{et } R = \begin{bmatrix} p_2 & p_3 & p_4 & p_5 & p_6 & p_7 & p_8 & p_9 & p_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{12} & p_{13} & p_{14} & p_{15} & p_{16} & p_{17} & p_{18} & p_{19} & p_{20} \end{bmatrix}$$

La matrice fondamentale N s'écrit :

$$N = [I_2 - Q]^{-1}$$

$$\text{Soit ici : } N = \begin{bmatrix} 1 & -p_1 \\ 0 & 1-p_{11} \end{bmatrix}^{-1}$$

$$\text{Soit } N = \frac{1}{1-p_{11}} \begin{bmatrix} 1-p_{11} & p_1 \\ 0 & 1 \end{bmatrix}$$

Probabilités d'états, probabilités de détection :

La matrice B = [bij] est telle que bij est la probabilité que le processus soit absorbé en j en partant de l'état i. Elle est donnée par : B = NR

On obtient donc :

$$\text{Pr } [K_{\text{COI}} = 0, @] = \text{Pr } (2) = b_{1,1} = p_2$$

$$\text{Pr } [K_{\text{AE}} = 0, @] = \text{Pr } (3) = b_{1,2} = p_3$$

$$\text{Pr } [K_{\text{PM}} = 0, @] = \text{Pr } (4) = b_{1,3} = p_4$$

$$\text{Pr } [K_{\text{CG}} = 0, @] = \text{Pr } (5) = b_{1,4} = p_5$$

$$\text{Pr } [K_{\text{COI}} = 0, \neg @] = \text{Pr } (6) = b_{1,5} = p_6$$

$$\text{Pr } [K_{\text{AE}} = 0, \neg @] = \text{Pr } (7) = b_{1,6} = p_7$$

$$\text{Pr } [K_{\text{PM}} = 0, \neg @] = \text{Pr } (8) = b_{1,7} = p_8$$

$$\text{Pr } [K_{\text{CG}} = 0, \neg @] = \text{Pr } (9) = b_{1,8} = p_9$$

$$\text{Pr } [\text{d\u00e9tection en } 0] = \text{Pr } (10) = b_{1,9} = p_{10}$$

$$\text{Pr } [\text{d\u00e9tection COI, } K_{\text{COI}} > 0, @] = \text{Pr } (12) = b_{1,10} = \frac{p_1 p_{12}}{1-p_{11}}$$

$$\text{Pr } [\text{d\u00e9tection AE, } K_{\text{AE}} > 0, @] = \text{Pr } (13) = b_{1,11} = \frac{p_1 p_{13}}{1-p_{11}}$$

$$\text{Pr } [\text{d\u00e9tection PM, } K_{\text{PM}} > 0, @] = \text{Pr } (14) = b_{1,12} = \frac{p_1 p_{14}}{1-p_{11}}$$

$$\text{Pr } [\text{d\u00e9tection CG, } K_{\text{CG}} > 0, @] = \text{Pr } (15) = b_{1,13} = \frac{p_1 p_{15}}{1-p_{11}}$$

$$\text{Pr } [\text{d\u00e9tection COI, } K_{\text{COI}} > 0, \neg @] = \text{Pr } (16) = b_{1,14} = \frac{p_1 p_{16}}{1-p_{11}}$$

$$\text{Pr } [\text{d\u00e9tection AE, } K_{\text{AE}} > 0, \neg @] = \text{Pr } (17) = b_{1,15} = \frac{p_1 p_{17}}{1-p_{11}}$$

$$\text{Pr } [\text{d\u00e9tection PM, } K_{\text{PM}} > 0, \neg @] = \text{Pr } (18) = b_{1,16} = \frac{p_1 p_{18}}{1-p_{11}}$$

$$\text{Pr } [\text{d\u00e9tection CG, } K_{\text{CG}} > 0, \neg @] = \text{Pr } (19) = b_{1,17} = \frac{p_1 p_{19}}{1-p_{11}}$$

$$\text{Pr } [\text{non d\u00e9tection } > 0] = \text{Pr } (20) = b_{1,18} = \frac{p_1 p_{20}}{1-p_{11}}$$

D'o\u00f9 l'on tire les probabilit\u00e9s de d\u00e9tection :

$$\text{Pr } [\text{d\u00e9tection COI}] = p_2 + p_6 + (p_{12} + p_{16}) \frac{p_1}{1-p_{11}}$$

$$\text{Pr } [\text{d\u00e9tection AE}] = p_3 + p_7 + (p_{13} + p_{17}) \frac{p_1}{1-p_{11}}$$

$$\text{Pr [détection PM]} = p_4 + p_8 + (p_{14} + p_{18}) \frac{p_1}{1-p_{11}}$$

$$\text{Pr [détection CG]} = p_5 + p_9 + (p_{15} + p_{19}) \frac{p_1}{1-p_{11}}$$

$$\text{Pr [non détection]} = p_{10} + \frac{p_1 p_{20}}{1-p_{11}}$$

- Moments de détection :

τ et τ_2 représentent les moyennes et variances du temps passé avant absorption avec :

$$\tau = N \xi \quad \text{et} \quad \tau_2 = (2N - I_2) \tau - \tau \text{sq}$$

où ξ est le vecteur colonne à éléments égaux à 1.

et τsq est la matrice obtenue en élevant chaque élément de τ au carré.

On obtient donc :

$$\tau = \frac{1}{1-p_{11}} \begin{bmatrix} 1 - p_{11} + p_1 \\ 1 \end{bmatrix}$$

$$\tau_2 = \frac{1}{(1-p_{11})^2} \begin{bmatrix} p_1 (1-p_1 + p_{11}) \\ p_{11} \end{bmatrix}$$

D'où l'on tire ; en partant de l'état 0 :

$$E [\text{temps passé avant absorption}] = \frac{1 - p_{11} + p_1}{1-p_{11}}$$

$$= E [K_{\text{COI}}] + 1 = E [K_{\text{AE}}] + 1 = E [K_{\text{PM}}] + 1$$

$$\rightarrow E [K_{\text{COI}}] = E [K_{\text{AE}}] = E [K_{\text{PM}}] = \frac{p_1}{1-p_{11}}$$

et en partant de l'état 1 :

$$E [\text{temps passé avant absorption}] = \frac{1}{1-p_{11}}$$

$$= E [K_{\text{COI}}, K_{\text{COI}} > 0] = E [K_{\text{AE}}, K_{\text{AE}} > 0] =$$

$$E [K_{\text{PM}}, K_{\text{PM}} > 0]$$

$$\rightarrow E [K_{COI}, K_{COI} > 0] = E [K_{AE}, K_{AE} > 0] = E [K_{PM}, K_{PM} > 0] = \frac{1}{1-p11}$$

$$\sigma^2 [\text{temps passé avant absorption en partant de 0}] = \frac{p1 (1-p1 + p11)}{(1-p11)^2}$$

$$\sigma^2 [K_{COI}] = \sigma^2 [K_{AE}] = \sigma^2 [K_{PM}] = \frac{p1 (1-p1 + p11)}{(1-p11)^2}$$

$$\sigma^2 [\text{temps passé avant absorption en partant de 1}] = \frac{p11}{(1-p11)^2}$$

$$\sigma^2 [K_{COI}, K_{COI} > 0] = \sigma^2 [K_{AE}, K_{AE} > 0] = \sigma^2 [K_{PM}, K_{PM} > 0] = \frac{p11}{(1-p11)}$$

2.2. Second Modèle

Pour ce modèle l'exécution de la première instruction n'est pas différenciée des autres. On obtient alors le modèle de la figure 3. Pour les mêmes raisons que pour le premier modèle statique, le modèle adopté pour l'étude sera celui de la figure 4.

2.2.1. Accès matriciel :

Les expressions déjà calculées vont se trouver simplifiées avec ce modèle :

- Probabilités d'états :

$$\begin{aligned} \text{Pr (1)} (i + 1) &= p11 \text{ Pr (1)} (i) \\ \text{Pr (2)} (i + 1) &= p2 \text{ Pr (1)} (i) + \text{Pr (2)} (i) \\ \text{Pr (3)} (i + 1) &= p3 \text{ Pr (1)} (i) + \text{Pr (3)} (i) \\ \text{Pr (4)} (i + 1) &= p4 \text{ Pr (1)} (i) + \text{Pr (4)} (i) \\ \text{Pr (5)} (i + 1) &= p5 \text{ Pr (1)} (i) + \text{Pr (5)} (i) \\ \text{Pr (6)} (i + 1) &= p6 \text{ Pr (1)} (i) + \text{Pr (6)} (i) \\ \text{Pr (7)} (i + 1) &= p7 \text{ Pr (1)} (i) + \text{Pr (7)} (i) \\ \text{Pr (8)} (i + 1) &= p8 \text{ Pr (1)} (i) + \text{Pr (8)} (i) \\ \text{Pr (9)} (i + 1) &= p9 \text{ Pr (1)} (i) + \text{Pr (9)} (i) \end{aligned}$$

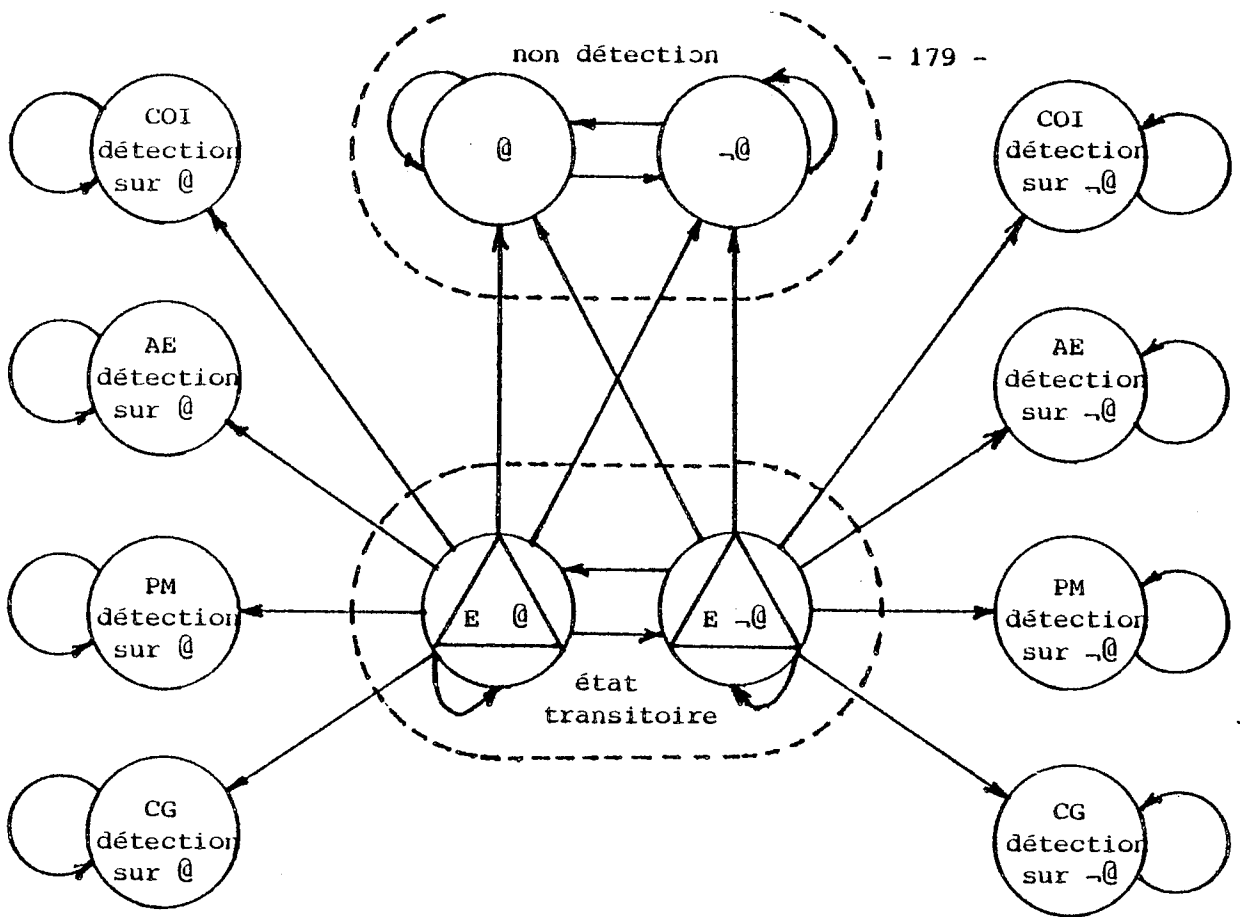
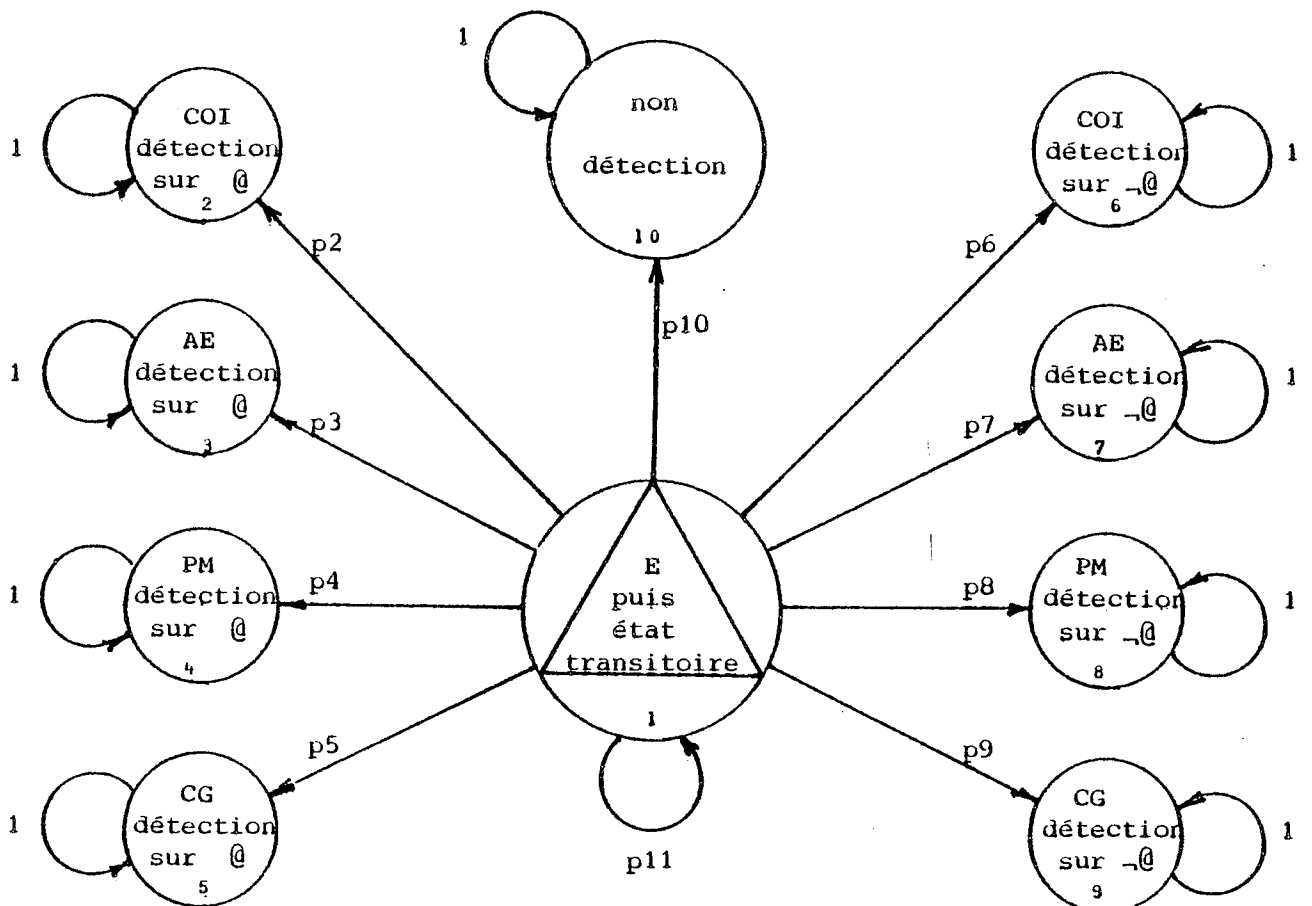


Figure 3 - Modèle théorique du comportement statique sans différenciation de la première instruction.



$$p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 + p10 + p11 = 1$$

$$\text{Pr } (10)^{(i+1)} = p_{10} \text{Pr } (1)^{(i)} + \text{Pr } (10)^{(i)}$$

avec les conditions initiales

$$\text{Pr } (1)^{(1)} = 1$$

$$\text{Pr } (k)^{(1)} = 0 \quad \forall k \neq 1$$

On obtient facilement les probabilités :

$$\text{Pr } (1)^{(i)} = p_{11}^i$$

$$\text{Pr } (2)^{(i)} = p_2 (1 + p_{11} + p_{11}^2 + \dots + p_{11}^{i-1}) = p_2 \frac{1-p_{11}^i}{1-p_{11}}$$

$$\text{Pr } (3)^{(i)} = p_3 \frac{1-p_{11}^i}{1-p_{11}}$$

$$\text{Pr } (4)^{(i)} = p_4 \frac{1-p_{11}^i}{1-p_{11}}$$

$$\text{Pr } (5)^{(i)} = p_5 \frac{1-p_{11}^i}{1-p_{11}}$$

$$\text{Pr } (6)^{(i)} = p_6 \frac{1-p_{11}^i}{1-p_{11}}$$

$$\text{Pr } (7)^{(i)} = p_7 \frac{1-p_{11}^i}{1-p_{11}}$$

$$\text{Pr } (8)^{(i)} = p_8 \frac{1-p_{11}^i}{1-p_{11}}$$

$$\text{Pr } (9)^{(i)} = p_9 \frac{1-p_{11}^i}{1-p_{11}}$$

$$\text{Pr } (10)^{(i)} = p_{10} \frac{1-p_{11}^i}{1-p_{11}}$$

- Distributions, densités, probabilités limites :

On obtient les distributions, à partir des probabilités d'états, pour $k \geq 0$

$$\begin{aligned} \text{Pr } [K_{\text{COI}} \leq k, @] &= \text{Pr } (2)^{(k+1)} \\ &= p_2 \frac{1-p_{11}^{k+1}}{1-p_{11}} \end{aligned}$$

$$\begin{aligned} \Pr [K_{COI} \leq k, \neg @] &= \Pr (6)^{(k+1)} \\ &= p_6 \frac{1-p_{11}}{1-p_{11}}^{k+1} \end{aligned}$$

$$\Pr [K_{AE} \leq k, @] = \Pr (3)^{(k+1)} = p_3 \frac{1-p_{11}}{1-p_{11}}^{k+1}$$

$$\Pr [K_{AE} \leq k, \neg @] = \Pr (7)^{(k+1)} = p_7 \frac{1-p_{11}}{1-p_{11}}^{k+1}$$

$$\Pr [K_{PM} \leq k, @] = \Pr (4)^{(k+1)} = p_4 \frac{1-p_{11}}{1-p_{11}}^{k+1}$$

$$\Pr [K_{PM} \leq k, \neg @] = \Pr (8)^{(k+1)} = p_8 \frac{1-p_{11}}{1-p_{11}}^{k+1}$$

On en tire les densités de détection par la formule,
pour $k > 0$: $\Pr [K = k] = \Pr [K \leq k] - \Pr [K \leq k-1]$

et donc ; pour les bornes d'adresse :

$$\Pr [K_{COI} = k, @] = p_2 p_{11}^k$$

$$\Pr [K_{AE} = k, @] = p_3 p_{11}^k$$

$$\Pr [K_{PM} = k, @] = p_4 p_{11}^k$$

pour les bornes de "non-adresse" :

$$\Pr [K_{COI} = k, \neg @] = p_6 p_{11}^k$$

$$\Pr [K_{AE} = k, \neg @] = p_7 p_{11}^k$$

$$\Pr [K_{PM} = k, \neg @] = p_8 p_{11}^k$$

Par passage à la limite, on en tire les probabilités de détection :

Pour les bornes d'adresse :

$$\Pr [\text{détection COI}, @] = \frac{p_2}{1-p_{11}}$$

$$\Pr [\text{détection AE}, @] = \frac{p_3}{1-p_{11}}$$

$$\text{Pr}[\text{détectio} \text{ PM}, @] = \frac{p_4}{1-p_{11}}$$

$$\text{Pr}[\text{détectio} \text{ CG}, @] = \frac{p_5}{1-p_{11}}$$

Pour les bornes de "non adresse" :

$$\text{Pr}[\text{détectio} \text{ COI}, -@] = \frac{p_6}{1-p_{11}}$$

$$\text{Pr}[\text{détectio} \text{ AE}, -@] = \frac{p_7}{1-p_{11}}$$

$$\text{Pr}[\text{détectio} \text{ PM}, -@] = \frac{p_8}{1-p_{11}}$$

$$\text{Pr}[\text{détectio} \text{ CG}, -@] = \frac{p_9}{1-p_{11}}$$

et pour la non-détection :

$$\text{Pr}[\text{non détectio}] = \frac{p_{10}}{1-p_{11}}$$

- Moments de détection :

Le premier moment s'obtient à partir des densités :

$$\begin{aligned} E[K_{\text{COI}}, @] &= \sum_k k \text{Pr}[K_{\text{COI}} = k, @] \frac{1}{\text{Pr}[\text{détectio} \text{ COI}, @]} \\ &= p_2 \sum_k k p_{11}^k \cdot \frac{1}{\frac{p_2}{1-p_{11}}} \\ &= \frac{p_{11}}{1-p_{11}} \end{aligned}$$

$$\begin{aligned} E[K_{\text{COI}}, -@] &= \sum_k k \text{Pr}[K_{\text{COI}} = k, -@] \frac{1}{\text{Pr}[\text{détectio} \text{ COI}, -@]} \\ &= p_6 \sum_k k p_{11}^k \cdot \frac{1}{\frac{p_6}{1-p_{11}}} \\ &= \frac{p_{11}}{1-p_{11}} \end{aligned}$$

Il en est de même pour K_{AE} et K_{PM} , on peut donc écrire :

$$E [K_{AE, @}] = \frac{p11}{1-p11}$$

$$E [K_{AE, @}] = \frac{p11}{1-p11}$$

$$E [K_{PM, @}] = \frac{p11}{1-p11}$$

$$E [K_{PM, @}] = \frac{p11}{1-p11}$$

De même :

$$E [K_{COI}] = E [K_{AE}] = E [K_{PM}] = \frac{p11}{1-p11}$$

Il vient ensuite :

$$\begin{aligned} \sigma^2 [K_{COI, @}] &= \sum_k k^2 \Pr [K_{COI} = k, @] \frac{1}{\Pr [\text{détection COI, @}]} - E [K_{COI, @}]^2 \\ &= p2 \sum_k k^2 p11^k \frac{1}{\frac{p2}{1-p11}} - \left(\frac{p11}{1-p11} \right)^2 \\ &= \frac{p11 + p11^2}{(1-p11)^2} - \frac{p11^2}{(1-p11)^2} \\ &= \frac{p11}{(1-p11)^2} \end{aligned}$$

Il en est de même pour chaque détecteur on peut donc écrire :

$$\sigma^2 [K_{COI, @}] = \sigma^2 [K_{AE, @}] = \sigma^2 [K_{PM, @}] = \frac{p11}{(1-p11)^2}$$

$$\text{et } \sigma^2 [K_{COI, -@}] = \sigma^2 [K_{AE, -@}] = \sigma^2 [K_{PM, -@}] = \frac{p11}{(1-p11)^2}$$

De même,

$$\sigma^2 [K_{COI}] = \sigma^2 [K_{AE}] = \sigma^2 [K_{PM}] = \frac{p11}{(1-p11)^2}$$

2.2.2. Accès direct :

La matrice de transition P s'écrit dans cet exemple (selon l'ordre des états 2, 3, 4, 5, 6, 7, 8, 9, 10, 1)

$$P = \begin{bmatrix} I_9 & 0 \\ R & Q \end{bmatrix}$$

où $Q = [p_{11}]$

et $R = [p_2 \ p_3 \ p_4 \ p_5 \ p_6 \ p_7 \ p_8 \ p_9 \ p_{10}]$

La matrice fondamentale N s'écrit :

$$N = [I_1 - Q]^{-1}$$

Soit ici : $N = \begin{bmatrix} 1 \\ 1-p_{11} \end{bmatrix}$

- Probabilité d'état, probabilité de détection :

La matrice B donne les probabilités d'états soit :

$B = NR$ on obtient :

Pour les bornes d'adresse :

$$\text{Pr [détection COI, @]} = \text{Pr (2)} = b_{1, 1} = \frac{p_2}{1-p_{11}}$$

$$\text{Pr [détection AE, @]} = \text{Pr (3)} = b_{1, 2} = \frac{p_3}{1-p_{11}}$$

$$\text{Pr [détection PM, @]} = \text{Pr (4)} = b_{1, 3} = \frac{p_4}{1-p_{11}}$$

$$\text{Pr [détection CG, @]} = \text{Pr (5)} = b_{1, 4} = \frac{p_5}{1-p_{11}}$$

Pour les bornes de "non adresse" :

$$\text{Pr [détection, -@]} = \text{Pr (6)} = b_{1, 5} = \frac{p_6}{1-p_{11}}$$

$$\text{Pr [détection AE, } -\theta] = \text{Pr (7)} = b_1, 6 = \frac{p_7}{1-p_{11}}$$

$$\text{Pr [détection PM, } -\theta] = \text{Pr (8)} = b_1, 7 = \frac{p_8}{1-p_{11}}$$

$$\text{Pr [détection CG, } -\theta] = \text{Pr (9)} = b_1, 8 = \frac{p_9}{1-p_{11}}$$

Pour la non détection :

$$\text{Pr [non détection]} = \text{Pr (10)} = b_1, 9 = \frac{p_{10}}{1-p_{11}}$$

- Moments de détection

τ et τ_2 représentent les moyenne et variance du temps passé avant absorption avec :

$$\tau = N \xi \quad \text{et} \quad \tau_2 = (2N - I_1')\tau - \tau sq$$

où ξ et τsq ont les mêmes définitions que précédemment :

$$\tau = \frac{1}{1-p_{11}}$$

$$\tau_2 = \frac{p_{11}}{(1-p_{11})}$$

D'où l'on tire :

$$\begin{aligned} E [\text{temps avant absorption}] &= E [K_{COI}] + 1 \\ &= E [K_{AE}] + 1 \\ &= E [K_{PM}] + 1 \\ &= \frac{1}{1-p_{11}} \end{aligned}$$

$$E [K_{COI}] = \frac{p_{11}}{1-p_{11}}$$

$$E [K_{AE}] = \frac{p_{11}}{1-p_{11}}$$

$$E [K_{PM}] = \frac{p_{11}}{1-p_{11}}$$

$$\text{et } \sigma^2 [K_{\text{COI}}] = \frac{p11}{(1-p11)^2}$$

$$\sigma^2 [K_{\text{AE}}] = \frac{p11}{(1-p11)^2}$$

$$\sigma^2 [K_{\text{PM}}] = \frac{p11}{(1-p11)^2}$$

2.3. Récapitulation

Dans le tableau I et le tableau II sont regroupées les valeurs caractéristiques des deux modèles en séparant les détections selon qu'elles sont faites sur les bornes d'adresses et sur les bornes de "non adresses". Dans le tableau III, on trouvera les valeurs globales pour les deux modèles.

Les mêmes remarques que pour les 6800 peuvent être faites. En effet, les caractéristiques concernant la détection par COI et par AE sont absolument générales (ne dépendant pas du programme spécifique exécuté par le CPU). Les caractéristiques concernant la détection par PM, sont, elles, dépendantes de l'application i.e. des adresses d'implantations RAM, ROM spécifiques au programme.

Les paramètres de détection par CG sont également dépendants de l'application ; Un cas idéalisé, comme dans l'application considérée ici, est celui où l'espace mémoire occupé par le programme est parcouru séquentiellement, avec retour au point de départ et relance du chien de garde. Si l'événement E est tel que le compteur ordinal est uniformément réparti, après E, sur l'espace mémoire occupé, alors la probabilité de détection par CG et la probabilité de non détection doivent être égales. Mais ce cas est très irréaliste, car un programme d'application est tel que le temps réel séparant l'instant de relance (sans événement E) ne sera pas constant, en raison de la difficulté voire de l'impossibilité, de déterminer de tels points de relance dans l'algorithme.

Le modèle de la figure 2 reflète mieux le comportement du système que celui de la figure 4. Nous verrons lors de l'expérimentation que les valeurs des paramètres p_2 , p_{12} , p_6 et p_{16} de la figure 2 sont très différentes p_2 , p_6 (resp p_3 p_7) de la figure 4 étant des valeurs intermédiaires.

On peut remarquer que le modèle de la figure 2 et celui de la figure 4 sont équivalents pour $p_{11} = p_1$, $p_{12} = p_2$, $p_{13} = p_3$, $p_{14} = p_4$, $p_{15} = p_5$, $p_{16} = p_6$, $p_{17} = p_7$, $p_{18} = p_8$, $p_{19} = p_9$, $p_{20} = p_{10}$.

$\Pr [K_{COI} = 0, @] = p_2$ $\Pr [K_{COI} = k, @] = p_1 p_{12} p_{11}^{k-1}$ $\Pr [K_{COI} \leq k, @] = p_2 + p_1 p_{12} \frac{1-p_{11}^k}{1-p_{11}}$ $\Pr [\text{Détection COI}, @] = p_2 + \frac{p_1 p_{12}}{1-p_{11}}$ $E [K_{COI}, @] = \frac{p_1 p_{12}}{(1-p_{11})[p_2 (1-p_{11}) + p_1 p_{12}]}$ $\sigma^2 [K_{COI}, @] = \frac{p_1 p_{12} [p_2 (1-p_{11}^2) + p_1 p_{11} p_{12}]}{(1-p_{11})^2 [p_2 (1-p_{11}) + p_1 p_{12}]^2}$	$\Pr [K_{COI} = 0, -@] = p_6$ $\Pr [K_{COI} = k, -@] = p_1 p_{16} p_{11}^{k-1}$ $\Pr [K_{COI} \leq k, -@] = p_6 + p_1 p_{16} \frac{1-p_{11}^k}{1-p_{11}}$ $\Pr [\text{détection COI}, -@] = p_6 + \frac{p_1 p_{16}}{1-p_{11}}$ $E [K_{COI}, -@] = \frac{p_1 p_{16}}{(1-p_{11}) [p_6 (1-p_{11}) + p_1 p_{16}]}$ $\sigma^2 [K_{COI}, -@] = \frac{p_1 p_{16} [p_6 (1-p_{11}^2) + p_1 p_{11} p_{16}]}{(1-p_{11})^2 [p_6 (1-p_{11}) + p_1 p_{16}]^2}$
$\Pr [K_{AE} = 0, @] = p_3$ $\Pr [K_{AE} = k, @] = p_1 p_{13} p_{11}^{k-1}$ $\Pr [K_{AE} \leq k, @] = p_3 + p_1 p_{13} \frac{1-p_{11}^k}{1-p_{11}}$ $\Pr [\text{détection AE}, @] = p_3 + \frac{p_1 p_{13}}{1-p_{11}}$ $E [K_{AE}, @] = \frac{p_1 p_{13}}{(1-p_{11}) [p_3 (1-p_{11}) + p_1 p_{13}]}$ $\sigma^2 [K_{AE}, @] = \frac{p_1 p_{13} [p_3 (1-p_{11}^2) + p_1 p_{11} p_{13}]}{(1-p_{11})^2 [p_3 (1-p_{11}) + p_1 p_{13}]^2}$	$\Pr [K_{AE} = 0, -@] = p_7$ $\Pr [K_{AE} = k, -@] = p_1 p_{17} p_{11}^{k-1}$ $\Pr [K_{AE} \leq k, -@] = p_7 + p_1 p_{17} \frac{1-p_{11}^k}{1-p_{11}}$ $\Pr [\text{détection AE}, -@] = p_7 + \frac{p_1 p_{17}}{1-p_{11}}$ $E [K_{AE}, -@] = \frac{p_1 p_{17}}{(1-p_{11}) [p_7 (1-p_{11}) + p_1 p_{17}]}$ $\sigma^2 [K_{AE}, -@] = \frac{p_1 p_{17} [p_7 (1-p_{11}^2) + p_1 p_{11} p_{17}]}{(1-p_{11})^2 [p_7 (1-p_{11}) + p_1 p_{17}]^2}$
$\Pr [K_{PM} = 0, @] = p_4$ $\Pr [K_{PM} = k, @] = p_1 p_{14} p_{11}^{k-1}$ $\Pr [K_{PM} < k, @] = p_4 + p_1 p_{14} \frac{1-p_{11}^k}{1-p_{11}}$ $\Pr [\text{détection PM}, @] = p_4 + \frac{p_1 p_{14}}{1-p_{11}}$ $E [K_{PM}, @] = \frac{p_1 p_{14}}{(1-p_{11}) [p_4 (1-p_{11}) + p_1 p_{14}]}$ $\sigma^2 [K_{PM}, @] = \frac{p_1 p_{14} [p_4 (1-p_{11}^2) + p_1 p_{11} p_{14}]}{(1-p_{11})^2 [p_4 (1-p_{11}) + p_1 p_{14}]^2}$	$\Pr [K_{PM} = 0, -@] = p_8$ $\Pr [K_{PM} = k, -@] = p_1 p_{18} p_{11}^{k-1}$ $\Pr [K_{PM} < k, -@] = p_8 + p_1 p_{18} \frac{1-p_{11}^k}{1-p_{11}}$ $\Pr [\text{détection PM}, -@] = p_8 + \frac{p_1 p_{18}}{1-p_{11}}$ $E [K_{PM}, -@] = \frac{p_1 p_{18}}{(1-p_{11}) [p_8 (1-p_{11}) + p_1 p_{18}]}$ $\sigma^2 [K_{PM}, -@] = \frac{p_1 p_{18} [p_8 (1-p_{11}^2) + p_1 p_{11} p_{18}]}{(1-p_{11})^2 [p_8 (1-p_{11}) + p_1 p_{18}]^2}$
$\Pr [K_{CG} = 0, @] = p_5$ $\Pr [K_{CG} = k, @] = p_1 p_{15} p_{11}^{k-1}$ $\Pr [K_{CG} \leq k, @] = p_5 + p_1 p_{15} \frac{1-p_{11}^k}{1-p_{11}}$ $\Pr [\text{Détection CG}, @] = p_5 + \frac{p_1 p_{15}}{1-p_{11}}$	$\Pr [K_{PM} = 0, -@] = p_9$ $\Pr [K_{PM} = k, -@] = p_1 p_{19} p_{11}^{k-1}$ $\Pr [K_{PM} \leq k, -@] = p_9 + p_1 p_{19} \frac{1-p_{11}^k}{1-p_{11}}$ $\Pr [\text{détection CG}, -@] = p_9 + \frac{p_1 p_{19}}{1-p_{11}}$
<p>A</p>	<p>B</p>

Tableau I - Valeurs caractéristiques sur les bornes d'adresse (A) et sur les bornes de "non adresse" (B) pour le modèle de la figure 2 .

$\Pr [K_{COI} = 0, @] = p2$ $\Pr [K_{COI} = k, @] = p2 p11^k$ $\Pr [K_{COI} \leq k, @] = p2 \frac{1-p11^{k+1}}{1-p11}$ $\Pr [\text{détection COI}, @] = \frac{p2}{1-p11}$ $E [K_{COI}, @] = \frac{p11}{1-p11}$ $\sigma^2 [K_{COI}, @] = \frac{p11}{(1-p11)^2}$	$\Pr [K_{COI} = 0, -@] = p6$ $\Pr [K_{COI} = k, -@] = p6 p11^k$ $\Pr [K_{COI} \leq k, -@] = p6 \frac{1-p11^{k+1}}{1-p11}$ $\Pr [\text{détection COI}, -@] = \frac{p6}{1-p11}$ $E [K_{COI}, -@] = \frac{p11}{1-p11}$ $\sigma^2 [K_{COI}, -@] = \frac{p11}{(1-p11)^2}$
$\Pr [K_{AE} = 0, @] = p3$ $\Pr [K_{AE} = k, @] = p3 p11^k$ $\Pr [K_{AE} \leq k, @] = p3 \frac{1-p11^{k+1}}{1-p11}$ $\Pr [\text{détection AE}, @] = \frac{p3}{1-p11}$ $E [K_{AE}, @] = \frac{p11}{1-p11}$ $\sigma^2 [K_{AE}, @] = \frac{p11}{(1-p11)^2}$	$\Pr [K_{AE} = 0, -@] = p7$ $\Pr [K_{AE} = k, -@] = p7 p11^k$ $\Pr [K_{AE} \leq k, -@] = p7 \frac{1-p11^{k+1}}{1-p11}$ $\Pr [\text{détection AE}, -@] = \frac{p7}{1-p11}$ $E [K_{AE}, -@] = \frac{p11}{1-p11}$ $\sigma^2 [K_{AE}, -@] = \frac{p11}{(1-p11)^2}$
$\Pr [K_{PM} = 0, @] = p4$ $\Pr [K_{PM} = k, @] = p4 p11^k$ $\Pr [K_{PM} \leq k, @] = p4 \frac{1-p11^{k+1}}{1-p11}$ $\Pr [\text{détection PM}, @] = \frac{p4}{1-p11}$ $E [K_{PM}, @] = \frac{p11}{1-p11}$ $\sigma^2 [K_{PM}, @] = \frac{p11}{(1-p11)^2}$	$\Pr [K_{PM} = 0, -@] = p8$ $\Pr [K_{PM} = k, -@] = p8 p11^k$ $\Pr [K_{PM} \leq k, -@] = p8 \frac{1-p11^{k+1}}{1-p11}$ $\Pr [\text{détection PM}, -@] = \frac{p8}{1-p11}$ $E [K_{PM}, -@] = \frac{p11}{1-p11}$ $\sigma^2 [K_{PM}, -@] = \frac{p11}{(1-p11)^2}$
$\Pr [K_{CG} = 0, @] = p5$ $\Pr [K_{CG} = k, @] = p5 p11^k$ $\Pr [K_{CG} \leq k, @] = p5 \frac{1-p11^{k+1}}{1-p11}$ $\Pr [\text{Détection CG}, @] = \frac{p5}{1-p11}$	$\Pr [K_{CG} = 0, -@] = p9$ $\Pr [K_{CG} = k, -@] = p9 p11^k$ $\Pr [K_{CG} \leq k, -@] = p9 \frac{1-p11^{k+1}}{1-p11}$ $\Pr [\text{détection CG}, -@] = \frac{p9}{1-p11}$
<p style="text-align: center;">A</p>	<p style="text-align: center;">B</p>

Tableau II - Valeurs caractéristiques sur les bornes d'adresse (A) et sur les bornes de "non adresse" (B) pour le modèle de la figure 4 .

$\Pr [K_{COI} = 0] = p_2 + p_6$ $\Pr [K_{COI} = k] = p_1 (p_{12} + p_{16}) p_{11}^{k-1}$ $\Pr [K_{COI} \leq k] = p_2 + p_6 + p_1 (p_{12} + p_{16}) \frac{1 - p_{11}^k}{1 - p_{11}}$ $\Pr [\text{détection COI}] = p_2 + p_6 + \frac{p_1 (p_{12} + p_{16})}{1 - p_{11}}$ $E [K_{COI}] = \frac{p_1}{(1 - p_{11})}$ $\sigma^2 [K_{COI}] = \frac{p_{11}}{(1 - p_{11})^2}$	$\Pr [K_{COI} = 0] = p_2 + p_6$ $\Pr [K_{COI} = k] = (p_2 + p_6) p_{11}^k$ $\Pr [K_{COI} \leq k] = (p_2 + p_6) \frac{1 - p_{11}^{k+1}}{1 - p_{11}}$ $\Pr [\text{détection COI}] = \frac{p_2 + p_6}{1 - p_{11}}$ $E [K_{COI}] = \frac{p_{11}}{(1 - p_{11})}$ $\sigma^2 [K_{COI}] = \frac{p_{11}}{(1 - p_{11})^2}$
$\Pr [K_{AE} = 0] = p_3 + p_7$ $\Pr [K_{AE} = k] = p_1 (p_{13} + p_{17}) p_{11}^{k-1}$ $\Pr [K_{AE} \leq k] = p_3 + p_7 + p_1 (p_{13} + p_{17}) \frac{1 - p_{11}^k}{1 - p_{11}}$ $\Pr [\text{détection AE}] = p_3 + p_7 + \frac{p_1 (p_{13} + p_{17})}{1 - p_{11}}$ $E [K_{AE}] = \frac{p_1}{1 - p_{11}}$ $\sigma^2 [K_{AE}] = \frac{p_{11}}{(1 - p_{11})^2}$	$\Pr [K_{AE} = 0] = p_3 + p_7$ $\Pr [K_{AE} = k] = (p_3 + p_7) p_{11}^k$ $\Pr [K_{AE} \leq k] = (p_3 + p_7) \frac{1 - p_{11}^{k+1}}{1 - p_{11}}$ $\Pr [\text{détection AE}] = \frac{p_3 + p_7}{1 - p_{11}}$ $E [K_{AE}] = \frac{p_{11}}{1 - p_{11}}$ $\sigma^2 [K_{AE}] = \frac{p_{11}}{(1 - p_{11})^2}$
$\Pr [K_{PM} = 0] = p_4 + p_8$ $\Pr [K_{PM} = k] = p_1 (p_{14} + p_{18}) p_{11}^{k-1}$ $\Pr [K_{PM} \leq k] = p_4 + p_8 + p_1 (p_{14} + p_{18}) \frac{1 - p_{11}^k}{1 - p_{11}}$ $\Pr [\text{Détection PM}] = p_4 + p_8 + \frac{p_1 (p_{14} + p_{18})}{1 - p_{11}}$ $E [K_{PM}] = \frac{p_1}{1 - p_{11}}$ $\sigma^2 [K_{PM}] = \frac{p_{11}}{(1 - p_{11})^2}$	$\Pr [K_{PM} = 0] = p_4 + p_8$ $\Pr [K_{PM} = k] = (p_4 + p_8) p_{11}^k$ $\Pr [K_{PM} \leq k] = (p_4 + p_8) \frac{1 - p_{11}^{k+1}}{1 - p_{11}}$ $\Pr [\text{détection PM}] = \frac{p_4 + p_8}{1 - p_{11}}$ $E [K_{PM}] = \frac{p_{11}}{1 - p_{11}}$ $\sigma^2 [K_{PM}] = \frac{p_{11}}{(1 - p_{11})^2}$
$\Pr [K_{CG} = 0] = p_5 + p_9$ $\Pr [K_{CG} = k] = p_1 (p_{15} + p_{19}) p_{11}^{k-1}$ $\Pr [K_{CG} \leq k] = p_5 + p_9 + p_1 (p_{15} + p_{19}) \frac{1 - p_{11}^k}{1 - p_{11}}$ $\Pr [\text{détection CG}] = p_5 + p_9 + \frac{p_1 (p_{15} + p_{19})}{1 - p_{11}}$ $\Pr [\text{non détection}] = p_{10} + \frac{p_1 p_{20}}{1 - p_{11}}$	$\Pr [K_{CG} = 0] = p_5 + p_9$ $\Pr [K_{CG} = k] = (p_5 + p_9) p_{11}^k$ $\Pr [K_{CG} \leq k] = (p_5 + p_9) \frac{1 - p_{11}^{k+1}}{1 - p_{11}}$ $\Pr [\text{détection CG}] = \frac{p_5 + p_{11}}{1 - p_{11}}$ $\Pr [\text{non détection}] = \frac{p_{20}}{1 - p_{11}}$
<p>(A)</p>	<p>(B)</p>

Tableau III - Valeurs caractéristiques globales (adresse + non adresse) pour les modèles de la figure 2 (A) et de la figure 4 (B).

III. RESULTATS EXPERIMENTAUX

Les pannes simulées ici sont la modification, à un instant aléatoire uniformément réparti sur une période T_0 , de l'adresse de la prochaine instruction. Cette modification s'effectue par l'altération d'un ou plusieurs bits de cette adresse. Seules les modifications non triviales sont considérées : les altérations modifiant l'adresse de façon à diriger le compteur ordinal en dehors de zones mémoire contenant des programmes ne sont pas prises en compte. Chaque modification est répétée 30 fois.

Les résultats bruts de ces expérimentations sont rassemblés dans le tableau IV. Le numéro de bits signifie que le ou les bits considérés ont été complémentés à l'instant de l'altération. Environ 170 000 instructions 68000 ont été simulées pour ces expérimentations.

Le tableau IV appelle les mêmes commentaires que les pannes de la périphérie, ainsi :

- On ne considère que le premier détecteur, et donc un détecteur, peu efficace peut le devenir si d'autres sont supprimés, mais bien sûr avec des temps de détection plus longs.

Il n'y a pas de distorsion selon l'événement E considéré : un ou plusieurs bits affectés, poids des bits affectés... mais ceci est correct car les événements E à conséquences triviales ont été éliminés.

Le temps de détection, quand il y a détection par CG, se rapproche des valeurs théoriques $E = 170$ et $\sigma = 98$ lorsque l'échantillon est suffisamment important. Et si l'on considère l'échantillon global de détection par CG on obtient les valeurs $E = 176$ et $\sigma = 93$, ce qui dénote une assez bonne répartition de l'événement E sur l'intervalle de $T_0 = 340$ instructions.

N° bit	D E T E C T I O N				Non
	COI E [K _{COI}] # σ [K _{COI}]	AE E [K _{AE}] # σ [K _{AE}]	PM E [K _{PM}] # σ [K _{PM}]	CG E [K _{CG}] # σ [K _{CG}]	Détection #
0		E = 0 30 σ = 0			
1	E = 0 5 σ = 0		E = 9 3 σ = 0	E = 133 4 σ = 72	18
2	E = 0 3 σ = 1	E = 9 2 σ = 0	(E = 10) 1	E = 187 8 σ = 68	16
3	(E = 0) 1	E = 2 2 σ = 1	E = 19 2 σ = 21	E = 191 12 σ = 106	13
4	E = 1 6 σ = 1	(E = 2) 1		E = 258 4 σ = 62	12
5	E = 0 5 σ = 0		E = 4 2 σ = 1	E = 165 8 σ = 140	15
6	E = 0 6 σ = 0	E = 86 3 σ = 19	E = 4 5 σ = 3	E = 126 9 σ = 102	7
7	E = 0 9 σ = 0			E = 167 14 σ = 86	7
8	E = 0 3 σ = 0		(E = 6) 1	E = 179 24 σ = 91	2
1,2	E = 0 5 σ = 0	E = 7 2 σ = 5	E = 15 2 σ = 6	E = 202 10 σ = 60	11
1,3	E = 10 9 σ = 29	E = 1 3 σ = 1	(E = 9) 1	E = 145 7 σ = 118	10
1,4	E = 1 6 σ = 1		E = 45 2 σ = 61	E = 184 10 σ = 66	12
1,5	E = 0 5 σ = 0	(E = 3) 1	E = 8 6 σ = 6	E = 170 7 σ = 131	10
1,6	E = 0 6 σ = 0	E = 3 2 σ = 1	E = 6 6 σ = 4	E = 170 13 σ = 108	3
1,7	E = 0 9 σ = 0			E = 155 13 σ = 86	8

Tableau IV - Caractéristiques statiques expérimentales après occurrence d'un événement E (To = 340 instructions). (début)

N° bit	DETECTION				Non Détection
	COI E [K _{COI}] # σ [K _{COI}]	AE E [K _{AE}] # σ [K _{AE}]	PM E [K _{PM}] # σ [K _{PM}]	CG E [K _{CG}] # σ [K _{CG}]	#
2,3	E = 0 8 σ = 0	E = 4 2 σ = 0	(E = 4) 1	E = 192 8 σ = 86	11
2,4	E = 0 4 σ = 0		E = 12 2 σ = 14	E = 180 13 σ = 86	11
2,5	E = 0 6 σ = 0		E = 4 3 σ = 2	E = 124 10 σ = 111	11
3,4	E = 0 7 σ = 0		E = 8 3 σ = 5	E = 197 11 σ = 89	9

Tableau IV - Caractéristiques statistiques expérimentales après occurrence d'un événement E (T₀ = 340 instructions). (fin)

Les tableaux V et VI donnent les valeurs des caractéristiques statiques générales mais différenciées sur les détections immédiates ou non immédiates et sur le fait que la détection s'est faite sur une adresse ou une "non adresse".

Le tableau V donne les moments ainsi que le nombre de détections, le tableau VI donne les probabilités de détection :

A D R E S S E S				N O N A D R E S S E S			
Détection par	K = 0	K > 0	Total	K = 0	K > 0	Total	
	#	# E σ	# E σ		#	# E σ	# E σ
COI	-	-	-	92	E = 8 12 σ = 25	E = 1 104 σ = 9	
AE	-	E = 24 12 σ = 39	E = 24 12 σ = 39	30	E = 5 6 σ = 4	E = 1 36 σ = 3	
PM	-	E = 15 37 σ = 24	E = 15 37 σ = 24	-	E = 6 10 σ = 3	E = 6 10 σ = 3	
CG	-	E = 187 159 σ = 90	E = 187 159 σ = 90	-	E = 109 26 σ = 79	E = 109 26 σ = 79	
Non Détection	186						

Tableau V - Caractéristiques de détection immédiates et non immédiates différenciées sur les bornes de détection (adresse ou non adresse).

	A D R E S S E	N O N A D R E S S E
Pr [détection COI]	0	0. 182
Pr [détection AE]	0.021	0. 063
Pr [détection PM]	0.065	0. 018
Pr [détection CG]	0.279	0. 046
Pr [non détection]	0. 326	

Tableau VI - Probabilité de chaque détecteur selon la borne de détection (adresse ou non adresse).

Ces tableaux amènent les commentaires suivants :

- lorsque la détection se fait sur la borne de non-adresse, la détection est plus rapide, moyenne et écart-type plus faibles.
- La détection immédiate se fait uniquement sur les bornes de non adresse.
- A partir de l'échantillon de taille 570, les caractéristiques expérimentales globales du tableau VII peuvent être déterminées (il n'y a pas de pondération à effectuer, contrairement à ce qui a été fait pour les pannes de la périphérie, puisque les coefficients ne dépendent pas de l'événement E particulier. On notera que seules les caractéristiques concernant K_{COI} et K_{AE} seront d'un intérêt général pour la suite, car étant les seules indépendantes de toute application.

Quelques commentaires peuvent être faits sur la distribution du temps de détection par COI. La valeur forte de $Pr [K_{COI} = 0]$ ne provient pas de ce que l'adresse de la prochaine instruction est "envoyée" dans une zone remplie de codes invalides. Cette adresse n'est envoyée que dans des zones où figurent des programmes/données. La valeur $0.182 * 0.885 = 0.161$ est censée représenter la proportion de codes opérations invalides à une adresse quelconque en mémoire. Or il y a 30 % de codes invalides. En fait, les détections immédiates ne se font comme on l'a remarqué plus haut, que sur les bornes de non-adresse.

Une analyse du programme spécifique a montré que la répartition d'instructions 68000 était : 68 % étant des instructions d'un mot, 24 % à 2 mots, 6% à 3 mots et 1% à 4 mots ; ce qui nous donne : 16,75 % de codes quelconques. Or, il y a 30 % de codes invalides et donc la probabilité de détecter immédiatement un code invalide est de 5% ce qui est beaucoup plus faible que les 16 % que l'on trouve par expérience. Ceci vient du fait que la répartition des valeurs sur les mots n'étant pas des codes d'opérations, n'est pas uniforme. Cette considération sera reprise pour l'analyse dynamique.

Les valeurs de $E [K_{COI}]$ et $\sigma [K_{COI}]$ uniquement lorsque $K_{COI} \neq 0$, seront alors nécessaires. Ces valeurs sont respectivement : 8 et 25 à partir de l'échantillon. Elles sont représentatives exactement du nombre total d'instructions nécessaires pour passer de l'état 1 à l'état 12 ou 16.

DETECTION COI	DETECTION AE	DETECTION PM	DETECTION CG.	NON DETECTION
$E [K_{COI}]$ $\# \sigma [K_{COI}]$	$E [K_{AE}]$ $\sigma [K_{AE}]$	$E [K_{PM}]$ $\sigma [K_{PM}]$	$E [K_{CG}]$ $\sigma [K_{CG}]$	#
1	7	13	176	
104	48	47	185	186
		Pr [détection COI] = 0. 182 Pr [détection AE] = 0. 084 Pr [détection PM] = 0. 082 Pr [détection CG] = 0. 325 Pr [non détection] = 0. 326		
$Pr [K_{COI} = 0] = 0.885$ $Pr [K_{AE} = 0] = 0.625$ $Pr [K_{PM} = 0] = 0$ $Pr [K_{CG} = 0] = 0$			$Pr [K_{COI} > 0] = 0.115$ $Pr [K_{AE} > 0] = 0.375$ $Pr [K_{PM} > 0] = 1$ $Pr [K_{CG} > 0] = 1$	

Tableau VII - Caractéristiques expérimentales globales du modèle statique.

Les coefficients correspondant aux modèles des figures 2 et 4 peuvent être calculés pour l'application considérée.

Pour le modèle de la figure 2 :

$$\begin{aligned}
 \text{Pr [détection COI, } K_{\text{COI}} = 0, @]} &= 0.182 * 0 = p_2 \\
 \text{Pr [détection AE, } K_{\text{AE}} = 0, @]} &= 0.084 * 0 = p_3 \\
 \text{Pr [détection PM, } K_{\text{PM}} = 0, @]} &= 0.082 * 0 = p_4 \\
 \text{Pr [détection CG, } K_{\text{CG}} = 0, @]} &= 0.325 * 0 = p_5 \\
 \text{Pr [détection COI, } K_{\text{COI}} = 0, -@]} &= 0.182 * 0.885 = p_6 \\
 \text{Pr [détection AE, } K_{\text{AE}} = 0, -@]} &= 0.084 * 0.625 = p_7 \\
 \text{Pr [détection PM, } K_{\text{PM}} = 0, -@]} &= 0.082 * 0 = p_8 \\
 \text{Pr [détection CG, } K_{\text{CG}} = 0, -@]} &= 0.325 * 0 = p_9 \\
 \text{Pr [détection COI, } K_{\text{COI}} > 0, @]} &= 0 = \frac{p_1 p_{12}}{1-p_{11}} \\
 \text{Pr [détection AE, } K_{\text{AE}} > 0, @]} &= 0.021 = \frac{p_1 p_{13}}{1-p_{11}} \\
 \text{Pr [détection PM, } K_{\text{PM}} > 0, @]} &= 0.065 = \frac{p_1 p_{14}}{1-p_{11}} \\
 \text{Pr [détection CG, } K_{\text{CG}} > 0, @]} &= 0.279 = \frac{p_1 p_{15}}{1-p_{11}} \\
 \text{Pr [détection COI, } K_{\text{COI}} > 0, -@]} &= 0.021 = \frac{p_1 p_{16}}{1-p_{11}} \\
 \text{Pr [détection AE, } K_{\text{AE}} > 0, -@]} &= 0.011 = \frac{p_1 p_{17}}{1-p_{11}} \\
 \text{Pr [détection PM, } K_{\text{PM}} > 0, -@]} &= 0.018 = \frac{p_1 p_{12}}{1-p_{11}} \\
 \text{Pr [détection CG, } K_{\text{CG}} > 0, -@]} &= 0.046 = \frac{p_1 p_{19}}{1-p_{11}} \\
 E [K_{\text{COI}} + AE + PM, K_{\text{COI}} + AE + PM > 0] &= 13.4 = \frac{1}{1-p_{11}}
 \end{aligned}$$

On notera que la divergence des résultats expérimentaux de $E [K_{\text{COI}}]$, $E [K_{\text{AE}}]$ et $E [K_{\text{PM}}]$ a fait considérer alors leur moyenne : $E [K_{\text{COI}} + AE + PM] = 13.4$

On supposera pour résoudre le problème que la probabilité de non détection est constante quelque soit le nombre d'instructions exécutées donc :

$p_{10} = p_{20}$. On obtient alors les valeurs :

$p_1 = 0.757$	$p_{11} = 0.925$
$p_2 = 0$	$p_{12} = 0$
$p_3 = 0$	$p_{13} = 0.002$
$p_4 = 0$	$p_{14} = 0.006$
$p_5 = 0$	$p_{15} = 0.028$
$p_6 = 0.161$	$p_{16} = 0.002$
$p_7 = 0.053$	$p_{17} = 0.001$
$p_8 = 0$	$p_{18} = 0.002$
$p_9 = 0$	$p_{19} = 0.005$
$p_{10} = 0.029$	$p_{20} = 0.029$

Pour le modèle de la figure 4 les valeurs sont :

$p_2 = 0$	$p_6 = 0.029$
$p_3 = 0.003$	$p_7 = 0.010$
$p_4 = 0.010$	$p_8 = 0.003$
$p_5 = 0.044$	$p_9 = 0.007$

$$p_{10} = 0.052$$

$$p_{11} = 0.841$$

Ces valeurs sont rappelées dans la figure 5 pour le modèle avec différenciation de la première instruction (a) ainsi que pour le modèle sans différenciation de la première instruction (b).

On peut alors conclure pour le modèle statique, en notant que les valeurs obtenues ne sont importantes qu'en ce qui concerne des détecteurs indépendants de l'application comme COI et AE. Enfin le modèle de la figure 2 est plus exact que celui de la figure 4 .

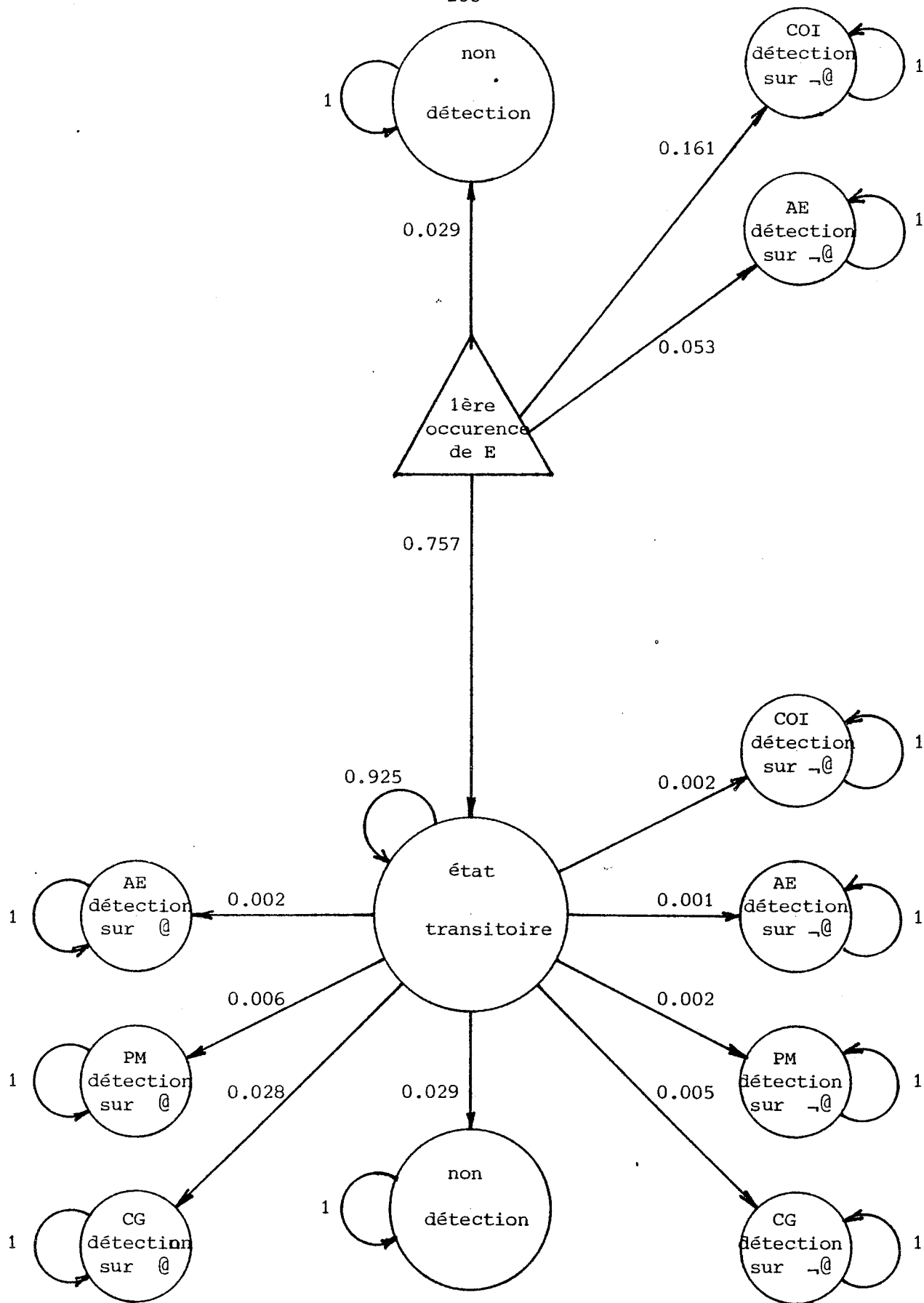


Figure 5 (a) - Valeurs caractéristiques des paramètres du modèle statique avec différenciation de la première instruction.

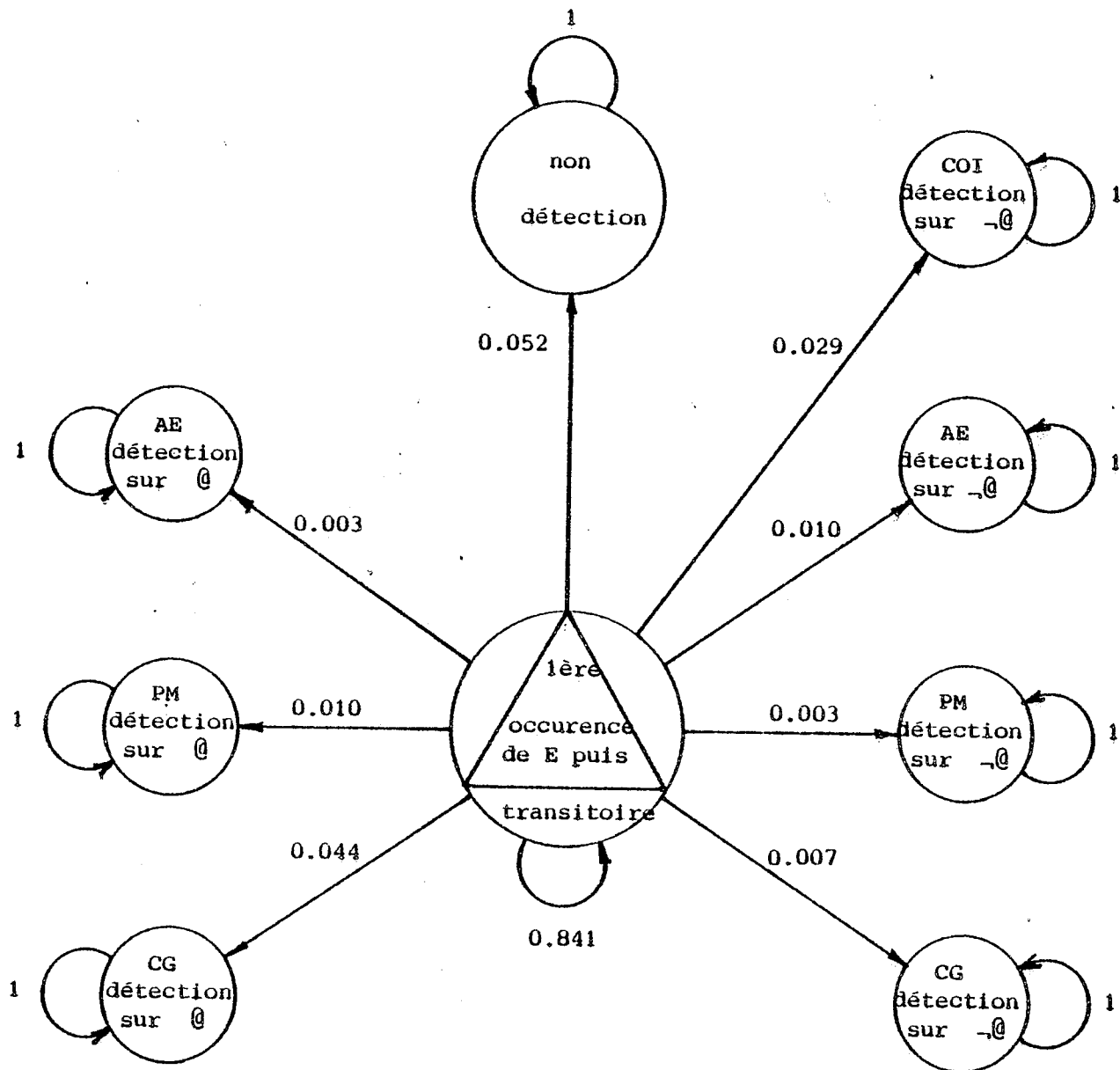


Figure 5 (b) - Valeurs caractéristiques des paramètres du modèle statique sans différenciation de la première instruction.

IV. ANALYSE DYNAMIQUE MODELISATION

Si l'analyse statique ne considère qu'un et un seul événement E, l'analyse dynamique considère quant à elle, l'occurrence de l'événement E avec une certaine fréquence. Chaque occurrence de E étant séparée par un intervalle, intervalle de durée moyenne MTB-E (MTB-E = mean time between E).

Appelons $1/f$ cet intervalle, en nombre d'instructions, entre deux occurrences de l'événement E. Chaque instruction a la probabilité f (i.e. $1/MTB-E$) de générer un événement E.

Dans toute l'analyse dynamique, on ne tiendra pas compte de la détection par chien de garde, les résultats ainsi obtenus seront pessimistes par rapport aux cas réels. Cette considération provient du fait que, la relance du chien de garde n'étant pas régulière, les résultats obtenus sont donc entachés d'erreurs. La détection par chien de garde sera donc considérée comme non détection.

Deux cas sont à distinguer : les pannes créant l'événement E avec une faible fréquence (MTB-E grand) et les pannes créant l'événement E avec une grande fréquence (MTB-E petit).

Le paragraphe IV.1 concerne les pannes créant l'événement E avec de faibles fréquences ; les pannes créant l'événement E avec de grandes fréquences pour lesquelles les modèles dynamiques seront développés, seront considérées dans les paragraphes suivants, comme pour l'analyse statique, les modèles seront étudiés sous forme matricielle puis un accès direct sera obtenu par le modèle de Markov.

1. Evénement E de faibles fréquences

Ce cas n'est pas le plus important, en effet, les pannes créant un événement E de faible fréquence seront normalement testées par d'autres moyens : programmes de test, comme pour les pannes ne créant pas d'événement E.

Soit le programme de test peut être déroulé, soit il provoque la génération d'un événement E. Dans ce cas, pour chaque génération de E, le temps de détection se fait à partir des probabilités P1, P2, P3, P4 où :

$$\begin{aligned} P1 &= \text{Pr [détection COI]} \\ P2 &= \text{Pr [détection AE]} \\ P3 &= \text{Pr [détection PM]} \\ P4 &= \text{Pr [non détection] (y compris CG)}. \end{aligned}$$

Si I est la variable représentant le nombre de générations de l'événement E, on obtient :

$$\text{Pr [I = i]} = P4^{i-1} (1-P4)$$

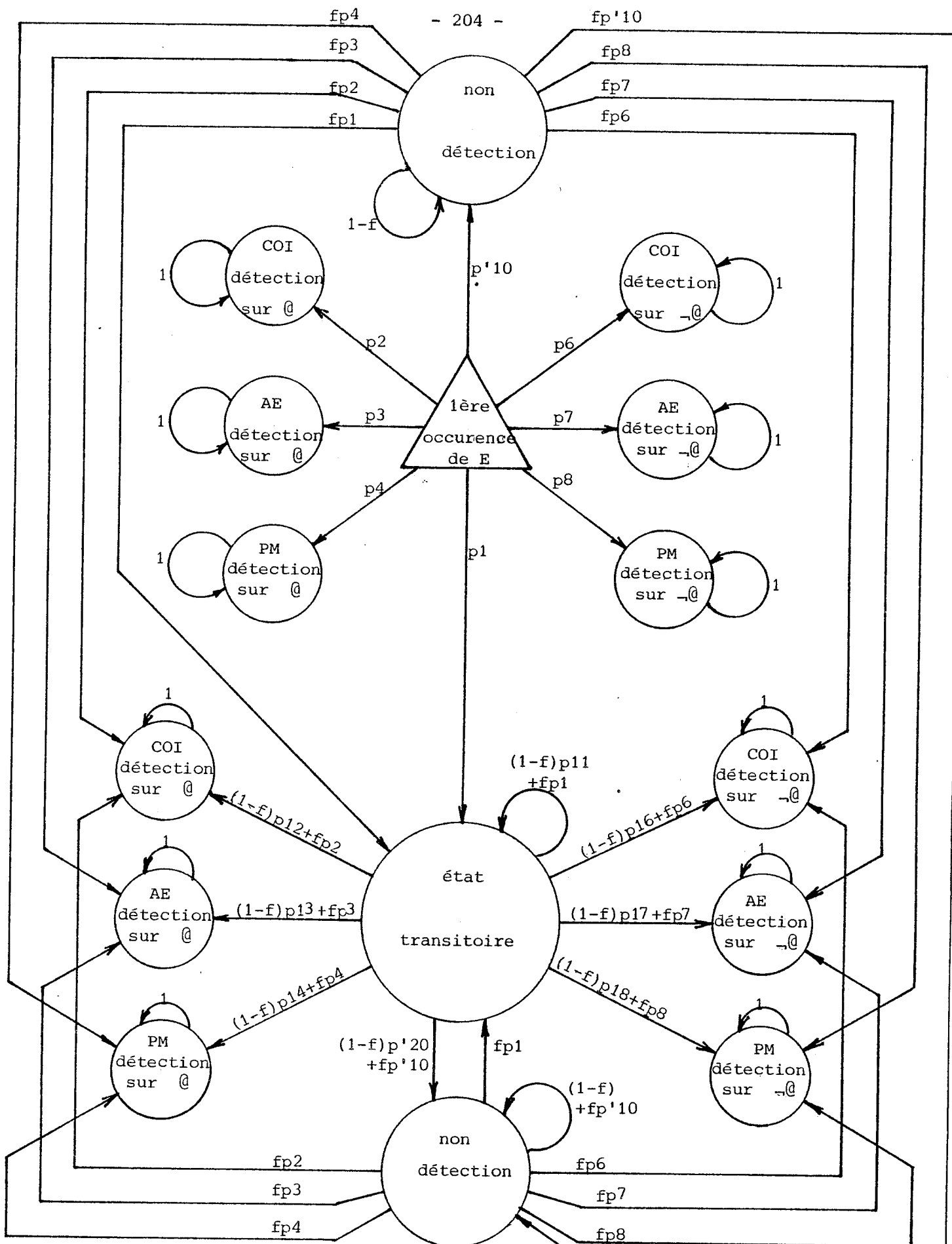
Le temps de détection peut donc être approché par (i-1) générations de E, augmentées de la valeur d'un intervalle To. Mais la distribution de ce temps dépend de la distribution du nombre de générations de E en l'occurrence d'une loi binomiale si l'on suppose f constant pour chaque instruction exécutée. La détermination de cette distribution du temps de détection n'a pas été faite en raison de son peu d'intérêt pratique.

2. Evénements E de grandes fréquences. Définitions des modèles dynamiques

On se place donc dans le cas de fréquences de E grandes. Les modèles dynamiques vont être considérés à partir des modèles statiques, mais pour lesquels la détection par chien de garde sera regroupée avec la non détection.

2.1. Premier modèle :

Le modèle de la figure 6 s'obtient à partir du modèle de la figure 2, en regroupant les états de détection par chien de garde et de non détection. Chaque état non absorbant pouvant devenir état d'occurrence des événements E suivants.



$$\begin{aligned}
 p'10 &= p5(\text{fig.17}) + p9(\text{fig.17}) + p10(\text{fig.17}) \\
 p'20 &= p15(\text{fig.17}) + p19(\text{fig.17}) + p20(\text{fig.17}) \\
 p1 + p2 + p3 + p4 + p6 + p7 + p8 + p'10 &= 1 \\
 p11 + p12 + p13 + p14 + p16 + p17 + p18 + p'20 &= 1
 \end{aligned}$$

Figure 6 - Modèle dynamique avec différenciation de la première instruction.

2.1.1. Accès matriciel

La complexité du système empêchant sa résolution matricielle, on ne trouvera ici que les probabilités d'états du système. Pour ce modèle on n'étudiera que la résolution directe par modèle de Markov.

- PROBABILITES D'ETAT

$$\begin{aligned}
 \text{Pr}(1)^{(i+1)} &= (1-f) p_{11} \text{Pr}(1)^{(i)} + fp_1 [\text{Pr}(1)^{(i)} + \text{Pr}(10)^{(i)} + \text{Pr}(20)^{(i)}] \\
 \text{Pr}(12)^{(i+1)} &= (1-f) p_{12} \text{Pr}(1)^{(i)} + fp_2 [\text{Pr}(1)^{(i)} + \text{Pr}(10)^{(i)} + \text{Pr}(20)^{(i)}] + \text{Pr}(12)^{(i)} \\
 \text{Pr}(13)^{(i+1)} &= (1-f) p_{13} \text{Pr}(1)^{(i)} + fp_3 [\text{Pr}(1)^{(i)} + \text{Pr}(10)^{(i)} + \text{Pr}(20)^{(i)}] + \text{Pr}(13)^{(i)} \\
 \text{Pr}(14)^{(i+1)} &= (1-f) p_{14} \text{Pr}(1)^{(i)} + fp_4 [\text{Pr}(1)^{(i)} + \text{Pr}(10)^{(i)} + \text{Pr}(20)^{(i)}] + \text{Pr}(14)^{(i)} \\
 \text{Pr}(16)^{(i+1)} &= (1-f) p_{16} \text{Pr}(1)^{(i)} + fp_6 [\text{Pr}(1)^{(i)} + \text{Pr}(10)^{(i)} + \text{Pr}(20)^{(i)}] + \text{Pr}(16)^{(i)} \\
 \text{Pr}(17)^{(i+1)} &= (1-f) p_{17} \text{Pr}(1)^{(i)} + fp_7 [\text{Pr}(1)^{(i)} + \text{Pr}(10)^{(i)} + \text{Pr}(20)^{(i)}] + \text{Pr}(17)^{(i)} \\
 \text{Pr}(18)^{(i+1)} &= (1-f) p_{18} \text{Pr}(1)^{(i)} + fp_8 [\text{Pr}(1)^{(i)} + \text{Pr}(10)^{(i)} + \text{Pr}(20)^{(i)}] + \text{Pr}(18)^{(i)} \\
 \text{Pr}(10)^{(i+1)} &= (1-f) \text{Pr}(10)^{(i)} \\
 \text{Pr}(20)^{(i+1)} &= (1-f) p_{20} \text{Pr}(1)^{(i)} + fp_{10} [\text{Pr}(1)^{(i)} + \text{Pr}(10)^{(i)} + \text{Pr}(20)^{(i)}] + (1-f) \text{Pr}(20)^{(i)}
 \end{aligned}$$

avec l'état initial :

$$\text{Pr}(1)^{(0)} = 1, \text{Pr}(i)^{(0)} = 0 \forall i \neq 1$$

2.1.2. Accès direct

La matrice de transition P, organisée en fonction des états transitoires et absorbants, s'écrit (l'ordre des états sur les lignes et les colonnes étant 2, 3, 4, 6, 7, 8, 12, 13, 14, 16, 17, 18, 0, 1, 10, 20)

$$P = \begin{bmatrix} I_{12} & O \\ R & Q \end{bmatrix}$$

où

$$Q = \begin{bmatrix} O & A_1 & A_2 & A_3 \\ O & B_1 & B_2 & B_3 \\ O & C_1 & C_2 & C_3 \\ O & D_1 & D_2 & D_3 \end{bmatrix}$$

$$R = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \\ 0 & 0 & 0 & 0 & 0 & 0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \end{bmatrix}$$

avec : $A_1 = p1$

$A_2 = p'10$

$A_3 = 0$

$B_1 = (1-f) p11 + fp1$

$B_2 = 0$

$B_3 = (1-f) p'20 + fp'10$

$C_1 = fp1$

$C_2 = 1-f$

$C_3 = fp'10$

$D_1 = fp1$

$D_2 = 0$

$D_3 = 1-f + fp'10$

et :

$a_1 = p2$

$a_4 = p6$

$a_2 = p3$

$a_5 = p7$

$a_3 = p4$

$a_6 = p8$

$b_1 = (1-f) p12 + fp2$

$b_4 = (1-f) p16 + fp6$

$b_2 = (1-f) p13 + fp3$

$b_5 = (1-f) p17 + fp7$

$b_3 = (1-f) p14 + fp4$

$b_6 = (1-f) p18 + fp8$

$c_1 = fp2$

$c_4 = fp6$

$c_2 = fp3$

$c_5 = fp7$

$c_3 = fp4$

$c_6 = fp8$

$d_1 = fp2$

$d_4 = fp6$

$d_2 = fp3$

$d_5 = fp7$

$d_3 = fp4$

$d_6 = fp8$

La matrice fondamentale N s'écrit alors :

$N = [I_4 - Q]^{-1}$

Soit $N = \frac{1}{\text{dét.}(I-Q)} \begin{bmatrix} n11 & n12 & n13 & n14 \\ n21 & n22 & n23 & n24 \\ n31 & n32 & n33 & n34 \\ n41 & n42 & n43 & n44 \end{bmatrix}$

avec

$$\det(I-Q) = (1-B_1)(1-C_2)(1-D_3) - B_3 D_1 (1-C_2) - B_2 C_1 (1-D_3) - C_3 D_2 (1-B_1) - B_2 C_3 D_1 - B_3 C_1 D_2$$

et

$$n_{11} = (1-B_1)(1-C_2)(1-D_3) - B_3 D_1 (1-C_2) - B_2 C_1 (1-D_3) - C_3 D_2 (1-B_1) - B_2 C_3 D_1 - B_3 C_1 D_2$$

$$n_{12} = A_1 (1-C_2)(1-D_3) + A_3 D_1 (1-C_2) + A_2 C_1 (1-D_3) + A_3 C_1 D_2 + A_2 C_3 D_1 - A_1 C_3 D_2$$

$$n_{13} = A_2 (1-B_1)(1-D_3) + A_3 D_2 (1-B_1) + A_1 B_2 (1-D_3) + A_3 B_2 D_1 + A_1 B_3 D_2 - A_2 B_3 D_1$$

$$n_{14} = A_3 (1-B_1)(1-C_2) + A_2 C_3 (1-B_1) + A_1 B_3 (1-C_2) + A_1 B_2 C_3 + A_2 B_3 C_1 - A_3 B_2 C_1$$

$$n_{21} = 0$$

$$n_{22} = (1-C_2)(1-D_3) - C_3 D_2$$

$$n_{23} = B_2 (1-D_3) + B_3 D_2$$

$$n_{24} = B_3 (1-C_2) + B_2 C_3$$

$$n_{31} = 0$$

$$n_{32} = C_1 (1-D_3) + C_3 D_1$$

$$n_{33} = (1-B_1)(1-D_3) - B_3 D_1$$

$$n_{34} = C_3 (1-B_1) + B_3 C_1$$

$$n_{41} = 0$$

$$n_{42} = D_1 (1-C_2) + C_1 D_2$$

$$n_{43} = D_2 (1-B_1) + B_2 D_1$$

$$n_{44} = (1-B_1)(1-C_2) - B_2 C_1$$

- PROBABILITES D'ETAT, PROBABILITES DE DETECTION

La matrice $B = [b_{ij}]$ est telle que b_{ij} est la probabilité que le processus soit absorbé en j en partant de l'état i . Elle est donnée par $B = NR$

On obtient donc :

$$\text{Pr} [K_{\text{COI}} = 0, @] = \text{Pr} (2) = b_{1,1} = p_2$$

$$\text{Pr} [K_{\text{AE}} = 0, @] = \text{Pr} (3) = b_{1,2} = p_3$$

$$\text{Pr} [K_{\text{PM}} = 0, @] = \text{Pr} (4) = b_{1,3} = p_4$$

$$\text{Pr} [K_{\text{COI}} = 0, -@] = \text{Pr} (6) = b_{1,4} = p_6$$

$$\text{Pr} [K_{\text{AE}} = 0, -@] = \text{Pr} (7) = b_{1,5} = p_7$$

$$\text{Pr} [K_{\text{PM}} = 0, -@] = \text{Pr} (8) = b_{1,6} = p_8$$

$$\text{Pr} [\text{détection COI}, K_{\text{COI}} > 0, @] = \text{Pr} (12) = b_{1,7} = \frac{1}{\det(I-Q)} [b_1^{n_{12}} + c_1^{n_{13}} + d_1^{n_{14}}]$$

$$\text{Pr} [\text{détection AE}, K_{\text{AE}} > 0, @] = \text{Pr} (13) = b_{1,8} = \frac{1}{\det(I-Q)} [b_2^{n_{12}} + c_2^{n_{13}} + d_2^{n_{14}}]$$

$$\text{Pr} [\text{détection PM}, K_{\text{PM}} > 0, @] = \text{Pr} (14) = b_{1,9} = \frac{1}{\det(I-Q)} [b_3^{n_{12}} + c_3^{n_{13}} + d_3^{n_{14}}]$$

$$\text{Pr} [\text{détection COI}, K_{\text{COI}} > 0, -@] = \text{Pr} (16) = b_{1,10} = \frac{1}{\det(I-Q)} [b_4^{n_{12}} + c_4^{n_{13}} + d_4^{n_{14}}]$$

$$\text{Pr} [\text{détection AE}, K_{\text{AE}} > 0, -@] = \text{Pr} (17) = b_{1,11} = \frac{1}{\det(I-Q)} [b_5^{n_{12}} + c_5^{n_{13}} + d_5^{n_{14}}]$$

$$\text{Pr} [\text{détection PM}, K_{\text{PM}} > 0, -@] = \text{Pr} (18) = b_{1,12} = \frac{1}{\det(I-Q)} [b_6^{n_{12}} + c_6^{n_{13}} + d_6^{n_{14}}]$$

En remplaçant les paramètres par leurs valeurs on obtient :

$$\text{Pr} [\text{détection COI}, K_{\text{COI}} > 0, @] = \frac{(1-f)p_1p_{12}+p_2[fp_1+p'_{10}(1-p_{11}+fp_{11})+(1-f)p_1p'_{20}]}{(1-p_{11}+fp_{11})(1-p'_{10})-p_1(p'_{20}-fp'_{20}+f)}$$

$$\text{Pr} [\text{détection AE}, K_{\text{AE}} > 0, @] = \frac{(1-f)p_1p_{13}+p_3[fp_1+p'_{10}(1-p_{11}+fp_{11})+(1-f)p_1p'_{20}]}{(1-p_{11}+fp_{11})(1-p'_{10})-p_1(p'_{20}-fp'_{20}+f)}$$

$$\text{Pr} [\text{détection PM}, K_{\text{PM}} > 0, @] = \frac{(1-f)p_1p_{14}+p_4[fp_1+p'_{10}(1-p_{11}+fp_{11})+(1-f)p_1p'_{20}]}{(1-p_{11}+fp_{11})(1-p'_{10})-p_1(p'_{20}-fp'_{20}+f)}$$

$$\text{Pr} [\text{détection COI}, K_{\text{COI}} > 0, -@] = \frac{(1-f)p_1p_{16}+p_6[fp_1+p'_{10}(1-p_{11}+fp_{11})+(1-f)p_1p'_{20}]}{(1-p_{11}+fp_{11})(1-p'_{10})-p_1(p'_{20}-fp'_{20}+f)}$$

$$\text{Pr} [\text{détection AE}, K_{\text{AE}} > 0, -@] = \frac{(1-f)p_1p_{17}+p_7[fp_1+p'_{10}(1-p_{11}+fp_{11})+(1-f)p_1p'_{20}]}{(1-p_{11}+fp_{11})(1-p'_{10})-p_1(p'_{20}-fp'_{20}+f)}$$

$$\text{Pr} [\text{détection PM}, K_{\text{PM}} > 0, -@] = \frac{(1-f)p_1p_{18}+p_8[fp_1+p'_{10}(1-p_{11}+fp_{11})+(1-f)p_1p'_{20}]}{(1-p_{11}+fp_{11})(1-p'_{10})-p_1(p'_{20}-fp'_{20}+f)}$$

On peut vérifier que la somme des probabilités de détection est égale à 1 ;

$$\text{Pr}(2)+\text{Pr}(3)+\text{Pr}(4)+\text{Pr}(6)+\text{Pr}(7)+\text{Pr}(8)+\text{Pr}(12)+\text{Pr}(13)+\text{Pr}(14)+\text{Pr}(16)+\text{Pr}(17)+\text{Pr}(18) = 1$$

- MOMENTS DU TEMPS DE DETECTION

T et T_2 représentent les moyennes et variances du temps passé avant absorption avec :

$$T = N \xi \text{ et } T_2 = (2N - I_4) T - T \text{sq}$$

où ξ est le vecteur colonne à éléments égaux à 1.

et $T \text{sq}$ est la matrice obtenue par élévation au carré de chaque élément de

On obtient :

$$\begin{aligned} T_1 &= E (\text{temps passé avant absorption en partant de l'état } 0) \\ &= \frac{n_{11} + n_{12} + n_{13} + n_{14}}{\det (I-Q)} \\ &= \frac{+A2B3(C1-D1)+A2C3D1}{(1-B1)(1-C2)(1-D3) - B3D1(1-C2)} \\ &= \frac{[(1-B1)(1-C2)+A1(1-C2)+A2(1-B1)+A2C1](1-D3)+A2C3(1-B1)+(A1B3-B3D1)(1-C2)}{(1-B1)(1-C2)(1-D3) - B3D1(1-C2)} \\ &= \frac{f [(1-p_{11}+fp_{11})(p'_{10}+f(1-p'_{10})) + (1-f) p_1 (p'_{20}+f(1-p'_{20}))]}{f^2 [(1-p_{11}+fp_{11})(1-p'_{10}) - p_1 (p'_{20} + f (1-p'_{20}))]} \\ &= \frac{(1-p_{11}+fp_{11})(p'_{10}+f(1-p'_{10}))+ (1-f) p_1 (p'_{20}+f(1-p'_{20}))}{f [(1-p_{11}+fp_{11})(1-p'_{10}) - p_1 (p'_{20}+f(1-p'_{20}))]} \\ &= 1 + \frac{p'_{10}(1-p_{11}+fp_{11}) + p_1 (p'_{20} + f (1-p'_{20}))}{f [(1-p_{11}+fp_{11})(1-p'_{10}) - p_1 (p'_{20}+f(1-p'_{20}))]} \\ &= E [K_{COI}, @] + 1 = E [K_{AE}, @] + 1 = E [K_{PM}, @] + 1 \\ &= E [K_{COI}, -@] + 1 = E [K_{AE}, -@] + 1 = E [K_{PM}, -@] + 1 \end{aligned}$$

De même on obtient :

E [temps passé avant absorption en partant de 1] =

$$\frac{p'20 + f (1-p'20)}{f [(1-p'10) (1-p11+fp11) - p1 (p'20+f (1-p'20))]}$$

E [temps passé avant absorption en partant de 10] =

$$\frac{1 - p11 + fp11 + 2p1 (fp'10 + (1-f) p'20)}{f [(1-p'10) (1-p11+fp11)-p1 (p'20 + f(1-p'20))]}$$

E [temps passé avant absorption en partant de 20]

$$\frac{1 - p11 + fp11}{f [(1-p'10) (1-p11+fp11)-p1 (p'20+f (1-p'20))]}$$

On obtient pour le second moment :

$$\sigma^2 [K_{COI}] = \sigma^2 [K_{AE}] = \sigma^2 [K_{PM}] =$$

$$\frac{1}{f^4 [(1-p'10) (1-p11+fp11)-p1 (f+(1-f)p'20)]^2} *$$

$$\begin{aligned} & [-f^2 p'10 (f(1-p'10)+2-p'10) (1-p11+fp11)^2 + f^2 p1^2 (f-p'10^2) (f+(1-f)p'20)^2 \\ & + (1-p11+fp11) [f^2 (f+(1-f)p'20) (2p1p'10(f+p'10(1-p'10)-f))] \\ & + 2f^2 p1p'10 [(2-f) p1 p'10(1-p'10) + f(p'10^2-2p'10-2)] \\ & + 2f^2 (1-f)p'20 [-p1p'10(1-p'10)+1-p'10 (1+2p'10)]] \\ & + f^2 p1 [-f^2 p1-p1 (fp'10+(1-f) (1-p'10)p'20)^2 + 2f(f+(1-f)p'20) (1+p1p'10) \\ & - 2 p1 p'10 (f+ (1-f) p'20) [(1f) (1-p'10) p'20 + p'10 (2-f)] \\ & - 2 fp1 (fp'10 + (1-p'10) (1-f) p'20)]] \end{aligned}$$

que l'on peut mettre sous la forme :

$$\begin{aligned} & \sigma^2 [K_{COI}] = \\ & \left\{ f^5 [p1^2 - 2 p1^2 p'20 - p1 p11 + 2 p1 p11 p'10 - p11^2 p'10 \right. \\ & \quad \left. + p11^2 p'10^2 - 2p1 p11 p'10 p'20 + p1 p11 p'20 + p1^2 p'20^2] \right. \\ & \left. + f^4 [p1 + 2 p1 p'10 + p1 p11 - 4 p1 p11 p'10 + 4p1^2 p'20 - p1^2 - p1 p'20 \right. \\ & \quad \left. - 2 p11 p'10 + 4 p11^2 p'10 - 3 p11^2 p'10^2 + 6 p1 p11 p'10 p'20 \right. \\ & \quad \left. + 2 p11 p'10^2 - 2 p1 p'10 p'20 - 4 p1 p11 p'20 - 3 p1^2 p'20^2] \right\} \end{aligned}$$

$$\begin{aligned}
 &+f^3 [-p_1 p'_{20} - 2p_1 p'_{10} + 2p_1 p_{11} p'_{10} - 2 p_1^2 p'_{20} + 6 p_{11} p'_{10} \\
 &\quad - 5 p_{11}^2 p'_{10} + 3p_{11}^2 p'_{10}{}^2 - 4 p_{11} p'_{10}{}^2 - 6 p_1 p_{11} p'_{10} p'_{20} \\
 &\quad + 4 p_1 p'_{10} p'_{20} - p'_{10} + p'_{10}{}^2 + 5 p_1 p_{11} p'_{20} + 3 p_1^2 p'_{20}{}^2] \\
 &+f^2 [2 p'_{10} - 4p_{11} p'_{10} + 2p_{11} p'_{10}{}^2 - p'_{10}{}^2 - 2 p_1 p'_{10} p'_{20} \\
 &+2 p_{11}^2 p'_{10} - p_{11}^2 p'_{10}{}^2 + 2p_1 p_{11} p'_{10} p'_{20} + 2p_1 p'_{20} - 2p_1 p_{11} p'_{20} \\
 &- p_1^2 p'_{20}{}^2] \left\{ * \frac{1}{f^4 [(1-p'_{10})(1-p_{11}+fp_{11}) - p_1 (f+(1-f) p'_{20})^2]} \right.
 \end{aligned}$$

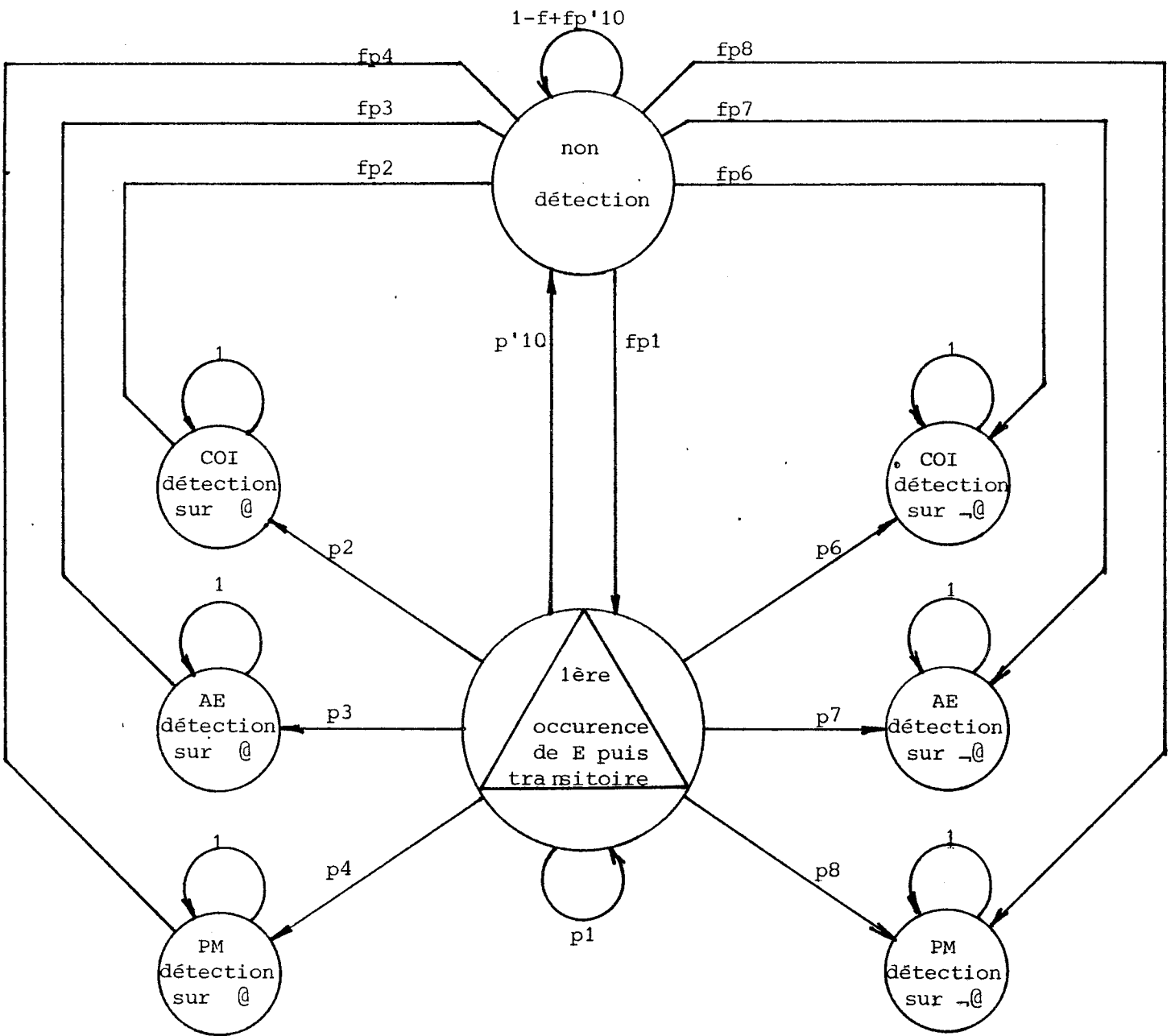
2.2. Second Modèle

Pour ce second modèle, c'est le modèle de la figure 4 qui est repris ici pour un ensemble d'événements E dont la probabilité d'occurrence par instruction est f. On obtient alors la figure 7 .

Pour les mêmes raisons que dans le paragraphe précédent, on ne tiendra pas compte de la détection par chien de garde qui sera considérée comme non détection.

2.2.1. Accès matriciel

- Probabilités d'Etat



$$p'10 = p5 \text{ (fig.19)} + p9 \text{ (fig.19)} + p10 \text{ (fig.19)}$$

$$p1 + p2 + p3 + p4 + p6 + p7 + p8 + p'10 = 1$$

Figure 7 - Modèle dynamique adopté sans différenciation de la première instruction.

$$\begin{aligned}
 \text{Pr}(1)^{(i+1)} &= p_1 \text{Pr}(1)^{(i)} + fp_1 \text{Pr}(10)^{(i)} \\
 \text{Pr}(2)^{(i+1)} &= p_2 \text{Pr}(1)^{(i)} + fp_2 \text{Pr}(10)^{(i)} + \text{Pr}(2)^{(i)} \\
 \text{Pr}(3)^{(i+1)} &= p_3 \text{Pr}(1)^{(i)} + fp_3 \text{Pr}(10)^{(i)} + \text{Pr}(3)^{(i)} \\
 \text{Pr}(4)^{(i+1)} &= p_4 \text{Pr}(1)^{(i)} + fp_4 \text{Pr}(10)^{(i)} + \text{Pr}(4)^{(i)} \\
 \text{Pr}(6)^{(i+1)} &= p_6 \text{Pr}(1)^{(i)} + fp_6 \text{Pr}(10)^{(i)} + \text{Pr}(6)^{(i)} \\
 \text{Pr}(7)^{(i+1)} &= p_7 \text{Pr}(1)^{(i)} + fp_7 \text{Pr}(10)^{(i)} + \text{Pr}(7)^{(i)} \\
 \text{Pr}(8)^{(i+1)} &= p_8 \text{Pr}(1)^{(i)} + fp_8 \text{Pr}(10)^{(i)} + \text{Pr}(8)^{(i)} \\
 \text{Pr}(10)^{(i+1)} &= p'_{10} \text{Pr}(1)^{(i)} + (1-f(1-p'_{10})) \text{Pr}(10)^{(i)}
 \end{aligned}$$

Avec $\text{Pr}(1)^{(0)} = 1$

$$\text{Pr}(i)^{(0)} = 0 \quad \forall i \neq 1$$

à l'instant 0 apparaît la première occurrence de E.

Il est donc nécessaire de calculer la puissance ième de la matrice S :

$$S = \begin{bmatrix}
 p_1 & 0 & 0 & 0 & 0 & 0 & 0 & fp_1 \\
 p_2 & 1 & 0 & 0 & 0 & 0 & 0 & fp_2 \\
 p_3 & 0 & 1 & 0 & 0 & 0 & 0 & fp_3 \\
 p_4 & 0 & 0 & 1 & 0 & 0 & 0 & fp_4 \\
 p_6 & 0 & 0 & 0 & 1 & 0 & 0 & fp_6 \\
 p_7 & 0 & 0 & 0 & 0 & 1 & 0 & fp_7 \\
 p_8 & 0 & 0 & 0 & 0 & 0 & 1 & fp_8 \\
 p'_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 1-f(1-p'_{10})
 \end{bmatrix}$$

S est de la forme :

$$\begin{bmatrix}
 a & 0 & 0 & 0 & 0 & 0 & 0 & A \\
 b & 1 & 0 & 0 & 0 & 0 & 0 & B \\
 c & 0 & 1 & 0 & 0 & 0 & 0 & C \\
 d & 0 & 0 & 1 & 0 & 0 & 0 & D \\
 e & 0 & 0 & 0 & 1 & 0 & 0 & E \\
 g & 0 & 0 & 0 & 0 & 1 & 0 & G \\
 h & 0 & 0 & 0 & 0 & 0 & 1 & H \\
 l & 0 & 0 & 0 & 0 & 0 & 0 & L
 \end{bmatrix}$$

Il est aisé de voir que S^n , la nième puissance de S , est de la forme :

$$\begin{bmatrix} a_n & 0 & 0 & 0 & 0 & 0 & 0 & A_n \\ b_n & 1 & 0 & 0 & 0 & 0 & 0 & B_n \\ c_n & 0 & 1 & 0 & 0 & 0 & 0 & C_n \\ d_n & 0 & 0 & 1 & 0 & 0 & 0 & D_n \\ e_n & 0 & 0 & 0 & 1 & 0 & 0 & E_n \\ g_n & 0 & 0 & 0 & 0 & 1 & 0 & G_n \\ h_n & 0 & 0 & 0 & 0 & 0 & 1 & H_n \\ l_n & 0 & 0 & 0 & 0 & 0 & 0 & L_n \end{bmatrix}$$

avec les relations de récurrence :

$$\begin{array}{ll} a_{n+1} = aa_n + lA_n & A_{n+1} = Aa_n + LA_n \\ b_{n+1} = ab_n + lB_n + b & B_{n+1} = Ab_n + LB_n + B \\ c_{n+1} = ac_n + lC_n + c & C_{n+1} = Ac_n + LC_n + C \\ d_{n+1} = ad_n + lD_n + d & D_{n+1} = Ad_n + LD_n + D \\ e_{n+1} = ae_n + lE_n + e & E_{n+1} = Ae_n + LE_n + E \\ g_{n+1} = ag_n + lG_n + g & G_{n+1} = Ag_n + LG_n + G \\ h_{n+1} = ah_n + lH_n + h & H_{n+1} = Ah_n + LH_n + H \\ l_{n+1} = al_n + lL_n & L_{n+1} = Al_n + LL_n \end{array}$$

Il faut donc résoudre les deux types de systèmes récurrents, qui sont semblables à une translation près :

$$\begin{cases} a_{n+1} = aa_n + lA_n \\ A_{n+1} = Aa_n + LA_n \end{cases} \quad \text{et} \quad \begin{cases} b_{n+1} = ab_n + lB_n + b \\ B_{n+1} = Ab_n + LB_n + B \end{cases}$$

Pour le premier système ; en posant :

$$X_n = \begin{bmatrix} a_n \\ A_n \end{bmatrix} \quad X_1 = \begin{bmatrix} a \\ A \end{bmatrix}$$

On a :

$$\begin{aligned} X_{n+1} &= \begin{bmatrix} a & l \\ A & L \end{bmatrix} X_n \\ &= \begin{bmatrix} a & l \\ A & L \end{bmatrix}^n X_1 = M^n X_1 \end{aligned}$$

Or M peut s'écrire $JM'J^{-1}$ où M' est diagonale.

On obtient alors :

$$J = \begin{bmatrix} 1 & 1 \\ \lambda_1 - a & \lambda_2 - a \end{bmatrix}$$

$$J^{-1} = \frac{1}{(\lambda_2 - \lambda_1)} \begin{bmatrix} \lambda_2 - a & -1 \\ a - \lambda_1 & 1 \end{bmatrix}$$

$$M' = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

et l'on a finalement :

$$X_{n+1} = JM'^n J^{-1} X_1 = \frac{1}{(\lambda_2 - \lambda_1)} \begin{bmatrix} 1 & 1 \\ \lambda_1 - a & \lambda_2 - a \end{bmatrix} \begin{bmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{bmatrix} \begin{bmatrix} \lambda_2 - a - 1 \\ a - \lambda_1 & 1 \end{bmatrix} X_1$$

$$\text{avec } X_1 = \begin{bmatrix} a \\ A \end{bmatrix}$$

et λ_1, λ_2 solutions de

$$\lambda^2 - (a+L)\lambda + aL - Al = 0$$

Ceci donne immédiatement :

$$\begin{bmatrix} a_{n+1} \\ A_{n+1} \end{bmatrix} = JM'^n J^{-1} \begin{bmatrix} a \\ A \end{bmatrix}$$

$$\begin{bmatrix} l_{n+1} \\ L_{n+1} \end{bmatrix} = JM'^n J^{-1} \begin{bmatrix} 1 \\ L \end{bmatrix}$$

Le second système récurrent est du même type que le précédent :

En posant :

$$Y_n = \begin{bmatrix} b_n \\ B_n \end{bmatrix} \quad Y_1 = \begin{bmatrix} b \\ B \end{bmatrix}$$

on a :

$$Y_{n+1} = M Y_n + Y_1$$

De manière classique, on élimine Y_1 par translation :

$$Y = Z + K_1$$

soit :

$$\begin{aligned} Z_{n+1} + K_1 &= M (Z_n + K_1) + Y_1 \\ &= M Z_n + M K_1 + Y_1 \end{aligned}$$

On se ramène alors au premier système en posant :

$$K_1 = M K_1 + Y_1$$

On a, immédiatement :

$$K_1 = \begin{bmatrix} \det \begin{bmatrix} -b & 1 \\ -B & L-1 \end{bmatrix} \\ \det \begin{bmatrix} a-1 & 1 \\ A & L-1 \end{bmatrix} \\ \det \begin{bmatrix} a-1 & -b \\ A & -B \end{bmatrix} \\ \det \begin{bmatrix} a-1 & 1 \\ A & L-1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \frac{b(1-L) + B1}{(a-1)(L-1) - A1} \\ \frac{B(1-a) + Ab}{(a-1)(L-1) - A1} \end{bmatrix}$$

$$\text{Par suite : } Z_{n+1} = M^n Z_1 \quad Z_1 = Y_1 - K_1$$

$$\text{Donc } Y_{n+1} = M^n Z_1 + K_1$$

Soit finalement :

$$\begin{bmatrix} b_{n+1} \\ B_{n+1} \end{bmatrix} = JM^n J^{-1} \begin{bmatrix} \frac{ab(L-1) - 1(bA+B)}{(a-1)(L-1) - A1} \\ \frac{BL(a-1) - A(B1+b)}{(a-1)(L-1) - A1} \end{bmatrix} + \begin{bmatrix} \frac{b(1-L) + B1}{(a-1)(L-1) - A1} \\ \frac{B(1-a) + Ab}{(a-1)(L-1) - A1} \end{bmatrix}$$

$$\begin{bmatrix} c_{n+1} \\ C_{n+1} \end{bmatrix} = JM^n J^{-1} \begin{bmatrix} \frac{ac(L-1) - 1(cA+C)}{(a-1)(L-1) - A1} \\ \frac{CL(a-1) - A(C1+c)}{(a-1)(L-1) - A1} \end{bmatrix} + \begin{bmatrix} \frac{c(1-L) + C1}{(a-1)(L-1) - A1} \\ \frac{C(1-a) + Ac}{(a-1)(L-1) - A1} \end{bmatrix}$$

$$\begin{bmatrix} d_{n+1} \\ D_{n+1} \end{bmatrix} = JM^n J^{-1} \begin{bmatrix} \frac{ad(L-1) - 1(dA+D)}{(a-1)(L-1) - A1} \\ \frac{DL(a-1) - A(D1+d)}{(a-1)(L-1) - A1} \end{bmatrix} + \begin{bmatrix} \frac{d(1-L) + D1}{(a-1)(L-1) - A1} \\ \frac{D(1-a) + Ad}{(a-1)(L-1) - A1} \end{bmatrix}$$

$$\begin{bmatrix} e_{n+1} \\ E_{n+1} \end{bmatrix} = JM^n J^{-1} \begin{bmatrix} \frac{ae(L-1) - 1(eA+E)}{(a-1)(L-1) - A1} \\ \frac{EL(a-1) - A(EL+e)}{(a-1)(L-1) - A1} \end{bmatrix} + \begin{bmatrix} \frac{e(1-L) + E1}{(a-1)(L-1) - A1} \\ \frac{E(1-a) + Ae}{(a-1)(L-1) - A1} \end{bmatrix}$$

$$\begin{bmatrix} g_{n+1} \\ G_{n+1} \end{bmatrix} = JM^n J^{-1} \begin{bmatrix} \frac{ag(L-1) - 1(gA+G)}{(a-1)(L-1) - A1} \\ \frac{GL(a-1) - A(G1+g)}{(a-1)(L-1) - A1} \end{bmatrix} + \begin{bmatrix} \frac{g(1-L) + G1}{(a-1)(L-1) - A1} \\ \frac{G(1-a) + Ag}{(a-1)(L-1) - A1} \end{bmatrix}$$

$$\begin{bmatrix} h_{n+1} \\ H_{n+1} \end{bmatrix} = JM^n J^{-1} \begin{bmatrix} \frac{ah(L-1) - 1(hA+H)}{(a-1)(L-1) - A1} \\ \frac{HL(a-1) - A(H1+h)}{(a-1)(L-1) - A1} \end{bmatrix} + \begin{bmatrix} \frac{h(1-L) + H1}{(a-1)(L-1) - A1} \\ \frac{H(1-a) + Ah}{(a-1)(L-1) - A1} \end{bmatrix}$$

Ayant les a_i, b_i, \dots, l_i et les A_i, B_i, \dots, L_i les probabilités d'états suivent par :

$$\begin{aligned} \text{Pr}(2)^{(n)} &= \text{Pr} [\text{Détection par COI, @, après } n \text{ exécutions d'instructions} \\ &\quad \text{suivant la 1ère occurrence de E}] \\ &= b_n \end{aligned}$$

$$\text{Pr}(3)^{(n)} = c_n$$

$$\text{Pr}(4)^{(n)} = d_n$$

$$\text{Pr}(6)^{(n)} = e_n$$

$$\text{Pr}(7)^{(n)} = g_n$$

$$\text{Pr}(8)^{(n)} = h_n$$

Naturellement :

Pr [détection par COI après n exécutions d'instructions suivant la 1ère occurrence de E]

$$= b_n + e_n$$

Pr [détection par AE après n exécutions d'instructions suivant la 1ère occurrence de E]

$$= c_n + g_n$$

Pr [détection par PM après n exécutions d'instructions suivant la 1ère occurrence de E]

$$= d_n + h_n$$

De ces probabilités d'état, il est aisé de déterminer les distributions, densités et moments du temps de détection.

- Distributions, densités et probabilités limites

Les distributions de K_{COI} , K_{AE} et K_{PM} sont :

$$\text{Pr} [K_{COI} \leq k, @] = \text{Pr}(2)^{(k+1)}$$

$$= \frac{b_{k+1}}{1 - b_{k+1}}$$

$$\text{Pr} [K_{COI} \leq k, -@] = \text{Pr}(6)^{(k+1)}$$

$$\text{Pr} [K_{AE} \leq k, @] = \text{Pr}(3)^{(k+1)}$$

$$\text{Pr} [K_{AE} \leq k, -@] = \text{Pr}(7)^{(k+1)}$$

$$\text{Pr} [K_{PM} \leq k, @] = \text{Pr}(4)^{(k+1)}$$

$$\text{Pr} [K_{PM} \leq k, -@] = \text{Pr}(8)^{(k+1)}$$

Les densités de K_{COI} , K_{AE} , et K_{PM} s'obtiennent par :

$$\begin{aligned} \Pr [K_{COI} = 0, @] &= b \\ \Pr [K_{COI} = k, @] &= \Pr [K_{COI} \leq k, @] - \Pr [K_{COI} \leq k-1, @] \\ &= b_{k+1} - b_k \\ \Pr [K_{COI} = 0, \neg @] &= 1 \\ \Pr [K_{COI} = k, \neg @] &= \Pr [K_{COI} \leq k, \neg @] - \Pr [K_{COI} \leq k-1, \neg @] \\ &= 1_{k+1} - 1_k \\ \Pr [K_{AE} = 0, @] &= c \\ \Pr [K_{AE} = k, @] &= \Pr [K_{AE} \leq k, @] - \Pr [K_{AE} \leq k-1, @] \\ &= c_{k+1} - c_k \\ \Pr [K_{AE} = 0, \neg @] &= g \\ \Pr [K_{AE} = k, \neg @] &= \Pr [K_{AE} \leq k, \neg @] - \Pr [K_{AE} \leq k-1, \neg @] \\ &= g_{k+1} - g_k \\ \Pr [K_{PM} = 0, @] &= d \\ \Pr [K_{PM} = k, @] &= \Pr [K_{PM} \leq k, @] - \Pr [K_{PM} \leq k-1, @] \\ &= d_{k+1} - d_k \\ \Pr [K_{PM} = 0, \neg @] &= h \\ \Pr [K_{PM} = k, \neg @] &= \Pr [K_{PM} \leq k, \neg @] - \Pr [K_{PM} \leq k-1, \neg @] \\ &= h_{k+1} - h_k \end{aligned}$$

Les probabilités limites sont obtenues alors :

$$\begin{aligned} \Pr [\text{détection par COI}, @] &= \lim_{n \rightarrow \infty} b_n \\ \Pr [\text{détection par COI}, \neg @] &= \lim_{n \rightarrow \infty} e_n \\ \Pr [\text{détection par AE}, @] &= \lim_{n \rightarrow \infty} c_n \\ \Pr [\text{détection par AE}, \neg @] &= \lim_{n \rightarrow \infty} g_n \\ \Pr [\text{détection par PM}, @] &= \lim_{n \rightarrow \infty} d_n \\ \Pr [\text{détection par PM}, \neg @] &= \lim_{n \rightarrow \infty} h_n \end{aligned}$$

et l'on doit avoir :

$$\lim_{n \rightarrow \infty} b_n + \lim_{n \rightarrow \infty} c_n + \lim_{n \rightarrow \infty} d_n + \lim_{n \rightarrow \infty} e_n + \lim_{n \rightarrow \infty} g_n + \lim_{n \rightarrow \infty} h_n = 1$$

Ces limites s'obtiennent facilement par passage à la limite directement dans les systèmes récurrents.

On a ainsi :

$$\text{Pr [détection COI, } @] = \lim_{n \rightarrow \infty} b_n = \frac{b(1-L) + B1}{(a-1)(L-1) - A1} = \frac{p2}{1-p1-p'10}$$

$$\text{Pr [détection AE, } @] = \lim_{n \rightarrow \infty} c_n = \frac{c(1-L) + C1}{(a-1)(L-1) - A1} = \frac{p3}{1-p1-p'10}$$

$$\text{Pr [détection PM, } @] = \lim_{n \rightarrow \infty} d_n = \frac{d(1-L) + D1}{(a-1)(L-1) - A1} = \frac{p4}{1-p1-p'10}$$

$$\text{Pr [détection COI, } \neg @] = \lim_{n \rightarrow \infty} e_n = \frac{e(1-L) + E1}{(a-1)(L-1) - A1} = \frac{p6}{1-p1-p'10}$$

$$\text{Pr [détection AE, } \neg @] = \lim_{n \rightarrow \infty} g_n = \frac{g(1-L) + G1}{(a-1)(L-1) - A1} = \frac{p7}{1-p1-p'10}$$

$$\text{Pr [détection PM, } \neg @] = \lim_{n \rightarrow \infty} h_n = \frac{h(1-L) + H1}{(a-1)(L-1) - A1} = \frac{p8}{1-p1-p'10}$$

et on a bien :

$$\begin{aligned} & \lim_{n \rightarrow \infty} b_n + \lim_{n \rightarrow \infty} c_n + \lim_{n \rightarrow \infty} d_n + \lim_{n \rightarrow \infty} e_n + \lim_{n \rightarrow \infty} g_n + \lim_{n \rightarrow \infty} h_n \\ &= \frac{(b+c+d+e+g+h)(1-L) + (B+C+D+E+G+H)1}{(a-1)(L-1) - A1} \\ &= \frac{(1-a-1)(1-L) + (1-A-L)1}{(a-1)(L-1) - A1} \\ &= 1 \end{aligned}$$

- Moments du temps de détection :

Les premiers moments de détection s'obtiennent à partir de :

$$E [K_{COI}, @] = \sum_{k=1}^{\infty} k \Pr [K_{COI} = k, @] \frac{1}{\Pr [\text{détection COI}, @]}$$

Soit :

$$\begin{aligned} E [K_{COI}, @] &= k (b \Pr (1)^k + B \Pr (10)^k) \cdot \frac{1}{\lim_{n \rightarrow \infty} b_n} \\ &= [b \sum_1^{\infty} k a_k + B \sum_1^{\infty} k l_k] \cdot \frac{1}{\lim_{n \rightarrow \infty} b_n} \end{aligned}$$

On obtient de même :

$$E [K_{COI}, -@] = \frac{e}{\lim_{n \rightarrow \infty} e_n} \sum_1^{\infty} k a_k + \frac{E}{\lim_{n \rightarrow \infty} e_n} \sum_1^{\infty} k l_k$$

$$E [K_{AE}, @] = \frac{c}{\lim_{n \rightarrow \infty} c_n} \sum_1^{\infty} k a_k + \frac{C}{\lim_{n \rightarrow \infty} c_n} \sum_1^{\infty} k l_k$$

$$E [K_{AE}, -@] = \frac{g}{\lim_{n \rightarrow \infty} g_n} \sum_1^{\infty} k a_k + \frac{G}{\lim_{n \rightarrow \infty} g_n} \sum_1^{\infty} k l_k$$

$$E [K_{PM}, @] = \frac{d}{\lim_{n \rightarrow \infty} d_n} \sum_1^{\infty} k a_k + \frac{D}{\lim_{n \rightarrow \infty} d_n} \sum_1^{\infty} k l_k$$

$$E [K_{PM}, -@] = \frac{h}{\lim_{n \rightarrow \infty} h_n} \sum_1^{\infty} k a_k + \frac{H}{\lim_{n \rightarrow \infty} h_n} \sum_1^{\infty} k l_k$$

Les $\sum_1^{\infty} k a_k$ et $\sum_1^{\infty} k l_k$ s'obtiennent directement à partir des systèmes précédents :

$$X_k = \begin{bmatrix} a_k \\ A_k \end{bmatrix}$$

$$\begin{aligned} \sum k x_k &= \sum k J M'^{k-1} J^{-1} x_1 \\ &= J (\sum k M'^{k-1}) J^{-1} x_1 \end{aligned}$$

or $\sum k M'^{k-1} = [(I_2 - M')^2]^{-1}$

Et donc en posant :

$$\begin{aligned} U &= [1 \ 0] J [(I_2 - M')^2]^{-1} J^{-1} \begin{bmatrix} a \\ A \end{bmatrix} \\ \text{et } U' &= [1 \ 0] J [(I_2 - M')^2]^{-1} J^{-1} \begin{bmatrix} l \\ L \end{bmatrix} \end{aligned}$$

on obtient finalement :

$$E [K_{COI}, @] = \frac{bU + BU'}{\lim_{n \rightarrow \infty} b_n}$$

$$E [K_{COI}, -@] = \frac{eU + EU'}{\lim_{n \rightarrow \infty} e_n}$$

$$E [K_{AE}, @] = \frac{cU + CU'}{\lim_{n \rightarrow \infty} c_n}$$

$$E [K_{AE}, -@] = \frac{gU + GU'}{\lim_{n \rightarrow \infty} g_n}$$

$$E [K_{PM}, @] = \frac{dU + DU'}{\lim_{n \rightarrow \infty} d_n}$$

$$E [K_{PM}, -@] = \frac{hU + HU'}{\lim_{n \rightarrow \infty} h_n}$$

Les seconds moments de détection sont obtenus à partir de :

$$\begin{aligned} \sigma^2 [K_{COI}, @] &= \sum_1^{\infty} k^2 \Pr [K_{COI} = k, @] \cdot \frac{1}{\Pr [\text{détection COI}, @]} - E [K_{COI}, @]^2 \\ &= \sum_1^{\infty} k^2 (b \Pr(1)^k + B \Pr(10)^k) \cdot \frac{1}{\lim_{n \rightarrow \infty} b_n} - E [K_{COI}, @]^2 \\ &= [b \sum_1^{\infty} k^2 a_k + B \sum_1^{\infty} k l_k] \cdot \frac{1}{\lim_{n \rightarrow \infty} b_n} - E [K_{COI}, @]^2 \end{aligned}$$

2.2.2. Accès Direct

La matrice de transition P, pour le modèle de la figure 7 organisée selon les états absorbants non absorbants selon l'ordre des états :

2, 3, 4, 6, 7, 8, 1, 10 s'écrit :

$$P = \begin{bmatrix} I_6 & O \\ R & Q \end{bmatrix}$$

avec $R = \begin{bmatrix} p_2 & p_3 & p_4 & p_6 & p_7 & p_8 \\ fp_2 & fp_3 & pf_4 & pf_6 & fp_7 & fp_8 \end{bmatrix}$

et $Q = \begin{bmatrix} p_1 & p'_{10} \\ fp_1 & 1-f + fp'_{10} \end{bmatrix}$

- Probabilités d'état et Probabilités de détection :

La matrice fondamentale du système, vaut donc ici :

$$N = [I_2 - Q]^{-1} = \begin{bmatrix} 1-p_1 & -p'_{10} \\ -fp_1 & f(1-p'_{10}) \end{bmatrix}^{-1}$$

$$= \frac{1}{f(1-p_1-p'_{10})} \begin{bmatrix} f(1-p'_{10}) & p'_{10} \\ fp_1 & 1-p_1 \end{bmatrix}$$

On a alors :

$B = NR = \{b_{ij}\} = \text{Pr} [\text{le système est absorbé en } j \text{ en partant de } i]$

$\text{Pr} [\text{détection COI, } @] = \text{Pr} (2) = b_{11} = \frac{p_2}{1-p_1-p'_{10}}$

$\text{Pr} [\text{détection AE, } @] = \text{Pr} (3) = b_{12} = \frac{p_3}{1-p_1-p'_{10}}$

$\text{Pr} [\text{détection PM, } @] = \text{Pr} (4) = b_{13} = \frac{p_4}{1-p_1-p'_{10}}$

$\text{Pr} [\text{détection COI, } -@] = \text{Pr} (6) = b_{14} = \frac{p_6}{1-p_1-p'_{10}}$

$\text{Pr} [\text{détection AE, } -@] = \text{Pr} (7) = b_{15} = \frac{p_7}{1-p_1-p'_{10}}$

$\text{Pr} [\text{détection PM, } -@] = \text{Pr} (8) = b_{16} = \frac{p_8}{1-p_1-p'_{10}}$

- Moments du temps de détection :

De même, on obtient τ et τ_2 représentant respectivement les moyennes et variances du temps passé avant absorption en fonction de l'état de départ :

$$\tau = N \xi$$

$$\tau_2 = (2N - I_2) \tau - \tau_{sq}$$

où ξ est le vecteur colonne à éléments égaux à 1.

τ_{sq} est la matrice obtenue par élévation au carré de chaque élément de

$$\tau = \frac{1}{f(1-p_1-p'_1)} \begin{bmatrix} f(1-p'_1) + p'_1 \\ 1-p_1 + fp_1 \end{bmatrix}$$

$$\begin{aligned} \text{soit } \tau_{11} &= E [K_{COI}, @] + 1 = E [K_{AE}, @] + 1 = E [K_{PM}, @] + 1 \\ &= E [K_{COI}, -@] + 1 = E [K_{AE}, -@] + 1 = E [K_{PM}, -@] + 1 \\ &= E [K_{COI}] + 1 = E [K_{AE}] + 1 = E [K_{PM}] + 1 \end{aligned}$$

$$\rightarrow E [K_{COI}, @] = E [K_{COI}, -@] = E [K_{COI}] = \frac{fp_1 + p'_1}{f(1-p_1-p'_1)}$$

$$E [K_{AE}, @] = E [K_{AE}, -@] = E [K_{AE}] = \frac{fp_1 + p'_1}{f(1-p_1-p'_1)}$$

$$E [K_{PM}, @] = E [K_{PM}, -@] = E [K_{PM}] = \frac{fp_1 + p'_1}{f(1-p_1-p'_1)}$$

La variance est donnée par :

$$\tau_2 = \frac{1}{f^2(1-p_1-p'_1)^2} \left[\begin{aligned} & f^2 p_1 (1-p'_1) - fp'_1(1-p'_1-3p_1) - p'_1(2p_1-2 + p'_1) \\ & f^2 p_1 (1-p'_1) - f [(1-p_1)^2 - p'_1(1+p_1)] + (1-p_1)^2 \end{aligned} \right]$$

Les $\sum_1^{\infty} k^2 a_k$ et $\sum_1^{\infty} k^2 l_k$ s'obtiennent, comme précédemment directement dans les systèmes :

$$x_k = \begin{bmatrix} a_k \\ A_k \end{bmatrix}$$

$$\begin{aligned} \sum k^2 x_k &= \sum k^2 J M^{k-1} J^{-1} x_1 \\ &= J (\sum k^2 M^{k-1}) J^{-1} x_1 \end{aligned}$$

$$\text{or } \sum k^2 M^{k-1} = (I_2 + M') [(I_2 - M')^3]^{-1}$$

et donc en posant :

$$\begin{aligned} v &= [1 \ 0] J (I_2 + M') [(I_2 - M')^3]^{-1} J^{-1} \begin{bmatrix} a \\ A \end{bmatrix} \\ v' &= [1 \ 0] J (I_2 + M') [(I_2 - M')^3]^{-1} J^{-1} \begin{bmatrix} l \\ L \end{bmatrix} \end{aligned}$$

on obtient :

$$\sigma^2 [K_{COI}, @] = \frac{bv + BV'}{\lim_{n \rightarrow \infty} b_n} - E [K_{COI}, @]^2$$

$$\sigma^2 [K_{COI}, -@] = \frac{ev + EV'}{\lim_{n \rightarrow \infty} e_n} - E [K_{COI}, -@]^2$$

$$\sigma^2 [K_{AE}, @] = \frac{cv + CV'}{\lim_{n \rightarrow \infty} c_n} - E [K_{AE}, @]^2$$

$$\sigma^2 [K_{AE}, -@] = \frac{gv + GV'}{\lim_{n \rightarrow \infty} g_n} - E [K_{AE}, -@]^2$$

$$\sigma^2 [K_{PM}, @] = \frac{dv + DV'}{\lim_{n \rightarrow \infty} d_n} - E [K_{PM}, @]^2$$

$$\sigma^2 [K_{PM}, -@] = \frac{hv + HV'}{\lim_{n \rightarrow \infty} h_n} - E [K_{PM}, -@]^2$$

2.3. Récapitulation et discussion :

2.3.1. Récapitulation

En fonction des paramètres des modèles représentés par les figures 6 et 7, les caractéristiques dynamiques du temps de détection sont rassemblées dans les tableaux VIII et IX .

Il est aisé de vérifier que :

- si pour le modèle de la figure 6, on a :

$$p_1 = p_{11}, p_2 = p_{12}, p_3 = p_{13}, p_4 = p_{14}, p_6 = p_{16}, p_7 = p_{17}, p_8 = p_{18}, \\ p'_{10} = p'_{20}$$

toutes les expressions obtenues se ramènent à celles de la figure 7 .

- quand $f \rightarrow 1$:

- . Les probabilités de détection de la figure 6 tendent vers les probabilités de détection de la figure 7 .
- . Les espérances et variances des figures 6 et 7 tendent respectivement vers

$$\frac{p_1 + p'_{10}}{1 - p_1 - p'_{10}} \text{ et } \frac{p_1 + p'_{10}}{(1 - p_1 - p'_{10})^2} \text{ donc } E [K_{COI}] + 1 \text{ et } \sigma [K_{COI}] \text{ suivent des} \\ \text{lois géométriques de paramètres } (1 - p_1 - p'_{10})$$

- quand $f \rightarrow 0$:

- . Les espérances et variances, aussi bien pour le modèle de la figure 6 que pour celui de la figure 7 tendent vers l'infini, ce qui est normal.
- . On retombe sur les modèles statiques en s'empêchant d'entrer dans les états de non détection 10 et 20, ce qui a comme effet d'annuler les sorties possibles de ces états.

Pr [K _{COI} = 0]	p2		p6
Pr [K _{COI} = k]	*		*
Pr [K _{COI} ≤ k]	*		*
Pr [détection COI]	$\frac{(1-f)p_1 p_{12} p_2 (1-p_{11}+fp_{11})}{(1-p'_{10})(1-p_{11}+fp_{11})-p_1 (p'_{20}-fp'_{20}+f)}$		$\frac{(1-f)p_1 p_{16} p_6 (1-p_{11}+fp_{11})}{(1-p'_{10})(1-p_{11}+fp_{11})-p_1 (p'_{20}-fp'_{20}+f)}$
E [K _{COI}]	$\frac{f(p_{11} p'_{10} + p_1 - p_1 p'_{10}) + p'_{10} - p_{11} p'_{10} + p_1 p'_{20}}{f [(1-p_{11}+fp_{11})(1-p'_{10}) - p_1 (p'_{20}+f(1-p'_{20}))]}$		$\frac{f(p_{11} p'_{10} + p_1 - p_1 p'_{10}) + p'_{10} - p_{11} p'_{10} + p_1 p'_{20}}{f [(1-p_{11}+fp_{11})(1-p'_{10}) - p_1 (p'_{20}+f(1-p'_{20}))]}$
σ ² [K _{COI}]	cf. § IV. 2.1.2.		cf § IV.2.1.2.
Pr [K _{AE} = 0]	p3		p7
Pr [K _{AE} = k]	*		*
Pr [K _{AE} ≤ k]	*		*
Pr [détection AE]	$\frac{(1-f)p_1 p_{13} + p_3 (1-p_{11}+fp_{11})}{(1-p'_{10})(1-p_{11}+fp_{11})-p_1 (p'_{20}-fp'_{20}+f)}$		$\frac{(1-f) p_1 p_{17} + p_7 (1-p_{11} + fp_{11})}{(1-p'_{10})(1-p_{11}+fp_{11})-p_1 (p'_{20}-fp'_{20}+f)}$
E [K _{AE}]	E [K _{COI}]		E [K _{COI}]
σ ² [K _{AE}]	σ ² [K _{COI}]		σ ² [K _{COI}]
Pr [K _{PM} = 0]	p4		p8
Pr [K _{PM} = k]	*		*
Pr [K _{PM} ≤ k]	*		*
Pr [détection PM]	$\frac{(1-f)p_1 p_{14} + p_4 (1-p_{11}+fp_{11})}{(1-p'_{10})(1-p_{11}+fp_{11})-p_1 (p'_{20}-fp'_{20}+f)}$		$\frac{(1-f)p_1 p_{18} + p_8 (1-p_{11}+fp_{11})}{(1-p'_{10})(1-p_{11}+fp_{11})-p_1 (p'_{20}-fp'_{20}+f)}$
E [K _{PM}]	E [K _{COI}]		E [K _{COI}]
σ ² [K _{PM}]	σ ² [K _{COI}]		σ ² [K _{COI}]
(a)		(b)	

* non calculés

Tableau VIII - Valeurs caractéristiques sur les bornes d'adresse (a) et sur les bornes de 'non adresse' (b) pour le modèle de la figure 6 .

Pr [K _{COI} = 0]	p2		p6
Pr [K _{COI} = k]	*		*
Pr [K _{COI} ≤ k]	*		*
Pr [détection COI]	$\frac{(1-f)p_1 p_{12} p_2 (1-p_{11}+fp_{11})}{(1-p'_{10})(1-p_{11}+fp_{11})-p_1(p'_{20}-fp'_{20}+f)}$		$\frac{(1-f)p_1 p_{16} p_6 (1-p_{11}+fp_{11})}{(1-p'_{10})(1-p_{11}+fp_{11})-p_1(p'_{20}-fp'_{20}+f)}$
E [K _{COI}]	$\frac{f(p_{11} p'_{10} + p_1 - p_1 p'_{10}) + p'_{10} - p_{11} p'_{10} + p_1 p'_{20}}{f [(1-p_{11}+fp_{11})(1-p'_{10}) - p_1(p'_{20}+f(1-p'_{20}))]}$		$\frac{f(p_{11} p'_{10} + p_1 - p_1 p'_{10}) + p'_{10} - p_{11} p'_{10} + p_1 p'_{20}}{f [(1-p_{11}+fp_{11})(1-p'_{10}) - p_1(p'_{20}+f(1-p'_{20}))]}$
σ ² [K _{COI}]	cf. § IV. 2.1.2.		cf § IV.2.1.2.
Pr [K _{AE} = 0]	p3		p7
Pr [K _{AE} = k]	*		*
Pr [K _{AE} ≤ k]	*		*
Pr [détection AE]	$\frac{(1-f)p_1 p_{13} + p_3 (1-p_{11}+fp_{11})}{(1-p'_{10})(1-p_{11}+fp_{11})-p_1(p'_{20}-fp'_{20}+f)}$		$\frac{(1-f) p_1 p_{17} + p_7 (1-p_{11} + fp_{11})}{(1-p'_{10})(1-p_{11}+fp_{11})-p_1(p'_{20}-fp'_{20}+f)}$
E [K _{AE}]	E [K _{COI}]		E [K _{COI}]
σ ² [K _{AE}]	σ ² [K _{COI}]		σ ² [K _{COI}]
Pr [K _{PM} = 0]	p4		p8
Pr [K _{PM} = k]	*		*
Pr [K _{PM} ≤ k]	*		*
Pr [détection PM]	$\frac{(1-f)p_1 p_{14} + p_4 (1-p_{11}+fp_{11})}{(1-p'_{10})(1-p_{11}+fp_{11})-p_1(p'_{20}-fp'_{20}+f)}$		$\frac{(1-f) p_1 p_{18} + p_8 (1-p_{11} + fp_{11})}{(1-p'_{10})(1-p_{11}+fp_{11})-p_1(p'_{20}-fp'_{20}+f)}$
E [K _{PM}]	E [K _{COI}]		E [K _{COI}]
σ ² [K _{PM}]	σ ² [K _{COI}]		σ ² [K _{COI}]
(a)		(b)	

* non calculés

Tableau VIII - Valeurs caractéristiques sur les bornes d'adresse (a) et sur les bornes de 'non adresse' (b) pour le modèle de la figure 6 .

Les $\sum_1^{\infty} k^2 a_k$ et $\sum_1^{\infty} k^2 l_k$ s'obtiennent, comme précédemment directement dans les systèmes :

$$x_k = \begin{bmatrix} a_k \\ A_k \end{bmatrix}$$

$$\begin{aligned} \sum k^2 x_k &= \sum k^2 J M^{k-1} J^{-1} x_1 \\ &= J (\sum k^2 M^{k-1}) J^{-1} x_1 \end{aligned}$$

$$\text{or } \sum k^2 M^{k-1} = (I_2 + M') [(I_2 - M')^3]^{-1}$$

et donc en posant :

$$\begin{aligned} v &= [1 \ 0] J (I_2 + M') [(I_2 - M')^3]^{-1} J^{-1} \begin{bmatrix} a \\ A \end{bmatrix} \\ v' &= [1 \ 0] J (I_2 + M') [(I_2 - M')^3]^{-1} J^{-1} \begin{bmatrix} l \\ L \end{bmatrix} \end{aligned}$$

on obtient :

$$\sigma^2 [K_{COI}, @] = \frac{bv + BV'}{\lim_{n \rightarrow \infty} b_n} - E [K_{COI}, @]^2$$

$$\sigma^2 [K_{COI}, -@] = \frac{ev + EV'}{\lim_{n \rightarrow \infty} e_n} - E [K_{COI}, -@]^2$$

$$\sigma^2 [K_{AE}, @] = \frac{cv + CV'}{\lim_{n \rightarrow \infty} c_n} - E [K_{AE}, @]^2$$

$$\sigma^2 [K_{AE}, -@] = \frac{gv + GV'}{\lim_{n \rightarrow \infty} g_n} - E [K_{AE}, -@]^2$$

$$\sigma^2 [K_{PM}, @] = \frac{dv + DV'}{\lim_{n \rightarrow \infty} d_n} - E [K_{PM}, @]^2$$

$$\sigma^2 [K_{PM}, -@] = \frac{hv + HV'}{\lim_{n \rightarrow \infty} h_n} - E [K_{PM}, -@]^2$$

2.2.2. Accès Direct

La matrice de transition P, pour le modèle de la figure 7 organisée selon les états absorbants non absorbants selon l'ordre des états :

2, 3, 4, 6, 7, 8, 1, 10 s'écrit :

$$P = \begin{bmatrix} I_6 & O \\ R & Q \end{bmatrix}$$

avec $R = \begin{bmatrix} p_2 & p_3 & p_4 & p_6 & p_7 & p_8 \\ fp_2 & fp_3 & pf_4 & pf_6 & fp_7 & fp_8 \end{bmatrix}$

et $Q = \begin{bmatrix} p_1 & p'10 \\ fp_1 & 1-f + fp'10 \end{bmatrix}$

- Probabilités d'état et Probabilités de détection :

La matrice fondamentale du système, vaut donc ici :

$$N = [I_2 - Q]^{-1} = \begin{bmatrix} 1-p_1 & -p'10 \\ -fp_1 & f(1-p'10) \end{bmatrix}^{-1}$$

$$= \frac{1}{f(1-p_1-p'10)} \begin{bmatrix} f(1-p'10) & p'10 \\ fp_1 & 1-p_1 \end{bmatrix}$$

On a alors :

$B = NR = \{b_{ij}\} = \text{Pr} [\text{le système est absorbé en } j \text{ en partant de } i]$

$\text{Pr} [\text{détection COI, } @] = \text{Pr} (2) = b_{11} = \frac{p_2}{1-p_1-p'10}$

$\text{Pr} [\text{détection AE, } @] = \text{Pr} (3) = b_{12} = \frac{p_3}{1-p_1-p'10}$

$\text{Pr} [\text{détection PM, } @] = \text{Pr} (4) = b_{13} = \frac{p_4}{1-p_1-p'10}$

$\text{Pr} [\text{détection COI, } -@] = \text{Pr} (6) = b_{14} = \frac{p_6}{1-p_1-p'10}$

$\text{Pr} [\text{détection AE, } -@] = \text{Pr} (7) = b_{15} = \frac{p_7}{1-p_1-p'10}$

$\text{Pr} [\text{détection PM, } -@] = \text{Pr} (8) = b_{16} = \frac{p_8}{1-p_1-p'10}$

- Moments du temps de détection :

De même, on obtient τ et τ_2 représentant respectivement les moyennes et variances du temps passé avant absorption en fonction de l'état de départ :

$$\tau = N \xi$$

$$\tau_2 = (2N - I_2) \tau - \tau_{sq}$$

où ξ est le vecteur colonne à éléments égaux à 1.

τ_{sq} est la matrice obtenue par élévation au carré de chaque élément de

$$\tau = \frac{1}{f(1-p_1-p'_{10})} \begin{bmatrix} f(1-p'_{10}) + p'_{10} \\ 1-p_1 + fp_1 \end{bmatrix}$$

$$\begin{aligned} \text{soit } \tau_{11} &= E [K_{COI}, @] + 1 = E [K_{AE}, @] + 1 = E [K_{PM}, @] + 1 \\ &= E [K_{COI}, -@] + 1 = E [K_{AE}, -@] + 1 = E [K_{PM}, -@] + 1 \\ &= E [K_{COI}] + 1 = E [K_{AE}] + 1 = E [K_{PM}] + 1 \end{aligned}$$

$$\rightarrow E [K_{COI}, @] = E [K_{COI}, -@] = E [K_{COI}] = \frac{fp_1 + p'_{10}}{f(1-p_1-p'_{10})}$$

$$E [K_{AE}, @] = E [K_{AE}, -@] = E [K_{AE}] = \frac{fp_1 + p'_{10}}{f(1-p_1-p'_{10})}$$

$$E [K_{PM}, @] = E [K_{PM}, -@] = E [K_{PM}] = \frac{fp_1 + p'_{10}}{f(1-p_1-p'_{10})}$$

La variance est donnée par :

$$\tau_2 = \frac{1}{f^2(1-p_1-p'_{10})^2} \begin{bmatrix} f^2 p_1 (1-p'_{10}) - fp'_{10}(1-p'_{10}-3p_1) - p'_{10} (2p_1-2 + p'_{10}) \\ f^2 p_1 (1-p'_{10}) - f [(1-p_1)^2 - p'_{10} (1+p_1)] + (1-p_1)^2 \end{bmatrix}$$

2.3. Récapitulation et discussion :

2.3.1. Récapitulation

En fonction des paramètres des modèles représentés par les figures 6 et 7, les caractéristiques dynamiques du temps de détection sont rassemblées dans les tableaux VIII et IX .

Il est aisé de vérifier que :

- si pour le modèle de la figure 6, on a :

$$p_1 = p_{11}, p_2 = p_{12}, p_3 = p_{13}, p_4 = p_{14}, p_6 = p_{16}, p_7 = p_{17}, p_8 = p_{18}, \\ p'_{10} = p'_{20}$$

toutes les expressions obtenues se ramènent à celles de la figure 7 .

- quand $f \rightarrow 1$:

. Les probabilités de détection de la figure 6 tendent vers les probabilités de détection de la figure 7 .

. Les espérances et variances des figures 6 et 7 tendent respectivement vers

$$\frac{p_1 + p'_{10}}{1 - p_1 - p'_{10}} \quad \text{et} \quad \frac{p_1 + p'_{10}}{(1 - p_1 - p'_{10})^2} \quad \text{donc } E[K_{COI}] + 1 \text{ et } \sigma[K_{COI}] \text{ suivent des} \\ \text{lois géométriques de paramètres } (1 - p_1 - p'_{10})$$

- quand $f \rightarrow 0$:

. Les espérances et variances, aussi bien pour le modèle de la figure 6 que pour celui de la figure 7 tendent vers l'infini, ce qui est normal.

. On retombe sur les modèles statiques en s'empêchant d'entrer dans les états de non détection 10 et 20, ce qui a comme effet d'annuler les sorties possibles de ces états.

Pr [K _{COI} = 0]	p2		p6
Pr [K _{COI} = k]	*		*
Pr [K _{COI} ≤ k]	*		*
Pr [détection COI]	$\frac{(1-f)p_1 p_{12} + p_2 (1-p_{11} + fp_{11})}{(1-p'10) (1-p_{11} + fp_{11}) - p_1 (p'20 - fp'20 + f)}$		$\frac{(1-f)p_1 p_{16} + p_6 (1-p_{11} + fp_{11})}{(1-p'10) (1-p_{11} + fp_{11}) - p_1 (p'20 - fp'20 + f)}$
E [K _{COI}]	$\frac{f(p_{11} p'10 + p_1 - p_1 p'20) + p'10 - p_{11} p'10 + p_1 p'20}{f [(1-p_{11} + fp_{11}) (1-p'10) - p_1 (p'20 + f (1-p'20))]}$		$\frac{f(p_{11} p'10 + p_1 - p_1 p'20) + p'10 - p_{11} p'10 + p_1 p'20}{f [(1-p_{11} + fp_{11}) (1-p'10) - p_1 (p'20 + f (1-p'20))]}$
σ ² [K _{COI}]	cf. § IV. 2.1.2.		cf § IV.2.1.2.
Pr [K _{AE} = 0]	p3		p7
Pr [K _{AE} = k]	*		*
Pr [K _{AE} ≤ k]	*		*
Pr [détection AE]	$\frac{(1-f)p_1 p_{13} + p_3 (1-p_{11} + fp_{11})}{(1-p'10) (1-p_{11} + fp_{11}) - p_1 (p'20 - fp'20 + f)}$		$\frac{(1-f) p_1 p_{17} + p_7 (1-p_{11} + fp_{11})}{(1-p'10) (1-p_{11} + fp_{11}) - p_1 (p'20 - fp'20 + f)}$
E [K _{AE}]	E [K _{COI}]		E [K _{COI}]
σ ² [K _{AE}]	σ ² [K _{COI}]		σ ² [K _{COI}]
Pr [K _{PM} = 0]	p4		p8
Pr [K _{PM} = k]	*		*
Pr [K _{PM} ≤ k]	*		*
Pr [détection PM]	$\frac{(1-f)p_1 p_{14} + p_4 (1-p_{11} + fp_{11})}{(1-p'10) (1-p_{11} + fp_{11}) - p_1 (p'20 - fp'20 + f)}$		$\frac{(1-f)p_1 p_{18} + p_8 (1-p_{11} + fp_{11})}{(1-p'10) (1-p_{11} + fp_{11}) - p_1 (p'20 - fp'20 + f)}$
E [K _{PM}]	E [K _{COI}]		E [K _{COI}]
σ ² [K _{PM}]	σ ² [K _{COI}]		σ ² [K _{COI}]
		(a)	(b)

* non calculés

Tableau VIII - Valeurs caractéristiques sur les bornes d'adresse (a) et sur les bornes de non adresse' (b) pour le modèle de la figure 6 .

Pr [$K_{COI} = 0$]	p_2	p_6
Pr [$K_{COI} = k$]	$p_2 a_k + fp_2 l_k$	$p_6 a_k + fp_6 l_k$
Pr [$K_{COI} \leq k$]	$b_{k+1} \frac{p_2}{1-p_1-p'10}$	$e_{k+1} \frac{p_6}{1-p_1-p'10}$
Pr [détection COI]	$\frac{fp_1 + p'10}{f(1-p_1-p'10)}$	$\frac{fp_1 + p'10}{f(1-p_1-p'10)}$
$E [K_{COI}]$	$\frac{f^2 p_1(1-p'10) - fp'10(1-p'10-3p_1) - p'10(2p_1-2+p'10)}{f^2(1-p_1-p'10)}$	$\frac{f^2 p_1(1-p'10) - fp'10(1-p'10-3p_1) - p'10(2p_1-2+p'10)}{f^2(1-p_1-p'10)}$
$f^2 [K_{COI}]$		
Pr [$K_{AE} = 0$]	p_3	p_7
Pr [$K_{AE} = k$]	$p_3 a_k + fp_3 l_k$	$p_7 a_k + fp_7 l_k$
Pr [$K_{AE} \leq k$]	$c_{k+1} \frac{p_3}{1-p_1-p'10}$	$g_{k+1} \frac{p_7}{1-p_1-p'10}$
Pr [détection AE]	$\frac{fp_1+p'10}{f(1-p_1-p'10)}$	$\frac{fp_1 + p'10}{f(1-p_1-p'10)}$
$E [K_{AE}]$	$\frac{f^2 p_1(1-p'10) - fp'10(1-p'10-3p_1) - p'10(2p_1-2+p'10)}{f^2(1-p_1-p'10)}$	$\frac{f^2 p_1(1-p'10) - fp'10(1-p'10-3p_1) - p'10(2p_1-2+p'10)}{f^2(1-p_1-p'10)}$
$f^2 [K_{AE}]$		
Pr [$K_{PM} = 0$]	p_4	p_8
Pr [$K_{PM} = k$]	$p_4 a_k + fp_4 l_k$	$p_8 a_k + fp_8 l_k$
Pr [$K_{PM} \leq k$]	$d_{k+1} \frac{p_4}{1-p_1-p'10}$	$h_{k+1} \frac{p_8}{1-p_1-p'10}$
Pr [détection PM]	$\frac{fp_1 + p'10}{f(1-p_1-p'10)}$	$\frac{fp_1 + p'10}{f(1-p_1-p'10)}$
$E [K_{PM}]$	$\frac{f^2(1-p'10) - fp'10(1-p'10-3p_1) - p'10(2p_1-2+p'10)}{f^2(1-p_1-p'10)}$	$\frac{f^2 p_1(1-p'10) - fp'10(1-p'10-3p_1) - p'10(2p_1-2+p'10)}{f^2(1-p_1-p'10)}$
$f^2 [K_{PM}]$		

ec : $\begin{bmatrix} a_{k+1} \\ A_{k+1} \end{bmatrix} = JM^k J^{-1} \begin{bmatrix} p_1 \\ fp_1 \end{bmatrix}, \begin{bmatrix} l_{k+1} \\ L_{k+1} \end{bmatrix} = JM^k J^{-1} \begin{bmatrix} p'10 \\ 1-f(1-p'10) \end{bmatrix}, J = \begin{bmatrix} \lambda_1^{-a} & 1 \\ 1 & \lambda_2^{-a} \end{bmatrix}, J^{-1} = \frac{1}{1(\lambda_2^{-a} - \lambda_1^{-a})} \begin{bmatrix} \lambda_2^{-a} & -1 \\ a - \lambda_1 & 1 \end{bmatrix}, M' = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$

λ_1, λ_2 solutions de $\lambda^2 - (a+L)\lambda + aL - A1 = 0$ les $b_{k+1}, c_{k+1}, d_{k+1}, e_{k+1}, g_{k+1}, h_{k+1}$ se trouvent au paragraphe S2.2.1.1.

Tableau IX - Valeurs caractéristiques sur les bornes d'adresse (a) et sur les bornes de "non adresse" (b) pour le modèle de la figure 7.

Pour les modèles des figures 6 et 7, les moments de K_{COI} , K_{AE} , et K_{PM} représentent le temps de détection (au contraire des figures 2 et 4). Ce temps de détection, pour une application et un processeur donnés, est fonction de la fréquence $1/f$ de génération de l'événement E. Une valeur caractéristique de la fréquence est celle pour laquelle, si cette valeur existe la moyenne du temps de détection est inférieure à la fréquence $1/f$.

2.3.2. Valeur caractéristique de la fréquence de E. :

- Modèle de la figure 6

L'expression de la moyenne du temps de détection pour le modèle de la figure 6 est donné par :

$$E [K] = \frac{f(p_{11} p'_{10} + p_1 - p_1 p'_{20}) + p'_{10} - p_{11} p'_{10} + p_1 p'_{20}}{f [(1-p_{11} + fp_{11})(1-p'_{10}) - p_1(p'_{20} + f(1-p'_{20}))]}$$

La valeur caractéristique de la fréquence est donnée pour $E [K] = \frac{1}{f}$ par :

$\frac{1}{f} = \frac{p_1}{(1-p_{11})(1-2p'_{10}) - 2p_1 p'_{20}}$

- Modèle de la figure 7

Pour le modèle simplifié de la figure 7, on a :

$$E [K] = \frac{fp_1 + p'_{10}}{f(1-p_1-p'_{10})}$$

En fonction de f , $E [K]$ est donc représenté par un segment d'hyperbole. On préférera considérer $E [K]$ en fonction de $1/f$ (production d'un événement E toutes les $1/f$ instructions).

$E [K]$ est linéaire en $1/f$ selon la relation :

$$E [K] = \frac{p'_{10}}{1-p_1-p'_{10}} \frac{1}{f} + \frac{p_1}{1-p_1-p'_{10}}$$

Pour les grandes valeurs de $1/f$, $E[K]$ tend donc vers la valeur $\frac{p'10}{1-p1-p'10}$.
On sera néanmoins prudent dans l'utilisation de cette valeur, car les hypothèses faites sont que la fréquence $1/f$ est petite par rapport à la période T_0 d'un chien de garde.

La valeur caractéristique de la fréquence à partir de laquelle on peut borner $E[K]$ par la dite fréquence $1/f$ est telle que :

$$E[K] = \frac{1}{f} \quad \text{soit :}$$

$$\frac{1}{f} = \frac{p1}{1-p1-2p'10}$$

On vérifie aisément que la valeur obtenue pour le modèle de la figure 6 donne la valeur obtenue pour le modèle de la figure 7, en faisant $p_{11} = p_1$ et $p'10 = p'20$.

V.- DYNAMISATION DES RESULTATS EXPERIMENTAUX

Comme pour le M6800, les résultats expérimentaux statiques obtenus pour une application sur le M68000 vont permettre d'évaluer, sans procéder à des simulations, les caractéristiques du temps de détection en fonction de la fréquence $1/f$ de l'événement E.

Compte-tenu des valeurs des coefficients déterminés au paragraphe III, le tableau X peut-être donné. Les expressions les plus complexes ne sont pas reprises.

On se gardera bien de vouloir rapprocher directement les probabilités de détection, des résultats des modèles statiques (figures 2 et 4), lorsque f tend vers zéro, les écarts provenant du passage par les états 10 et/ou 20. Le rapprochement ne peut être fait que pour des valeurs relatives les unes par rapport aux autres, pour les modèles des figures 2 et 4. Le point important est de considérer quelques valeurs numériques. Ainsi, selon le modèle simplifié de la figure 7, pour une fréquence de l'événement E de $1/f = 100$, le premier moment du temps de détection est égal à 199 pour un écart-type de 275. Mais le modèle 7 reflète moins bien la réalité que celui de la fig. 6. Comparativement, la moyenne donnée par le modèle de la figure 6 est égale à 202 et l'écart type à 282.

Pr [détection COI, @]	0	0
Pr [détection COI, -@]	$\frac{0.14f + 0.014}{0.188f + 0.026}$	0.518
Pr [détection AE, @]	$\frac{-0.002f + 0.002}{0.188f + 0.026}$	0.054
Pr [détection AE, -@]	$\frac{0.048f + 0.005}{0.188f + 0.026}$	0.179
Pr [détection PM, @]	$\frac{-0.005f + 0.005}{0.188f + 0.026}$	0.179
Pr [détection PM, -@]	$\frac{-0.002f + 0.002}{0.188f + 0.026}$	0.054
E [K _{COI}] E [K _{AE}] E [K _{PM}]	$\frac{0.737f + 0.049}{0.188f^2 + 0.026f}$	$\frac{0.841f + 0.103}{0.056 f}$
σ^2 [K _{COI}] σ^2 [K _{AE}] σ^2 [K _{PM}]	$\frac{-0.139f^3 + 0.813f^2 + 0.108f + 0.005}{(0.188f^2 + 0.026f)^2}$	$\frac{0.754f^2 + 0.167f + 0.022}{(0.056f)^2}$
	(a)	(b)

Tableau X - Caractéristiques dynamiques du temps de détection pour l'application expérimentale considérée sur le MC 68000 pour le modèle de la figure 6 (a) et pour le modèle de la figure 7 (b).

Il est aisé de vérifier sur le tableau X que si $f = 1$, $E [K_{COI}] + 1$ et $\sigma^2 [K_{COI}]$ correspondent bien aux moments de lois géométriques aussi bien pour le modèle de la figure 6 que pour celui de la figure 7 ; les paramètres de réussite de ces lois géométriques étant : 0.214 pour la figure 6 et 0.056 pour la figure 7 .

Le modèle de la figure 4, peut donc être utilisé pour approcher la valeur de la moyenne ainsi que l'écart-type, mais son emploi sera délicat en ce qui concerne les faibles fréquences.

La valeur caractéristique de $1/f$ à partir de laquelle on peut borner la moyenne du temps de détection par $1/f$ n'existe pas pour l'application considérée, que ce soit pour le modèle de la figure 6 ou pour celui de la figure 7 qui n'indiquent que des valeurs négatives.

Le tableau XI donne les valeurs limites, pour $f = 1$ et $f \rightarrow 0$, pour les deux modèles, en fonction de la fréquence $1/f$.

$f = 1$	
$E [K_{COI}] = 3.67$ $\sigma [K_{COI}] = 4.15$ $\frac{\sigma [K_{COI}]}{E [K_{COI}]} = 1.13$	$E [K_{COI}] = 16.86$ $\sigma [K_{COI}] = 17.34$ $\frac{\sigma [K_{COI}]}{E [K_{COI}]} = 1.03$
$f \rightarrow 0$	
$E [K_{COI}] \rightarrow 1.88 \frac{1}{f}$ $\sigma [K_{COI}] \rightarrow 2.72 \frac{1}{f}$ $\frac{\sigma [K_{COI}]}{E [K_{COI}]} \rightarrow 1.44$	$E [K_{COI}] \rightarrow 1.84 \frac{1}{f}$ $\sigma [K_{COI}] \rightarrow 2.65 \frac{1}{f}$ $\frac{\sigma [K_{COI}]}{E [K_{COI}]} \rightarrow 1.44$
(a)	(b)

Tableau XI - Valeurs limites des moments de détection pour le modèle de la figure 6 (a) et pour celui de la figure 7 (b), pour l'application expérimentale considérée.

VI.- GENERALISATION A UNE APPLICATION QUELCONQUE

Les résultats expérimentaux, présentés précédemment, concernent une application particulière et les paramètres déterminés peuvent évidemment ne pas correspondre à une application quelconque. Une application quelconque peut ne pas posséder de protection mémoire si tout l'espace est occupé par de la RAM. Ainsi que déjà mentionné, la valeur 0.161 du paramètre p6 de la figure 2 correspondant à la détection immédiate par COI, peut paraître exagérée. Par contre, la valeur de p16 correspond à une application quelconque. Ainsi que déterminé au paragraphe III, une valeur de 0.05 pour p6 est plus réaliste.

On peut remarquer encore que le paramètre p3 caractérisant la détection immédiate par une adresse erreur devrait être plus importante pour une application quelconque ; conserver cette valeur correspond donc à un résultat pessimiste par rapport à ce qu'il devrait être. On obtient alors les valeurs des paramètres (figure 6) :

p1 = 0.868	p6 = 0.050	p11 = 0.933	p16 = 0.002
p2 = 0	p7 = 0.053	p12 = 0	p17 = 0.001
p3 = 0	p8 = 0	p13 = 0.002	p18 = 0
p4 = 0	p9 = 0	p14 = 0.028	p19 = 0.005
p5 = 0	p10 = 0.029	p15 = 0.028	p20 = 0.029

En ce qui concerne la figure 7, il faut extraire les cas de détection lorsque $K_{COI} = 0$, des résultats expérimentaux, pour recalculer les nouveaux résultats "expérimentaux" en conséquence, toujours en admettant une probabilité 0.05 et non 0.161 de détection avec $K_{COI} = 0$. Il est ainsi possible de déterminer la moyenne du temps de détection $E [K_{COI}]$:

$$E [K_{COI}] = 13.4 * Pr [K_{COI} > 0]$$

On va donc recalculer la probabilité de détection par COI à partir du modèle 2 avec les coefficients d'une application quelconque, on obtient alors :

$$\begin{aligned} \text{Pr [détection COI]} &= 0.050 + \frac{0.868 * 0.003}{(1-0.933)} \\ &= 0.089 \end{aligned}$$

$$\text{On a donc Pr } [K_{\text{COI}} = 0] = \frac{0.050}{0.089} = 0.563$$

$$\text{et par suite Pr } [K_{\text{COI}} > 0] = 0.437$$

$$\begin{aligned} \text{et donc E } [K_{\text{COI}}] &= 13.4 * 0.437 \\ &= 5.9 \end{aligned}$$

On obtient alors les coefficients suivants pour la figure 7 et une application quelconque :

$$\begin{array}{ll} p_2 = 0 & p_6 = 0.013 \\ p_3 = 0.003 & p_7 = 0.010 \\ p_4 = 0 & p_8 = 0 \end{array}$$

$$p'_{10} = 0.120$$

$$p_{11} = 0.854$$

1.- Distribution et densité du temps de détection

Ces paramètres sont déterminés à partir du modèle simplifié de la figure 7 car ils seraient trop complexes à déterminer pour le modèle de la figure 6. A partir des résultats du tableau IX, on obtient :

$$\text{Pr} [K_{\text{COI}} < k] = b_{k+1} + e_{k+1}$$

$$\text{Pr} [K_{\text{COI}} = k] = (p_2+p_6) a_k + f (p_2+p_6) l_k$$

$$\text{Pr} [K_{\text{AE}} < k] = c_{k+1} + g_{k+1}$$

$$\text{Pr} [K_{\text{AE}} = k] = (p_3+p_7) a_k + f (p_3+p_7) l_k$$

$$\text{Pr} [K_{\text{COI} + \text{AE}} \leq k] = b_{k+1} + e_{k+1} + c_{k+1} + g_{k+1}$$

$$\text{Pr} [K_{\text{COI} + \text{AE}} = k] = (p_2+p_3+p_6+p_7) a_k + f (p_2+p_3+p_6+p_7) l_k$$

Les valeurs propres λ_1 et λ_2 sont solutions de :

$$\lambda^2 - [1.854 - 0.880 f] \lambda + 0.854 - 0.854 f = 0$$

Ces valeurs propres vont être calculées pour les cas limites ($f = 1$ et $f \rightarrow 0$), ce qui donne :

1.1. Valeurs caractéristiques pour $f = 1$:

Si $f = 1$ λ_1 et λ_2 sont solutions de

$$\lambda^2 - 0.974 \lambda = 0 \quad \rightarrow \quad \begin{cases} \lambda_1 = 0 \\ \lambda_2 = 0.974 \end{cases}$$

Ce qui nous donne :

$$J = \begin{bmatrix} 0.120 & 0.120 \\ -0.854 & 0.120 \end{bmatrix}$$

et donc :

$$J^{-1} = \begin{bmatrix} 1.027 & -1.027 \\ 7.307 & 1.027 \end{bmatrix}$$

$$M' = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$\text{donc } JM^k J^{-1} = \begin{bmatrix} 0.123 \lambda_1^k + 0.877 \lambda_2^k & -0.123 \lambda_1^k + 0.123 \lambda_2^k \\ -0.877 \lambda_1^k + 0.877 \lambda_2^k & 0.877 \lambda_1^k + 0.123 \lambda_2^k \end{bmatrix}$$

$$\rightarrow a_{k+1} = (0.123 \lambda_1^k + 0.877 \lambda_2^k) * 0.854 + (-0.123 \lambda_1^k + 0.123 \lambda_2^k) * 0.854$$

$$a_{k+1} = 0.85 \lambda_2^k$$

les $b_{k+1} \dots h_{k+1}$ sont tirés des formules du § IV.2.2.1.

$$b_{k+1} = 0$$

$$c_{k+1} = -0.112 \lambda_2^k + 0.115$$

$$d_{k+1} = 0$$

$$e_{k+1} = -0.487 \lambda_2^k + 0.500$$

$$g_{k+1} = -0.375 \lambda_2^k + 0.385$$

$$h_{k+1} = 0$$

$$l_{k+1} = (0.123 \lambda_1^k + 0.877 \lambda_2^k) * 0.120 + (-0.123 \lambda_1^k + 0.123 \lambda_2^k) * 0.120$$

$$l_{k+1} = 0.120 \lambda_2^k$$

On vérifie aisément que les probabilités correspondent à la réalité pour $k = 0$.

On peut maintenant calculer la distribution :

$$\Pr [K_{COI} + AE \leq k] = 1 - 0.974 \lambda_2^k$$

$$\text{On a bien } \Pr [K_{COI} + AE \leq 0] = 0.026$$

On obtient alors facilement la densité :

$$\Pr [K_{COI} + AE = k] = 0.025 \lambda_2^{k-1}$$

$$\text{Ce qui donne bien } \Pr [K_{COI} + AE = 0] = 0.026$$

1.2. - Valeurs caractéristiques pour $f \rightarrow 0$:

Pour f tendant vers 0, λ_1 et λ_2 sont solutions de :

$$\lambda^2 - 1.854 \lambda + 0.854 = 0$$

Soit $\lambda_1 = 1$ et $\lambda_2 = 0.854$

Ce qui donne :

$$J = \begin{bmatrix} 0.120 & 0.120 \\ 0.146 & 0 \end{bmatrix}$$

et

$$J^{-1} = \begin{bmatrix} 0 & 6.849 \\ 8.333 & -6.849 \end{bmatrix}$$

$$M' = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$JM'^k J^{-1} = \begin{bmatrix} \lambda_2^k & 0.822 \lambda_1^k - 0.822 \lambda_2^k \\ 0 & \lambda_1^k \end{bmatrix}$$

Ce qui donne les coefficients :

$$a_{k+1} = 0.854 \lambda_2^k$$

$$l_{k+1} = 0.822 \lambda_1^k - 0.702 \lambda_2^k$$

les b_{k+1} , c_{k+1} h_{k+1} , s'obtiennent directement sur la matrice de départ modifiée par la valeur de $f = 0$, ce qui donne :

$$b_{k+1} = 0$$

$$c_{k+1} = 0.021 - 0.018 \lambda_2^k$$

$$d_{k+1} = 0$$

$$l_{k+1} = 0.089 - 0.067 \lambda_2^k$$

$$g_{k+1} = 0.068 - 0.058 \lambda_2^k$$

$$h_{k+1} = 0$$

On a bien, comme précédemment le coefficient de probabilité, correspondant au modèle, pour $k = 0$.

On peut calculer alors la distribution du temps de détection :

$$\Pr [K_{\text{COI} + \text{AE}} \leq k] = 0.178 - 0.152 \lambda_2^k$$

On a bien $\Pr [K_{\text{COI} + \text{AE}} \leq 0] = 0.026$

On obtient alors facilement :

$$\Pr [K_{\text{COI} + \text{AE}} = k] = 0.022 \lambda_2^{k-1}$$

Ce qui donne bien $\Pr [K_{\text{COI} + \text{AE}} = 0] = 0.026$

On trouvera à la figure 8, les valeurs limites de la distribution du temps de détection pour une application quelconque s'exécutant sur le M68000.

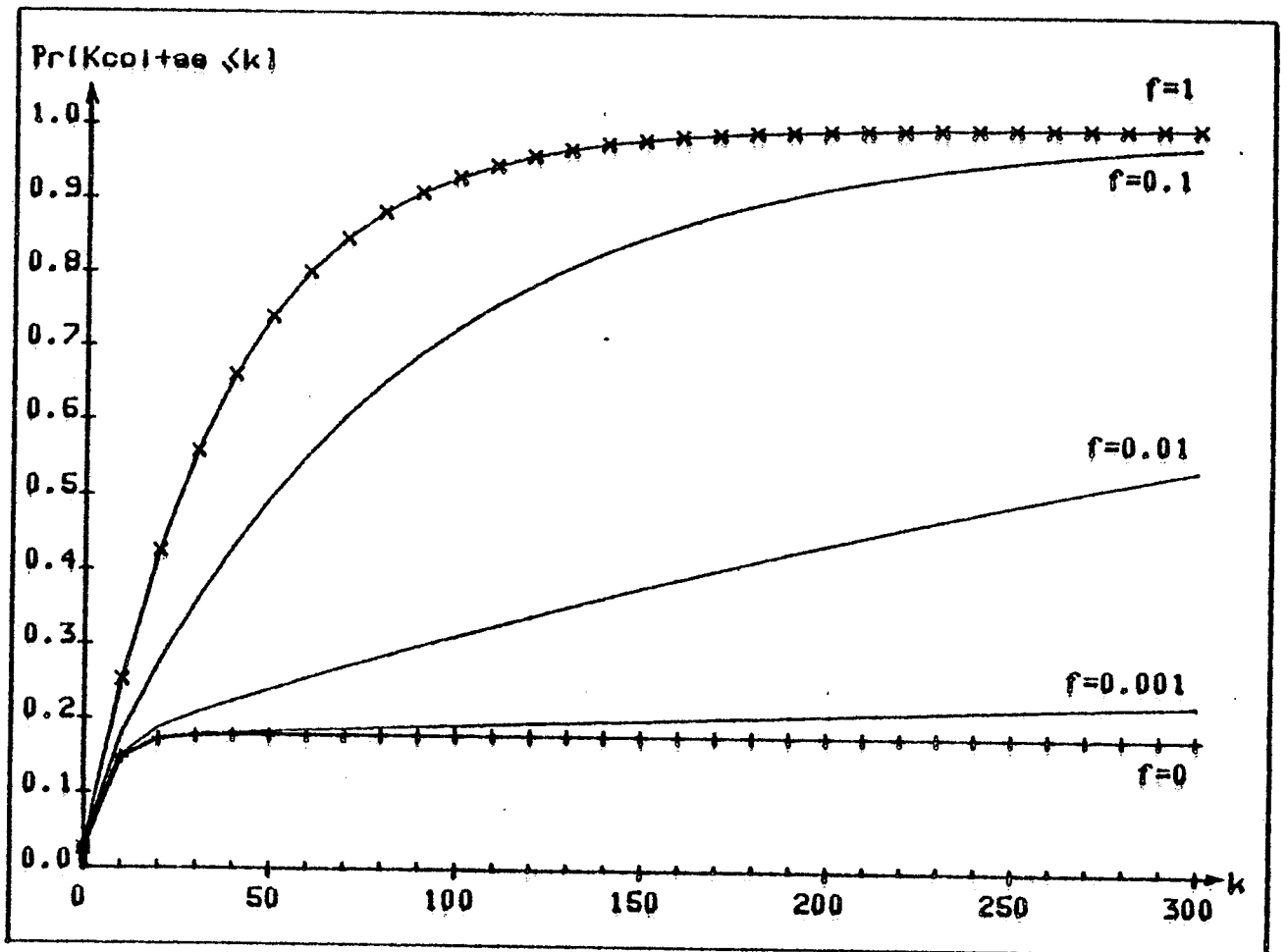


Figure 8 - Valeurs caractéristiques de la distribution du temps de détection pour une application quelconque s'exécutant sur le MC 68000.

2.- Moments du temps de détection en fonction de la fréquence :

Ces moments seront déterminés d'abord pour le modèle de la figure 6 puis pour le modèle simplifié de la figure 7.

Pour le modèle de la figure 6 :

$$E [K] = \frac{0.841f + 0.056}{0.092 f^2 + 0.011f}$$

$$\sigma^2 [K] = \frac{-0.077f^3 + 0.856f^2 + 0.114f + 0.004}{(0.092f^2 + 0.011f)^2}$$

Pour le modèle de la figure 7 :

$$E [K] = \frac{0.854 f + 0.120}{0.026 f}$$

$$\sigma^2 [K] = \frac{0.752f^2 + 0.202f + 0.021}{(0.026f)^2}$$

On remarque que dans les deux modèles, $E [K] + 1$ et $\sigma^2 [K]$ correspondent bien aux premier et second moments de lois géométriques de paramètres : 0.103 pour la figure 6 et 0.026 pour la figure 7, pour $f = 1$.

Les valeurs limites sont reportées dans le tableau XII et des valeurs numériques dans le tableau XIII.

Les courbes $E [K]$ et $\sigma [K]$ en fonction de $1/f$ sont représentées sur la figure 9.

$f = 1$	
$E [K] = 8.7$ $\sigma [K] = 9$ $\frac{\sigma [K]}{E [K]} = 1.06$	$E [K] = 37.5$ $\sigma [K] = 38$ $\frac{\sigma [K]}{E [K]} = 1.01$
$f \rightarrow 0$	
$E [K] = 5.1 \frac{1}{f}$ $\sigma [K] = 5.75 \frac{1}{f}$ $\sigma [K] = 1.28$	$E [K] = 4.6 \frac{1}{f}$ $\sigma [K] = 6 \frac{1}{f}$ $\frac{\sigma [K]}{E [K]} = 1.21$
Modèle figure 6.	Modèle figure 7

Tableau XII - Caractéristiques limites des moments de détection pour une application quelconque sur le MC 68000.

f	$1/f$	MODELE FIGURE 6		MODELE FIGURE 7	
		E [K]	[K]	[K]	[K]
0.1	10	69	77	79	85
0.05	20	126	139	125	140
0.02	50	284	317	264	306
0.01	100	540	606	494	585
0.005	200	1051	1183	955	1141
0.002	500	2579	2908	2341	2813
0.001	1000	5125	5783	4648	5600
0.0005	2000	10216	11532	9264	11174
0.0002	5000	25488	28782	23110	27895
0.0001	10000	50943	57530	46187	55765

Tableau XIII - Caractéristiques numériques des moments de détection pour une application quelconque sur le MC 68000.

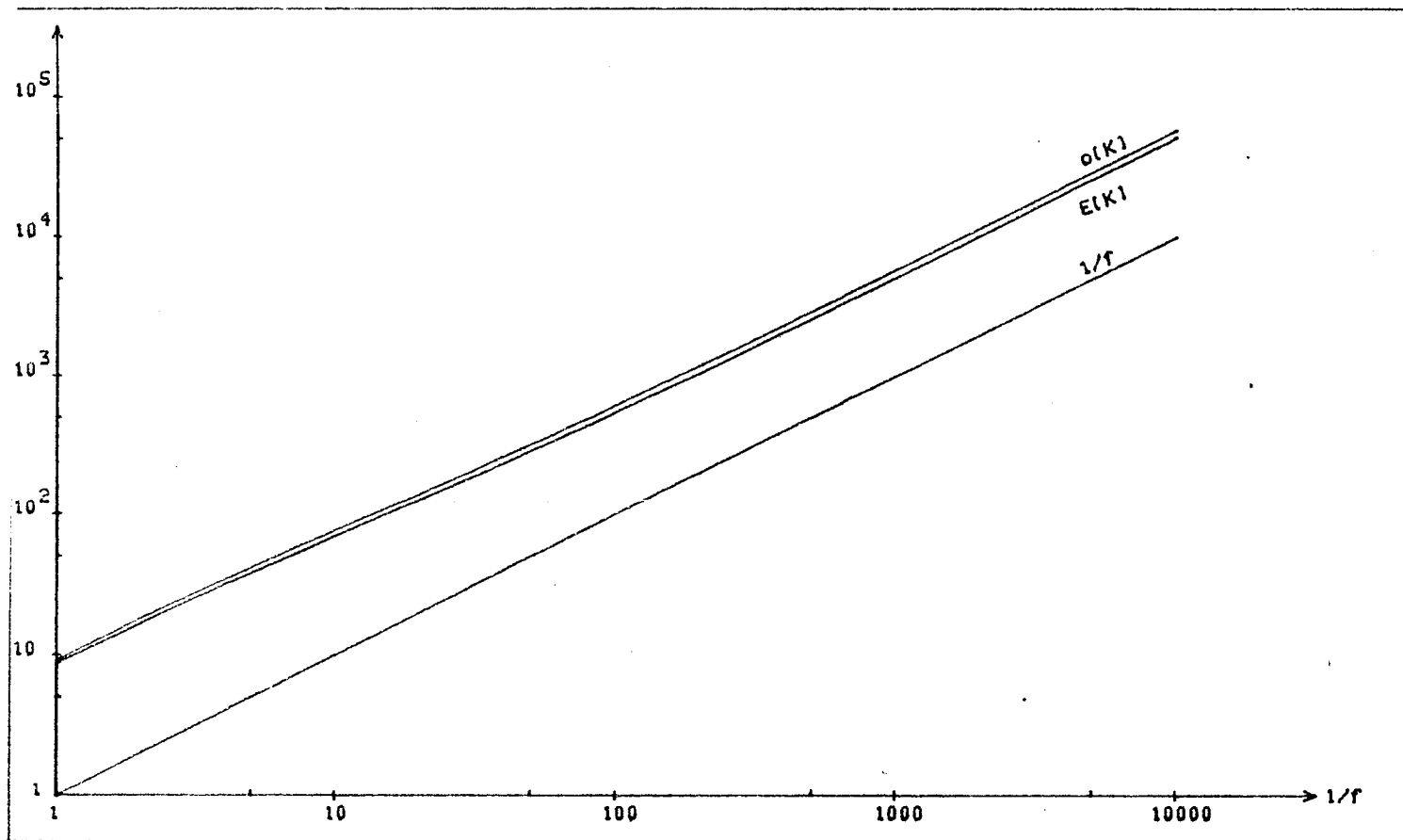


Figure 9 - Moyenne et écart-type du temps de détection pour une application quelconque s'exécutant sur le MC 68000, en fonction de 1/f.

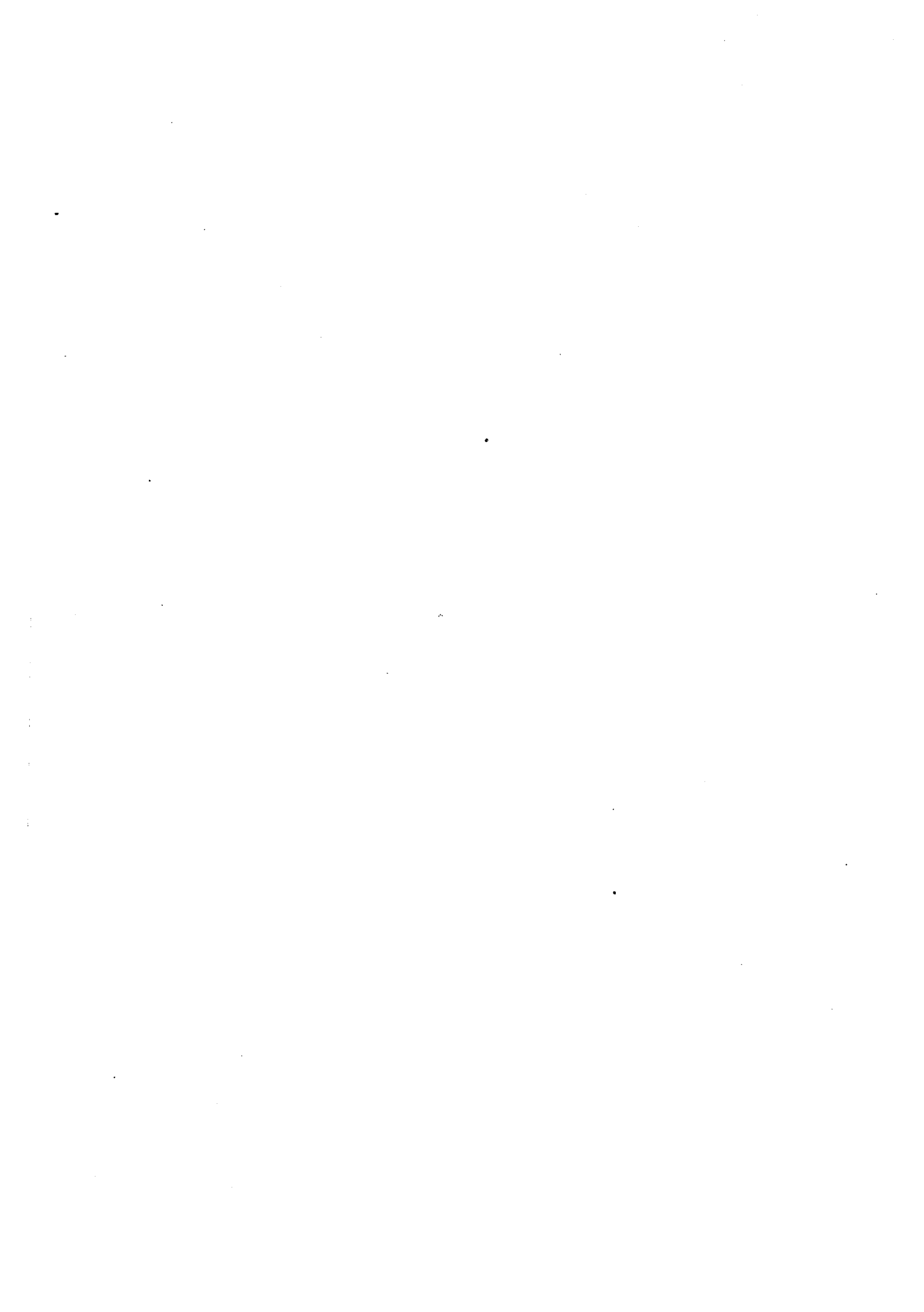
VII - CONCLUSION

Les caractéristiques du temps de détection des pannes du coeur, provoquant un déséquencement du compteur ordinal, pour une application quelconque s'exécutant sur le MC68000 montrent que ce temps de détection est confiné dans des valeurs très acceptables.

De plus, les résultats numériques présentés sont pessimistes par rapport à ce qu'ils seraient si par exemple on devait considérer la détection par chien de garde. Malheureusement, l'impossibilité quasi-certaine d'insérer des points de relance d'un chien de garde de manière parfaitement périodique dans des logiciels quelconques et la grande difficulté d'évaluation mènent à une impossibilité à obtenir des résultats généraux.

Comme il a déjà été dit pour les pannes de la périphérie de la pastille, les résultats généraux obtenus le sont sans adjonction de matériel puisque les détecteurs considérés (COI-AE) sont internes à la pastille.

Une fois encore, il faut souligner la portée de tels résultats car ils sont nécessaires aussi bien au test en ligne qu'au test hors ligne, puisqu'un "programme de test" ne peut seul détecter l'apparition d'événements E. Ce n'est que dans le cas de duplication de CPUS aux fins de comparaison que les résultats présentés dans les paragraphes précédents sont inutiles.



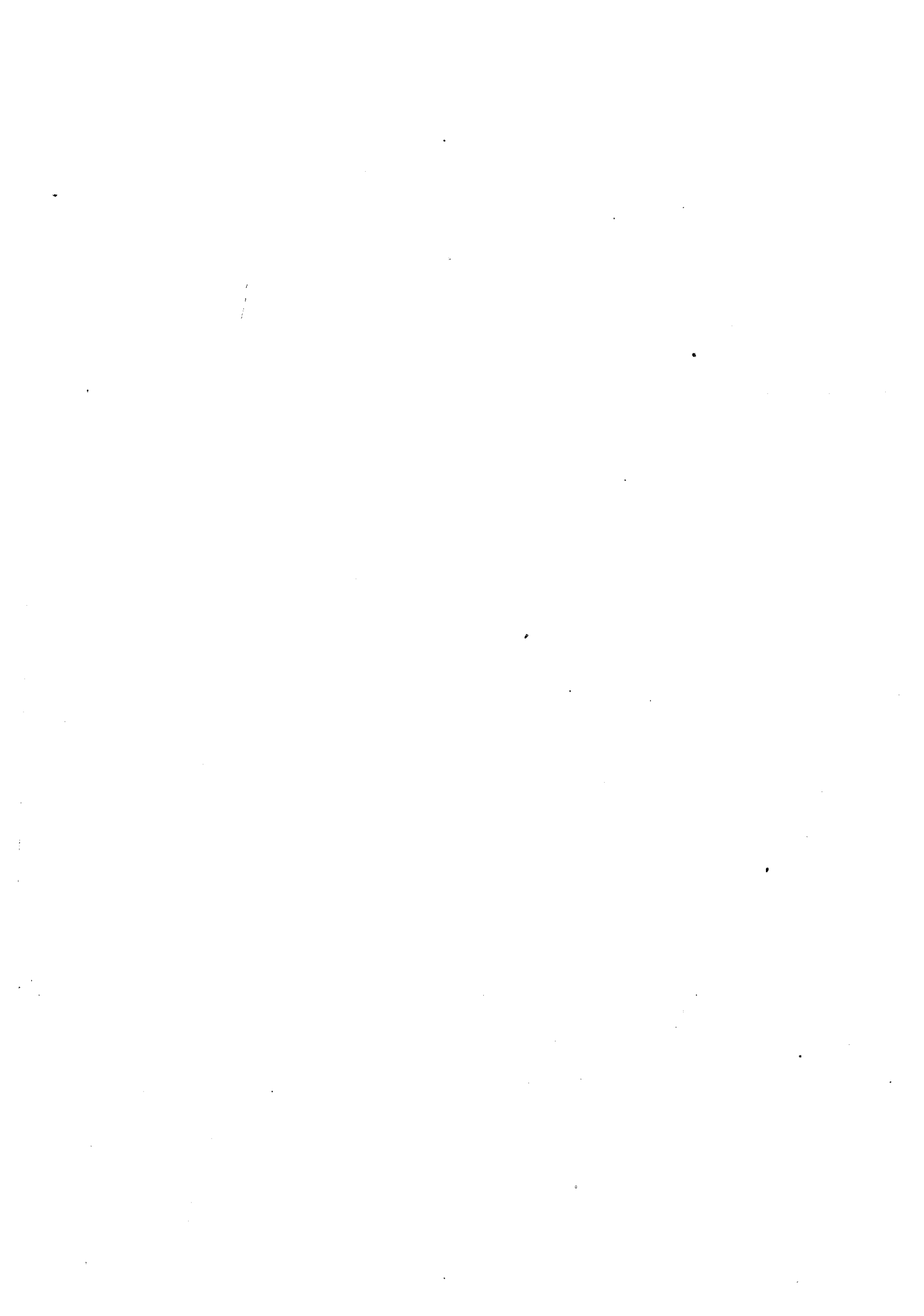
NOTES BIBLIOGRAPHIQUES

-:-:-:-:-:-:-:-

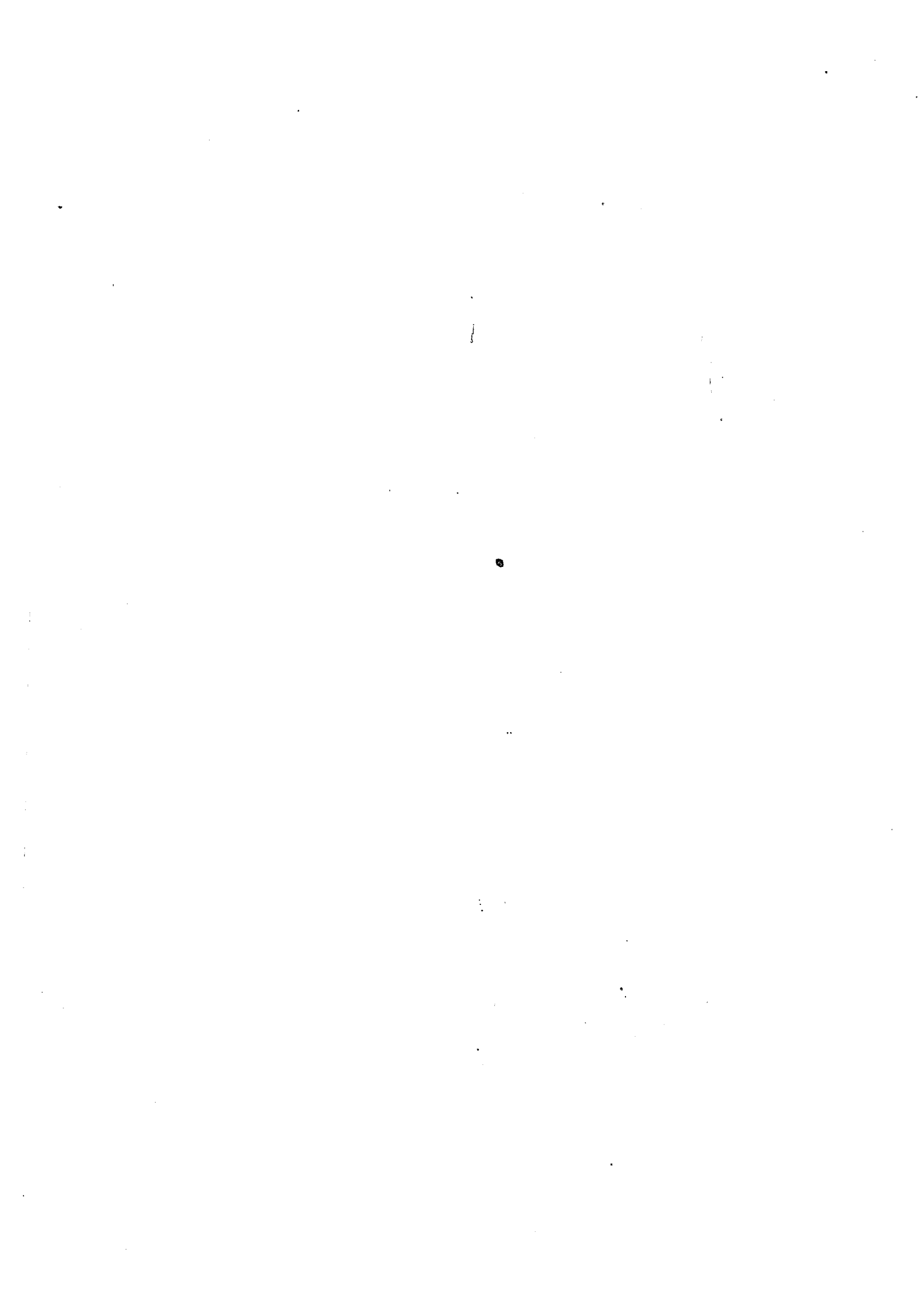
La méthodologie développée pour les modèles statiques et modèles dynamiques pour le M6800 pourra être trouvée dans [COU -80] [COU - 81]
En ce qui concerne les résultats généraux et formules sur les chaînes de Markov on se reportera à [KEM 76]

R E F E R E N C E S

- [COU - 80] - COURTOIS B. "Modélisation du temps de détection de pannes
du coeur d'un CPU"
Rapport IMAG RR 213, Octobre 1980
- [COU - 81] - COURTOIS B. "Test et LSI"
Thèse de Docteur d'Etat, Grenoble 1981
- [KEM - 76] - KEMENY J.G. & SNELL J.L. "Finite Markov chains"
Springer - Verlag, 1976.



CHAPITRE VI
COEUR NE CRÉANT PAS E
PARTIE OPÉRATIVE
Etape (v)



CHAPITRE VI - COEUR NE CREANT PAS E

PARTIE OPERATIVE

Etape (v)

I	- INTRODUCTION	257
	I.1. Stratégie du test	258
	I.2. Hypothèses de pannes envisagées pour le test analytique	258
	I.2.1. Classe 0	259
	I.2.2. Classe I	259
	I.2.3. Classe II	259
	I.3. Dérivation des vecteurs de test	259
	I.3.1. Dérivation des vecteurs de test de classe 0	260
	I.3.2. Dérivation des vecteurs de test de classe II limitée géographiquement.....	261
II	- PRESENTATION DE LA PARTIE OPERATIVE DU MC 68000	263
III	- ETUDE ET TEST DES OPERATEURS DE LA PARTIE OPERATIVE	267
	III.1. Unité arithmétique et logique (ALU). Opérateur de calcul de données	267
	III.1.1. Présentation de l'opérateur	267
	III.1.2. Test de classe 0 de l'opérateur.	268
	III.1.3. Test de classe II limitée géographiquement de l'opérateur.	279
	III.2. Etude et test des prédicats de l'opérateur de calcul	282
	III.3. Acquisition des opérandes sur l'ALU.	284
	III.3.1. Présentation de l'entrée A de l'ALU.....	284
	III.3.2. Test de l'entrée A de l'ALU.	285
	III.3.3. Cas particulier du bit 7 : extension octet.	286
	III.3.4. Décalage : cas particulier des bits 0, 7 et 15.....	288
	III.3.5. Présentation de l'entrée D de l'ALU.	293
	III.3.6. Test de l'entrée D de l'ALU.	294
	III.4. Opérateur décimal (BCD)	295

III.4.1. Présentation de l'opérateur	295
III.4.2. Test de l'opérateur.	296
III.5. Extension de l'opérateur de calcul (ALU-E)	299
III.5.1. Présentation de l'opérateur	299
III.5.2. Test de l'opérateur ALU-E	300
III.5.3. Cas particuliers des bits 0, 1 et 15.	301
III.6. Opérateur de multiplexage : MUX I/O	303
III.6.1. Présentation de l'opérateur	303
III.6.2. Test de l'opérateur.	304
III.7. Opérateur registre décodeur (DCR)	305
III.7.1. Présentation de l'opérateur	305
III.7.2. Test de l'opérateur	309
III.7.3. Acquisition et sortie de l'opérande.	310
III.8. Opérateur Fool-it	311
III.8.1. Présentation de l'opérateur	311
III.8.2. Test de l'opérateur.	314
III.9. Opérateur PREN (Priority Register Encoder)	315
III.9.1. Présentation de l'opérateur	315
III.9.2. Test de l'opérateur.	319
III.10. Opérateur de calcul d'adresses (AUH-AUL)	320
III.10.1. Présentation de l'opérateur	320
III.10.2. Test de l'opérateur.	323
III.11. Opérateur d'extension de signe (SIGNEX)	326
III.11.1. Présentation de l'opérateur	326
III.11.2. Test de l'opérateur.	326
IV - ETUDE ET TEST DU BUS	329
IV.1. Présentation du bus	329
IV.2. Test de la précharge	332
IV.3. Test des résistances	334
IV.4. Test de l'amplificateur	335
IV.5. Test des switches	335
IV.6. Test des coupures de bus.	336

V	- ETUDE ET TEST DES REGISTRES DE LA PARTIE OPERATIVE	337
V.1.	Test des registres organisés en RAM.	338
V.1.1.	Présentation du point de RAM	339
V.1.2.	Test du point de RAM	340
V.1.3.	Cas particulier des registres DT et PC.	343
V.2.	Test des registres autres que ceux de la RAM	344
V.2.1.	Test du registre AOB (Buffer de sortie adresse)	344
V.2.2.	Test du registre AUR (Latch d'AU)	346
V.2.3.	Test du registre AT (adresse temporaire)	348
V.2.4.	Test du registre du PREN	351
V.2.5.	Test du registre FTU (Field Translate Unit)	351
V.2.6.	Test du registre DCR (decode register)	354
V.2.7.	Test du registre DBIN (data input buffer)	354
V.2.8.	Test du registre DOB (data output buffer)	356
V.2.9.	Test du registre ALUE (ALU-Extender)	357
V.2.10.	Test du registre ALU-R (latch d'ALU)	357
V.2.11.	Test du registre ALU-B (buffer d'ALU).	359
VI	- CONCLUSION	361
REFERENCES	363



I. INTRODUCTION

Ce chapitre couvre la totalité du test analytique de la partie opérative du MC 68000 :

- test des opérateurs,
- test des bus,
- test des registres,
- test de la génération des signaux de compte-rendu.

On rappelle que les circuits du code-condition ainsi que leurs circuits de gestion sont considérés séparément. Ils n'appartiennent ni à la partie opérative, ni à la partie contrôle, ils feront l'objet d'un test particulier.

Ce test est abordé sans orientation particulière en-ligne ou hors-ligne. La séparation claire entre les différents tests pour les différentes parties est suffisante pour :

- opérer une sélection en vue d'un test en ligne (continu ou discontinu) : étape (v) de la méthodologie.
- prendre la totalité en vue d'un test hors-ligne (in-situ ou ex-situ).

Pour détecter les différentes pannes envisagées, les vecteurs de test sont fournis lorsqu'il n'y a pas d'ambiguïté entre le vecteur et l'instruction qui le générera. Par exemple pour le test du correcteur décimal (opérateur BCD) il est clair que l'instruction à lancer est une opération décimale codée binaire (ABCD = addition, SBCD = soustraction ou NBCD complément).

Pour certaines pannes, le lien entre le vecteur de test et l'instruction qui le générera n'est pas clair, il a donc été nécessaire de préciser l'instruction qui permettra de propager la panne afin de la détecter (par exemple pour le test de bus ou de registres).

La totalité des procédures n'ayant pas été programmée, il n'est donc pas possible de donner une taille à la totalité des programmes qui recouvrent le test analytique de la partie opérative.

I.1. - Stratégie de test :

Les vecteurs de test peuvent être obtenus de deux façons différentes, selon le fait que :

- le fonctionnement est considéré,
- la structure est considérée.

La partie contrôle de microprocesseur sera l'objet d'un test fonctionnel (dépendant de la fonction), tandis que la partie opérative d'un microprocesseur sera l'objet d'un test de type structurel, appelé test analytique (dépendant de la réalisation physique du circuit).

I.2. Hypothèses de pannes envisagées pour le test analytique

Le nombre de pannes possibles sur un circuit étant très important [COU 81], un regroupement en classe est nécessaire. Les trois classes utilisées ici, regroupent les différents types et nombres de défauts que l'on se bornera à considérer [COU 82].

1.2.1. Classe 0

Cette classe regroupe les défauts simples physiques, tels qu'un contact défaillant, un MOS défaillant, un conducteur coupé. Les courts-circuits ne font pas partie de cette classe. En cas de grille flottante, on considère que le mode de panne associé est un MOS S-ON ou S-OPEN.

1.2.2. Classe I

Cette classe regroupe la classe 0, et les défauts de court-circuits entre conducteur aluminium seulement et uniquement entre les deux conducteurs les plus proches géographiquement. Bien que le cas similaire entre deux diffusions soit sans doute moins probable, on le considérera néanmoins, car la difficulté du problème est comparable.

1.2.3. Classe II

Cette classe regroupe la classe I ainsi que les défauts tels que les courts-circuits entre un conducteur aluminium et un autre conducteur aluminium quelconque dans le circuit. Le cas de diffusion est étendu de façon semblable. Les défauts multiples sont considérés au sens classique du terme. Seuls les défauts classés "fabrication seulement" restent exclus (courts-circuits Aluminium-diffusion).

1.3. Dérivation des vecteurs de test

La détermination des vecteurs de test correspondant aux hypothèses de la classe 0 sera faite par sensibilisation du chemin électrique.

Un chemin électrique est constitué d'une ou plusieurs sources de courant et d'une ou plusieurs masses. Ce chemin électrique est dit sensible à une panne si la présence de la panne se manifeste par une erreur sur une ou plusieurs sorties observables.

La détermination des vecteurs de test de classe I et II implique la connaissance de l'implantation physique du circuit afin de pouvoir prendre en compte les seuls (classe I) ou tous les (classe II) courts-circuits possibles.

1.3.1. Dérivation des vecteurs de test de classe 0

La figure 1 représente une porte dont la sortie S est $\overline{A.B + C}$. Les vecteurs de test concernant ces défauts de la classe 0 s'obtiennent sans difficulté.

- . Le test du MOS signal de l'entrée A est fait à l'aide du chemin MOS de charge $\rightarrow A \rightarrow B \rightarrow$ masse. Ainsi B doit être passant et C bloqué.
- . Le test du MOS signal de l'entrée B est fait à l'aide du même chemin. A doit être passant et C bloqué.
- . Le test du MOS signal de l'entrée C est fait à l'aide du chemin MOS de charge $\rightarrow C \rightarrow$ masse. A ou B (ou les deux) doit être bloqué.

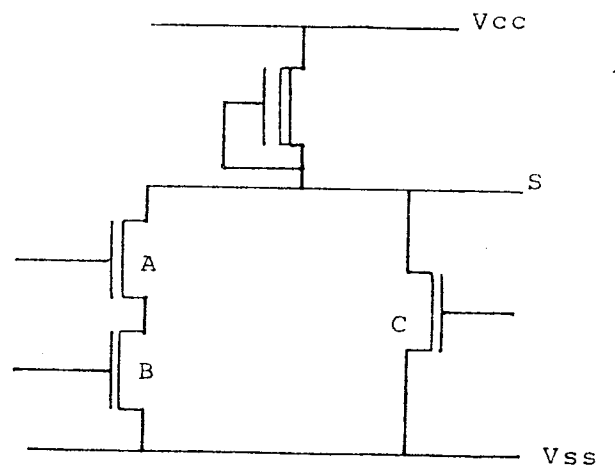


Figure 1. Porte standard : $S = \overline{A.B + C}$

Les vecteurs de test pour cette porte peuvent être ceux de la Table I.

A	B	C
0	1	0
1	1	0
1	0	0
1	1	0
0	0	0
0	0	1

Table 1. Vecteurs de test de la porte représenté figure 1.

I.3.2. Dérivation des vecteurs de test de classe II limitée géographiquement

Un test de classe II est possible par le biais d'un test exhaustif. En effet, la dérivation d'un test de classe II impliquant un examen de toutes les pannes physiques possibles, peut se faire en exécutant un test exhaustif des circuits concernés et donc finalement indépendant de la structure des circuits.

Une étude de test analytique pourrait donner un nombre de vecteurs de test moindre, mais au prix d'une étude plus longue.

Par exemple, le test de classe II limité géographiquement à la porte de la figure 1 donne les vecteurs de test de la table II

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Table I: Vecteurs de classe II limités géographiquement à la porte de figure 1.

Pour des raisons de clarté, la dérivation des vecteurs ne sera entièrement explicitée que pour l'opérateur arithmétique et logique (ALU). Pour les autres opérateurs, les bus et les registres, les tableaux présentant les vecteurs obtenus sont donnés. La dérivation de tous les vecteurs pour chaque opérateur, bus et registres a fait l'objet d'un rapport et pourra être trouvée dans [MAR 83].

II. PRESENTATION DE LA PARTIE OPERATIVE DU MC 68000

La figure 4 présente le schéma fonctionnel de la partie opérative du MC 68000.

Elle est organisée autour de deux bus bi-directionnels, bus "adresses" et bus "données", de 16 bits, segmentables en trois sections par des "switches" commandés par la partie contrôle.

On peut remarquer que chacun de ces deux bus peut effectuer en fait aussi bien des transferts d'adresses que de données.

Cette structure permet d'effectuer certains calculs en parallèle sur les poids forts et les poids faibles des registres adresses, et simultanément un calcul sur les registres données par l'intermédiaire de l'ALU.

Outre les registres adresses et données, la partie opérative comprend plusieurs registres tampons, ainsi que des circuits spécialisés.

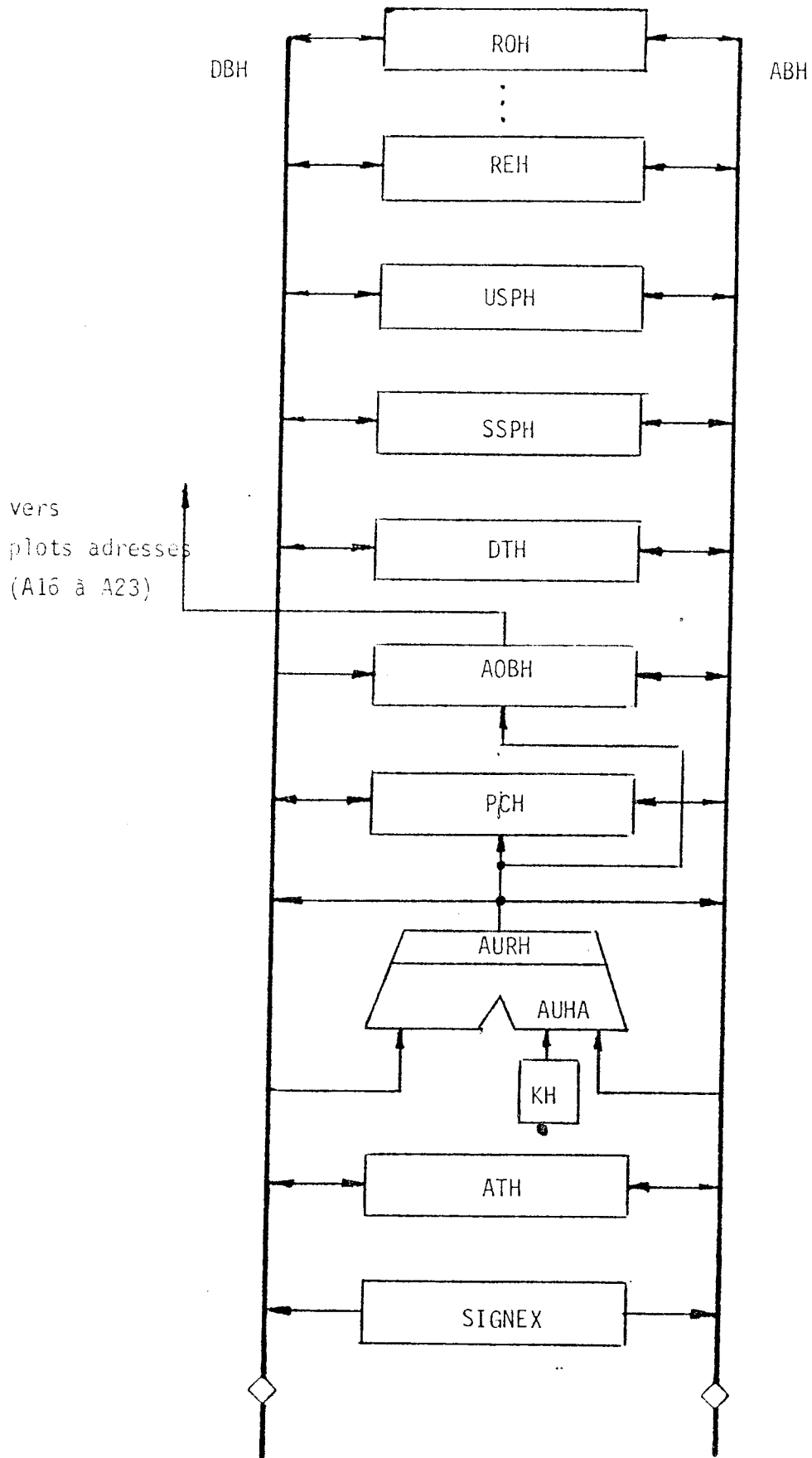


Figure 2. Schéma fonctionnel de la partie opérative du MC 68000

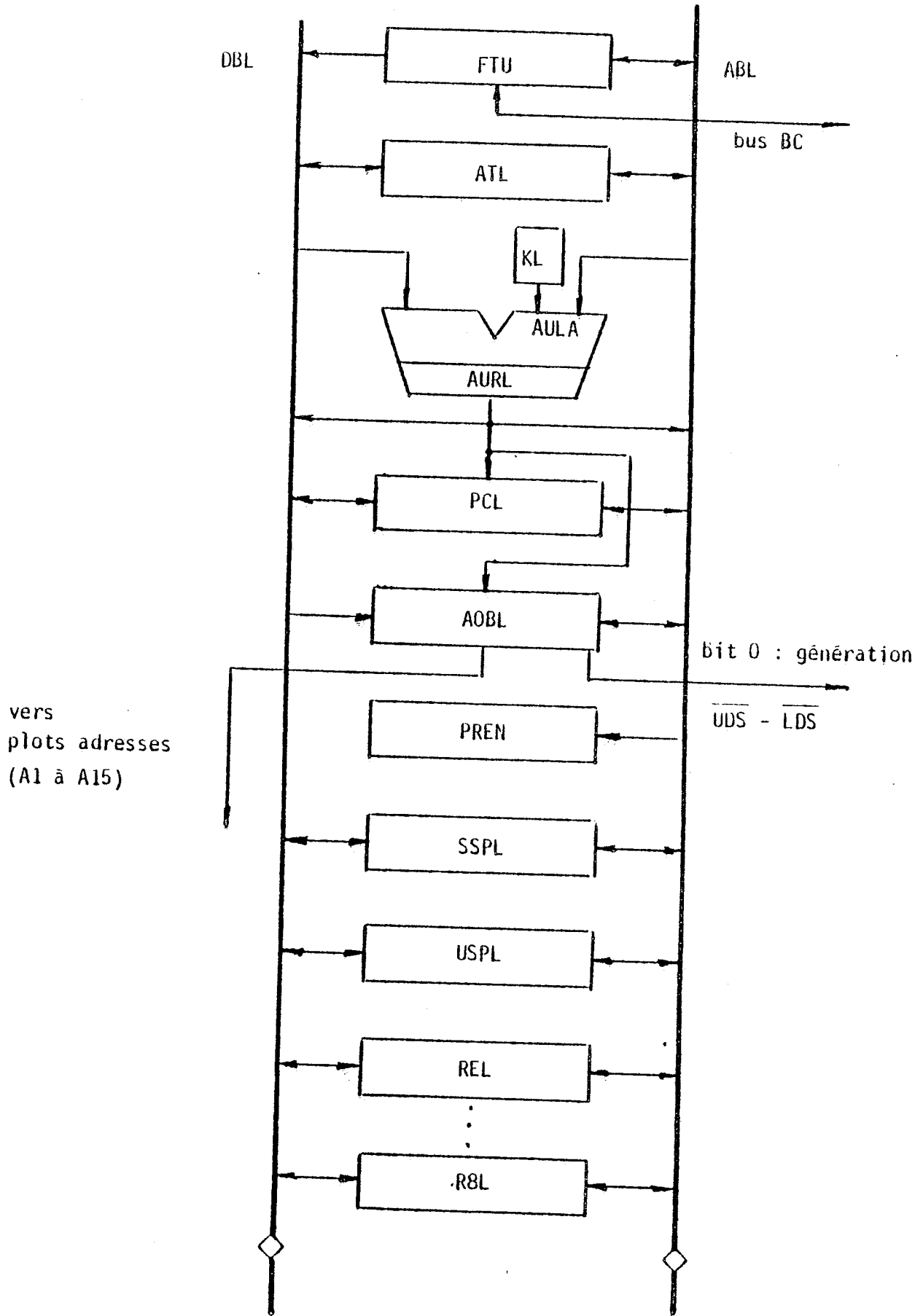


Figure 2. Schéma fonctionnel de la partie opérative du MC 68000 (suite)

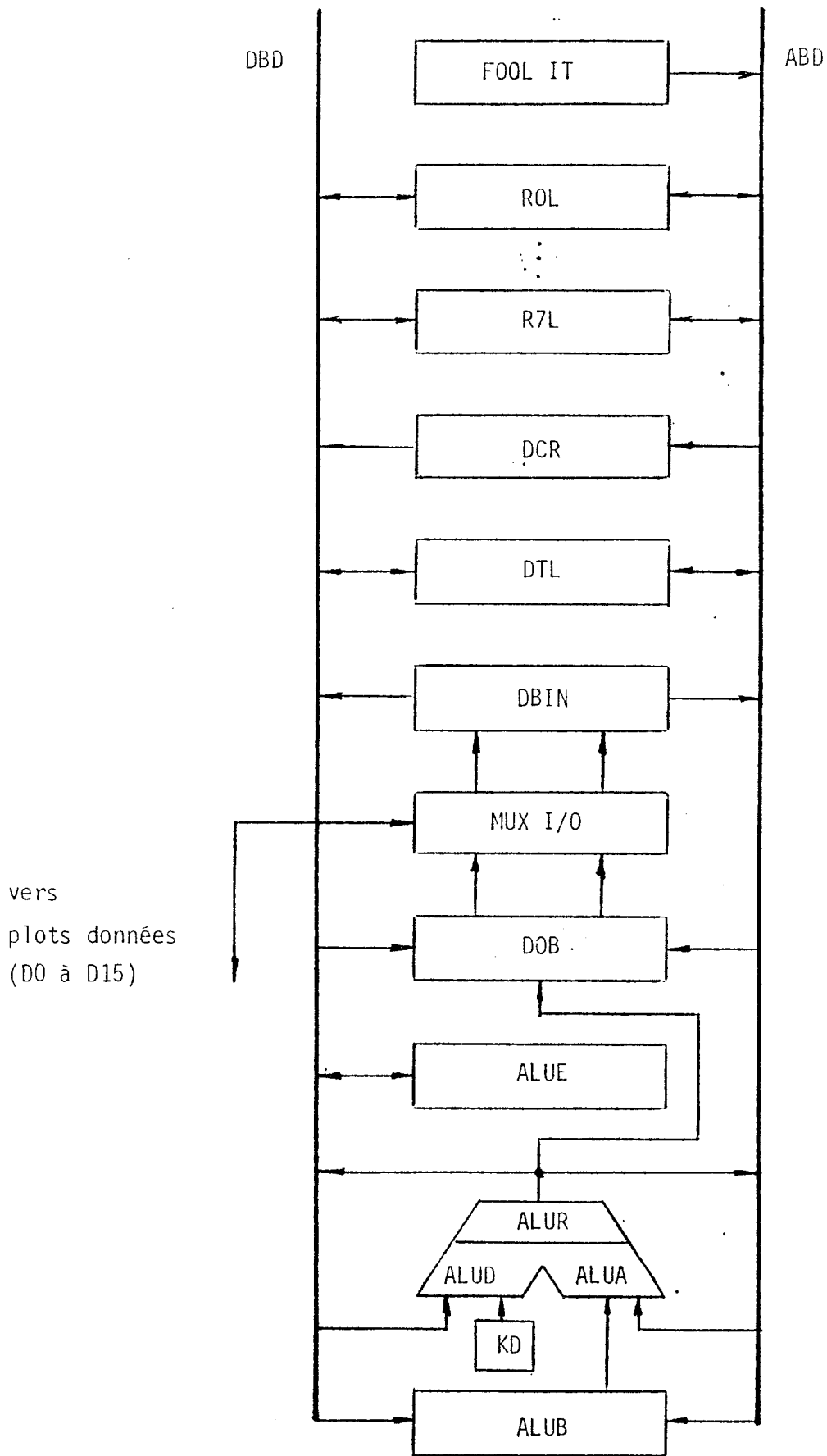


Figure 2. Schéma fonctionnel de la partie opérative du MC 68000 (suite)

III - ETUDE ET TEST DES OPERATEURS DE LA PARTIE OPERATIVE

La partie opérative contient 10 opérateurs :

- AOH : unité arithmétique (adresse poids forts) + tampon
- AUL : unité arithmétique (adresse poids faibles) + tampon
- ALU : unité arithmétique et logique (donnée) + tampon
- ALU-E : extension de l'ALU
- SIGNEX : extension de signe
- PREN : registre encodeur de priorité
- Fool-it : réécriture de l'octet poids faible dans un mot
- DCR : décodeur registre
- MUX I/O : manipulateur d'octets

III.1. Unité arithmétique et logique (ALU) - opérateur de calcul de données

III.1.1. Présentation de l'opérateur

La figure 5 donne le schéma d'un pas de l'opérateur.

Les commandes SL (décalage gauche), C-AND (et logique) $\overline{\text{C-OR}}$ (ou logique), $\overline{\text{C-XOR}}$ (ou exclusif), sont issues de la partie contrôle.

Les trois entrées possibles sont les opérandes A et D, et la retenue complémentée $\overline{\text{CIN}}$. Les acquisitions des opérandes A et D sont étudiées par la suite, mais nous pouvons dire maintenant qu'elles permettent avec les commandes de l'UAL de réaliser les sommes, les soustractions, les décalages, les "et" ou les "ou" logiques.

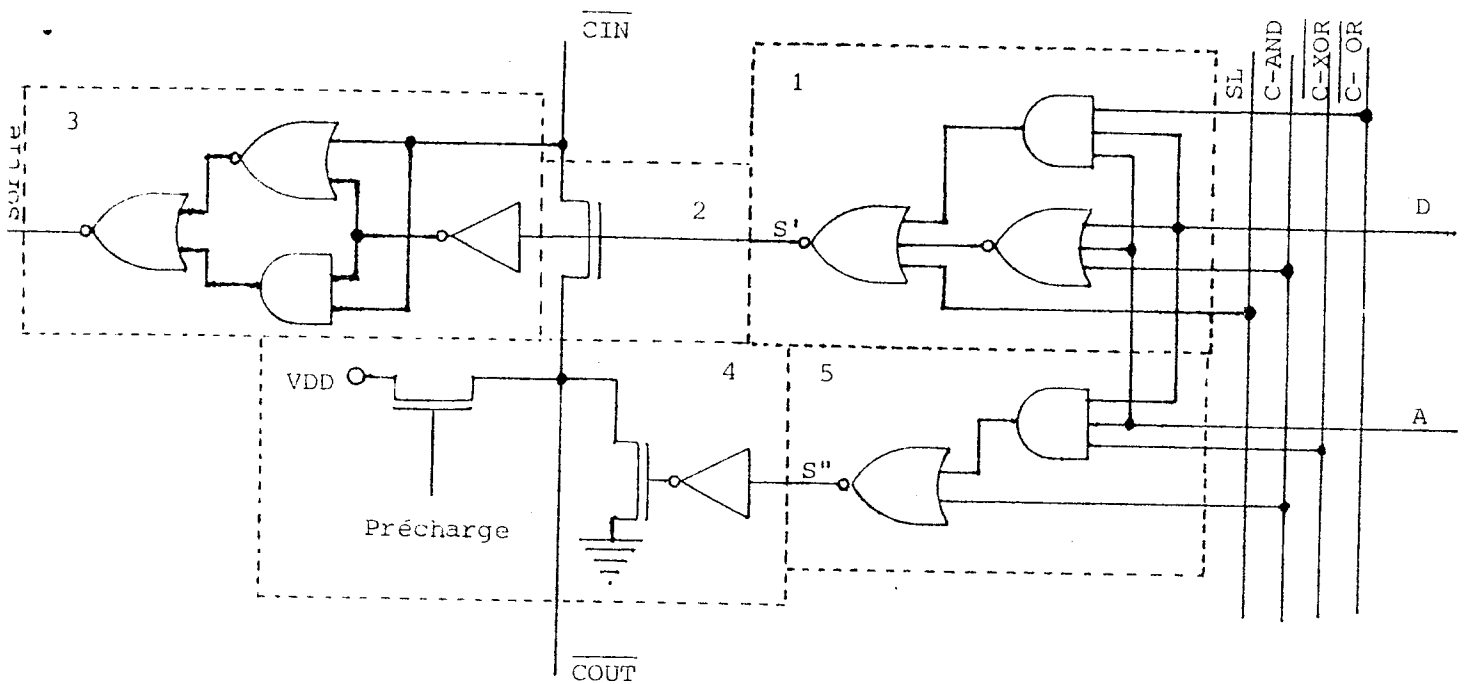


Figure 3. PAS de l'opérateur de calcul de donnée ALU du MC 68000.

Une étude préalable sur le MOTOROLA M 6800 nous indique que les pas d'opérateurs du M 6800 et MC 68000 sont les mêmes. Etant donné que le test du M 6800 a été étudié auparavant [COU 82], les résultats obtenus pour le M 6800 seront repris dans la mesure du possible.

III.1.2. Test de classe 0 de l'opérateur

Les hypothèses de panne de la classe 0 permettent de partitionner le circuit de la manière indiquée sur la figure précédente (5 blocs).

Le test va se dérouler en deux étapes, tout d'abord :

- une première partie va permettre de détecter les pannes affectant l'opérateur non compris les lignes de commandes du code de l'opération ;
- une seconde partie va permettre de détecter les pannes affectant les lignes de commandes du code-opération.

Dans la première partie, c'est l'addition qui va être utilisée, donc le code opération fixé est $SL=0$, $C-AND=0$, $\overline{C-XOR}=1$, $\overline{C-OR}=1$. La table III présente le fonctionnement de l'opérateur en addition.

A	D	S'	S''	\overline{CIN}	\overline{COUT}	S
0	0	0	1	1	1	0
0	1	1	1	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	0	0
0	0	0	1	0	1	1
0	1	1	1	0	0	0
1	0	1	1	0	0	0
1	1	0	0	0	0	1

Table III. Fonctionnement du pas de l'opérateur en addition ($SL=0$, $C-AND=0$, $\overline{C-XOR}=1$, $\overline{C-OR}=1$)

a) Etude du bloc 1

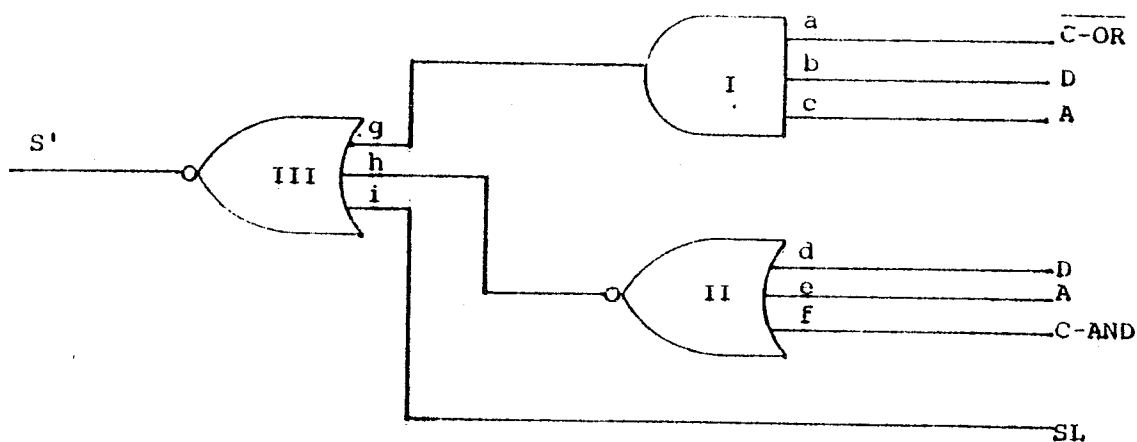


Figure 4. Bloc 1

Nous allons raisonner porte par porte, tout en sachant que si une panne est supposée pour une porte alors le vecteur de test doit être tel que la panne est propagée jusqu'en S' , (figure 4).

Exemple :

- collage de l'entrée b à 0, il faut $A = 1$ et $D = 1$ pour propager la panne jusqu'en S' comme le montre la figure 5. D'où le vecteur de test 11 :

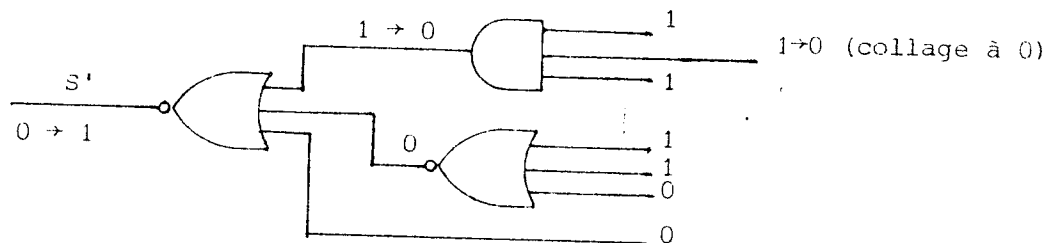


Figure 5. Propagation du collage de b à 0 jusqu'en S'

- collage de l'entrée b à 1, il faut $A = 1$ et $D = 0$, un raisonnement similaire montre que cette panne est propagée en S' d'où le vecteur de test 10.
- collage de l'entrée c à 0, $A = 1$, $D = 1$, déjà étudié.
Remarque : cela veut dire que le vecteur de test 11 teste 2 pannes.
- collage de l'entrée c à 1, $A = 0$, $D = 1$, la panne est propagée pour le vecteur de test 01.

Il n'est pas nécessaire de tester les collages de la sortie de la porte I (cf. figure 4) car les théories de l'équivalence de pannes permettent de démontrer que pour des portes standard comme celles-ci, il y a équivalence entre les pannes de collages des sorties et celles des entrées. Les vecteurs de test sont donc les mêmes que ceux obtenus.

- collage de l'entrée d à 1, pour $A = 0$ et $D = 0$ la panne est propagée jusqu'en S', donc le vecteur de test obtenu est 00.

Ces quatre vecteurs de test épuisent toutes les possibilités offertes sur les entrées A et D. Mais on peut vérifier que :

- collage de l'entrée d à 0, vecteur de test = 01
- collage de l'entrée e à 1, vecteur de test = 00
- collage de l'entrée e à 0, vecteur de test = 10
- collage de g à 1, vecteur de test = 01 ou 10
- collage de g à 0, vecteur de test = 11
- collage de h à 1, vecteur de test = 01 ou 10
- collage de h à 0, vecteur de test = 00

Les entrées A et D sont évidemment couvertes par ces vecteurs, en effet :

- collage de A à 0 : vecteurs de test 11 ou 10
- collage de A à 1 : vecteurs de test 01 ou 00
- collage de D à 0 : vecteurs de test 11 ou 01
- collage de D à 1 : vecteurs de test 10 ou 00

Il est rappelé que les collages des entrées a, f, i, seront étudiés plus loin.

Donc, pour tester le bloc 1 d'un pas d'ALU en addition, les vecteurs de test donnés par la table IV sont nécessaires.

A[I]	D[I]
0	0
0	1
1	0
1	1

Table IV. Vecteurs de test du bloc 1 de l'ALU pour le test en addition

b) Etude du bloc 2

Le chemin sensible est constitué par la décharge de la ligne vers la masse d'un pas précédent. On doit donc bloquer le chemin menant à l'autre masse, soit $S'' = 1$.

Les vecteurs de test sont donc :

$$\begin{array}{l} S' = 0 \quad \overline{CIN} = 0 \quad S'' = 1 \\ \text{et } S' = 1 \quad \overline{CIN} = 0 \quad S'' = 1 \end{array}$$

c) Etude du bloc 3

Le bloc 3 réalise la même fonction logique que le bloc 1 en addition. On trouve finalement les quatre vecteurs de test donnés par la table V.

\overline{CIN}	S'
0	0
0	1
1	0
1	1

Table V. Vecteurs de test du bloc 3 de l'ALU

d) Etude du bloc 4

- Transistor de précharge :

- . soit la précharge est impossible, d'où le vecteur de test $\overline{COUT} = 1$;
- . soit la précharge est continue et la consommation seule est anormale. Dans ce cas il y a un risque de claquage du MOS de précharge ou du MOS signal connecté à la masse ; si un de ces deux claquages se produit, on retombe alors dans les autres cas de pannes du bloc 4.

- Transistor de mise à la masse :

Le chemin sensible est constitué par la décharge de la ligne de report dans la masse incluse dans ce bloc.

On doit donc bloquer le chemin allant à une masse éventuelle par l'intermédiaire du pas précédent.

C'est-à-dire : soit $\overline{CIN} = 1$ ou $S' = 0$.

Les vecteurs de test sont donc :

$$S'' = 0 \quad \overline{CIN} = 1$$

$$S'' = 1 \quad \overline{CIN} = 1$$

ou

$$S'' = 0 \quad S' = 0$$

$$S'' = 1 \quad S' = 0$$

e) Etude du bloc 5

Pour les mêmes raisons qu'au bloc 4, il faut soit $\overline{CIN} = 1$, soit $S' = 0$, pour l'addition, un raisonnement similaire au bloc 1 donne les vecteurs de test :

$$A = 1 \quad D = 0$$

$$A = 0 \quad D = 1$$

$$A = 1 \quad D = 1$$

f) Minimisation du nombre de vecteurs de test

Bloc 1

$$A [I] = 0 \quad D [I] = 0 \quad VT 1$$

$$A [I] = 0 \quad D [I] = 1 \quad VT 2$$

$$A [I] = 1 \quad D [I] = 0 \quad VT 3$$

$$A [I] = 1 \quad D [I] = 1 \quad VT 4$$

Bloc 2

$S' = 0$	$\overline{CIN} = 0$	$S'' = 1$	VT 5
$S' = 1$	$\overline{CIN} = 0$	$S'' = 1$	VT 6

Bloc 3

$\overline{CIN} = 0$	$S' = 0$	VT 7
$\overline{CIN} = 0$	$S' = 1$	VT 8
$\overline{CIN} = 1$	$S' = 0$	VT 9
$\overline{CIN} = 1$	$S' = 1$	VT 10

Bloc 4

$\overline{COUT} = 1$		VT 11
$S'' = 0$	$\overline{CIN} = 1$	VT 12
ou $S'' = 0$	$\overline{CIN} = 0$	
$S'' = 1$	$\overline{CIN} = 1$	VT 13
ou $S'' = 1$	$S' = 0$	

Bloc 5

$A [I] = 1$	$D [I] = 0$	$\overline{CIN} = 1$	VT 14
ou $A [I] = 1$	$D [I] = 0$	$S' = 0$	
$A [I] = 0$	$D [I] = 1$	$\overline{CIN} = 1$	VT 15
ou $A [I] = 0$	$D [I] = 1$	$S' = 0$	
$A [I] = 1$	$D [I] = 1$	$\overline{CIN} = 1$	VT 16
ou $A [I] = 1$	$D [I] = 1$	$S' = 0$	

La Table VI présente les vecteurs de test obtenus pour l'ALU en addition.

A	D	S'	S''	$\overline{\text{CIN}}$	$\overline{\text{COUT}}$	S	Numéro des vecteurs
0	0	0	1	1	1	0	1 - 9 - 11 - 12
0	1	1	1	1	1	1	2 - 10 - 11 - 12 - 15
1	0	1	1	1	1	1	3 - 10 - 11 - 12 - 14
1	1	0	0	1	0	0	4 - 9 - 13 - 16
0	0	0	1	0	1	1	1 - 5 - 7 - 11 - 12
0	1	1	1	0	0	0	2 - 6 - 8
1	0	1	1	0	0	0	3 - 6 - 8
1	1	0	0	0	0	1	4 - 7 - 13 - 16

Table VI. Vecteurs de test obtenus pour l'ALU en addition.

Il est alors possible de tirer la table VII de la table précédente pour obtenir les cinq vecteurs de test suffisants couvrant le test complet de l'ALU en addition ;

A	D	$\overline{\text{CIN}}$
0	1	1
1	0	1
1	1	1
0	0	0
0	1	0

Table VII. Vecteurs de tests suffisant au test de l'ALU en addition.

Les vecteurs de test seront choisis de manière à minimiser le nombre total d'opérandes qui testeront toute l'UAL : en choisissant les cinq vecteurs de test définis plus haut, alors cinq opérandes suffisent à tester l'ALU en addition.

Maintenant que le test d'un pas d'opérateur est effectué pour une opération fixée, il faut maintenant vérifier la bonne entrée du code opération.

Commande SL

Collage SL à 1 (peut être testé pendant une addition) :

A [I] = 0	D [I] = 1
ou A [I] = 1	D [I] = 0

(ce collage est testé pendant l'addition)

Collage SL à 0 :

A [I] = 0	D [I] = 1
ou A [I] = 1	D [I] = 0

(mais nécessite commande SL à 1 ==> instruction ROL).

Commande C-AND

Collage C-AND à 1 (peut être testé pendant une addition) :

A [I] = 0	D [I] = 0
ou A [I] = 0	D [I] = 1
ou A [I] = 1	D [I] = 0

(ce collage est testé pendant l'addition)

Collage C-AND à 0 :

A [I] = 0	D [I] = 0
-----------	-----------

Dans le cas d'un "et" logique, la retenue entrante est automatiquement générée à zéro si $S' = 0$ est donc nécessaire (bloc 5) donc un second vecteur est nécessaire :

A [I] = 1	D [I] = 1
ou A [I] = 1	D [I] = 0
ou A [I] = 0	D [I] = 1

(mais nécessité de la commande C-AND à 1 ==> instruction AND).

Commande C-XOR

Collage C-XOR à 1 :

$$A [I] = 1 \qquad D [I] = 1$$

La retenue entrante est automatiquement générée à 1.

(mais nécessité de la commande C-XOR = 0 ==> instruction XOR).

Collage C-XOR à 0 (peut être testé pendant une addition) :

$$A [I] = 1 \qquad D [I] = 1$$

(ce collage est testé pendant l'addition).

Commande C-OR

Collage C-OR à 1 :

$$A [I] = 1 \qquad D [I] = 1$$

(mais nécessité de la commande C-OR = 0 ==> instruction OR).

Collage C-OR à 0 (peut être testé pendant l'addition) :

$$A [I] = 1 \qquad D [I] = 1$$

(ce collage est testé pendant l'addition).

La Table VIII résume les 10 vecteurs de test qui permettent de couvrir toutes les pannes de classe 0 de l'ALU du MC 68000.

Retenue entrante	PAS 0	PAS. 1	PAS 2	PAS 3...	OP A	OP D	SL	C-AND	C-XOR	C-OR	INSTRUCTION
0	01	01	01	01	0000	FFFF	0	0	1	1	
1	10	10	10	10	FFFF	0000	0	0	1	1	ADD
0	00	11	00	11	AAAA	AAAA	0	0	1	1	ou
1	11	00	11	00	5555	5555	0	0	1	1	SUB
1	01	01	01	01	0000	FFFF	0	0	1	1	
	11	11	11	11	FFFF	FFFF	0	0	1	0	OR
	00	00	00	00	0000	0000	0	1	1	1	AND
	11	11	11	11	FFFF	FFFF	0	1	1	1	AND
	11	11	11	11	FFFF	FFFF	0	0	0	1	X-OR
	0	0	0	0	0000	-	1	0	1	1	ROL

Table VIII. Vecteurs de test nécessaireset suffisantsau test des pannes de classe 0 de l'ALU du MC 68000

III.1.3. Test de classe II limité géographiquement de l'opérateur

Un test exhaustif d'un pas de l'opérateur peut aisément être déduit, car le nombre de configurations possibles des commandes SL, C-AND, C-XOR, C-OR est réduit à 5.

On obtient ainsi 24 vecteurs pour l'opérateur, valables pour la classe II réduite uniquement à un pas.

Ces 24 vecteurs se déterminent de la façon suivante :

a) Fonction ADD, SUB, ADDX, SUBX

Ces vecteurs sont ceux de la table IX et ils sont au nombre de 8. En effet les trois entrées A, D et \overline{CIN} définissent 2^3 possibilités.

b) Fonction OR

Dans ce cas, on a, en fonctionnement normal : $\overline{CIN} = 1$. On a donc quatre vecteurs composant les quatre possibilités sur les deux opérandes A et D.

c) Fonction AND

Dans ce cas, on a, en fonctionnement normal : $\overline{CIN} = 0$. On a donc aussi quatre vecteurs.

d) Fonction X-OR

On a $\overline{CIN} = 1$ et donc quatre vecteurs.

e) Fonction de DECALAGE A GAUCHE

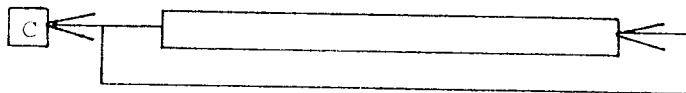
On a alors $\overline{CIN} = \overline{A_{I-1}}$, l'un des opérandes étant FFFF.

En utilisant l'instruction ROL (rotation circulaire gauche), les quatre vecteurs obtenus sont chainables.

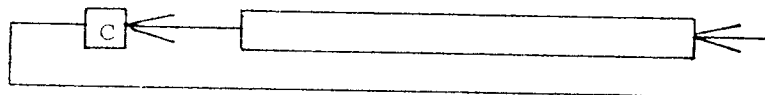
Il est à remarquer que l'instruction ROL du MC 68000 est différente de celle du M 6800.

En effet :

ROL Pour le MC 68000

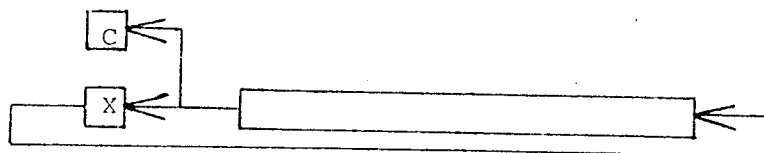


ROL Pour le M 6800



L'équivalent de l'instruction ROL du M 6800 est obtenu par l'intermédiaire du bit X d'extension du code condition, et par l'instruction ROXL du MC 68000 qui peut être représentée par :

ROXL



La Table IX regroupe les 24 vecteurs de test nécessaires au test de classe II limitée géographiquement et réduite uniquement à un pas d'ALU du MC 68000.

INSTRUCTION	PAS ₀ \overline{C}_{-1}	PAS 1	PAS 2	PAS 3.	OP1	OP2	Retenue entrante complémentée
ADD, SUB...	00 0	11	00	11	AAAA	AAAA	0
	11 1	00	11	00	5555	5555	1
	00 1	00	00	00	0000	0000	1
	01 1	01	01	01	0000	FFFF	1
	10 0	10	10	10	FFFF	0000	0
	10 1	10	10	10	FFFF	0000	1
	11 0	11	11	11	FFFF	FFFF	0
	01 0	01	01	01	0000	FFFF	0
OR	00 1	00	00	00	0000	0000	(1)
	01 1	01	01	01	0000	FFFF	(1)
	10 1	10	10	10	FFFF	0000	(1)
	11 1	11	11	11	FFFF	FFFF	(1)
AND	00 0	00	00	00	0000	0000	(0)
	01 0	01	01	01	0000	FFFF	(0)
	10 0	10	10	10	FFFF	0000	(0)
	11 0	11	11	11	FFFF	FFFF	(0)
X-OR	00 1	00	00	00	0000	0000	(1)
	01 1	01	01	01	0000	FFFF	(1)
	10 1	10	10	10	FFFF	0000	(1)
	11 1	11	11	11	FFFF	FFFF	(1)
ROL	0	0	0	0	0000	-	(1)
	1	1	1	1	FFFF	-	(1)
	0	1	0	1	AAAA	-	(1)
	1	0	1	0	5555	-	(1)

Table IX. Vecteurs de test de classe II limitée géographiquement et réduite à un pas pour l'opérateur ALU du MC 68000.

III.2. Etude et test des prédicats de l'opérateur de calcul

Ce test ne s'intéresse pas à la gestion des indicateurs par la partie contrôle, mais à la génération de ces comptes-rendus par l'opérateur de calcul.

Les prédicats se répartissent en deux types qui correspondent à la taille des opérations exécutées sur l'opérateur de calcul :

- Opérations sur les octets (8 bits) générant les prédicats octets :
BN, BC, BV, LBZ ;
- Opérations sur les mots (16 bits) générant les prédicats mots : WN,
WC, WV, HBZ.

Malgré tout, les opérations de l'opérateur de calcul sont toujours effectuées sur 16 bits, la taille de l'opération permettant de sélectionner et de ranger le prédicat correspondant dans les indicateurs N, C, V, Z ainsi que le rangement dans les registres de la partie (octet ou mot) correspondant à l'opération.

Un test de classe 0 est envisagé pour les prédicats de l'ALU. La table X regroupe les vecteurs de classe 0 des prédicats de l'opérateur de calcul, l'opération effectuée étant une addition.

	Entrée A	Entrée D	HBZ	WV	WN	WC	LBZ	BV	BN	BC
Opération 16 bits	0000	0000	1				1			
	0000	00FF	1		0		0		1	
	4000	4000	0		1		1		0	
	2000	2000	0				1			
	1000	1000	0				1			
	0800	0800	0				1			
	0400	0400	0				1			
	0200	0200	0				1			
	0100	0100	0				1			
	0100	0000	0				1			
	FF80	FF80		0				1		
	7FFF	7FFF		1				0		
	8000	8000		1		1		0		0
	0080	0080		0		0		1		1
Opération 8 bits	0000	FF00	0		1		1		0	
	0040	0040	1		0		0		1	
	0020	0020	1				0			
	0010	0010	1				0			
	0008	0008	1				0			
	0004	0004	1				0			
	0002	0002	1				0			
	0001	0001	1				0			
	0001	0000	1				0			
	7FFF	7FFF		1				0		
	007F	007F		0				1		
	0080	0080		0		0		1		1
	8000	8000		1		1		0		0

Table X Vecteurs de test de classe 0 pour le test des prédicats de l'opérateur ALU, l'opération effectuée sur les entrées A et D étant une addition.

III.3. Acquisition des opérandes sur l'ALU

III.3.1. Présentation de l'entrée A de l'ALU

La figure 6 représente le pas d'acquisition de l'opérande A sur l'ALU, le pas 7 qui diffère de celui-ci sera étudié plus loin.

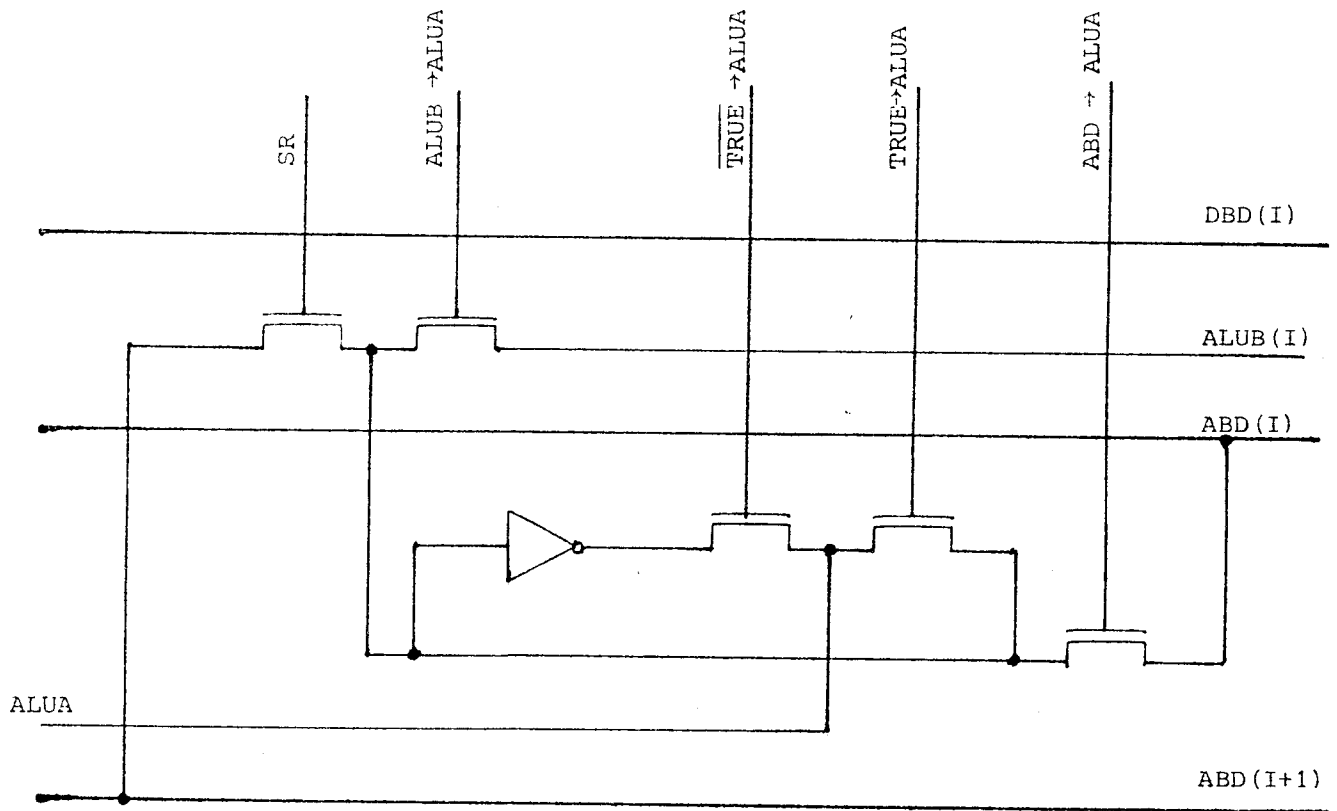


Figure 6. Acquisition de l'opérande A sur l'ALU pas I pour $I \in \{0, \dots, 6, 8, \dots, 15\}$

Remarque :

Lorsque la commande SR est active, l'entrée ABD (I+1), à prendre en compte pour le pas 15, est la ligne ALU[15]IN.

La commande ABD → ALUA permet d'entrer la valeur du bus ABD sur l'entrée A de l'ALU, elle est utilisée conjointement à l'une des deux commandes TRUE → ALUA ou $\overline{\text{TRUE}} \rightarrow \text{ALUA}$. Utilisée avec TRUE → ALUA, la commande ABD → ALUA permet d'effectuer les opérations ADD, OR, AND, X-OR, ROL sur la valeur du bus ABD.

Elle est utilisée avec $\overline{\text{TRUE}} \rightarrow \text{ALUA}$ pour effectuer la soustraction (complément à 1) pour laquelle la retenue entrante est égale à 1 ($\overline{\text{CIN}} = 0$) donnant ainsi la valeur du bus ABD en complément à deux.

La commande ALUB → ALUA permet d'entrer le contenu du buffer d'ALU sur l'entrée A de l'ALU.

La commande SR est utilisée pour le décalage à droite, elle est utilisée en même temps que dans l'ALUE (décalage de mot long).

III.3.2. Test de l'entrée A de l'ALU

Le test envisagé est un test de classe 0 réduite au défaut simple physique.

Le tableau XI regroupe les différents vecteurs nécessaires au test de l'acquisition de l'opérande A sur l'ALU.

$\overline{\text{CIN}}$	OP A	OP D	ABD	SL	C-AND	$\overline{\text{C-XOR}}$	$\overline{\text{C-OR}}$	SR
***0	0000	FFFF	(-)	0	0	1	11	(0)
0	FFFF	0000	(-)	0	0	1	1	(0)
** {	1 FFFF*	0000	(-)	0	0	1	1	(0)
	1 0000	FFFF	(-)	0	0	1	1	(0)
** {	1 0000*	FFFF	(-)	0	0	1	1	(0)
	1 FFFF	0000	(-)	0	0	1	1	(0)
(0)	(AAAA)	(FFFF)	5555	(0)	(0)	(1)	(1)	1

* Opérande en mémoire

** Ces deux vecteurs doivent se suivre dans cet ordre; ce sont les vecteurs 2 et 5 de la table VIII (test classe 0 de l'ALU).

*** Vecteur 1 de la table VIII. (test classe 0 de l'ALU).

Tableau XI. Vecteurs de test de classe 0 permettant de détecter les pannes sur l'acquisition de l'opérande A de l'ALU.

III.3.3. Cas particulier du bit 7 : extension octet

La figure 7 présente le cas particulier de l'acquisition de l'opérande A de l'ALU pour le bit 7.

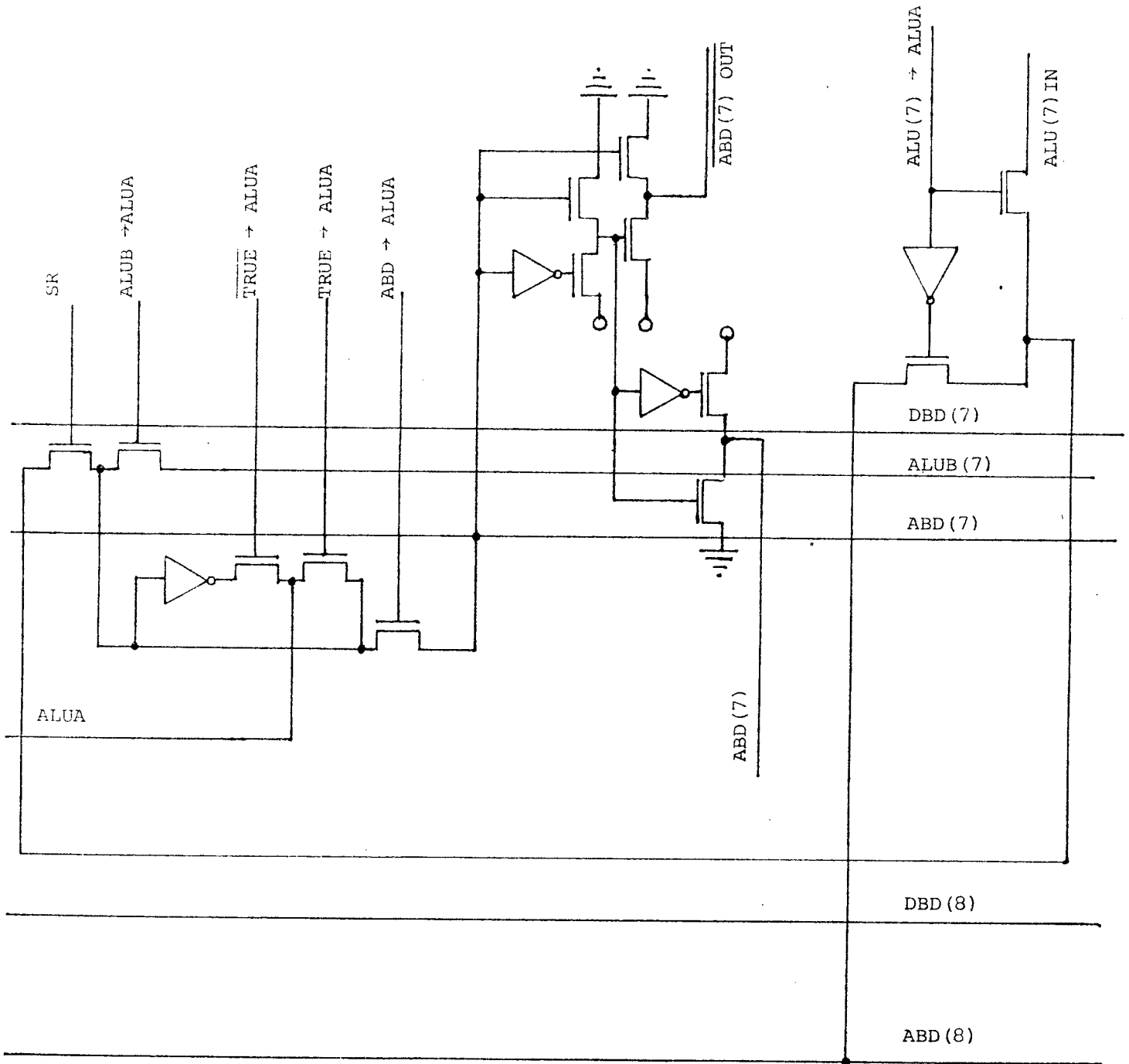


Figure 7. Acquisition de l'opérande A de l'ALU pour le bit 7

Un test de classe 0 est envisagé pour ce circuit. La table XII présente les vecteurs de classe 0 nécessaires pour le test du circuit d'extension octet.

INSTRUCTION	OP.	A	OP.	D
EXT.W	00	FF	-	
EXT.W	FF	00	-	
* ADD.W	0000		FFFF	
* SUB.W	0000		FFFF	
* Vecteurs 2 et 5 table VIII (test classe 0 de l'ALU)				

Table XII . Vecteurs de test de classe 0 du circuit d'extension octet du MC6800.

III.3.4. Décalage : cas particulier des bits 0, 7 et 15

Ce paragraphe va étudier particulièrement les entrées et sorties des bits 0, 7 et 15 lors des décalages, en effet pour ces opérations les entrées sont particulières pour ces bits, ils génèrent en outre des comptes-rendus pendant ces opérations. Pour ce test, nous emploierons un test de classe II, attaché au fonctionnement dans tous les cas possibles. La figure 9 regroupe les schémas des opérations de décalages et de rotations.

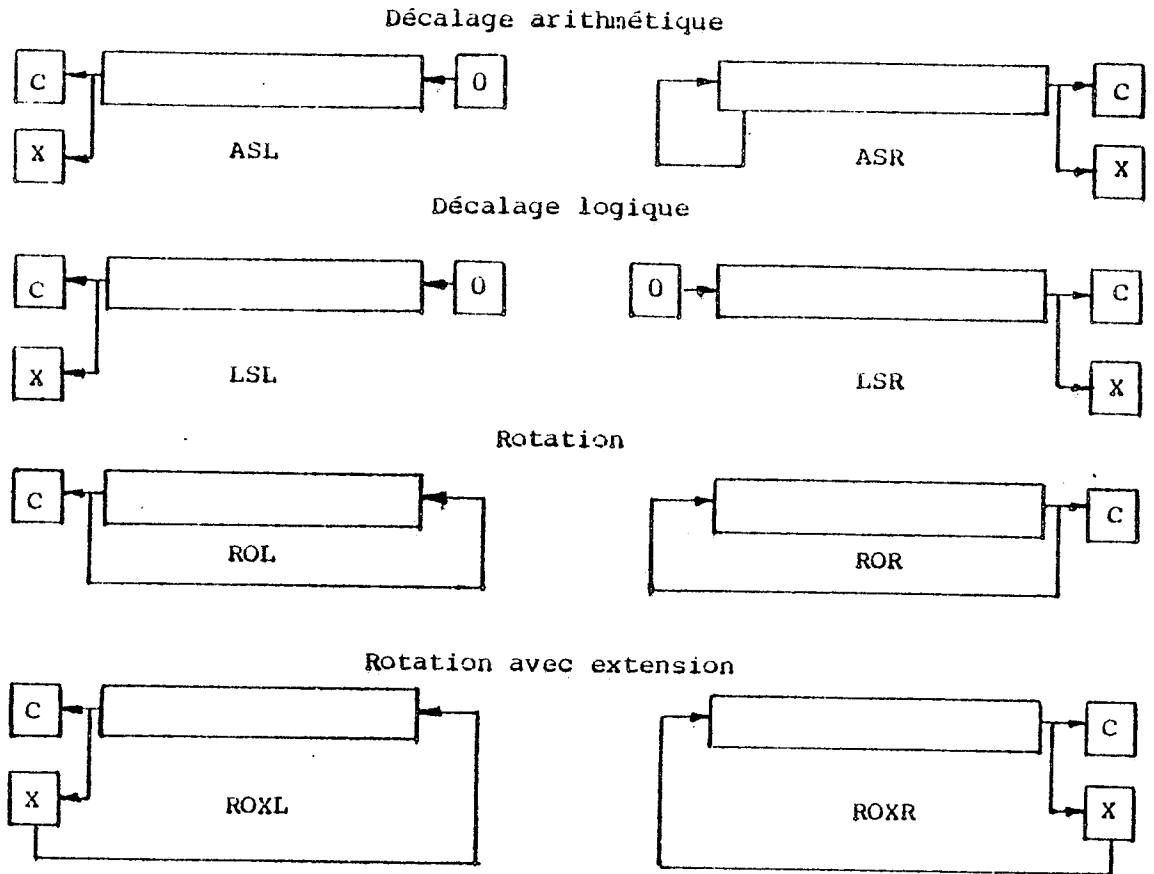


Figure 9. Schémas des opérations de décalages et de rotations du MC 68000

On peut remarquer, que l'ALU effectue les décalages sur 8 et 16 bits. Les décalages sur 32 bits sont obtenus par concaténation de l'ALU et de l'ALU-E (extension d'ALU traitée plus loin). En cas de décalage sur les registres les deux opérateurs sont chargés, l'opération est exécutée sur les deux, mais seul l'opérande de taille correcte est rangé dans le registre. Cette propriété va être utilisée pour le test qui nous intéresse. Seuls les bits 0, 7 et 15 sont testés ici les bits 16 et 31 sont abordés pour le test de ALUE, par contre leur influence sur les 0, 7 et 15 est testée dans ce paragraphe.

D'après la figure 9, on peut écrire :

- Le bit 0 peut prendre les valeurs : '0', X, bit 1, bit 7, bit 15, bit 31
- Le bit 7 peut prendre les valeurs : '0', X, bit 0, bit 6, bit 7, bit 8
- Le bit 15 peut prendre les valeurs : '0', X, bit 0, bit 14, bit 15, bit 16.

Un test de classe II limité géographiquement va être employé ici du fait de la non connaissance des commandes exactes manipulant ces bits.

La table XIII regroupe les vecteurs de test nécessaires au test de classe II limité géographiquement du circuit d'acquisition des bits 0, 7 et 15 et de compte-rendu pendant les décalages.

Les vecteurs sont classés selon les bits testés. Lorsque C et X sont mentionnés, le vecteur teste aussi le compte-rendu de l'opération. Les trois dernières paires de vecteurs testent uniquement le compte-rendu.

Instruction	Opérande en registre		X	C
ASL.L.	FFFF	FFFF	1	-
ROXL.L	FFFF	FFFF	0	-
	0000	0000	1	-
ASR.L	FFFF	FFFD	1	-
	0000	0002	0	-
ROL.B	FFFF	FF7F	1	1
	0000	0080	0	0
ROL.W	FFFF	7FFF	1	1
	0000	8000	0	0
LSR.B	FFFF	FFFF	1	-
ROXR.B	FFFF	FFFF	0	-
	0000	0000	1	-
ROR.B	FFFF	FFFE	1	-
	0000	0001	0	0
ASL.L	FFFF	FFBF	1	-
	0000	0040	0	-
ASR.B	FFFF	FF7F	1	-
	0000	0080	0	1

Table XVII. Vecteurs de test de classe II limitée géographiquement du circuit d'acquisition des bits 0, 7, 15 et des comptes-rendus pendant les décalages. (début).

Instruction	Opérande en registre		X	C
ASR.L	FFFF	FEFF	1	-
	0000	0100	0	-
LSR.W	FFFF	FFFF	1	-
ROXR.W	FFFF	FFFF	0	-
	0000	0000	1	-
ROR.W	FFFF	FFFE	1	1
	0000	0001	0	-
ASL.L.	FFFF	BFFF	1	-
	0000	4000	0	-
ASR.W	FFFF	7FFF	1	-
	0000	8000	0	-
ASR.L	FFFF	FFFE	1	1
	0000	0001	0	0
ROXL.B	FFFF	FF7F	1	1
	0000	0080	0	0
ROXL.W	FFFF	7FFF	1	1
	0000	8000	0	0

Table XVII. Vecteurs de test de classe II limités géographiquement du circuit d'acquisition des bits 0, 7, 15 et des comptes-rendus pendant les décalages. (fin).

III.3.5. Présentation de l'entrée D de l'ALU

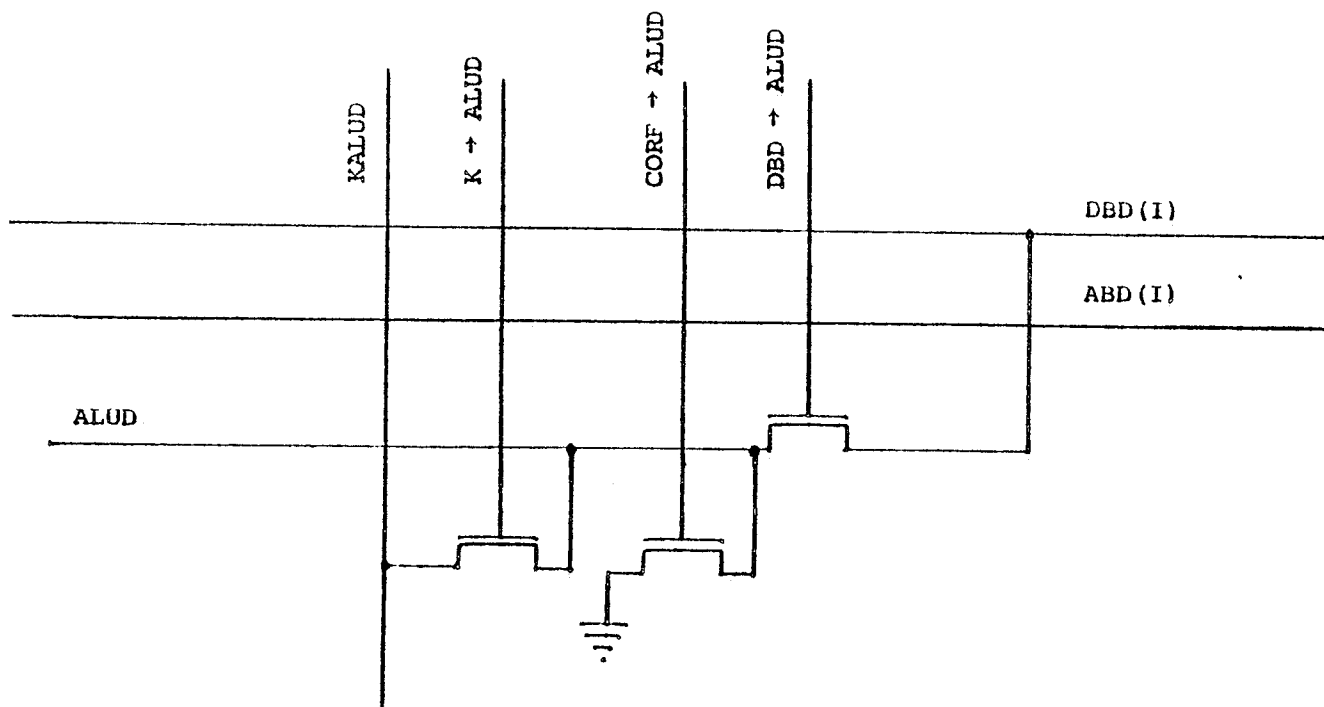


Figure 10. Acquisition de l'opérande D sur l'entrée de l'ALU pour les pas 8 à 15.

La figure 10 présente le pas de l'acquisition de l'opérande D sur l'entrée de l'ALU pour les pas 8 à 15. Pour les pas 0 à 7, l'acquisition reste la même seule la valeur de l'entrée en cas de commande CORF → ALUD active est modifiée et correspond à la sortie de l'opérateur BCD. Cet opérateur BCD sera étudié plus loin.

La commande DBD → ALUD valide l'entrée du but DBD sur l'entrée D de l'ALU permettant les opérations à deux opérands (l'opérande A était le premier opérande).

La commande CORF → ALUD active permet de générer la constante de correction décimale pour les calculs en DCB (opération sur octet seulement).

La commande K → ALUD active permet d'entrer la constante K dans l'ALU, cette constante permet les opérations à un opérande, telle que l'instruction ROL (constante K = 1) ou des instructions où l'opérande traverse seulement le circuit (constante K = 0) pour positionner les indicateurs, ect...

III.3.6. Test de l'entrée D de l'ALU

On s'intéresse à un test de classe 0 sur ce circuit.

La table XIV regroupe les vecteurs de test de classe 0 nécessaires au test de l'acquisition de l'opérande D de l'ALU. On remarque que seuls les 3 derniers vecteurs sont à rajouter à des vecteurs déjà obtenus. Les vecteurs choisis sont ceux du test de l'opérateur BCD (cf § III.4).

INSTRUCTION	OP . D	OP . D	VECTEUR/TABLE
ADD.W	0000	FFFF	1/VIII
SUB.W	FFFF	0000	2/VIII
ADD.W	AAAA	AAAA	3/VIII
SUB.W	5555	5555	4/VIII
* AND.W	0000	0000	7/VIII
* SUB.W	0000	FFFF	5/VIII
ASR.L	FFFF FFFD	-	4/XIII
ROL.W	FFFF 7FFF	-	8/XIII
EXT.W	00FF	-	1/XII
ABCD	4C	08	11/XV
ABCD	07	68	14/XV
ABCD	07	78	16/XV

* exécutées dans cet ordre, l'une derrière l'autre

Table XIV . Vecteurs de test de classe 0 de l'acquisition de l'opérande D sur l'ALU du MC68000.

III.4. Opérateur décimal (BCD)

III.4.1. Présentation de l'opérateur

La figure 11 présente l'opérateur décimal du MC 68000

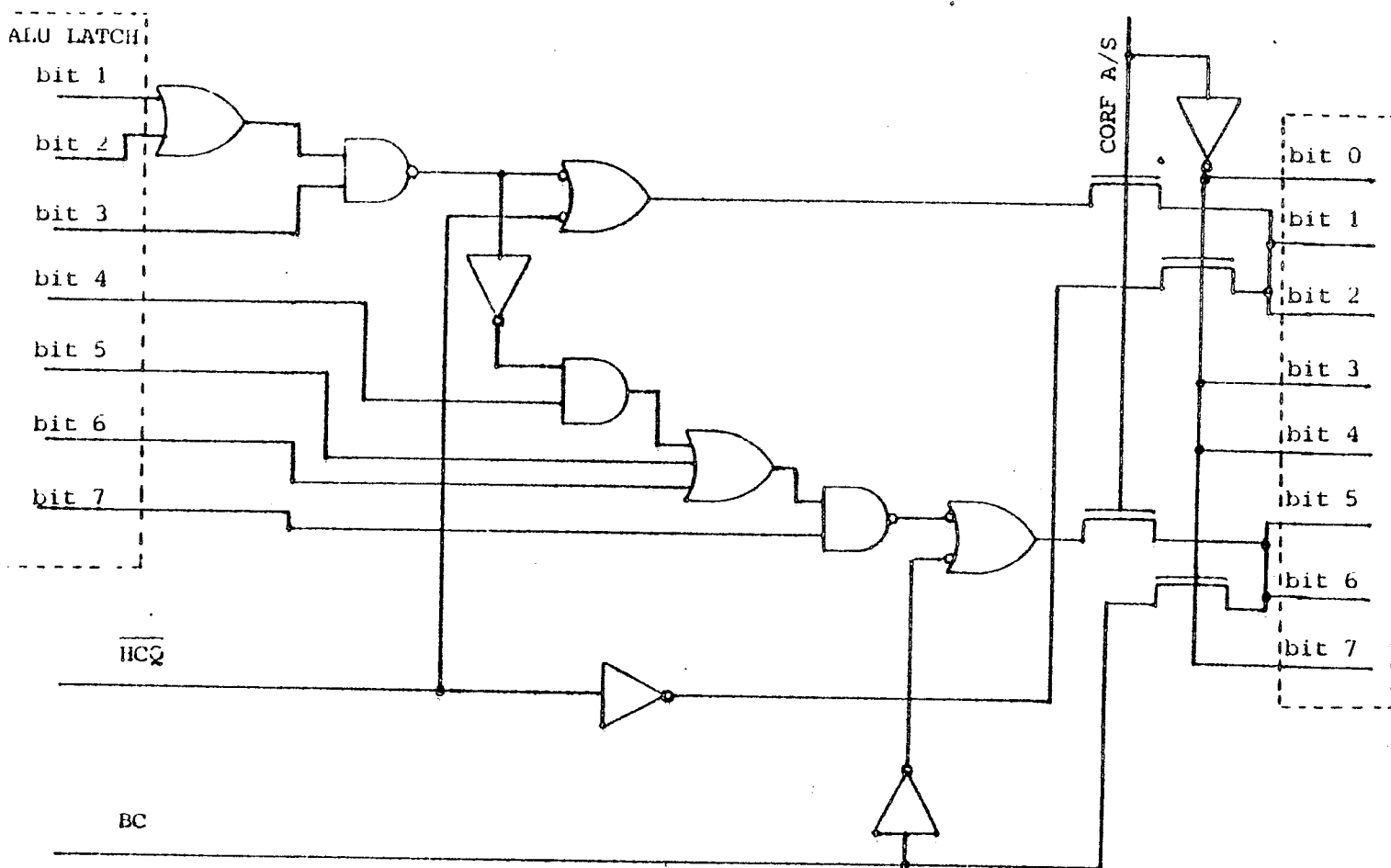


Figure 11. Opérateur décimal (BCD) du MC 68000

Les instructions décimales codées binaires (ABCD, SBCD, NBCD) nécessitent l'élaboration de constantes à ajouter au contenu de ALUR. La valeur de la constante à ajouter dépend de la valeur de l'opérande contenu dans ALUR et de la position des indicateurs HCQ (retenue bit 3) et BC (retenue bit 7).

Le circuit de génération de la constante est donc un circuit combinatoire recevant en entrée l'opérande contenu dans ALUR et les deux indicateurs.

Il est à noter que la taille de l'instruction DCB est l'octet. Ce circuit est identique à celui utilisé dans le M 6800 on reprendra donc le raisonnement utilisé pour le test du M 6800 [COU 81].

III.4.2. Test de l'opérateur

Un test de classe II limitée géographiquement est envisagé pour cet opérateur.

Le nombre de vecteurs de test à considérer pour un test exhaustif de ce circuit est grand. La structure de ce circuit permet de réduire fortement ce nombre.

La table XV présente les vecteurs de test de l'opérateur BCD pour le test de classe II limitée géographiquement.

OP1 et OP2 désignent les opérandes choisis pour les vecteurs de test du circuit correctif.

ALUR désigne le contenu de ALUR avant la correction décimale.

Résultat DCB désigne le contenu de ALU-R après la correction décimale.

Remarque :

On notera que tous les vecteurs de la table XV ne peuvent être obtenus par addition binaire de deux opérandes en DCB (par exemple le vecteur de test tel que le contenu de ALU-R soit égal à 38).

Chaque fois qu'il a été possible, les opérandes OP1 et OP2 ont été choisis de telle sorte qu'ils correspondent à des nombres DCB.

OP1	OP2	ALU-R(HEXA)	RESULTAT DCB
00	00	00	00
04	06	0C	12
08	08	10	16
0E	0E	1C	22
14	14	28	28
05	25	2A	30
2F	09	38	3E
2E	0C	3A	40
22	22	44	44
07	47	4E	54
4C	08	54	5A
4F	0F	5E	64
31	31	62	62
07	68	6F	75
09	69	72	78
07	78	7F	85
43	43	86	86
87	08	8F	95
8E	08	96	9C
97	08	9F	05
50	50	A0	00
57	58	AF	15
60	50	B0	10
67	58	BF	25
60	60	C0	20
68	62	CA	30
68	68	D0	36
78	62	DA	40
80	60	E0	40
88	64	EC	52
70	80	FO	50
78	87	FF	65
80	80	00	60

Table XX. Vecteurs de test de l'opérateur BCD du MC 68000 pour un test de classe II limitée géographiquement (début).

OP1	OP2	ALU-R(HEXA)	RESULTAT DCB
88	87	0F	75
90	80	10	70
98	87	1F	85
90	90	20	80
97	98	2F	35
98	92	30	36
BA	80	3A	A0
C0	80	40	A0
CA	80	4A	B0
D0	80	50	B0
DA	80	5A	C0
E0	80	60	C0
EA	80	6A	D0
F0	80	70	F0
FA	80	7A	E0
C0	C0	80	E0
CA	C0	8A	F0
D0	C0	90	F0
DA	C0	9A	00
E0	C0	A0	00
EA	C0	AA	10
F0	C0	B0	10
FA	C0	BA	20
E0	E0	C0	20
EA	E0	CA	30
F0	E0	D0	30
FA	E0	DA	40
F0	F0	E0	40
FA	F0	EA	50
F8	F8	F0	56
FE	FE	FC	62

Table XX. Vecteurs de test de l'opérateur BCD du MC 68000 pour un test de classe II limitée géographiquement (fin).

III.5. Extension de l'opérateur de calcul (ALU-E)

III.5.1. Présentation de l'opérateur

La figure 12 donne le schéma d'un pas de l'opérateur ALU-E.

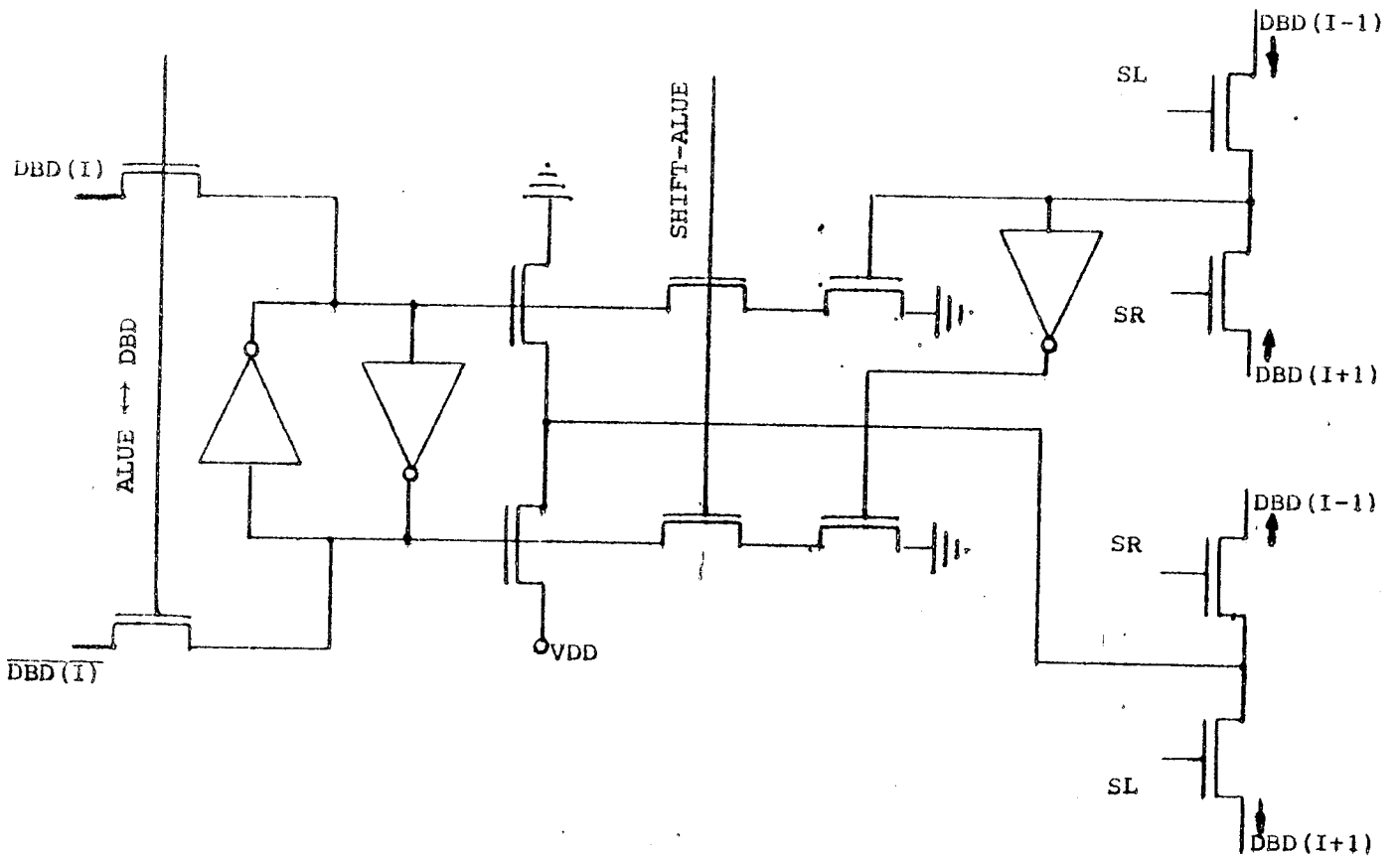


Figure 12. Pas de l'opérateur ALU-E.

Ce circuit est utilisé pour effectuer les décalages. En effet ceux-ci sont réalisés pour des opérandes en registres sur la concaténation ALUE + ALU quelle que soit la taille de l'opérande.

On distingue toutefois, toujours sur les registres, les opérations sur octet ou mot des opérations mot long. Les deux séquences distinctes ont même phase d'initialisation (chargement de ALUE et de ALU) mais différent par la nature du résultat qui se réduit au contenu de ALU latch dans le cas d'une opération sur octet ou mot. Pour les opérations avec opérande mémoire où la taille mot est la seule possible, seul l'ALU est utilisé.

La commande ALUE → DBD charge le bus DBD avec le contenu du registre ou bien réalise l'acquisition de l'opérande.

La commande SL ALUE génère les opérations de décalage gauche.

La commande SR ALUE génère les opérations de décalage droit.

La commande SHIFT ALUE valide le décalage.

Lorsque la commande SR ALUE est active (dans le cas d'un mot long) B 15 in ALUE désigne alors le bit entrant à la place du bit de poids fort dans ALUE.

Bo out ALUE désigne alors la valeur issue du bit n° 0 (de ALUE).

Lorsque la commande SL ALUE est active (dans le cas d'un mot long) B15 out ALUE désigne la valeur issue du bit n° 15.

Bo in ALUE désigne le bit entrant à la place du bit de poids faible de ALUE (bit 0).

III.5.2. Test de l'opérateur ALU-E

Un test de classe 0 est envisagé pour l'opérateur ALU-E.

La table XVI regroupe les vecteurs nécessaires du test de classe 0 limité à un pas de l'ALU-E.

Code	Opérande	Décalage de	Indicateur X
* ASL.L	FFFF FFFF	1	1
LSR.L	0000 0000 (registre)	1	-
LSR.L	AAAA AAAA	2	-
LSL.L	AAAA AAAA	2	-
ROL.L	0000 8000	17	-
ROR.L	0000 0001	17	-
ROXR.L	3333 3333	1	1
ROXR.L	CCCC CCCC	1	0
ROXL.L	3333 3333	1	1
ROXL.L	CCCC CCCC	1	0

* Vecteur 1 table XIII (test acquisition sur l'ALU).

Table XVI . Vecteurs de test de classe 0 d'un pas de l'opérateur ALU-E du MC68000.

III.5.3. Cas particuliers des bits 0, 1 et 15 :

Les bits 0 et 15 peuvent recevoir leur valeur de la partie contrôle ou des sorties de l'ALU. De plus les bits 0 et 1 génèrent des signaux de compte-rendu utilisés lors de la multiplication et de la division. Un test de classe II limité géographiquement est envisagé pour cette partie de circuit.

La table XVII regroupe les vecteurs de test nécessaires au test de classe II de l'entrée des bits 0 et 15 de l'ALUE ainsi que des signaux de compte-rendu associés. La dernière partie du tableau teste les comptes-rendus provenant du bit 0 et du bit 1.

Code	Opérande		X	C
ASL.L	FFFF	0000	-	-
ASL.L	0000	FFFF	-	-
ASR.L	0001	FFFF	-	-
ASR.L	FFFE	0000	-	-
LSR.L	FFFF	FFFF	1	-
ROXR.L	FFFF	FFFF	0	-
ROXR.L	0000	0000	1	-
ROR.L	FFFF	0000	1	-
ROR.L	0000	FFFF	0	-
ASL.L	8000	0001	1	-
ASL.L	7FFF	FFFE	0	-
ASR.L	7FFF	FFFF	1	-
ASR.L	8000	0000	0	-
ASR.L	0000	FFFF	1	-
ASR.L	FFFF	0000	0	-
ROL.L	0000	FFFF	1	-
ROL.L	FFFF	0000	0	-
ROL.L	0000	FFFF	1	1
ROL.L	FFFF	0000	0	0
ROXL.L	0000	FFFF	1	1
ROXL.L	FFFF	0000	0	0
Code	Opérande 1	Opérande 2	X	C
MUL.U	FFFF FFOC	FFFF FFOC	-	-
MUL.U	0000 000C	0000 000C	-	-
MUL.S	FFFF FFOC	FFFF FFOC	-	-
MUL.S	0000 000C	0000 000C	-	-

Table XVII. Vecteurs de test de classe II limités géographiquement des entrées BO IN ALUE et B15 IN ALUE ainsi que des signaux de compte-rendu de l'ALUE, lors des décalages et multiplications.

III.6. Opérateur de Multiplexage : MUX I/O

III.6.1. Présentation de l'opérateur

Ce mécanisme rend les opérations d'entrée/sortie d'octets indépendantes de la parité de l'adresse. Il recopie l'octet :

- poids fort, présent sur EDB8 à EDBF, sur EDB0 à EDB7

ou

- poids faible, présent sur EDB0 à EDB7, sur EDB8 à EDBF.

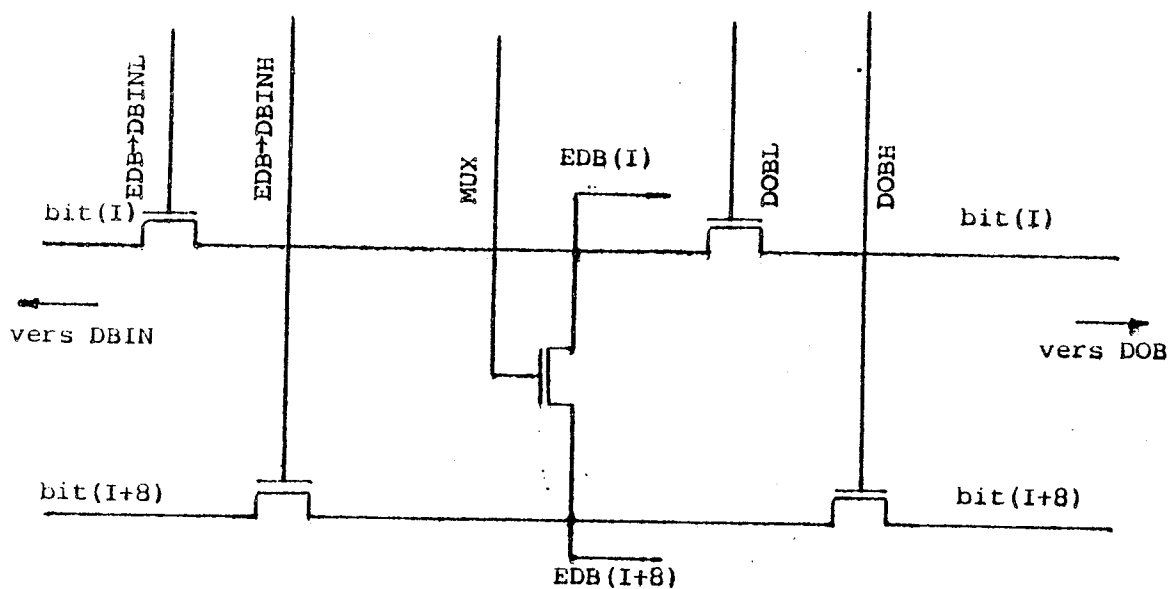


Figure 13. Pas de l'opérateur MUX I/O du MC 68000.

L'opérateur complet comprend 8 pas identiques à celui présenté à la figure 13, entrelacés.

Le fonctionnement est le suivant :

a) Lecture d'un mot (16 bits):

En lecture, les commandes DOBL et DOBH sont inactives. La donnée présente sur les plots arrive à l'opérateur par les lignes EDB(0) à EDB(15).

La commande MUX est inactive, les commandes d'entrée (EDB → DBINL et EDB → DBINH) sont actives et permettent à la donnée d'entrer dans le buffer d'entrée données (DBIN).

b) Lecture d'un octet (8 bits) :

En lecture, les commandes DOBL et DOBH sont inactives.

La donnée, présente sur les plots, arrive à l'opérateur par les lignes EDB(0) à EDB(7) si l'adresse est impaire, EDB(8) à EDB(15) si l'adresse est paire. Les lignes (EDB) non utilisées se trouvent à haute impédance. Dans ce cas la commande MUX est active, ainsi que la commande EDB → DBINL. La commande EDB → DBINH est inactive. Ainsi quelle que soit la parité de l'adresse, la donnée, présente soit sur les plots 0 à 7 ou 8 à 15, est entrée dans DBINL(0) à DBINL(7).

c) Ecriture d'un mot (16 bits) :

En écriture, les commandes DBINH et DBINL sont inactives.

La donnée présente sur les bits 0 à 15 dans le buffer de sortie données (DOB) est positionnée sur les lignes EDB(0) à EDB(15).

Les commandes DOBL et DOBH sont inactives, la commande MUX est inactive.

d) Ecriture d'un octet (8 bits) :

En écriture, les commandes DBINH et DBINL sont inactives.

La donnée présente sur les bits 0 à 7 dans le buffer de sortie (DOB) arrive à l'opérateur. La commande MUX est active, la commande DOBL l'est aussi, tandis que la commande DOBH ne l'est pas. La donnée est donc recopiée de EDB(0) à EDB(7) respectivement sur EDB(8) à EDB(15). Ainsi quelle que soit la parité de l'adresse, la donnée est accessible sur les plots 0 à 7 ainsi que sur les plots 8 à 15.

III.6.2. Test de l'opérateur

Un test de classe 0 est envisagé pour cet opérateur.

La table XVIII présente les vecteurs nécessaires au test de classe 0 de l'opérateur MUX I/O.

Entrée/ Sortie	MUX	DOBH	DOBL	DBINH	DBINL	EDB	DOB	AOB
S	0	1	1	-	-	-	FFFF	paire
S	1	0	1	-	-	-	0000	paire
S	1	0	1	-	-	-	FFFF	paire
S	0	1	1	-	-	-	00FF	paire
E	0	-	-	1	1	0000	-	paire
S*	0	1	1	-	-	-	0000	paire
E*	0	-	-	1	1	0000	-	paire

* ces deux vecteurs doivent être lancés dans cet ordre l'un derrière l'autre

Table XVIII. Vecteurs de test de classe 0 de l'opérateur MUX I/O du MC 68000

III.7. Opérateur registre décodeur (DCR)

III.7.1. Présentation de l'opérateur

La figure 14 présente le schéma de la partie registre de l'opérateur registre décodeur (DCR), la figure 15 présente la partie décodeur de cet opérateur.

Ce circuit est utilisé pour les instructions de manipulation de bits : BSET, BCLR, BCHG, BTST. Il fait la mise à 1 d'un bit du bus DBD parmi 16 dans le cas où l'on travaille sur le mot, parmi 32 dans le cas où l'on travaille sur le mot long.

Remarque :

Ce circuit fait la mise à 0 d'un bit sur le bus DBD (en parallèle).

La valeur du bit à modifier est codée sur les quatre bits ABD [0] à ABD [3] , ABD [4] étant utilisé pour indiquer si l'on travaille sur le mot ou le mot long.

La commande FOOL IT permet de distinguer l'opération sur octet (modulo 8) ou sur un mot (modulo 16), quand FOOL IT = 1 on travaille sur l'octet ; cette commande est alors utilisée par DCR pour ne pas prendre en compte le bit de poids forts ADB [3].

La commande ABD → DCR permet le chargement des cinq premiers bits du bus ABD dans le registre de DCR ainsi que la prise en compte du signal FOOL-IT.

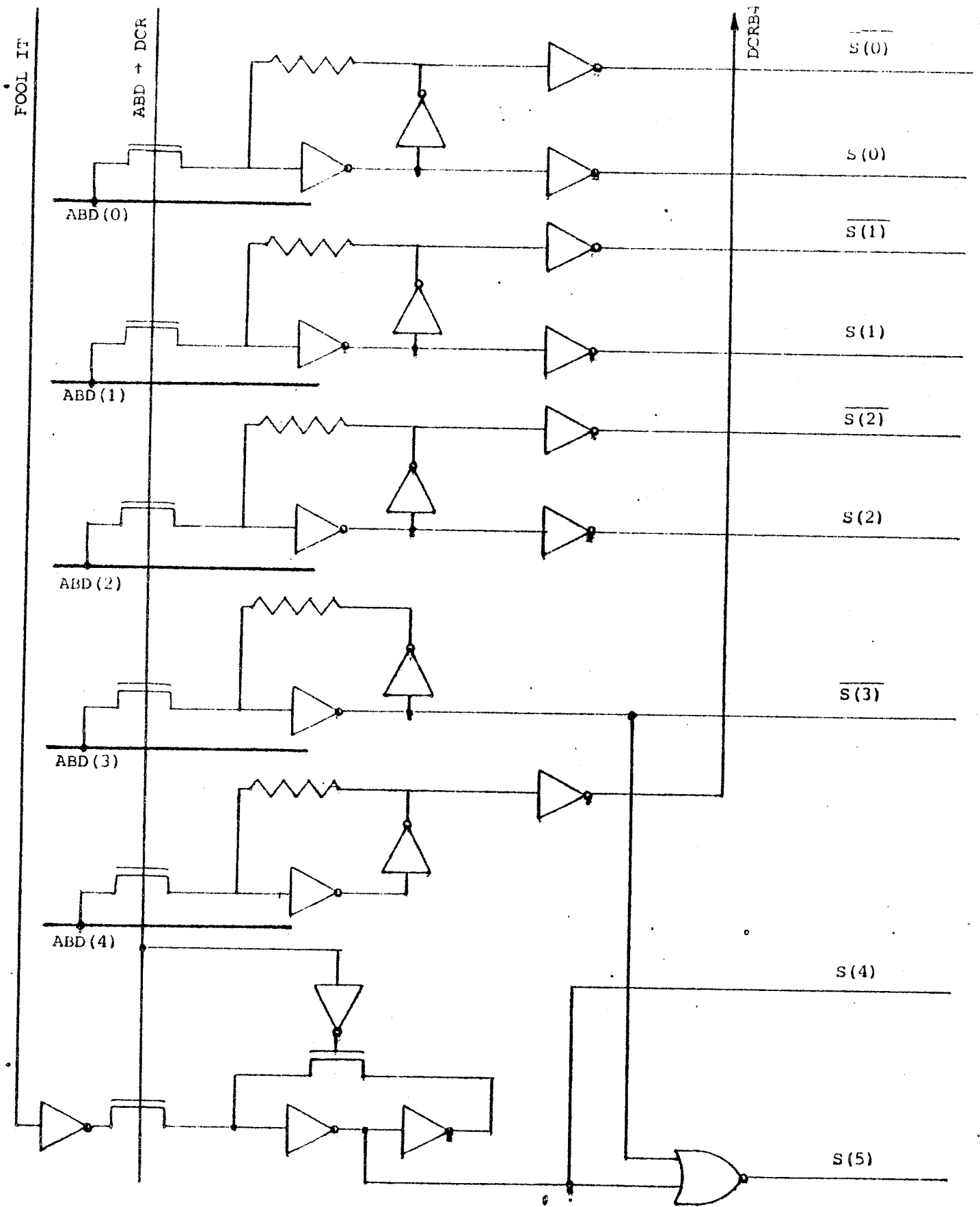


Figure 14. Schéma de la partie registre de l'opérateur DCR du MC 68000.

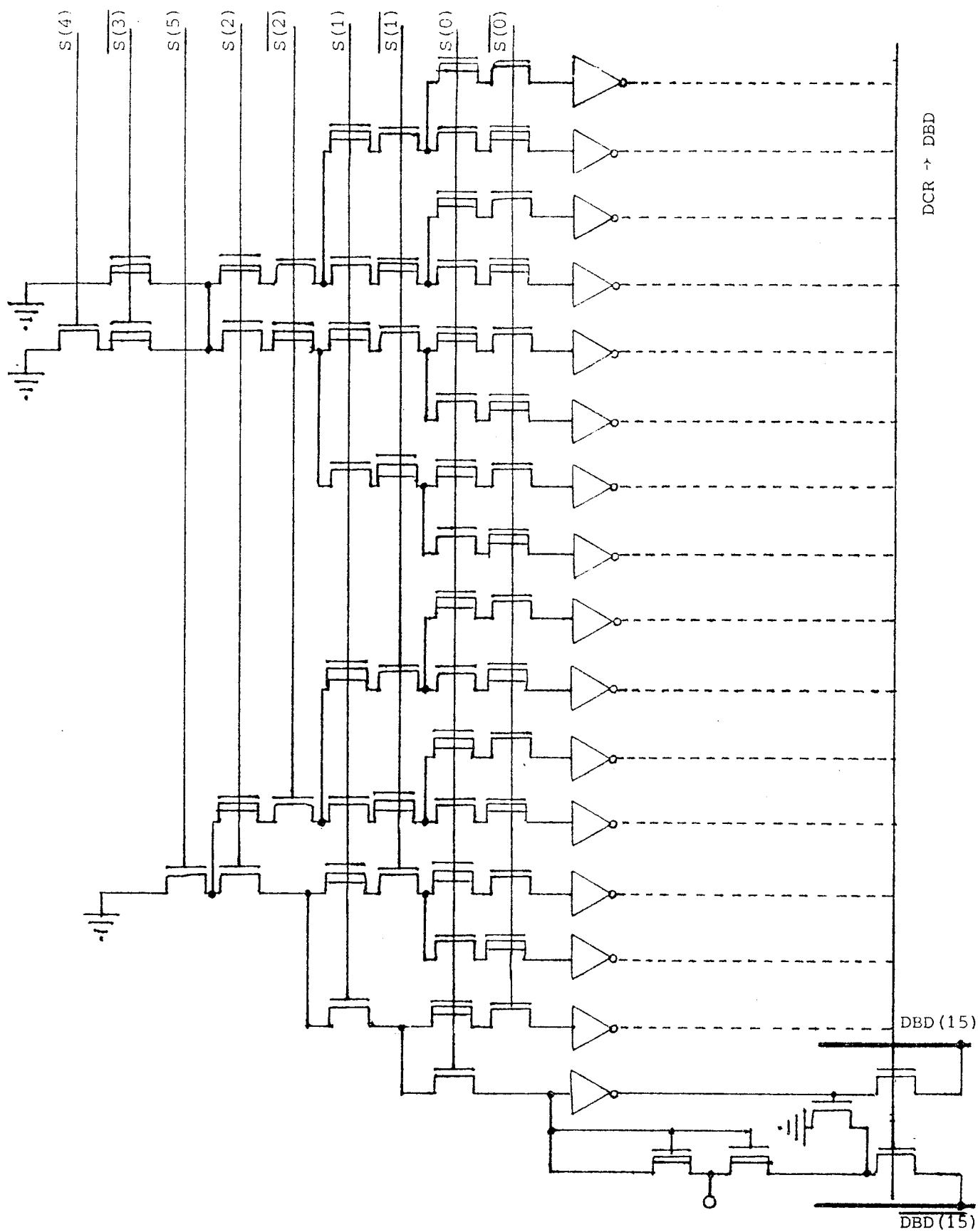


Figure 15 . Schéma de la partie décodeur de l'opérateur DCR du MC 68000.

Le signal DCR B4 est un compte-rendu de l'opérateur, précisant si l'on travaille sur le mot poids fort (bits 16 à 31) ou sur le mot poids faible (bits 0 à 15).

La commande DCR → DBD effectue la mise à 1, du bus complémenté ($\overline{\text{DBD}}$) et à 0 du bus direct (DBD).

REMARQUE

Ce circuit diffère selon l'époque de fabrication du MC 68000, mais le test utilisé ici étant un test de classe II, les légères modifications dans l'implantation de la fonction ne sont pas à prendre en compte.

III.7.2. Test de l'opérateur

Le test considéré ici est donc un test de classe II limitée géographiquement. Un test exhaustif de ce circuit serait important par le nombre de vecteurs à prendre en considération : 6 entrées, soit 2^6 vecteurs de test. La structure de ce circuit permet d'en réduire grandement le nombre.

En effet, seuls 18 vecteurs de test sont nécessaires, tels que présentés à la table XIX.

FOOL-IT	ABD [4]	ACD [3]	ABD [2]	ABD [1]	ABD [0]
0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	0	1	1
0	0	0	1	0	0
0	0	0	1	0	1
0	0	0	1	1	0
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	0	1
0	0	1	0	1	0
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	0	1
0	0	1	1	1	0
0	1	0	0	0	0
1	0	1	0	0	1

Table XIX. Vecteurs de test de classe II limitée géographiquement de l'opérateur DCR du MC 68000.

III.7.3. Acquisition et sortie de l'opérande :

Un test de classe 0 est envisagé pour l'acquisition des opérandes sur le bus A et la sortie des résultats sur le bus B. On s'aperçoit que les 18 vecteurs de test obtenus pour le test de l'opérateur lui-même sont nécessaires au test de l'acquisition de l'opérande.

Les vecteurs 1 et 2 du test de l'opérateur lui-même sont nécessaires au test de la sortie du résultat, à ces deux vecteurs vient s'ajouter l'instruction de la table XX.

La table XX ne regroupe que les deux instructions selon l'ordre dans lequel elles doivent être exécutées et ajoutées au test de l'opérateur pour effectuer le test complet de l'acquisition de l'opérande et de la sortie des résultats de l'opérateur DCR.

INSTRUCTION	OP 1	OP 2
* BTST R ₂ , 0	R ₂ = 0	0000
MOVE.W R ₀ , (R ₁)	R ₀ = 0	R ₁ = 0002
* Vecteur n° 2 table XIX.		

Table XX : Vecteurs de test de classe 0 à ajouter à tous les vecteurs de la table XIX pour le test de l'acquisition et sortie de l'opérateur DCR du MC 68000.

III.8. Opérateur FOOL-IT

III.8.1. Présentation de l'opérateur.

La figure 16 présente l'opérateur Fool-it du MC 68 000. Ce circuit est utilisé pour masquer les 8 bits poids forts du bus ABD, lorsque seuls les 8 bits poids faibles du bus ABD doivent être écrits dans l'octet poids faible d'un registre sans changer le contenu de l'octet poids fort de ce registre (opération sur un octet).

La commande LBYTE ABDSA active l'amplificateur de bus ABD [0] à ABD [7] en maintenant la ligne au niveau zéro, à la masse.

La commande HBYTE ABDSA active l'amplificateur de bus ABD [8] à ABD [15], elle joue le même rôle sur l'octet poids forts que la commande LBYTE ABDSA sur l'octet poids faibles.

La commande Fool-it crée une précharge sur l'octet poids fort, invalidant la donnée qui s'y trouve.

En effet, le fonctionnement de la lecture ou de l'écriture d'un registre est le suivant, en notant que la même commande effectue la lecture ou l'écriture :

- soit le registre est connecté sur un bus préchargé, il impose alors sa valeur au bus,
- soit le registre est connecté sur un bus amplifié, et ce dernier impose alors sa valeur au registre.

Donc ici,

Dans le cas où le registre doit être lu sur le bus ABD :

La précharge est activée sur tous les bits du bus (circuit non représenté sur la figure 22) et seule la commande ABD↔registre est valide.

Dans le cas où les 16 bits du bus ABD doivent être écrits dans le registre, les commandes LBYTE ABDSA, HBYTE ABDSA et ABD↔registre sont actives simultanément.

Dans le cas où seul l'octet poids faible du bus ABD doit être écrit dans le registre, les commandes Fool-it, LBYTE ABDSA et ABD↔registre sont actives simultanément.

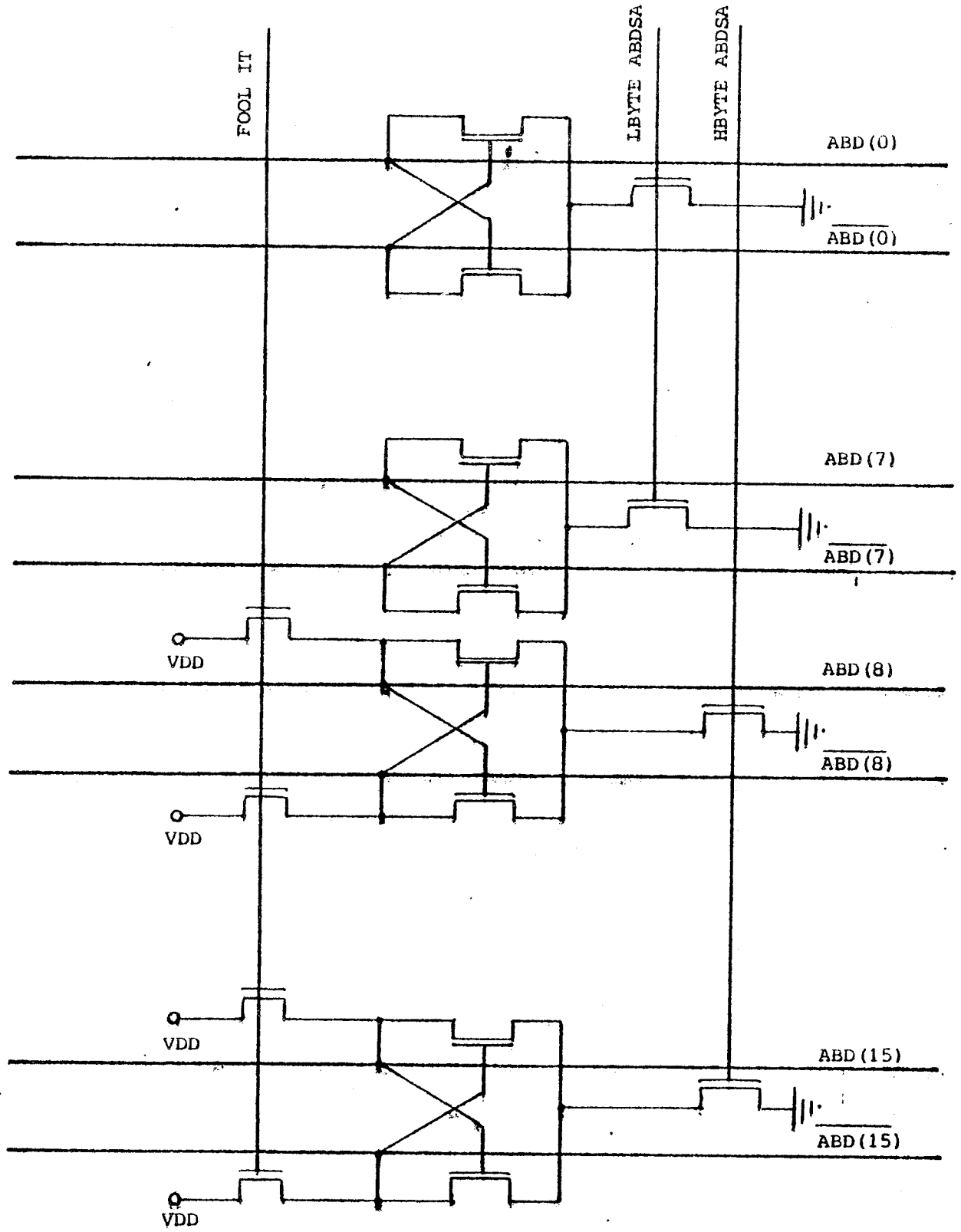


Figure 16 . Opérateur Fool-it du μ C 68000.

III.8.2. Test de l'opérateur

La lecture et l'écriture du registre (précharge et ampli) seront testées plus loin lors du test des registres. Ils seront donc supposés en bon état de fonctionnement.

Il reste à tester l'écriture de l'octet poids faible du bus ABD sur un registre quelconque, sans modification de l'octet poids fort de ce registre.

Un test de classe 0 est envisagé pour cet opérateur. En particulierisant les vecteurs de test, on retombe sur des vecteurs déjà obtenus (test des prédicats de l'ALU et test du MUX I/O) et ainsi, aucun vecteur nouveau n'est nécessaire.

La table XXI présente les vecteurs de classe 0 nécessaires au test de l'opérateur Fool-it.

INSTRUCTION	OP 1	OP 2	VECTEUR/TABLE
ADD.B	0000	FF00	15/X
MOVE.W R1, @	R1 = 0000	@ = 0	4 (ou 6)/XVIII
MOVE.W R1, @.	R1 = FF00	@ = 0	1/XVIII

Table XXI . Vecteurs de test nécessaires au test de classe 0 de l'opérateur Fool-it du MC68000.

III.9. L'opérateur PREN (Priority register encoder)

III.9.1. Présentation de l'opérateur

Le PREN est utilisé pour reconnaître le premier bit du bus ABL (en partant des poids faibles) qui est au niveau "1".

Ce circuit est constitué d'un registre 16 bits (dont un pas est représenté à la figure 17) suivi d'un encodeur sur 4 bits (cf. figure 18) qui indique (en binaire) à la partie contrôle, quel est le premier bit du bus ABL à "1".

Le PREN est utilisé essentiellement par l'instruction MOVEM (encodage du numéro de registre en fonction du masque de l'instruction).

La commande ABL + PREN permet de charger la valeur du bus ABL dans le registre.

La ligne ENO est une ligne de compte-rendu pour la partie contrôle. Elle vaut "1" lorsque tous les bits du registre sont à "1" (équivalent de tous les bits du bus ABL à "0"). Il a deux utilisations différentes :

- La sortie de l'encodeur (AR0, AR1, AR2, AR3) est à "1111" lorsque $ABL(15) = "1"$ ou lorsque tous les bits du registre sont à "1" (équivalent de tous les bits du bus ABL à "0"). Elle permet de différencier ces deux cas ;
- ENO sert de bit end pour indiquer la fin du traitement du masque.

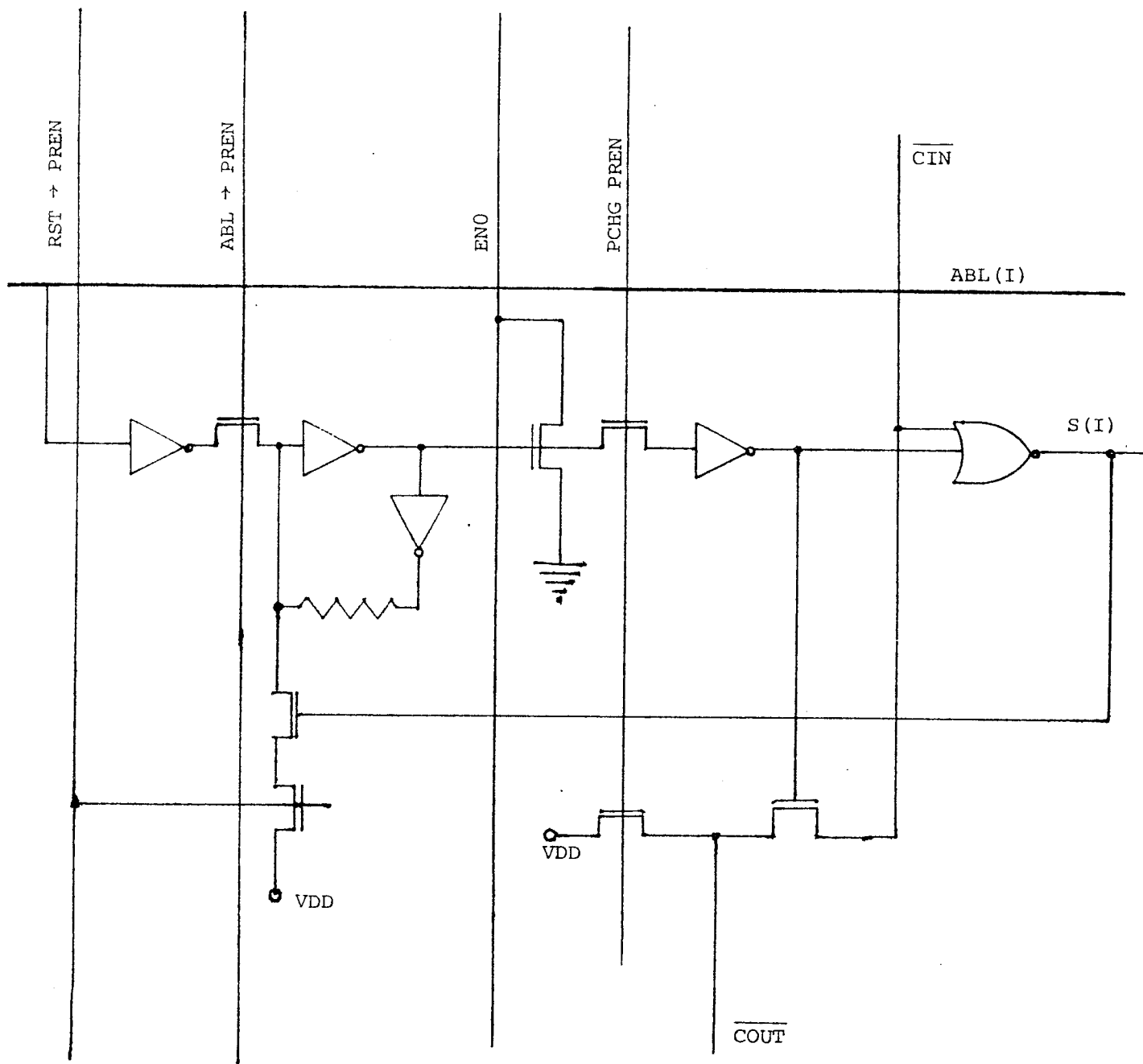


Figure 17. Pas de l'opérateur PREN du MC 68000 (côté registre).

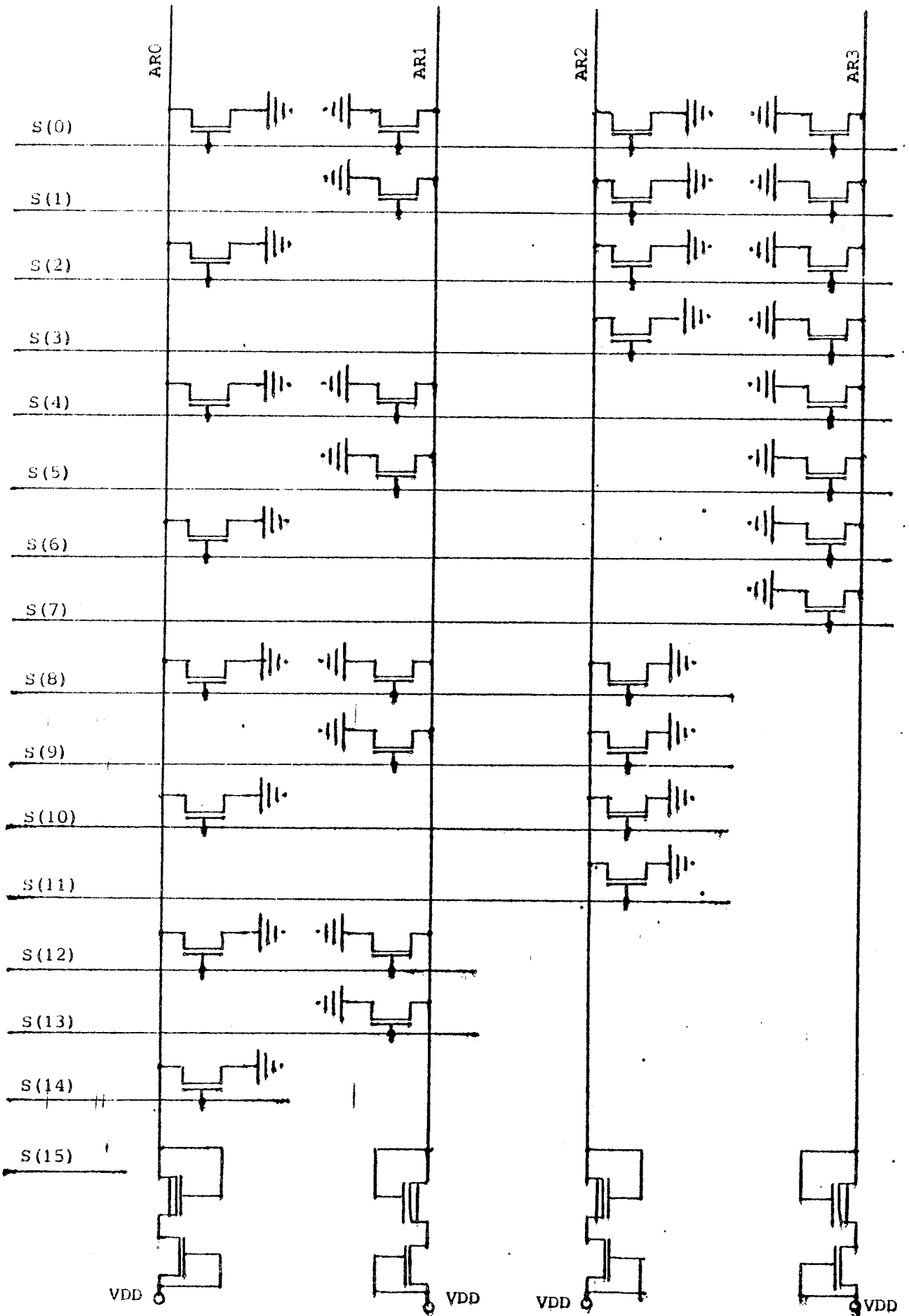


Figure 18. Pas de l'opérateur PREN du MC 68000 (côté encodeur).

La commande PCHG PREN permet de faire la précharge de chaque pas de l'opérateur.

La commande RST → PREN force à "1" le premier bit à "0" du registre quand elle est utilisée après la précharge.

Ainsi, par cette commande tous les bits du registre peuvent être examinés un par un en commençant par celui de poids le plus faible.

Il est à noter que $\overline{\text{CIN}}$ et $\overline{\text{COUT}}$ se comporte exactement comme $\overline{\text{CIN}}$ et $\overline{\text{COUT}}$ d'un pas d'ALU ($\overline{\text{CIN}}$ et à "1" quand il existe au moins un bit de ABL à "1" dans les étages précédents (de poids plus faible)).

Pour le pas 0, $\overline{\text{CIN}}$ est connectée à la masse.

La table **XXII** résume le fonctionnement d'un pas du registre du PREN.

Pour plus de clarté, c'est la valeur du bus qui est donnée (elle reflète la valeur de la sortie du registre qui est le complément de l'entrée de celui-ci).

On voit donc que :

$S(i) = 1$ si et seulement si le bit i du bus ABL est le premier bit (en partant des poids faibles) à "1".

ABL(1)	$\overline{\text{CIN}}(i)$	S	$\overline{\text{COUT}}(i)$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	0	1

Table **XXII** Fonctionnement d'un pas du registre du PREN.

III.9.2. Test de l'opérateur

Un test de classe 0 est envisagé pour l'opérateur PREN.

La table XXIII présente le test complet de classe 0 limité à un pas de l'opérateur PREN et de son prédicat.

Masque de l'instruction MOVEM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	*
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

* Le contenu du registre Ro = '0000 0000'

Table XXIII. Vecteurs de test de classe 0 limité à un pas du PREN et de son prédicat.

III.10. Opérateur de calcul d'adresses(AUH-AUL)

III.10.1. Présentation de l'opérateur

La figure 19 présente le pas de l'opérateur de calcul d'adresses. Les deux opérateurs AUH et AUL sont strictement identiques, AUH opérant sur les poids forts, AUL sur les poids faibles.

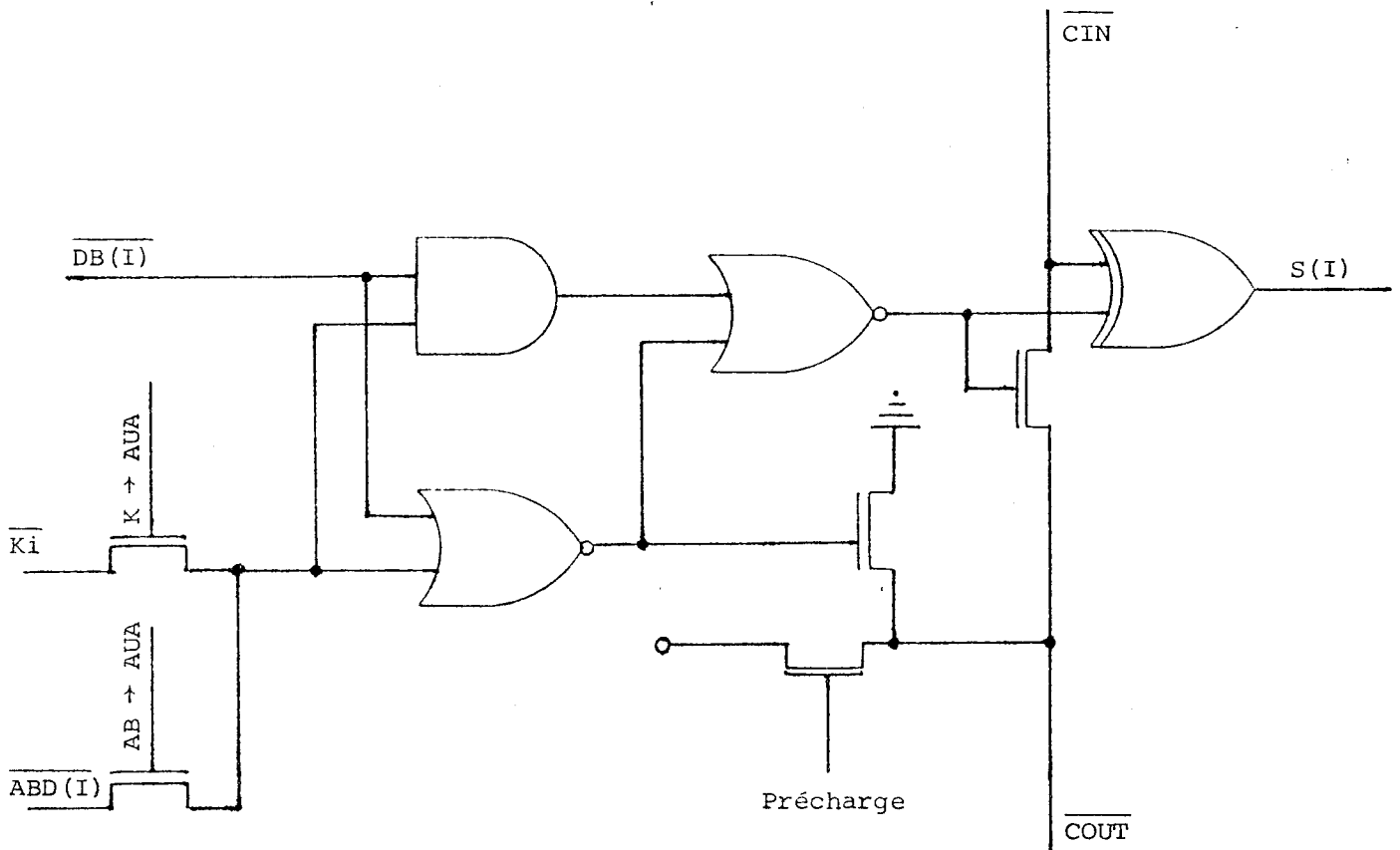


Figure 19. Pas de l'opérateur de calcul d'adresses (AUH, AUL) du MC 68000.

L'opérateur travaille en arithmétique complémentée, en addition par exemple le résultat est $\overline{a + b}$ où \overline{a} provient du bus $\overline{DBL(I)}$, et \overline{b} soit du bus $\overline{ABL(I)}$, soit de la constante \overline{Ki} .

La table **XXIV** présente le fonctionnement de l'opérateur de calcul d'adresses.

\overline{a}	\overline{b}	\overline{CIN}	\overline{COUT}	$S(I)$
0	0	0	0	0
0	0	1	0	1
1	1	0	1	0
1	1	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0

Table XXIV . Fonctionnement d'un pas de l'opérateur de calcul d'adresses.

Il est à noter que les unités arithmétiques AUH et AUL ne peuvent être utilisées séparément du fait de leur couplage les calculs ne s'effectuent que sur des mots longs (32 bits).

La retenue entrante de AUH est égale à la retenue sortante de AUL.

Bien que le bus d'adresses externe soit de 24 bits, le calcul d'adresses s'effectue sur 32 bits.

Il n'y a pas de retenue sortante sur le pas 31, les adresses sont donc calculées modulo 2^{32} . Et seuls les 24 premiers bits sont significatifs de l'adresse finale (sur les plots adresses).

Aucun dépassement de capacité n'est généré au niveau de AUL et AUH.

La figure 20 présente la juxtaposition des deux opérateurs de calcul d'adresses ainsi que la connexion des constantes.

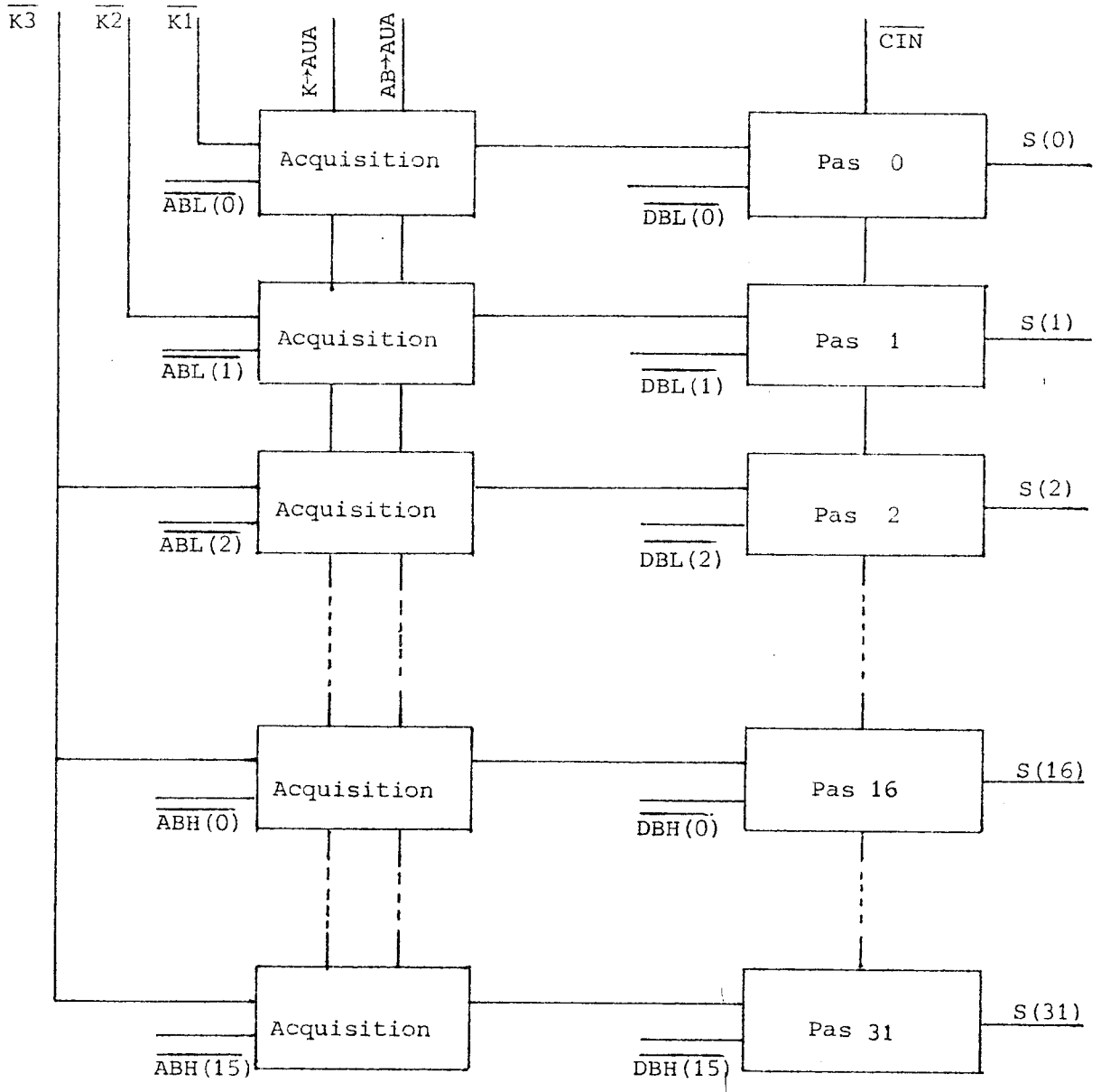


Figure 20. Fonctionnement de AUH + AUL et connexion des constantes.

Les constantes $\overline{K1}$, $\overline{K2}$ et $\overline{K3}$ sont utilisées pour incrémenter une adresse de +4, +2, +1, 0, -1, -2, -4 (pour accéder à l'octet, au mot, au bien, au mot long suivant).

(A une adresse mémoire correspond un octet).

La table XXV donne l'opération générée en fonction des constantes $\overline{K3}$, $\overline{K2}$, $\overline{K1}$ et de la retenue entrante \overline{CIN} .

$\overline{K3}$	$\overline{K2}$	$\overline{K1}$	\overline{CIN}	OPERATION
1	1	1	1	+0 → AUA
1	1	0	1	+1 → AUA
1	1	0	0	+2 → AUA
1	0	1	1	+2 → AUA
1	0	0	0	+4 → AUA
0	1	1	1	-4 → AUA
0	0	1	1	-2 → AUA
0	0	1	0	-1 → AUA
X	X	X	1	+AB → AUA

Table XXV . Opérations réalisées en fonction de la constante et de la retenue entrante.

De ce tableau il faut remarquer que l'on ne peut pas générer la retenue entrante \overline{CIN} à 0 lors de l'addition de deux adresses (AB + AUA).

III.10.2. Test de l'opérateur

Le signal de compte-rendu $AU = 0$, utilisé par la partie contrôle est généré à la sortie du latch de l'opérateur AU poids faible (AUL-Latch) et sera étudié lors du test de ce registre.

Le test utilisé va être un test de classe II limitée géographiquement, réduit uniquement à un pas de l'opérateur de calcul d'adresses AUH et AUL.

La table XXVI présente les vecteurs nécessaires au test de classe II limitée géographiquement et réduite uniquement à un pas de l'opérateur de calcul d'adresses. La table XXVII présente les vecteurs nécessaires au test de classe 0 d'un pas de l'opérateur de calcul y compris l'acquisition des opérands.

CIN	PAS 0	PAS 1	PAS 2	PAS 3	N° vecteur	OPER 1 DB	OPER 2 AB	RESULTAT
	K1	K2	K3	OPER 1				
1	0	0	0	0	1	0000	0000	0000 0000
1	0	1	0	1	2	AAAA	AAAA	5555 5555
1	1	0	1	0	3	5555	5555	AAAA AAAA
1	1	1	1	1	4	FFFF	FFFF	FFFF FFFF
1	0	0	1	0	5	0000	FFFE	FFFF FFFF
1	0	1	0	1	6	0000	FFFF	0000 0000
1	0	1	1	0	7	FFFF	0000	FFFF FFFF
1	1	0	1	1	8	FFFF	0000	0000 0000
CIN	K1	K2	K3			OPER 1	OPERATION	
0	0	X	X	X	9	XXXX	XXXX	OPER 1 +4
0	1	X	X	X	10	XXXX	XXXX	OPER 1 -1
0	0	X	X	X	11	XXXX	XXXX	OPER 1 -1
0	1	X	X	X	12	XXXX	XXXX	OPER 1 +4

Table XXVI Vecteurs de test nécessaires pour un test de classe II limitée géographiquement et réduite uniquement à un pas de l'opérateur de calcul d'adresses (AUH et AUL) du MC 68000.

\overline{CIN}	$\overline{K3}$ $\overline{K2}$ $\overline{K1}$	OPER	\overline{AB}	OPER	\overline{DB}
1	(1 1 1)	0000	0000	0000	0000
1		AAAA	AAAA	AAAA	AAAA
1		5555	5555	5555	5555
1		FFFF	FFFF	FFFF	FFFF
1		0000	0000	FFFF	FFFE
1		0000	0000	FFFF	FFFF
1		FFFF	FFFE	0000	0000
1		FFFF	FFFF	0000	0000
0	1 0 0			0000	0000
0	0 0 1			FFFF	FFFF
0	0 0 1			0000	0000
0	1 0 0			FFFF	FFFF
0	1 0 1	(0000	0002)	PC courant *	
1	(1 1 0)	0000	0001	0000	0000
1	(1 0 1)	0000	0002	0000	0000
1	(1 0 0)	0000	0003	0000	0000
1	(0 1 1)	FFFF	FFFC	0000	0000

* exécuté pendant adressage absolu long d'un mot

Table XXVII Vecteurs de test de classe 0 de l'opérateur de calcul d'adresses du MC 68000 (1 pas de l'opérateur et acquisition des opérands).

III.11. Opérateur d'extension de signe (SIGNEX)

III.11.1. Présentation de l'opérateur

La figure 21 présente l'opérateur SIGNEX du MC 68000, opérateur d'extension de signe du mot (16 bits) au mot long (32 bits).

Le mot long sera la concaténation des bus DBL (poids faibles) et DBH (poids forts) en cas d'extension de signe sur le bus DB, ou la concaténation des bus ABL (poids faibles) et ABH (poids forts) en cas d'extension de signe sur le bus AB.

ABSWHL et DBSWHL sont les commandes des switches de bus et permettent ou non la commutation des bus ABL et ABH d'une part, DBL et DBH d'autre part. Dans le cas où l'extension de signe doit être faite, il est bien évident que ces dernières commandes sont inactives de façon à ce que les bus ABH et ABL soient indépendants; idem pour DBH et DBL.

La commande SIGNEX → DBH permet d'effectuer l'extension de signe du bus DBL vers le bus DBH.

La commande SIGNEX → ABH permet d'effectuer l'extension de signe du bus ABL vers le bus ABH.

Lorsque la commande SIGNEX → DBH (resp. SIGNEX → ABH) est à '0' alors la valeur du bit 15 du bus DBL (resp. ABL) est reportée sur tous les bits du bus DBH (resp. ABH).

III.11.2. Test de l'opérateur

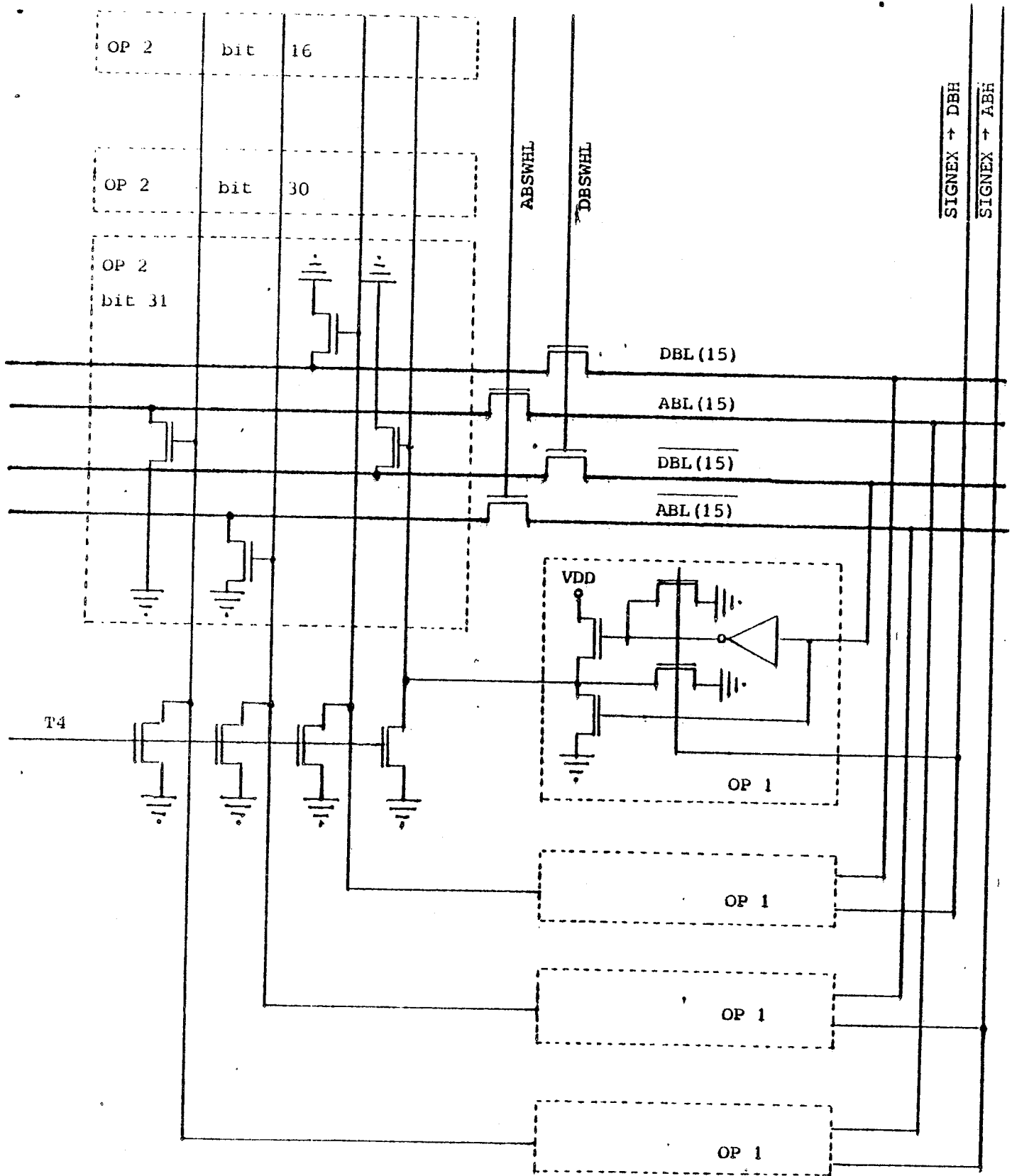


Figure 21. Schéma de l'opérateur d'extension de signe (SIGNEX) du MC 68000.

Pour chacun des quatre composants des circuits d'extension de signe, un test de classe II est envisagé.

La table XXVIII présente les vecteurs nécessaires au test de classe II limitée géographiquement de l'opérateur complet d'extension de signe.

: ABL [0] à ABL [14]	: ABL [15]	: Résultat i = 0 à 15
: 0	: 1	: ABH [i] = 1
: 1	: 0	: ABH [i] = 0
: DBL [0] à DBL [14]	: DBL [15]	: Résultat i = 0 à 15
: 0	: 1	: DBH [i] = 1
: 1	: 0	: DBH [i] = 0
: INSTRUCTION	: CONDITION	
: MOVE.L R0, (R1)	: R0 = 0000	0000
: CMP.L R0, R1	: R0 = FFFF	FFFF
	: R1 = FFFF	FFFF

Table XXVIII. Vecteurs de test de classe II limitée géographiquement de l'opérateur d'extension de signe (SIGNEX) du MC 68000.

IV. ETUDE ET TEST DU BUS

Il est nécessaire de tester le bus afin de savoir si les transferts se font correctement. En effet le principe de bus adopté pour le MC 68000 est complètement différent de celui du M 6800.

Le M 6800 possédait un bus unifilaire à précharge. Les points mémoire étant des points statiques dissymétriques (à cause de la résistance), possédaient un MOS de lecture et un MOS d'écriture.

Pour le MC 68000, le principe de bus utilisé est le bus bifilaire (ou bus différentiel), c'est-à-dire bus direct et bus complémenté. Il possède plusieurs types de points de mémorisation. Certains registres sont organisés en RAM, le point est symétrique et possède un seul MOS de lecture ou d'écriture, d'autres registres sont identiques à ceux du M 6800 et possèdent deux MOS, un de lecture, un d'écriture.

Chaque bus sera testé séparément des deux autres; en effet, certaines pan-
nes sur une des commandes de la gestion d'un bus transparentes en cas de
connexion de deux ou des trois bus, l'autre ou les deux autres corrigeant
l'effet de cette panne.

On rappelle que le bus EDB, de transfert des données vers (depuis) les plots données, le bus EAB, de transfert des adresses vers les plots adresses, le bus BC, de transfert de signaux de contrôle entre la partie contrôle et le registre FTU, ne sont pas pris en compte ici, mais sont considérés comme externes à la partie opérative.

IV.1. Présentation du bus

Le MC 68000 possède donc deux bus bifilaires de base, le bus AB et le bus DB chacun pouvant être segmenté en trois tronçons par des switches commandés depuis la partie contrôle.

Chaque bus possède deux switches qui lui confère quatre configurations possibles, comme le montre la figure 22, par exemple pour le bus AB :

<u>ABH</u>	<u>ABL</u>	<u>ABD</u>	Les deux switches ABSWML et ABSWLD inactifs
<u>ABH</u>	<u>ABL + ABD</u>		
<u>ABH + ABL</u>	<u>ABD</u>		Le switch ABSWML est inactif Le switch ABSWLD est actif
<u>ABH + ABL + ABD</u>			
			Le switch ABWHL est actif Le switch ABSWLD est inactif
			Les deux switches ABSWHL et ABSWLD actifs.

Figure 22. Configurations disponibles du bus par l'intermédiaire des switches de bus du MC 68000.

La figure 23 présente l'ensemble des circuits utilisés pour chaque tronçon de bus.

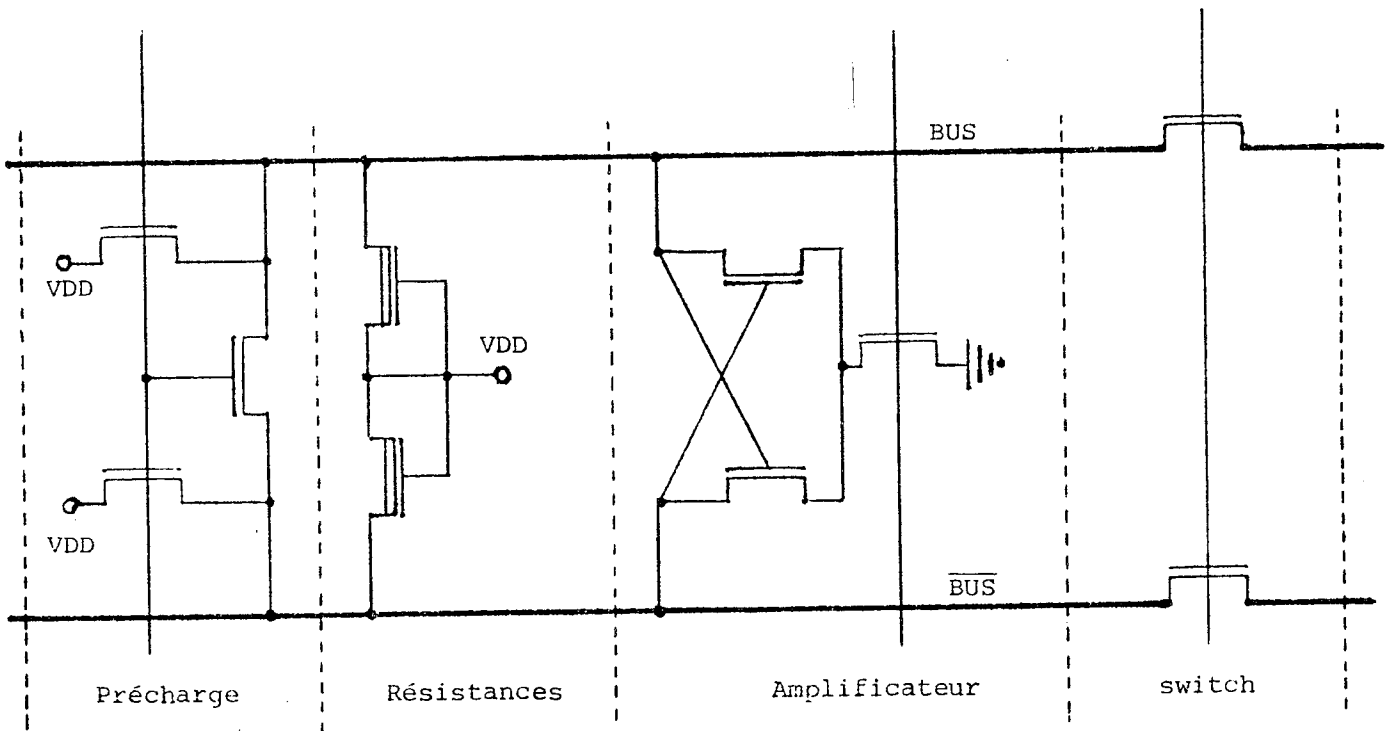


Figure 23. Composition des circuits utilisés pour la gestion des bus internes du MC 68000.

La figure 24 présente la connexion du point de mémorisation symétrique sur ce bus.

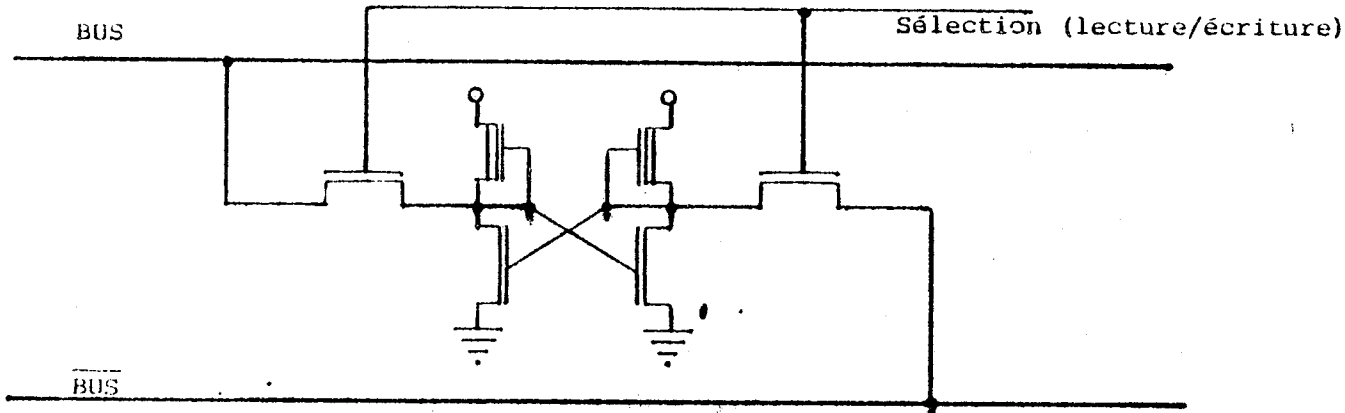


Figure 24. Connexion du point de mémorisation symétrique sur le bus.

La figure 25 présente le chronogramme de lecture et d'écriture d'un registre sur un bus.

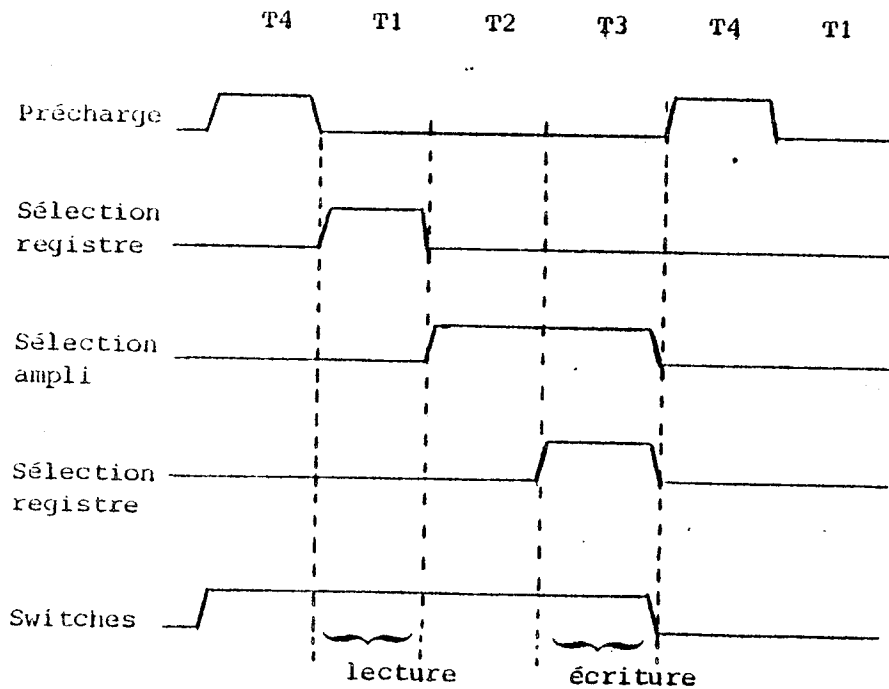


Figure 25. Chronogramme de l'utilisation des circuits du bus en lecture et en écriture pour le MC 68000.

Lorsque le registre est connecté sur un bus préchargé, il impose sa valeur au bus (lecture), s'il est connecté à un bus dont la valeur est maintenue par un ampli, l'ampli force la valeur du bus dans le registre (écriture).

IV.2. Test de la précharge

La précharge, comme indiquée à la figure 25, possède trois MOS, deux d'entre-eux relient respectivement les bus direct et complémenté à l'alimentation. Le troisième MOS sert à équilibrer les lignes ; en effet, à la suite d'un transfert une des lignes est à '1', l'autre à '0', grâce à ce troisième transistor, les deux MOS de précharge débitent autant l'un que l'autre pendant la précharge.

La table XXIX présente les instructions nécessaires au test de classe 0 du circuit de précharge des bus de la partie opérative du MC 68000.

Opération	Ro	R1	Répétition
LEA (Ro), R1 puis CMP.L R1, FFFF FFFF	FFFF FFFF	∅	2 fois
LEA (Ro), R1 puis MOVE.L R1, @ mémoire	0000 0000	∅	2 fois
MULU R1, Ro	∅∅∅∅ FFFF	∅∅∅∅ 1000	1 fois
MULU R1, Ro	∅∅∅∅ FFFF	∅∅∅∅ FFFF	1 fois
MOVE.L 0000 0000, R1	∅	∅	2 fois
MOVE.W (Ro), (R1)	∅ (Ro) = 0000	∅	2 fois
ADD.L Ro, R1	0000 0000	0000 0000	1 fois
ADD.L Ro, R1	FFFF FFFF	FFFF FFFF	1 fois
CMP.L Ro, R1	0000 0000	0000 0000	1 fois

Table XXIX . Vecteurs de test de classe 0 nécessaires au test du circuit de précharge des bus de la partie opérative du MC 68000.

IV.3. Test des résistances

Les résistances sont utilisées pour maintenir à '1' le bus qui s'y trouve après la lecture d'un registre, elles contre-balaçent les fuites du bus. Elles ont aussi tendance à faire monter légèrement le bus qui est à '0', mais dès que l'ampli est connecté, tout se rétablit correctement.

Il est à noter que la coupure des résistances ne provoque pas obligatoirement une erreur mais il y a un risque d'avoir une erreur. Si une erreur est provoquée, alors les vecteurs suivants la détecteront.

La table XXX regroupe les vecteurs de test de classe 0 nécessaires au test des résistances des lignes de bus de la partie opérative du MC 68000.

Opération	Ro	R1
LEA (Ro), R1	(Source) FFFF FFFF	∅
ADD.W Ro, R1	(Registre données) FFFF	(Registre données) FFFF
LEA (Ro), R1	(Source) 0000 0000	∅
MOVE.L 0000 0000, (Ro)	∅	-
MOVE.W (Ro), (R1)	∅ (Ro) = 0000	∅

Table XXX . Vecteurs de test de classe 0 des résistances de lignes du bus de la partie opérative du MC 68000.

IV.4. Test de l'amplificateur

L'amplificateur est utilisé pour écrire dans un registre ; une fois la lecture d'un registre terminée, l'amplificateur va permettre de bien séparer les lignes en maintenant une masse sur le bus à 'zéro', ce qui annule l'effet de la résistance correspondante, le registre destiné à être écrit est connecté sur le bus qui lui impose sa valeur par l'intermédiaire de l'ampli.

La table **XXXI** regroupe les vecteurs de test de classe 0 nécessaires au test du circuit d'amplification des lignes du bus de la partie opérative du MC 6800

Instructions	Ro.	R1	R2
LEA (Ro), R1	(source) FFFF FFFF	∅	∅
LEA (R2), R1	-	∅	(source) 0000 0000
MULU R1, R0	FFFF	1000	-
MULU R1, R2	-	FFFF	FFFF
LEA (R0), R1	(source) 0000 0000	∅	∅
LEA (R2), R1	-	∅	(source) FFFF FFFF
MOVE.L #0000 0000, (R1)	∅	∅	∅
MOVE.L #FFFF FFFF, (R1)	∅	∅	∅
MOVE.W (R0), (R1)	∅ (R0) = 0000	∅	-
MOVE.W (R2), (R1)	-	∅	∅ (R2) = FFFF
ADD.L Ro, R1	FFFF FFFF	FFFF FFFF	
MOVE.L (0000 0000.L), (Ro)	∅	∅	∅

Table **XXXI**. Instructions nécessaires au test de classe 0 du circuit d'amplification des lignes de bus de la partie opérative du MC 68000.

IV.5. Test des switches

Chacun des deux bus possède quatre switches qui lui permettent de partitionner ce bus en trois morceaux :

- SWHL et SWHL séparent BUSH et BUSL
- SWLD et SWLD séparent BUSL et BUSD.

La table **XXXII** regroupe les instructions nécessaires au test de classe 0 des switches de bus.

Bus	① instruction	instruction	condition
AB	\emptyset FFFE FFF8 FFFE FFFA	MOVE.L Ro, (R1) ROR.L 1, R2 MULU Ro, R1	Ro = 0000 0000 R2 = FFFF FFFF Ro = R1 = 0000
DB	\emptyset \emptyset \emptyset	MOVE.L (R1), Ro MOVE.L (Ro), FFFF.W ADD.L Ro, R1	Ro = FFFF FFFF (R1) = 0000 0000 (Ro) = 0000 0000 Ro = R1 = FFFF FFFF
AB + DB	FFFE FFF8 FFFE FFFE FFFE FFFC	MOVE.W R2, 0000FFFF EXG Ro, R1 CMP.L (Ro)+(R1)+	R2 = FFFF Ro = FFFF 0000 R1 = 0000 FFFF (Ro) = (R1) - FFFF FFFF R1 = FFFE FFFC

Table **XXXII**. Instructions nécessaires au test de classe 0 des switches de bus de la partie opérative du MC 68000.

V.6. Test des coupures de bus

On n'admettra que des pannes simples de coupures et non des doubles coupures sur les deux lignes du même bus.

La table **XXXIII** regroupe les instructions nécessaires au test de classe 0 d'une coupure sur les bus de la partie opérative du MC 68000.

Instructions	Conditions
* { MOVE.L (R1), (R2) MOVE.L Do, (Ro)	(R1) = FFFF FFFF Do = 0000 0000
* { MOVE.L (R1), (R2) MOVE.L Do, (Ro)	(R1) = 0000 0000 Do = FFFF FFFF
* { MOVE.L (R1), (R2) ROR.L Ro, Do	(R1) = 0000 0000 Do = FFFF 0000 Ro = 16
* { MOVE.L (R1), (R2) ROR.L Ro, Do	(R1) = FFFF FFFF Do = 0000 FFFF Ro = 16
* dans cet ordre	

Table **XXXIII**. Instructions nécessaires au test de classe 0 d'une coupure sur les bus de la partie opérative du MC 68000.

On peut remarquer que pour les registres pouvant être couplés (symbole ↔) il est possible de n'utiliser que la partie poids faible (16 bits) et de plus, parmi ceux-ci, les registres données offrent la possibilité de n'être chargés que sur les 8 bits poids faibles (cf. Fool-it, § III.8).

On peut remarquer, que le registre ALU-E est toujours considéré comme un registre poids forts bien qu'il soit connecté sur le bus ABD. Le registre ALU-B est aussi bien utilisé pour stocker les poids forts d'un opérande pendant l'opération sur les poids faibles, que pour stocker des poids faibles, comme dans la multiplication ou la division.

On rappelle que le registre de code condition, et le registre d'état (superviseur ou utilisateur, niveau d'interruption) ne font pas partie de la partie opérative, et ne sont donc pas considérés ici.

Le test des registres de la partie opérative se déroule en deux étapes. En effet, une partie des registres est organisée en RAM, tandis que les autres sont adressés séparément et indépendamment les uns des autres.

V.1. Test des registres organisés en RAM

La RAM est composée des registres connus du programmeur, ainsi que du registre de donnée temporaire, c'est-à-dire :

- Les registres données D0, D1, ..., D7,
- Les registres adresses A0, A1, ..., A6,

- Le pointeur de pile utilisateur A7 (USP),
- Le pointeur de pile superviseur A'7 (USP),
- Le compteur ordinal (PC)
- Le registre de données temporaire (DT).

Deux décodeurs identiques se situent l'un au-dessus des registres poids faibles, l'autre au-dessus des registres poids forts.

On peut remarquer que tous ces registres sont connectés en entrée et en sortie sur les bus Adresses et bus Données. Seul le compteur ordinal possède une entrée supplémentaire, provenant du latch de l'AU (AUR).

V.1.1. Présentation du point de RAM

La figure 26 présente le point de RAM ainsi que ses connexions sur les bus. On rappelle que les mêmes transistors servent en lecture et en écriture, et que le principe de bus bifilaire est utilisé.

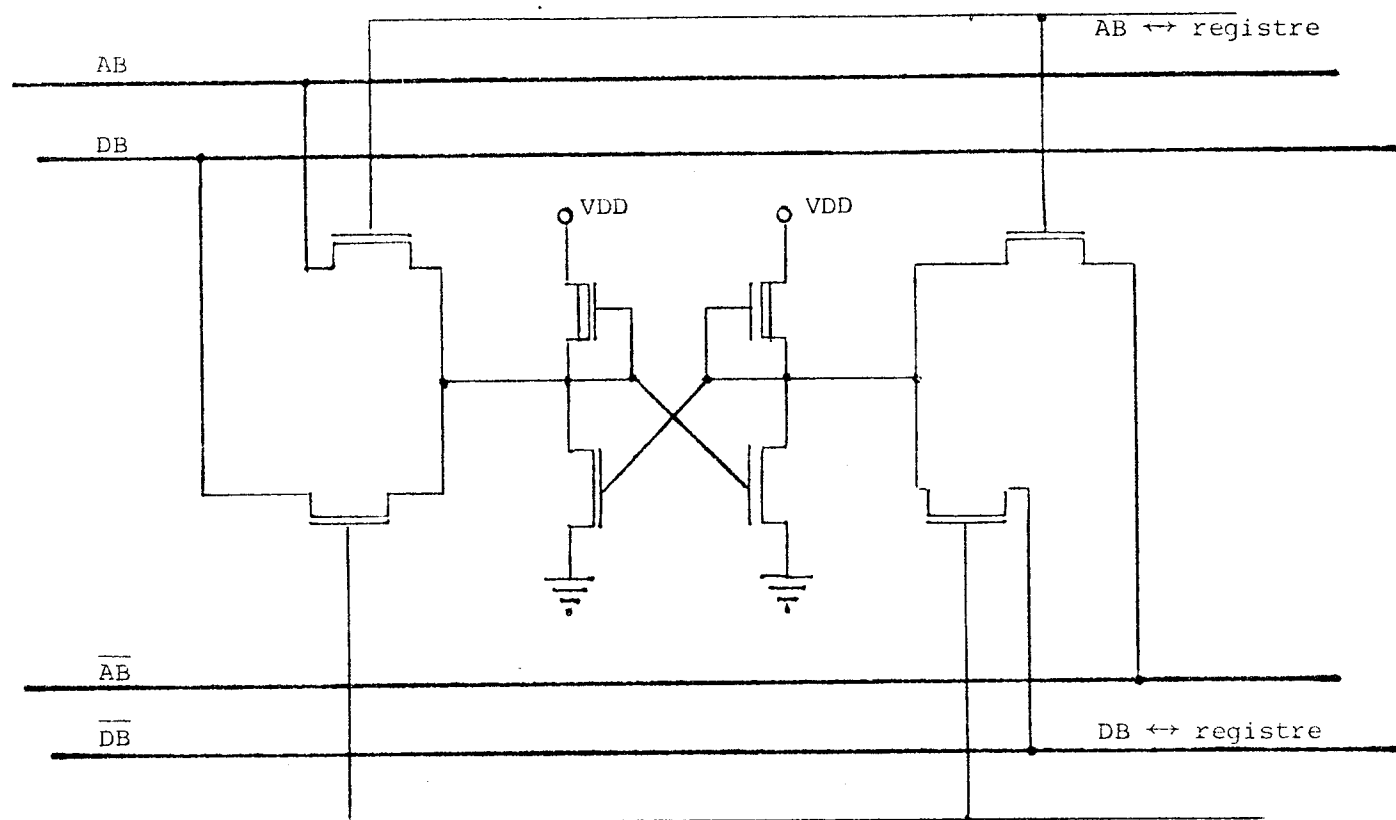


Figure 26. Le point de RAM et ses connexions sur les bus adresses et données du MC 68000.

V.1.2. Test du point de RAM

Un test tel que celui développé par [MAR 80], pour le test fonctionnel de mémoire vive à large couverture de pannes pourrait être envisagé. Ce test est un test fonctionnel, donc indépendant de la structure physique du circuit. Des travaux en cours semblent montrer que la classe de pannes couvertes par un tel algorithme serait équivalente à la classe I du test analytique utilisé ici.

On peut aussi envisager un test de classe 0 pour le point de RAM, tel que présenté ici.

Pour le test de la RAM du MC 68000, il faut remarquer que, bien que chaque registre puisse être connecté au deux bus, l'utilisation de connexions particulières imposera l'emploi d'instructions particulières. La table XXXV regroupe les transferts et tests nécessaires de la RAM du MC 68000. Le nombre d'accès aux points mémoire et le nombre de pannes testées sont données

En effet, il est possible d'effectuer les transferts en entrée ou en sortie avec la mémoire par le bus AB, mais pas par le bus DB.

Les vecteurs $X \times B \times P_i$ seront obtenus par l'utilisation de l'instruction MOVE pour charger les registres avec une valeur en mémoire.

Les vecteurs $P_i \times DB \times ALU$ seront obtenus lors d'addition de registres.

Les vecteurs $P_i \times \overline{DB} \times AU$ seront obtenus pendant un adressage indirect indexé (P_i correspond au registre d'index).

On obtient ainsi un algorithme détectant les pannes de classe 0 d'une RAM à double bus bifilaire en 6n. Mais, la programmation de cette algorithme entraînera un nombre d'accès plus grand, ceci est dû au fait que certains transferts ($P_i \times \overline{DB} \times AU$) impliquent un double accès mémoire (adressage indirect indexé où seul un des transferts est nécessaire).

On rappelle que l'accès à USP est obtenu lorsque le bit S du mot d'état est à 'zéro' (mode utilisateur), l'accès à SSP est obtenu lorsque le bit S du mot d'état est à 'un' (mode superviseur).

Vecteurs	Accès mémoire	Pannes testées
'1' > AB > P1	1	0
'0' > AB > P _i , (i > 1)	(n-1)	0
P1 > DB > ALU	1	0
Test 1	0	n
P2 > \overline{DB} > AU, DOB	1	0
Test 2	0	2
P _i , (i > 2) > \overline{DB} > AU, DOB	(n-2)	0
Test 3	0	1*(n-2)
'0' > AB > P1	1	0
'1' > AB > P _i , (i > 1)	(n-1)	0
P1 > \overline{DB} > AU, DOB	1	0
Test 4	0	n
P2 > DB > ALU	1	0
Test 5	0	2
P _i , (i > 2) > DB > ALU	(n-2)	0
Test 6	0	1*(n-2)
'1' > AB > P1	1	0
'0' > AB > P _i , (i > 1)	(n-1)	0
P1 > DB > ALU	1	0
Test 7	0	n
P2 > \overline{DB} > AU, DOB	1	0
Test 8	0	2
P _i , (i > 2) > \overline{DB} > AU, DOB	(n-2)	0
Test 9	0	1*(n-2)
Total	6n	6n

Table XXXV Vecteurs de test nécessaires au test de classe 0 de la RAM du MC 68000.

V.1.3. Cas particulier des registres DT et PC

L'accès au registre de données temporaires DT, est obtenu par l'instruction MOVEM avec adressage absolu court. (MOVEM XXX.W). Le registre DT est alors chargé par l'adresse absolue courte étendue (signe extend), puis celui-ci est transféré dans AU. Il faudra donc trois instructions MOVEM où la valeur de l'adresse de la seconde est le complément de celle de la première, la troisième étant identique à la première. Pour l'accès à l'ALU, une opération avec valeur immédiate empruntera le chemin désiré, la valeur immédiate est rangée dans DT, puis l'opérande est amené à l'ALU pour effectuer l'opération.

L'accès au registre PC est particulier du fait de son rôle de compteur ordinal. On utilisera deux instructions JMP avec l'adressage absolu long les deux adresses étant le complément l'une de l'autre (sauf pour le bit 0). Celui-ci pourra être testé seulement pour un test hors-ligne, car pour la valeur '1' il génèrera une interruption d'adresse erreur.

En ce qui concerne la connexion du latch d'AU (AU.R) et du PC la figure 27 présentant le pas du PC montre le type de connexion réalisé.

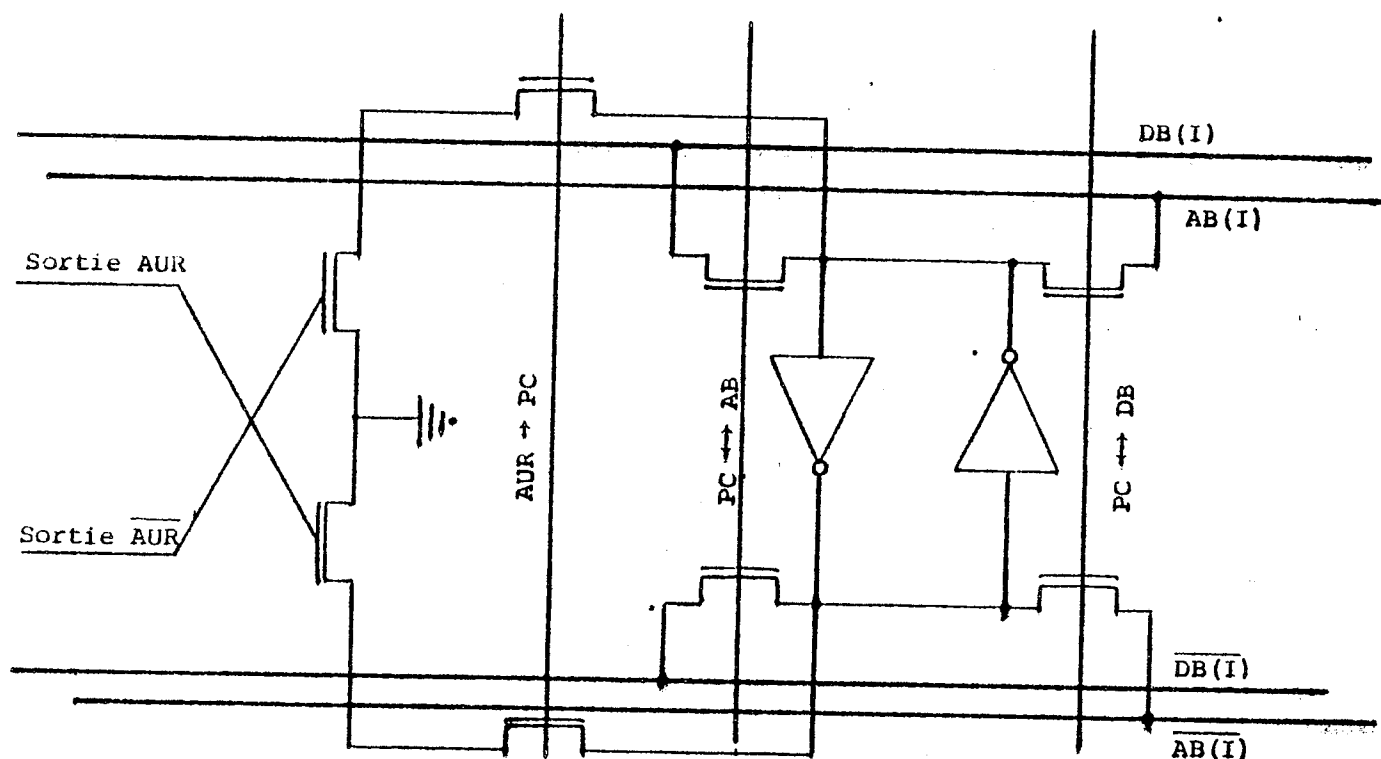


Figure 27. Pas du registre PC compteur ordinal du MC 68000.

La table XXXVI regroupe les instructions nécessaires au test de l'accès au registre PC par la sortie directe du latch de l'unité arithmétique AU.

@ INSTRUCTION	INSTRUCTION	CONDITION
0000	FFFC	NOP
0000	FFFE	NOP
FFFF	0000	NOP
FFFF	0000	MOVE.W d(An,ix), RO :d+An+ix = FFFF
0000	FFFE	MOVE.W d(An,ix), RO :d+An+ix = FFFF
FFFF	0000	MOVE.W d(An,ix), RO :d+An+ix = 0000
0000	FFFE	MOVE.W d(An,ix), RO :d+An+ix = 0000

Table XXXVI. Instructions nécessaires au test de classe 0 de l'accès au PC directement par le latch d'AU (AU-R) du MC 68000.

V.2. Test des registres autres que ceux de la RAM

V.2.1. Test du registre AOB (Buffer de sortie adresse)

a/ Présentation du registre

La figure 28 présente un pas du registre AOB. Celui-ci est donc composé des 16 bits du registre AOBL connectés sur les bus ABL et DBL et des 16 bits du registre AOBH connectés sur les bus ABH et DBH.

On peut noter que, bien que le registre ait 32 bits, seuls les 24 premiers sont connectés au bus adresse externe, le bit 0 servant à générer les signaux $\overline{\text{UDS}}$ (upper data strobe) et $\overline{\text{LDS}}$ (lower data strobe).

b/ Test du registre

Pour un test en-ligne, il faut utiliser un opérande de taille octet (.B) pour ne pas générer d'interruption d'adresse erreur quand le bit 0 est à 'un'. Seuls les 24 bits poids faibles seront testés (24 bits du bus adresse externes).

Dans le cas d'un test hors-ligne, on pourra tester les 32 bits du registre AOB en générant une adresse erreur pour chaque instruction (opérande de taille mot ou mot long et adresse impaire) le contenu de AOB étant alors sauvegardé en mémoire (SP-10, SP-12).

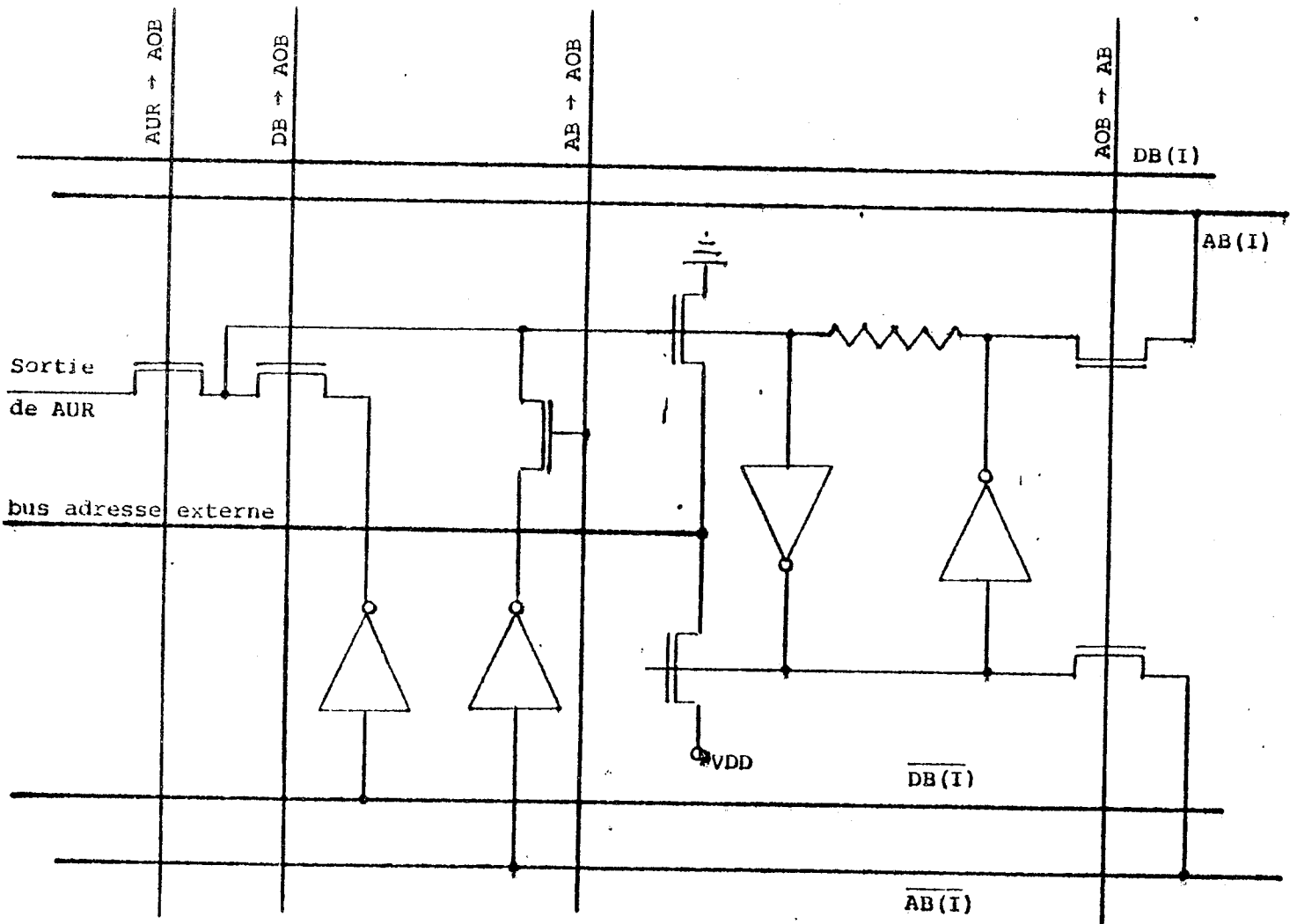


Figure 28. Pas du registre AOB avec connexion sur le bus adresse externe du MC 68000.

La table XXXVII regroupe les instructions nécessaires au test de la classe 0 correspondant au registre AOB. Pour un test hors-ligne, la taille de l'opération sera d'un mot ou mot long afin de générer une adresse erreur, on vérifiera ainsi les 32 bits du registre qui sont sauvegardés en mémoire.

Op de l'instruction	Instruction	Taille	Source	Destination
0000 00FB	MOVE	B/W	# FF00	(Ro) = 0000 00FF
FFFF FEFC	MOVE	B/W	# FF00	(Ro) = FFFF FF00
0000 FFFB	MOVE	B/W	XXXX.W	FFFF 0000.L
FFFE FFFA	MOVE	B/W	XXXX.W	0000 FFFF.L
0000 0000	ADD	W	(Ro)=0	FFFF 0000 (FFFF 0000)=0000 FFFF
0000 0000	MOVE	L	# 0000 00FF	(Ro) = FFFF FF00
0000 0000	MOVE	L	# FFFF FF00	(FFFF 0000)=0000 00FF
0000 0000	MOVE	B/W	# 00FF	(Ro) = FFFF FF00
B/W = taille byte pour le test en-ligne/taille word pour le test hors-ligne				

Table XXXVII - Instructions nécessaires au test de classe 0 correspondant au registre AOB du MC 68000.

V.2.2. Test du registre AUR (latch d'AU)

a/ Présentation du registre

La figure 29 présente un pas du registre AUR. Celui-ci est donc composé des 16 bits du registre AURL connectés sur les bus ABL et DBL et des 16 bits du registre AURH connectés sur les bus ABH et DBH.

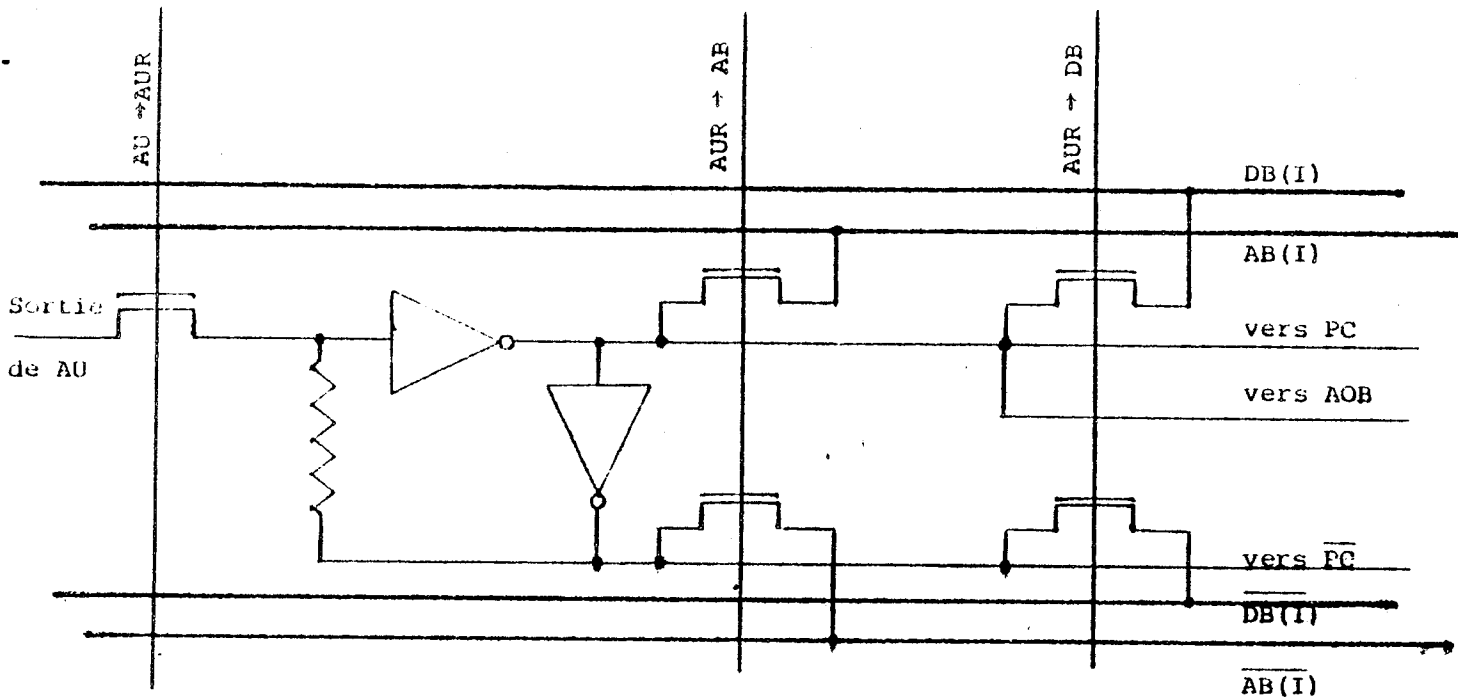


Figure 29 - Pas du registre AUR (32 bits) latch de l'AU du MC 68000.

La commande AU → AUR permet de charger le registre avec le contenu de l'AU.

La commande AUR → AB permet de charger le bus AB avec le contenu du registre AUR.

La commande AUR → DB permet de charger le bus DB avec le contenu du registre AUR.

La sortie vers le compteur ordinal (PC) est étudiée lors du test du registre PC et n'est pas reprise ici (cf. § V.1.3).

La sortie vers le registre AOB est étudiée lors du test du registre AOB et n'est pas reprise ici (cf. § V.2.1).

On remarquera que les 6 bits poids faibles du registre (AUR L(0), AUR L(1), AUR L(2), AUR L(3), AUR L(4), AUR L(5) génèrent un signal de compte-rendu appelé AU=0, utilisé pour les multiplications et divisions ainsi que pour les décalages dynamiques.

b/ Test du registre

La table XXXVIII regroupe les instructions nécessaires au test de classe 0 du registre AU-R et de sa ligne de compte-rendu AU = 0.

Adresse de l'instruction	Instruction	Condition
∅	MOVE.W d(Ro), d(R1)	d+Ro complément de d+R1
∅	MOVE.L (Ro), d(An, ix)	d+An = 0000 0000 ix = FFFF FFFF
∅	MOVE.L (Ro), d(An, ix)	d+An = FFFF FFFF ix = 0000 0000
FFFE FFFE 0000 FFFD	MOVE.L#0000 0000 (Ro) MOVE.L#0000 0000 (Ro)	∅
FFFE FFFE 0000 FFFD	MOVE.L FFFF FFFF.L, (Ro) MOVE.L FFFF FFFF.L, (Ro)	∅
∅	ASR.L Ro by R1	R1 = 32
∅	MOVE.L 0000 0000.L, (RO)	∅
∅ Quelconque		

Table XXXVIII. Instructions nécessaires au test de classe 0 du registre AU-R et de sa ligne de compte-rendu AU = 0.

V.2.3. Test du registre AT (adresse temporaire)

a/ Présentation du registre

La figure 30 présente un pas du registre AT. Celui-ci est composé des 16 bits du registre ATL connectés sur les bus ABL et DBL et des 16 bits du registre ATH connectés sur les bus ABH et DBH.

On remarque qu'il possède la même structure que le point de RAM, la seule différence réside dans le fait qu'il est adressé séparément et indépendamment des registres de la RAM.

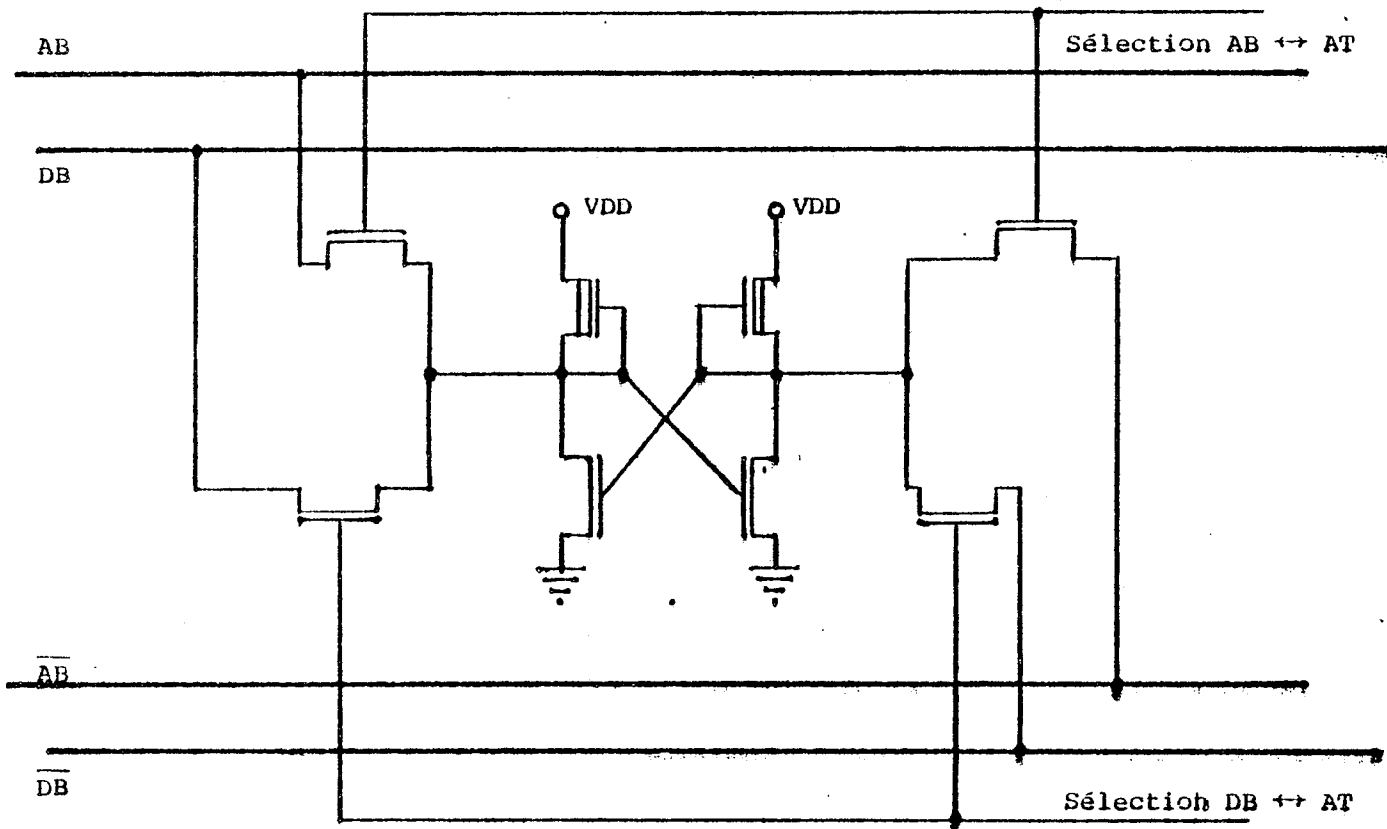


Figure 30 Pas du registre d'adresse temporaire AT du MC 68000.

La même séquence de test sera utilisée, seules les instructions générant les vecteurs seront différentes.

On peut remarquer, qu'un signal de compte-rendu est généré par le bit 15 du registre AT (ATL(15)) mais ce signal n'est pas connecté en partie contrôle.

b/ Test du registre

Deux instructions distinctes seront nécessaires pour le passage de l'ALU, une seule est suffisante pour le passage dans DOB. La table XXXIX regroupe les instructions nécessaires au test de classe 0 du registre AT. Ces instructions seront intégrées au test de la RAM. Le registre étant compté comme un Ri, $i > 2$.

Vecteurs	Instructions
'0' > DBE > AT ATL > AB > DOB ATH > AB > DOB Test 3bis	PEA 0000.W
'1' > DBE > AT ATL > AB > ALU Test 6bis	MOVE.L FFFF.W, FFFF.W
'1' > DBE > AT ATH > AB > ALU Test 6ter	ADDA.L # FFFF FFFF, Ao Ao quelconque
'0' > DBE > AT ATL > AB > DOB ATH > AB > DOB Test. 9bis	PEA 0000.W

Table XXXIX. Instructions nécessaires au test de classe 0 du registre d'adresses temporaire AT du MC68000. (à inclure dans le test du point de RAM).

V.2.4. Test du registre du PREN

Ce test est réalisé pendant le test de l'opérateur PREN, le registre faisant partie intégrante de l'opérateur (cf. § III.9).

V.2.5. Test du registre FTU (Field Translate Unit)

a/ Présentation du registre

La figure 38 présente un pas du FTU. Ce registre a de multiples utilisations, en effet, il est connecté à un bus bidirectionnel : le bus BC qui lui permet des transferts du mot d'état utilisateur ou superviseur (chargement ou rangement), transferts de valeurs immédiates provenant du PLA IRD (opération rapide : ADDQ, SUBQ, etc...) ou provenant du PLA TRAP (génération d'adresse particulière : RESET, etc...). Ce bus permet donc des échanges directs entre la partie contrôle et la partie opérative.

Les commandes suivantes, permettent :

- FTU ↔ ABL : le rangement du bus ABL dans le registre ou le chargement du registre sur le bus ABL .
- FTU ↔ DBL : le rangement du bus DBL dans le registre ou le chargement du registre sur le bus DBL
- LOADFTU : le rangement du bus BC dans le registre
- FTU → PSW : le chargement du registre sur le bus BC.

Le bus BC est un bus unifilaire, et sera considéré comme partie intégrante de la partie contrôle, il ne sera pas testé ici. Lors d'un rangement du bus dans le registre, le rangement s'effectue sur T3 comme pour une connexion sur les bus adresses et bus données. Lorsque le contenu du registre est écrit sur le bus BC, ce chargement est valide sur T4, T1, T2, T3 en effet ce bus unifilaire ne possède donc pas d'ampli de ligne de bus.

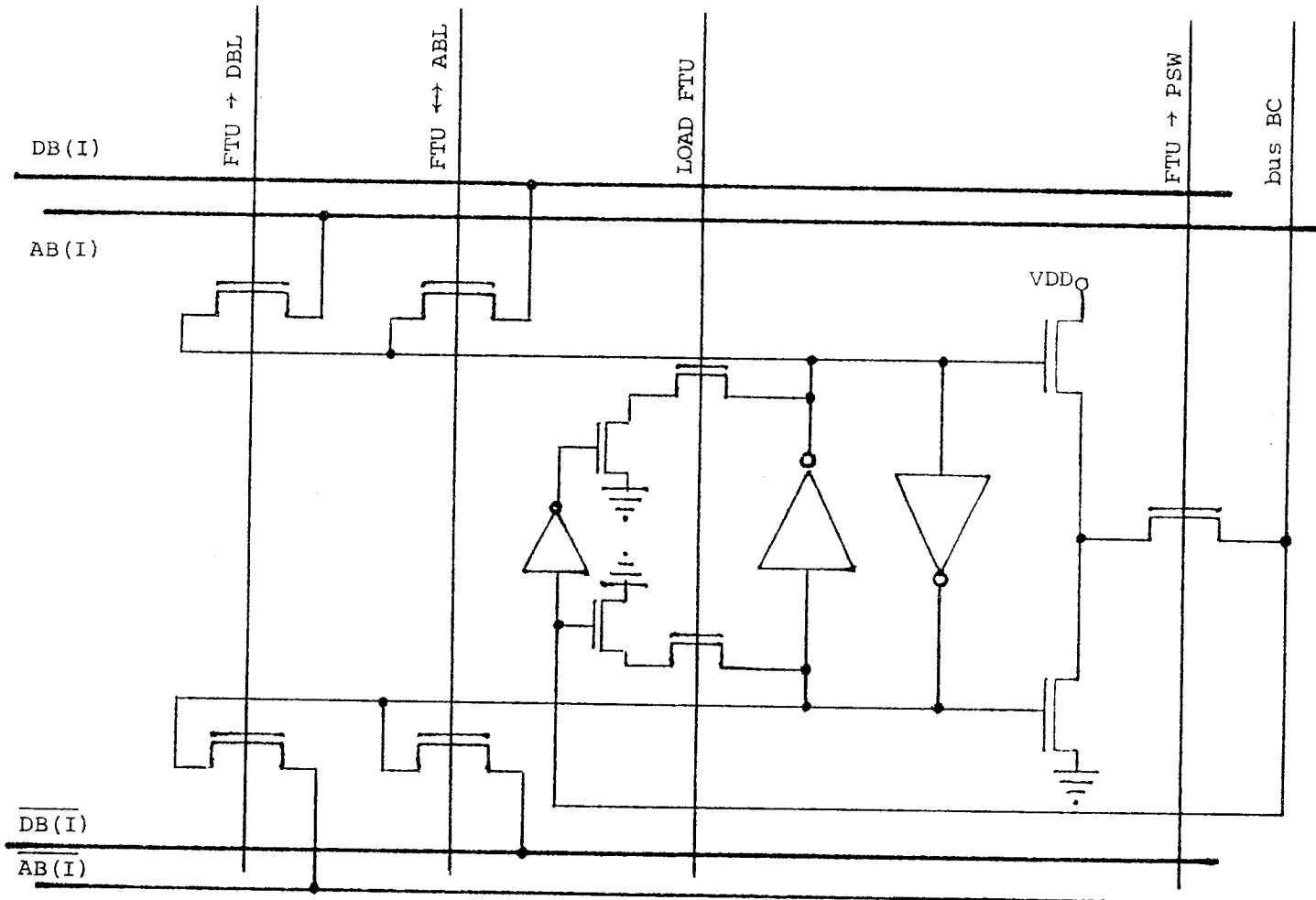


Figure 31 - Pas de l'opérateur FTU (Field Translate Unit) du MC 68000.

b/ Test du registre

La table XL regroupe les instructions nécessaires au test de classe 0 du registre de FTU du MC 68000.

INSTRUCTIONS	CONDITIONS
ORI FF, CR ↳ interruption	-
ADDQ # FF, Ro	-
ORI # FF, CR	-
MOVEQ # 00, Ro	-
MOVE.W (Ro), R1	(Ro) = FFFF
MOVE.L (Ro), XXX.W	(Ro) = FFFF FFFF
ORI # 00, CR	-
BEQ.B 7E	-
ORI # 00, CR	-
MOVEQ # FF, CR	-
MOVE.W (Ro), (R1)	-
MOVE.L # 0000 FFFF, Ro	(Ro) = 0000
MOVE.L # FFFF 0000, Ro	-
ORI # FF, CR	-
BEQ.B 80	-

Table XL. Instructions nécessaires au test de classe 0 du registre de FTU du MC 68000.

V.2.6. Test du registre DCR (decode register)

Ce test est réalisé pendant le test de l'opérateur DCR, le registre faisant partie intégrante de l'opérateur (cf. § III.7).

V.2.7. Test du registre DBIN (data input buffer)

a/ Présentation du registre

La figure 32 présente un pas du registre DBIN. La commande EDB → DBIN regroupe en fait deux commandes : EDB → DBINL pour l'octet poids faible, EDB → DBINH pour l'octet poids fort. Ces deux commandes sont testées lors du test du MUX I/O (cf. § III.6).

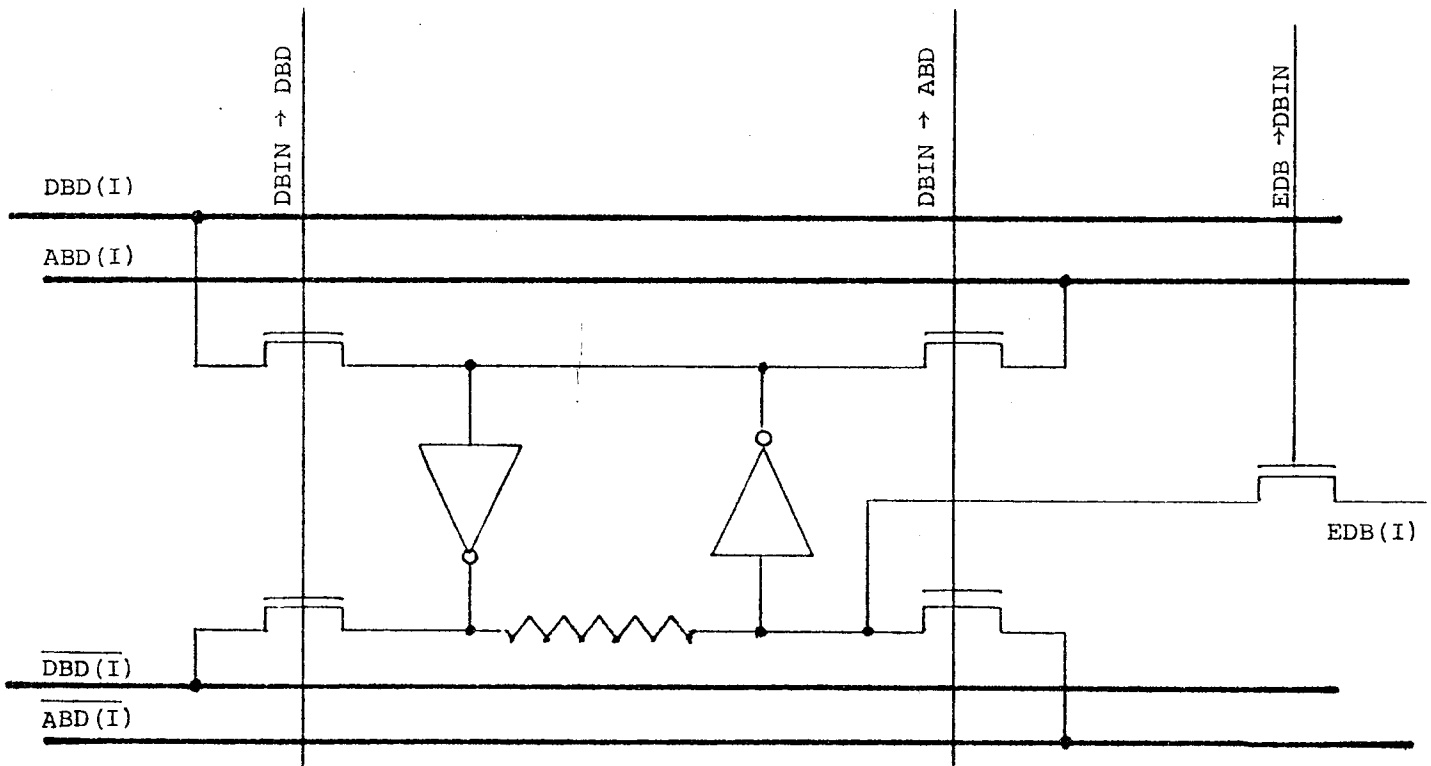


Figure 32 . Pas du registre buffer d'entrées de données DBIN du MC 68000.

La commande DBIN → DBD valide le transfert du contenu du registre sur le bus DBD.

La commande DBIN → ABD valide le transfert du contenu du registre sur le bus ABD.

b/ Test du registre

La table XLI regroupe les instructions nécessaires au test de classe 0 du registre buffer d'entrée données DBIN du MC 68000.

Instructions	Conditions
MOVE.W (Ro), (R1)	(Ro) = 0000
MOVE.L 0000 0000.L, (Ro)	∅
MOVE.L (Ro), XXXX.W	(Ro) = 0000 FFFF
MOVEP.W d(Ro), R1	d = FFFF Ro = 0000 0000
ADD.W (Ro), R1	(Ro) = FFFF
ADD.W (Ro), R1	(Ro) = 0000 R1 = FFFF
ADD.W R1, (Ro)	(Ro) = FFFF
ADD.W R1, (Ro)	(Ro) = 0000 R1 = FFFF

Table XLI. Instructions nécessaires au test de classe 0 du buffer d'entrée données DBIN du MC 68000.

V.2.8. Test du registre DOB (data output buffer)

a/ Présentation du registre

La figure 33 présente un pas du registre DOB. La commande DOB (H ou L) regroupe en fait les deux commandes, DOBH validant du transfert octet poids fort sur le bus données externe (EDB) et DOBL validant le transfert octet poids faible sur le bus données externe (EDB). Ces deux commandes sont testées lors du test du MUX I/O (cf. § III.6).

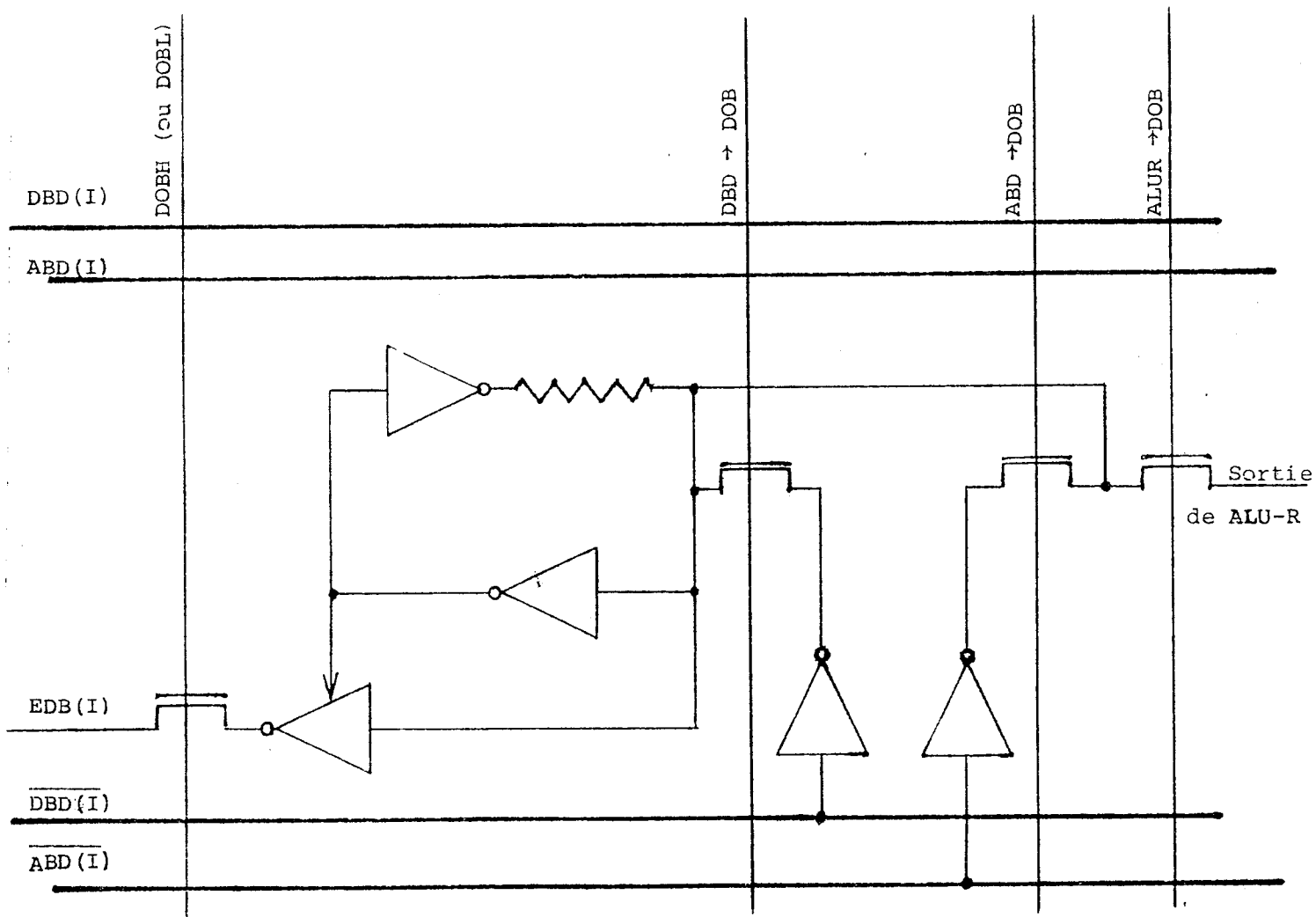


Figure 33. Pas du registre buffer de sortie données DOB du MC 68000.

La commande DBD → DOB permet le chargement du registre avec le contenu du bus DBD.

La commande ABD → DOB permet le chargement du registre avec le contenu du bus ABD.

La commande ALU-R → DOB permet le chargement du registre avec le contenu du latch d'ALU (ALU-R) cette commande sera testée lors du test de ce registre (cf. § V.2.10).

b/ Test du registre

La table XLII regroupe les instructions nécessaires au test de classe 0 du registre buffer de sortie données (DOB) du MC 68000.

INSTRUCTIONS	CONDITION
MOVE.W (Ro), (R1)	(Ro) = FFFF
MOVE.W (Ro), (R1)	(Ro) = 0000
MOVE.L (Ro), XXXX.W	(Ro) = FFFF FFFF
MOVE.L (Ro), XXXX.W	(Ro) = 0000 0000

Table XLII. Instructions nécessaires au test de classe 0 du registre buffer de sortie données DOB du MC 68000.

V.2.9. Test du registre ALUE (ALU-Extender)

Ce test est réalisé pendant le test de l'opérateur ALU-E, le registre faisant partie intégrante de l'opérateur (cf. § III.5).

V.2.10. Test du registre ALU-R (latch d'ALU)

a/ Présentation du registre

La figure 34 présente un pas du registre ALU-R, latch d'ALU.

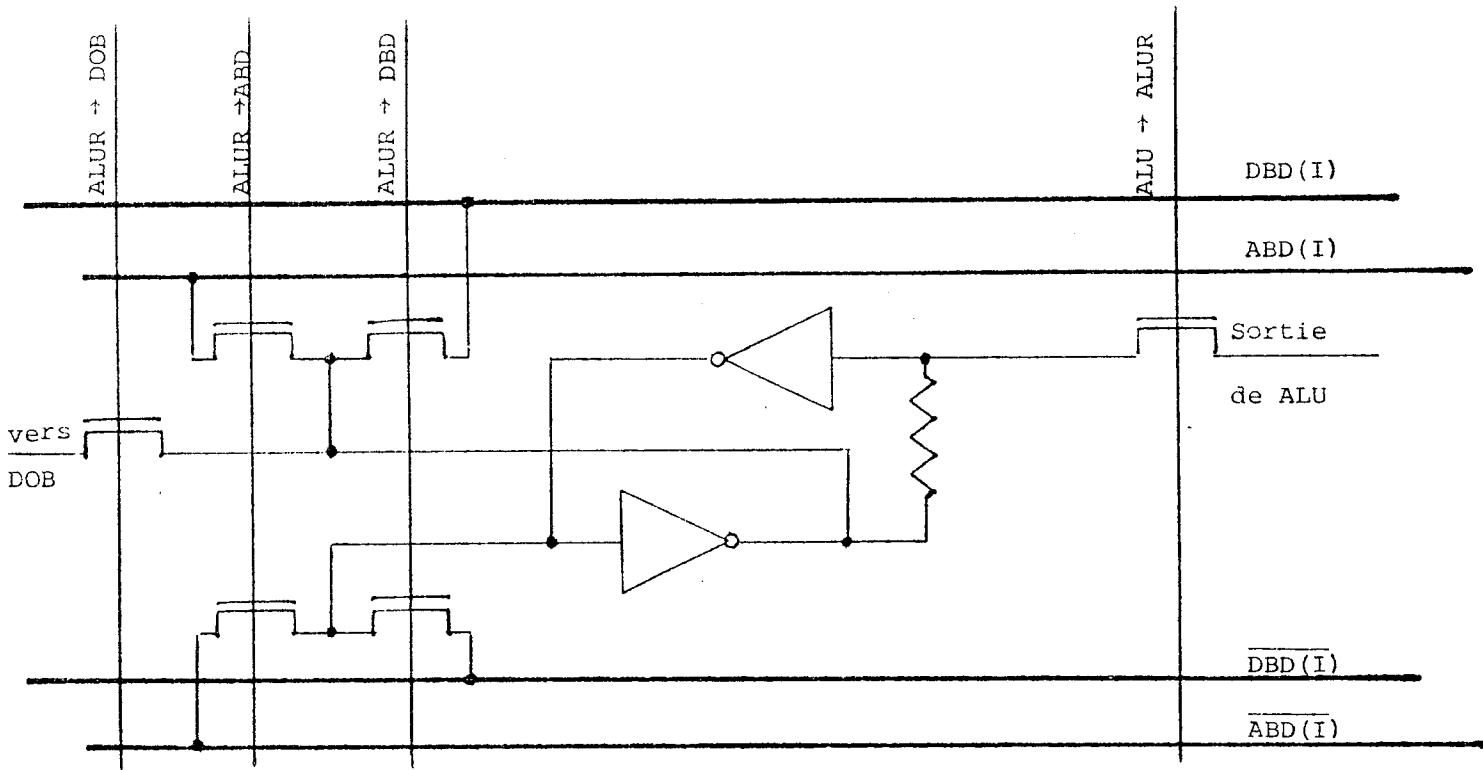


Figure 34. Pas du registre ALU-R (latch d'ALU) du MC 68000.

La commande $ALU \rightarrow ALU-R$ permet de charger le registre avec le résultat de l'ALU.

La commande $ALUR \rightarrow ABD$ permet de ranger le contenu du registre sur le bus ABD.

La commande $ALUR \rightarrow DBD$ permet de ranger le contenu du registre sur le bus DBD.

La commande $ALUR \rightarrow DOB$ permet de valider le transfert du registre ALUR vers le registre DOB.

b/ Test du registre

La table XLIII regroupe les instructions nécessaires au test de classe 0 du registre latch d'ALU (ALU-R) du MC 68000.

Instructions	Conditions
ADD.W Ro, (R1)	Ro = 0000 (R1) = 0000
ADD.W Ro, (R1)	Ro = FFFF (R1) = 0000
ADD.L Ro, (R1)	Ro = 0000 FFFF (R1) = 0000 FFFF
ADD.W d(An, ix), Ro	d = 0000
ADDX.L -(Ro), -(R1)	(Ro) = 0000 0000 (R1) = FFFF FFFF
ADDX.L -(Ro), -(R1)	(Ro) = FFFF FFFF (R1) = 0000 0000
MULU Ro, (R1)	Ro = FFFF (R1) = 1000
MULU Ro, (R1)	Ro = FFFF (R1) = FFFF
MOVE.W (Ro), 0000.W	∅
MOVE.L (Ro), (R1)	(Ro) = 0000 FFFF
MOVE.W (Ro), (R1) ORI.B #00, D0	(Ro) = FFFF
MOVE.W (Ro), FFFF 0000.L	(Ro) = FFFF
MOVE.L (Ro), FFFF.W	(Ro) = 0000 0000

Table XLIII. Instructions nécessaires au test de classe 0 du registre latch d'ALU (ALU-R) du MC 68000,

V.2.11. Test du registre ALU-B (Buffer d'ALU)

a/ Présentation du registre

La figure 35 présente le pas du registre ALU-B (Buffer d'ALU). Ce registre de 16 bits est utilisé soit pour stocker les 16 bits poids forts lors d'opérations de 32 bits à opérandes mémoires, soit pour stocker les 16 bits poids faibles lors d'opérations comme multiplication, division, etc...

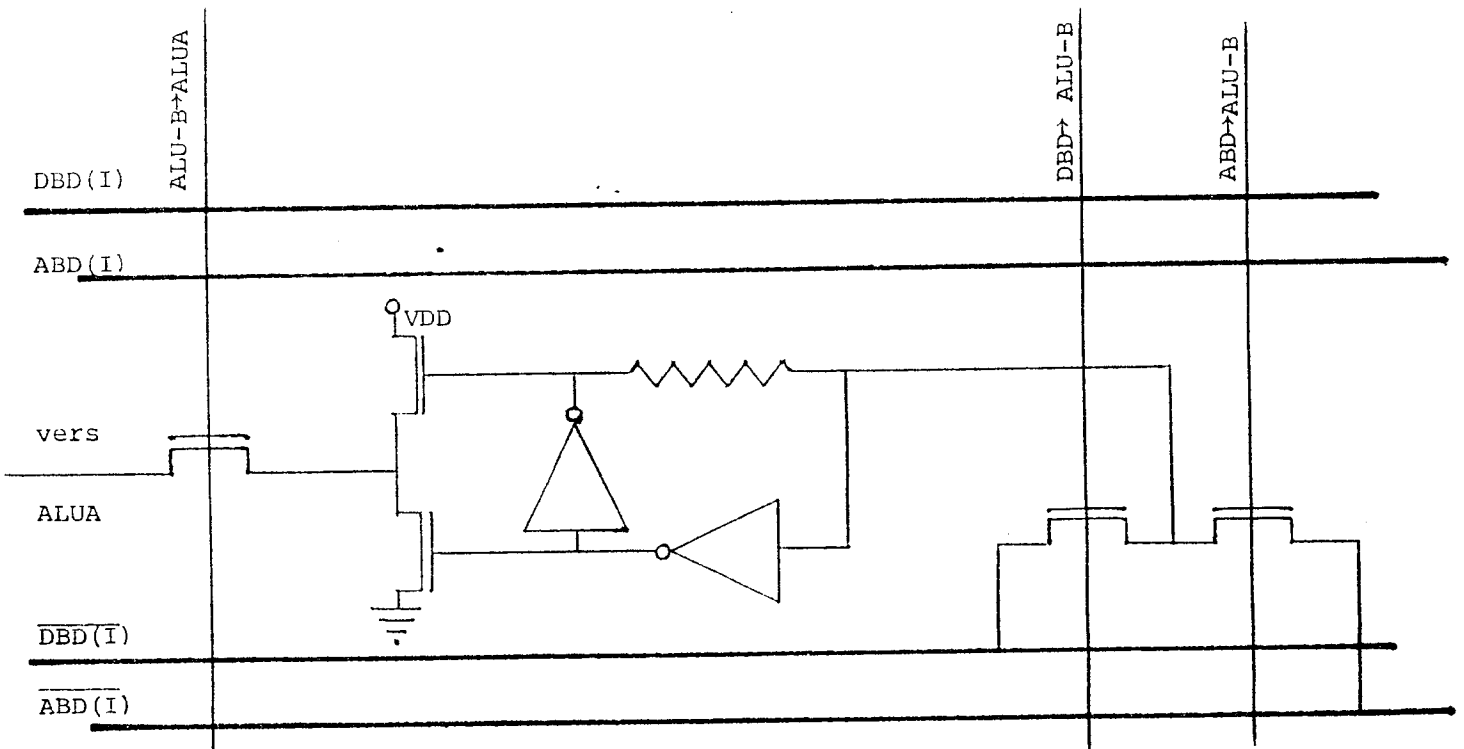


Figure 35. Pas du registre ALU-B (Buffer d'ALU) du MC 68000.

La commande DBD → ALUB permet le chargement du registre avec le contenu du bus DBD.

La commande ABD → ALUB permet le chargement du registre avec le contenu du bus ABD.

La commande ALUB → ALUA permet l'acquisition de l'opérande A de l'ALU à partir du registre ALUB. Cette commande est testée au cours de l'acquisition des opérandes sur l'ALU (cf. § III.3).

La table XLIV regroupe les instructions nécessaires au test de classe 0 du buffer de l'ALU : ALU-B du MC 68000.

Instructions	Conditions
MOVE.L (Ro), XXXX.W	(Ro) = 0000 0000
MULU Ro, (R1)	Ro = FFFF (R1) = 1000
MULU Ro, (R1)	Ro = 1000 (R1) = FFFF
MULU Ro, (R1)	Ro = 0000
MOVE.L Ro, XXX.W	(Ro) = FFFF FFFF

Table XLIV. Instructions nécessaires au test de classe 0 du registre buffer d'ALU, ALU-B du MC 68000.

VI. CONCLUSION :

On peut remarquer que l'excitation de certains circuits par les vecteurs de test ainsi que la propagation des erreurs jusqu'à un point observable sont dans certains cas problématiques, bien que des chemins existent pour les propager, mais ne sont pas utilisés dans le cas particulier où ils sont nécessaires.

Il serait possible, dans ce cas, de "reprogrammer" la ROM de micro-instructions pour améliorer le test. A ce niveau, deux choix sont possibles :

- reprogrammer des instructions existantes en utilisant des chemins différents,
- programmer les instructions "spéciales test" qui ne seraient utilisées que pour mettre en oeuvre les tests nécessaires, ceci étant fait en utilisant des instructions invalides.

On gagnerait ainsi non seulement sur le plan de la testabilité du circuit mais surtout sur la taille et le temps d'exécution des programmes de test nécessaires.

On s'aperçoit donc que plus l'échelle d'intégration augmente, plus les concepteurs introduisent des opérateurs complexes, qui auraient nécessité une surface de circuit trop grande préalablement. Le contenu fonctionnel est aussi augmenté : si l'on compare les parties opératives du M 6800 et du MC 68000, on s'aperçoit qu'il y a un facteur 5 dans l'augmentation du nombre des éléments fonctionnels.

M 6800	MC 68000
9 Registres	51 registres
2 Opérateurs	10 Opérateurs

Il va de soi que pour tester 5 fois plus de registres par exemple, il faudra au moins 5 fois plus de temps, ceci en tenant compte des étapes d'initialisation, et de test.

Il est donc nécessaire que des "primitives" spéciales pour le test soient disponibles à l'intérieur de ces microordinateurs, afin de diminuer le temps de test car, par exemple si un MC 68000 n'est que 4 fois plus rapide qu'un M 6800, le gain de puissance serait grandement diminué par la perte de temps dans le cas d'un test en ligne, continu ou discontinu.

REFERENCES

- [COU 81] COURTOIS B.
"Failure mechanisms, fault hypotheses and analytical testing of LSI-NMOS (HMOS) circuits", International Conference VLSI 81, Edinburg, 18-21/08/1981, England, Academic Press 1981.
- [COU 82] COURTOIS B.
"Test et LSI"
Thèse de doctorat d'état, 12/06/1982, Grenoble.
- [MAR 83] MARCHAL P.
"Test analytique de la partie opérative du microprocesseur MC 68000", Rapport de Recherche IMAG, RR n° 361, Mars 1983.
- [MAR 80] MARINESCU M.
"Test fonctionnel de mémoire vive à grande couverture de pannes", 2nd International Conference on Reliability and Maintainability, Perros-Guirec Trégastel, 8-12/09/1980.

CHAPITRE VII
COEUR NE CRÉANT PAS E
PARTIE CONTRÔLE
Etape (v)

CHAPITRE VII - COEUR NE CREANT PAS E

PARTIE CONTROLE

Etape (v)

I	- INTRODUCTION	369
II	- MODELE DE PANNE DE LA PARTIE CONTROLE	370
III	- HYPOTHESES DE PANNES DE LA COMMANDE D'OPERATEURS	373
	III.1. Operateur arithmétique et logique (ALU)	373
	III.2. Operateur arithmétique (AUH, AUL)	383
	III.3. Operateur d'extension d'ALU (ALUE)	388
	III.4. Operateur d'extension de signe (SIGNEX)	391
	III.5. Operateur encodeur de priorité (PREN)	393
	III.6. Operateur Fool-it	398
	III.7. Operateur de manipulation de bits (DCR)	398
	III.8. Operateur de multiplexage d'octets (MUX I/O)	402
IV	- HYPOTHESES DE PANNES DE LA GESTION DES TRANSFERTS	408
	IV.1. Pannes de gestion du bus	412
	IV.1.1. Pannes de précharge	412
	IV.1.1.1. Précharge inexistante	412
	IV.1.1.2. Précharge permanente	414
	IV.1.2. Pannes d'amplification	419
	IV.1.2.1. Amplification inexistante	421
	IV.1.2.2. Amplification permanente	421
	IV.1.3. Pannes de switches	421
	IV.1.4. Pannes de fool-it	425
	IV.2. Pannes de sélection de registres	427
	IV.2.1. Aucune source n'est sélectionnée	428
	IV.2.2. Plusieurs sources sont sélectionnées.	432

IV.3.	Hypothèses retenues pour les pannes de transferts	436
IV.4.	Erreurs possibles lors d'un transfert élémentaire	442
IV.5.	Erreurs possibles lors d'un transfert vers un opérateur	448
IV.5.1.	Opérateur PREN	448
IV.5.2.	Opérateur AU	452
IV.5.3.	Opérateur ALU	456
IV.5.4.	Opérateur ALUE	458
IV.5.5.	Opérateur DCR	459
IV.5.6.	Opérateur SIGNEX	463
IV.5.7.	Opérateurs non concernés par les pannes de transferts .	469
IV.6.	Erreurs possibles lorsqu'aucun transfert ne doit avoir lieu ...	469
IV.7.	Conclusion sur les hypothèses de pannes de la gestion des transferts	470
V	- TEST DU SEQUENCEMENT DES INSTRUCTIONS	474
V.1.	Définitions et classes de pannes (de test)	474
V.2.	Procédures de test utiles pour les classes 0, et II, sur un cycle	477
V.3.	Procédures de test utiles pour les classes I et III, sur un cycle	478
V.4.	Multiplicité temporelle, erreurs sur plusieurs cycles	479
V.5.	Multiplicité temporelle et spatiale, test d'une instruction ...	482
VI	- HYPOTHESES ET TEST DE LA GESTION DU MOT D'ETAT	486
VI.1.	Hypothèses et test de la gestion du code condition	487
VI.2.	Hypothèses et test de la gestion du mot superviseur	493
VI.3.	Hypothèses et test de la gestion des interruptions	494
VI.4.	Hypothèses et test des bascules S et T.	495
VII	- TEST DU BRANCHEMENT A BON ESCIENT	498
VII.1.	Test des alternatives de séquences.	498
VII.2.	Test de la seconde partie du PLA de branchement.	502
IX	- CONCLUSION	504
REFERENCES	507

I. INTRODUCTION

Ce chapitre couvre la totalité du test fonctionnel de la partie contrôle du MC 68000. Le modèle global des pannes de la partie contrôle est présenté. Des hypothèses de pannes concernant chaque partie sont examinées, ainsi que les méthodes de tests associés. Ce test fonctionnel de la partie contrôle est abordé de manière rigoureuse et sous des hypothèses de pannes plus fortes qu'il n'est fait habituellement :

- test des commandes d'opérateurs
- test de la gestion des transferts
- test du séquençement des commandes de chaque instruction
- test de la gestion du mot d'état
- test du branchement à bon escient.

Des procédures générales de test sont dégagées, en fonction de la gravité des conséquences entraînées sur le séquençement des programmes. Ce test est abordé sans orientation particulière en-ligne ou hors-ligne. La séparation claire entre les différents tests pour les différentes classes est suffisante pour :

- opérer une sélection en vue d'un test en-ligne (continu ou discontinu) : étape (v) de la méthodologie.
- prendre la totalité en vue d'un test hors-ligne (in-situ ou ex-situ).

L'ensemble de ces procédures constitue ce que beaucoup d'utilisateurs recherchent sous l'appellation de "test instruction".

II. MODELE DE PANNE DE LA PARTIE CONTROLE

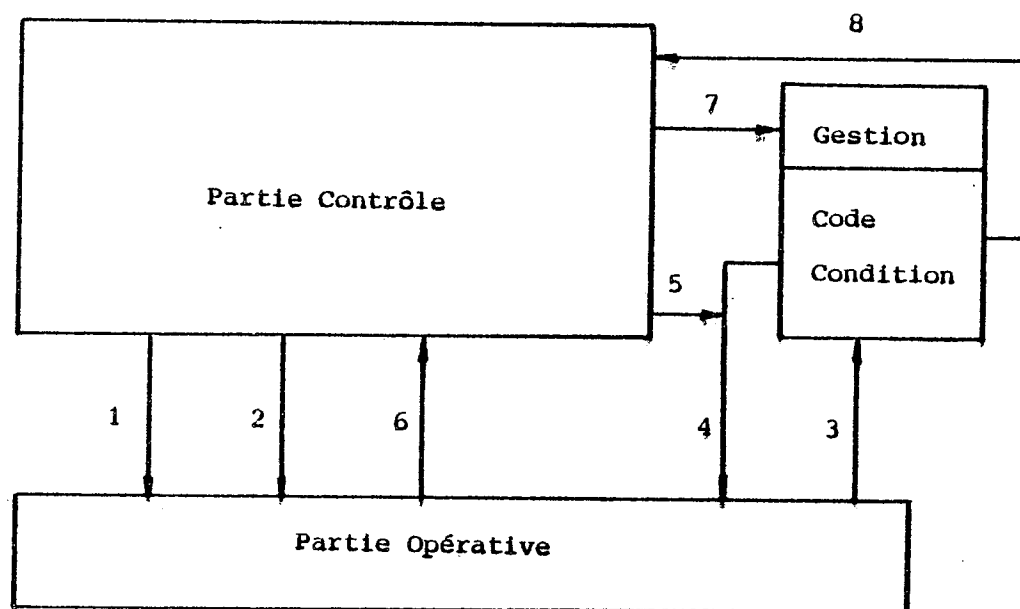
Le modèle adopté est celui de [COU 81] c'est-à-dire restreint à l'étude du test fonctionnel de la partie contrôle du MC 68000.

En ce qui concerne la partie opérative, comme il l'a été vu au Chapitre VI, un test analytique est préférable et ceci, plus en fonction du facteur de répétitivité que de la largeur des données. En effet, la conception des parties opératives est basée sur la notion de blocks, ainsi une partie opérative est construite sur un étage puis 8, 16 ou 32 de ces étages sont juxtaposés pour former l'ensemble selon la taille du chemin de données. Ainsi, les 9000 transistors se trouvant dans la partie opérative du MC 68000, ont un facteur de répétitivité de 38:1 (transistor existant : transistor dessiné). Ceci signifie que le test d'un block permettra de tester tous ceux qui ont été dupliqués à partir de ce block de base, donc, en moyenne, lorsque l'on teste la partie opérative du MC 68000 au niveau du transistor, le test (ou les vecteurs qui en découlent) d'un transistor permettra d'en tester environ 38. Ceci est encore plus net pour les registres de cette partie opérative, pour lesquels le facteur de répétitivité atteint 500:1 pour 5000 transistors environ.

Par contre, en ce qui concerne la partie contrôle, si les 50000 MOS contenus dans les ROM et PLAs ont un facteur de répétitivité de 50:1, et pour lesquels un test analytique serait fort bien envisageable, les 9000 transistors de la logique aléatoire (autant que dans toute la partie opérative) ont un facteur de répétitivité de 4:1, ce qui est vraiment très faible. Le test de l'influence d'une panne sur le fonctionnement des autres transistors est trop important pour considérer un test analytique. C'est pourquoi, pour la partie contrôle, un test fonctionnel est envisagé.

Ce test impose de connaître le fonctionnement de la partie contrôle et donc son "interface" vis-à-vis de l'extérieur (reste du circuit). La figure 1

représente cette "interface" contenant les commandes vers la partie opérative, contrôle des opérateurs, contrôle des transferts, etc....



- 1 - Commandes opérateur (§ III)
- 2 - Contrôle des transferts (§ IV)
- 3 - Compte rendu d'opérateurs et chargement (§ V)
- 4 - Valeur du code condition pour les opérations (§ V)
- 5 - Contrôle du code condition (§ V)
- 6 - Compte rendu d'opérateurs (§ V et VII)
- 7 - Gestion du code condition (§ VI)
- 8 - Valeur du code condition : branchement (§ VII)

Figure 1. - Interface entre la partie opérative la partie contrôle, le code condition.

Les 196 signaux transitant entre la partie opérative et la partie contrôle

du MC 68000 peuvent se décomposer de la façon suivante :

- 61 signaux de commandes d'opérateurs tels que code opération, multiplexage d'entrée etc.... Ces opérateurs peuvent être des unités arithmétiques (AU), des unités arithmétiques et logiques (ALU), des décaleurs etc...
- 123 signaux de commandes des transferts (précharges des bus, amplification, sélection lecture, sélection écriture etc....).
- 12 signaux de contrôle d'entrée du code condition vers la partie opérative. Ces signaux sont utilisés par les opérateurs, ils seront étudiés en même temps que ceux-ci.

En plus de ces signaux de contrôle et de commandes, les 21 signaux de compte-rendu sont à envisager, ces signaux transitant directement de la partie opérative vers la partie contrôle.

III. HYPOTHESES DE PANNES DE LA COMMANDE D'OPERATEURS

Ainsi qu'il a été dit précédemment, les hypothèses de pannes de la commande d'opérateurs sont des cas d'espèce et doivent être examinés pour chaque machine.

Le test envisagé est celui du séquençement des commandes des opérateurs : Il y a mauvais séquençement lorsqu'une commande est activée alors qu'elle ne doit pas l'être, ou bien lorsque cette commande n'est pas activée alors qu'elle devrait l'être. Il est donc nécessaire de déterminer qu'elle serait la fonction réalisée par l'opérateur en cas de panne sur ces commandes (code-op non existant, allié à une retenue entrante différente de celle prévue, ceci allié à un multiplexage des entrées non-correct).

L'exemple du MC 68000 sera explicite en ce qui concerne la réduction nécessaire des hypothèses.

Les différents opérateurs du MC 68000 vont donc être considérés ici.

III.1. Opérateur arithmétique et logique (UAL)

Les commandes de cet opérateur de calcul sont représentées à la figure 2.

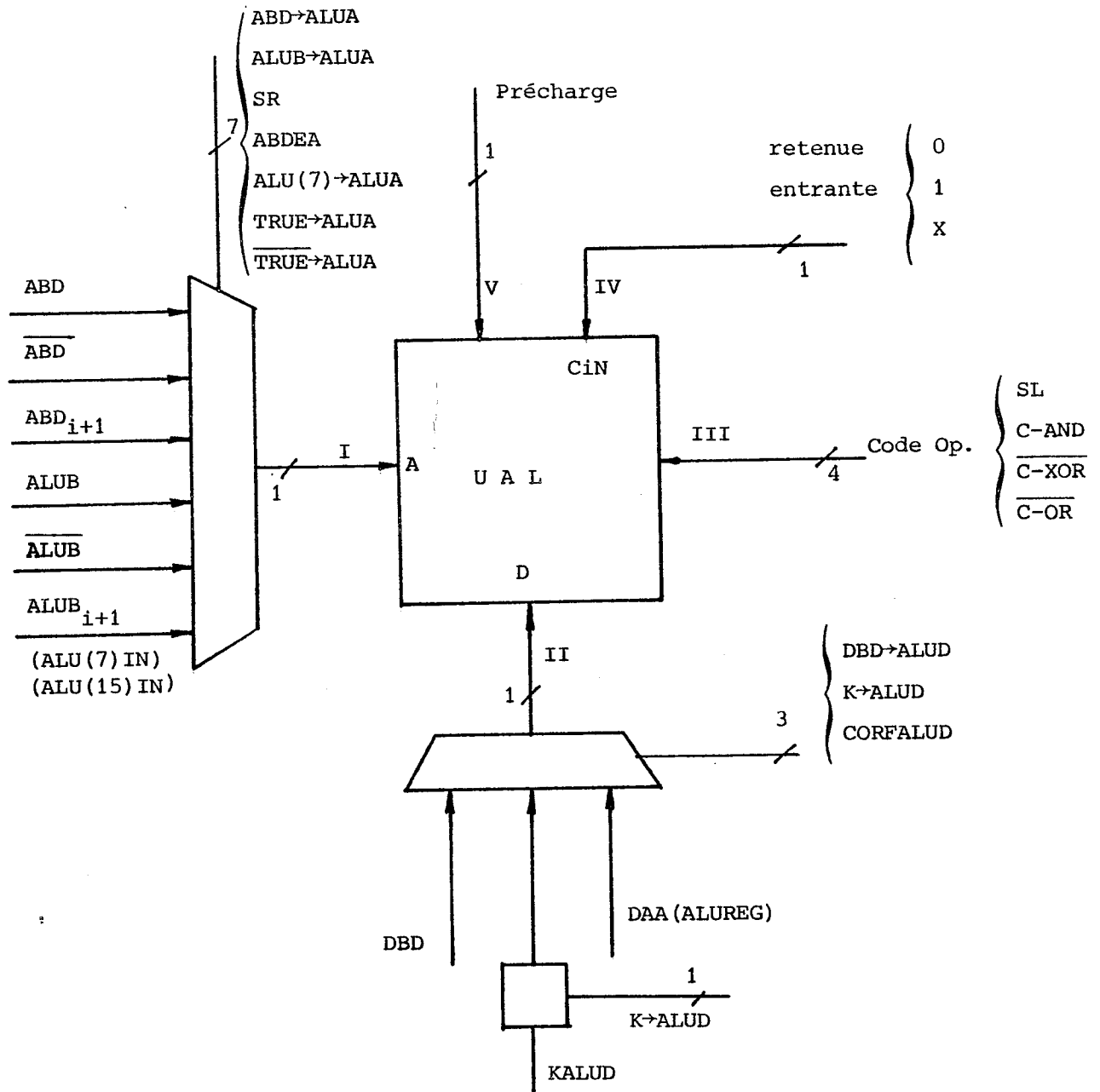


Figure 2. - Commandes de l'opérateur de calcul du MC 68000.

Sur l'entrée A de l'UAL sont connectés le bus ABD et le buffer d'UAL (ALUB). Cette entrée peut être directe, complémentée ou décalée.

Les différents registres peuvent être connectés dans le buffer d'UAL, par l'intermédiaire du bus ABD ou du bus DBD.

Sur l'entrée D se trouvent connecté le bus DBD direct, la constante KALUD (= 0 ou 1) et la valeur ajustée DCB du latch de l'ALU (ALU Reg.).

La retenue entrante est égale à 0, 1 ou x.

Enfin, de même que pour le M6800, le code-op de 4 bits permet de réaliser 5 opérations.

On comprend bien ici que des restrictions d'hypothèses de pannes doivent être faites. En effet, il faudrait sinon pouvoir évaluer le résultat obtenu en combinant des erreurs de tous les signaux : multiplexage entrée A, multiplexage entrée D, erreur du code-op, retenue entrante erronée, et ceci combiné au fait que les bus peuvent ne pas contenir les opérandes corrects correspondant au calcul désiré.

Les hypothèses retenues sont donc réduites comme il suit (tout en notant que des hypothèses plus larges pourraient être considérées, au prix d'étude et de temps de déroulement du test plus importants) :

Bloc I - Entrée A de l'UAL :

Dès qu'une, ou plusieurs origines sont sélectionnées sur le bus ABD, ou ont été sélectionnées pour être rangées dans le buffer d'ALU, la valeur correspondante pourrait être entrée directe, complémentée ou décalée. En cas d'erreur, plusieurs commandes de multiplexage pourraient être activées. On limite les hypothèses de la façon suivante :

- . Si l'origine attendue arrive à l'entrée du multiplexeur, elle passe le multiplexeur, seule la fonction de multiplexage (directe, complémentée, décalée) peut être erronée.
- . Si la donnée qui traverse le multiplexeur pour entrer dans l'UAL n'est pas l'origine attendue alors la fonction de multiplexage est correcte.

Bloc II - Entrée D de l'UAL :

Ce cas ressemble au précédent et on limite les hypothèses de la façon suivante :

- . Si l'origine sélectionnée sur le bus DBD est la bonne, alors une (seule) commande de multiplexage est activée, éventuellement à la place de la borne.
- . Si l'origine sélectionnée est fautive (plusieurs origines,....) les commandes de multiplexage sont correctes.

Bloc III - Code opération :

Sur les 16 valeurs possibles des 4 signaux, 5 seulement sont utilisées. Il est nécessaire de déterminer la fonction réalisée pour les 11 autres codages, si l'on désire tester que le code opération soit le bon (il est évident que sinon, pour certains opérandes on ne détecterait pas l'erreur). Ceci nécessite de connaître le circuit et non seulement la signification des signaux du code opération. Il est alors possible de dresser la table I, résumant les opérations effectuées par l'opérateur en cas d'erreur sur la génération du code-op.

SL	Code-op			Fonction réalisée	Sortie S0 à S31	
	CAND	COR	CXOR			
0	0	0	0	OR	$S_i = a_i + d_i$	
0	0	0	1	Anormal	$S_i = (a_i + d_i) \oplus C_{iN}$	$\overline{C_{out}} = a_i d_i \overline{C_{iN}}$
0	0	1	0	XOR	$S_i = a_i \oplus d_i$	
0	0	1	1	ADD	$S_i = (a_i \oplus d_i) \oplus C_{iN}$	$\overline{C_{out}} = \overline{a_i} \overline{d_i} + \overline{a_i d_i} C_{iN}$
0	1	0	0	Anormal	$S_i = 0$	
0	1	0	1	Anormal	$S_i = 0$	
0	1	1	0	Anormal	$S_i = a_i \cdot d_i$	
0	1	1	1	AND	$S_i = a_i \cdot d_i$	
1	0	0	0	Anormal	$S_i = 0$	
1	0	0	1	Anormal	$S_i = a_{i-1}, b_{i-1}$	
1	0	1	0	Anormal	$S_i = 0$	
1	0	1	1	Décalage gauche a_i	$S_i = a_{i-1} \cdot b_{i-1}$	
1	1	0	0	Anormal	$S_i = 1$	
1	1	0	1	Anormal	$S_i = 1$	
1	1	1	0	Anormal	$S_i = 1$	
1	1	1	1	Anormal	$S_i = 1$	

Le bit S0 dépend toujours de C_{iN} la retenue entrante.

Tableau I. Fonctions réalisées par l'opérateur de calcul du MC 68000 en cas d'erreur sur la génération du code opération uniquement.

Bloc IV - Retenue entrante :

Celle-ci peut prendre les valeurs 0, 1 ou x ; il est nécessaire de savoir ce que devient la fonction si cette retenue n'est pas correcte. Le tableau II résume le fonctionnement complet de l'ALU en fonction du code opération, ce dernier étant le code correct.

Code op				\overline{CiN}	Sortie S0 à S31	Retenue sortante COUT	Anormal / correct
SL	CAND	\overline{COR}	\overline{CXOR}				
0	0	0	0	1	$Si = ai + di$	1	correct (OR)
0	0	0	0	0	$Si = ai + di \oplus CiN$	$\overline{ai + di + CiN}$	Anormal
0	0	1	0	1	$Si = ai \oplus di$	1	(X-OR) correct
0	0	1	0	0	$Si = ai \oplus di \oplus CiN$	$\overline{ai \oplus di + CiN}$	Anormal
0	0	1	1	1	$Si = ai \oplus di \oplus \overline{CiN}$	$\overline{aidi} + \overline{aidi.CiN}$	Correct (ADD)
0	0	1	1	0	$Si = ai \oplus di \oplus \overline{CiN}$	$\overline{aidi} + \overline{aidi.CiN}$	Correct (ADD+X)
0	1	1	1	1	$Si = aidi \oplus \overline{CiN}$	0	Anormal
0	1	1	1	0	$Si = aidi$	0	Correct (AND)
1	0	1	1	1	So = 0 Si = CiN	\overline{aidi}	Correct (SL)
1	0	1	1	0	So = 1 Si = CiN	\overline{aidi}	Anormal

Tableau II. Fonctions réalisées par l'opérateur de calcul du MC 68000 en cas d'erreur sur la retenue entrante uniquement.

Bloc V - Précharge de l'opérateur :

Elle doit être valide pendant toute la durée de l'opération. C'est elle qui

permet la propagation de la retenue.

En cas de panne de précharge, au mieux la retenue sera propagée jusqu'au 3ème étage de l'opérateur, mais si un des trois premiers étages propage une retenue nulle, celle-ci sera propagée sur tout le reste de l'opérateur.

Le tableau III résume le fonctionnement obtenu en fonction du code opération (correct) et de la retenue entrante (correcte).

Code op				\overline{CIN}	Sortie avec panne de précharge			
SL	CAND	\overline{COR}	\overline{CXOR}		S0	S1	S2	$S_i, i > 2$
0	0	0	0	1	$a_0 + d_0$	$a_1 + d_1 \oplus \overline{S_0}$	$a_2 + d_2 \oplus \overline{S_0 S_1}$	$\overline{a_i + d_i}$
0	0	1	0	1	$a_0 \oplus d_0$	$a_1 \oplus d_1 \oplus \overline{S_0}$	$a_2 \oplus d_2 \oplus \overline{S_0 S_1}$	$\overline{a_i \oplus d_i}$
0	0	1	1	1	$a_0 \oplus d_0$	$a_1 \oplus d_1 \oplus \overline{S_0}$	$a_2 \oplus d_2 \oplus \overline{S_0 S_1}$	$\overline{a_i \oplus d_i}$
0	0	1	1	0	$\overline{a_0 \oplus d_0}$	$\overline{a_1 \oplus d_1}$	$\overline{a_2 \oplus d_2}$	$\overline{a_i \oplus d_i}$
0	1	1	1	0	$a_0 d_0$	$a_1 d_1$	$a_2 d_2$	$a_i d_i$
1	0	1	1	1	0	0	0	0

Tableau III. Fonctionnement de l'opérateur de calcul du MC 68000 en cas de panne de précharge uniquement.

Enfin la multiplicité d'erreur est limitée de la manière suivante :

- S'il y a une erreur concernant l'un des blocs I, II, III, IV et V (figure 2), il n'y en a pas concernant les quatre autres, à moins

qu'il y ait génération de commandes correspondant à une autre instruction du répertoire. Ainsi, une erreur multiple simultanément sur le code opération, la retenue entrante et le multiplexage d'entrée, ne peut correspondre qu'à une instruction existante dans le répertoire, par exemple une addition transformée en soustraction.

- S'il y a une erreur de sélection de l'origine sur l'un des bus, pour une instruction à 2 opérands, il n'y en a pas sur l'autre.
- S'il y a une erreur de sélection sur l'origine de l'un des bus, il n'y en a pas sur les commandes l'opérateur.

EXEMPLE :

1 - Opération à 1 opérande : décalage à gauche du registre Di :

L'opération de décalage à gauche sera notée SHIFT. Le résultat correct devant être $S_i = a_{i-1}.1$ si on se reporte aux notations des tableaux I, II ou III.

a) vérifier la bonne origine sur le bus ABD :

(cf. § IV - hypothèses de pannes de la gestion des transferts).

b) vérifier la bonne commande du bloc I :

L'entrée correcte est le bus ABDi direct.

- $R \neq \text{SHIFT (ALUB)}$ ((correspond à une mauvaise entrée donc bonne fonction du MUX).

- $R \neq \text{SHIFT } (\overline{D_i})$

- $R \neq \text{SHIFT } (D_i + 1)$

- $R \neq \text{SHIFT } (\overline{D_i + 1})$

} (Bonne entrée donc mauvaise fonction du MUX)

c) vérifier la bonne commande du bloc II :

L'entrée correcte est KALUD = 1.

- R \neq SHIFT ($D_i.0$)
 - R \neq SHIFT ($D_i.DAA$ (ALUREG))
 - R \neq SHIFT ($D_i - DBD_i$)
- } (mauvaise entrée sur bloc II
donc bonne entrée du bloc I).

d) vérifier la bonne commande du bloc III :

l'entrée correcte est $SL = 1$, $CAND = 0$, $\overline{COR} = 1$, $\overline{CXOR} = 1$.

R \neq 15 Résultats possibles donnés par la table I avec

$a_j = D_{ij}$ et $d_j = 1$.

e) vérifier la bonne commande du bloc IV :

la bonne retenue est $\overline{CiN} = 1$

R \neq SHIFT (D_i) avec $R_0 \neq 1$ Résultat donné par la table II avec

$a_j = D_{ij}$ et $d_j = 1$.

f) vérifier la bonne commande du bloc V :

La précharge doit être active.

R \neq 0 Résultat donné par la table III.

g) vérifier la bonne instruction :

R \neq tous les résultats possibles si une autre instruction modifiant D_i était générée.

h) vérifier la bonne destination du résultat R vers D_i .

(cf. § IV. Hypothèses de pannes de la gestion des transferts).

2 - Opération à 2 opérandes : ou logique des registres D_i et D_j ,

Résultat dans D_i :

L'opération sera notée $OU(D_i, D_j)$, le résultat correct devant être $R_i = D_{ii} + D_{ji}$ si on se reporte aux notations des tables I, II ou III.

- a) vérifier la bonne origine sur le bus ABD et la bonne origine sur le bus DBD (cf. § IV. Hypothèses de pannes de la gestion des transferts).
- b) vérifier la bonne commande du bloc I :
L'entrée correcte est le bus ABD_i direct :
 $R \neq$ OU (ALUB, D_j) mauvaise entrée du bloc I donc bonne entrée du bloc II.
 $R \neq$ OU ($\overline{D_i}$, D_j)
 $R \neq$ OU (D_{i+1} , D_j)
 $R \neq$ OU ($\overline{D_{i+1}}$, D_j) } bonne entrée du bloc II, mauvaise fonction du MUX.
- c) vérifier la bonne commande du bloc II :
L'entrée correcte est le bus DBD_i .
 $R \neq$ OU (D_i , KALUD = 0)
 $R \neq$ OU (D_i , KALUD = 1)
 $R \neq$ OU (D_i , DAA (ALUREG)) } mauvaise entrée du bloc II donc bonne entrée du bloc I.
- d) vérifier la bonne commande du bloc III :
L'entrée correcte est $SL = 1$, $CAND = 0$, $\overline{COR} = 1$, $\overline{CXOR} = 1$.
 $R \neq$ 15 Résultats possibles donnés par la table I avec
 $a_i = D_{ii}$ et $d_i = D_{ji}$.
- e) vérifier la bonne commande du bloc IV :
La bonne retenue est $\overline{CiN} = 1$.
 $R \neq$ résultat donné par la table II pour le code opération correcte.
- f) vérifier la bonne commande du bloc V :
La précharge doit être active.
 $R \neq$ résultat donné par la table III pour le code opération correcte.

g) vérifier la bonne instruction :

R ≠ tous les résultats possibles si une autre instruction modifiant D_i était générée.

h) vérification de la bonne destination du résultat R vers D_i (cf. § IV).

III.2. Opérateur arithmétique (AUH, AUL) :

Les commandes de cet opérateur de calcul d'adresses sont représentées à la figure 3.

Le MC 68000 comporte 2 opérateurs de calcul d'adresses : AUH et AUL identiques quant à leur fonctionnement, le premier opérant sur les poids faibles (bits 0 à 15), le second sur les poids forts (bits 16 à 31).

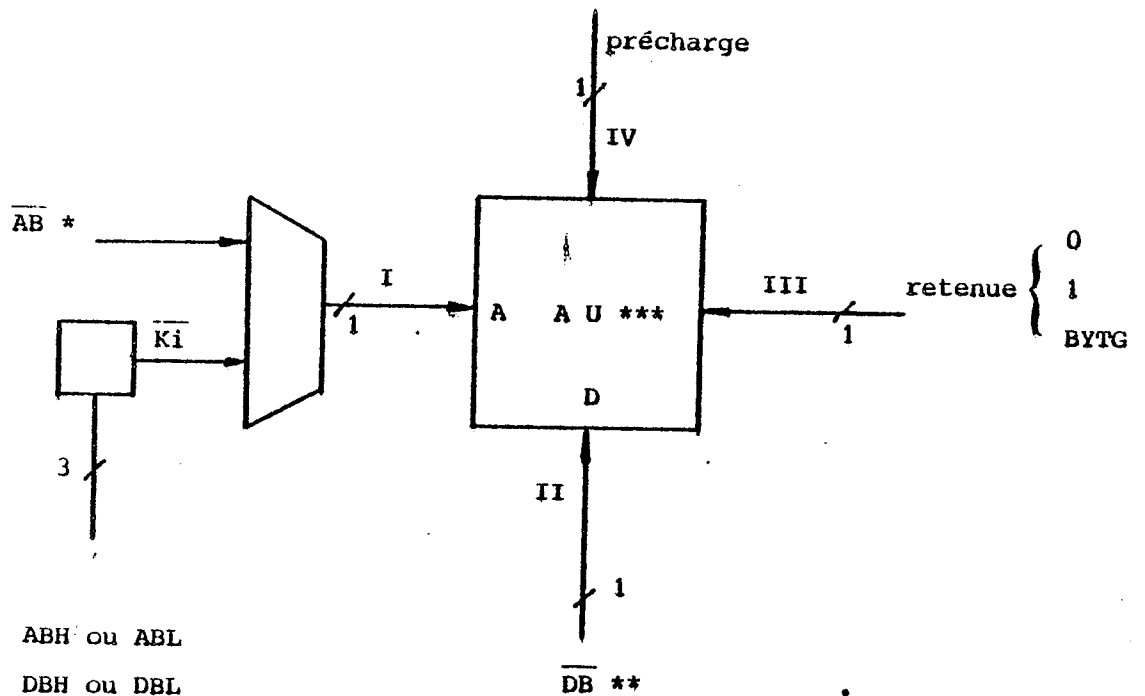


Figure 3. Commandes de l'opérateur de calcul d'adresses du MC 68000.

Les AU fonctionnent sur les valeurs complémentées. L'entrée A de l'AU est connectée sur le bus \overline{AB} ainsi que la constante $\overline{K_i}$ permettant d'incrémenter de +4, +2, +1, 0, -1, -2, -4 l'adresse présente à l'entrée D.

L'entrée D de l'AU n'est pas multiplexée et reçoit le bus DB.

La retenue entrante peut prendre la valeur 0, 1 ou BYTG, signal provenant de la partie contrôle permettant de paramétrer l'incrémentatation pour un octet ou un mot, c'est-à-dire que l'AU effectue, selon BYTG : +1/+2 et -1/-2.

La structure du circuit est telle que les diverses fonctions réalisables par l'opérateur sont celles indiqués par la table IV.

Entrée constante K_3	Entrée K_2	Entrée K_1	Entrée AB	Retenue entrante Cin	fonctionnement	Sortie sans panne précharge	Sortie avec panne précharge	
							S0 S1 S2	$S_i, i > 2$
1	1	1	-	1	Correct	D + 0	Idem colonne précédente	$S_i = \overline{D_i}$
1	1	1	-	0	Anormal	D + 1		$S_i = \overline{D_i}$
1	1	0	-	1	Correct	D + 1	Idem colonne précédente	$S_i = \overline{D_i}$
1	1	0	-	0	Correct	D + 2		$S_i = \overline{D_i}$
1	0	1	-	1	Correct	D + 2	Idem colonne précédente	$S_i = \overline{D_i}$
1	0	1	-	0	Anormal	D + 3		$S_i = \overline{D_i}$
1	0	0	-	1	Anormal	D + 3	Idem colonne précédente	$S_i = \overline{D_i}$
1	0	0	-	0	Correct	D + 4		$S_i = \overline{D_i}$
0	1	1	-	1	Correct	D - 4	Idem colonne précédente	$S_i = D_i$
0	1	1	-	0	Anormal	D - 3		$S_i = D_i$
0	1	0	-	1	Anormal	D - 3	Idem colonne précédente	$S_i = D_i$
0	1	0	-	0	Anormal	D - 2		$S_i = D_i$
0	0	1	-	1	Correct	D - 2	Idem colonne précédente	$S_i = D_i$
0	0	1	-	0	Correct	D - 1		$S_i = D_i$

Tableau IV. Fonctions réalisées par l'opérateur arithmétique du MC 68000 en cas d'erreur sur les commandes (début).

Entrée constante K_3	Entrée K_2	Entrée K_1	Entrée \overline{AB}	Retenue entrante $C_i N$	fonctionnement	Sortie	
						sans panne précharge	Sortie avec panne précharge
0	0	0	-	1	Anormal	D - 1	S0 S1 S2 Si, i > 2
0	0	0	-	0	Anormal	D - 0	Idem colonne précédente Si = Di
-	-	-	\overline{AB}_i	1	Correct	D + A	Idem colonne précédente Si = Ai + Di
-	-	-	\overline{AB}_i	0	Anormal	D + A + 1	Si = Ai + Di

Tableau IV. Fonctions réalisées par l'opérateur arithmétique du MC 68000 en cas d'erreur sur les commandes

(fin).

Il est aisé pour cet opérateur d'admettre des pannes multiples sur toutes les commandes : génération de constante, retenue entrante et précharge. La multiplicité d'erreurs est donc limitée de la manière suivante :

- Les erreurs sur les commandes de l'opérateur ne sont pas admises en plus d'erreur de sélection des origines sur les bus,
- Si il y a erreur de sélection sur l'origine sur l'un des bus, pour une opération à 2 opérandes, il n'y en a pas sur l'autre, ce pour chaque opération.
- Si il y a erreurs simultanées sur les 2 opérateurs AUH et AUH, les erreurs de commandes sont identiques pour les 2 opérateurs.

EXEMPLE :

Calcul de $D_i + 2$:

- a) vérifier la bonne origine sur le bus DBD (cf. § IV).
- b) vérifier les bonnes commandes du bloc AUL :
 - . R ≠ 15 Résultats possibles pour l'addition de constantes sans panne de précharge, donnés par la table IV.
 - . R ≠ 16 Résultats possibles pour l'addition de constantes avec panne de précharge, donnés par la table IV.
 - . R ≠ 2 Résultats possibles pour l'addition du bus AB sans panne de précharge, donnés par la table IV.
 - . R ≠ 2 Résultats possibles pour l'addition du bus AB avec panne de précharge, donnés par la table IV.
- c) vérifier les bonnes commandes du bloc AUH (idem § b).
- d) vérifier la bonne instruction ;

R \neq tous les résultats possibles si une autre instruction modifiant D_i était générée.

e) vérifier la bonne destination du résultat R_i vers D_i (cf. § IV).

III.3. Opérateur d'extension de l'ALU (ALUE) :

Les commandes de cet opérateur d'extension de l'ALU sont représentées à la figure 4. Ce circuit est utilisé pour effectuer les décalages. En effet, ceux-ci sont réalisés sur la concaténation ALUE + ALU quelle que soit la taille de l'opérande. Les 2 sont toujours chargés et le calcul s'effectue sur les 2, mais en cas d'opération sur l'octet ou le mot, seul le contenu de ALU LATCH est pris en compte.

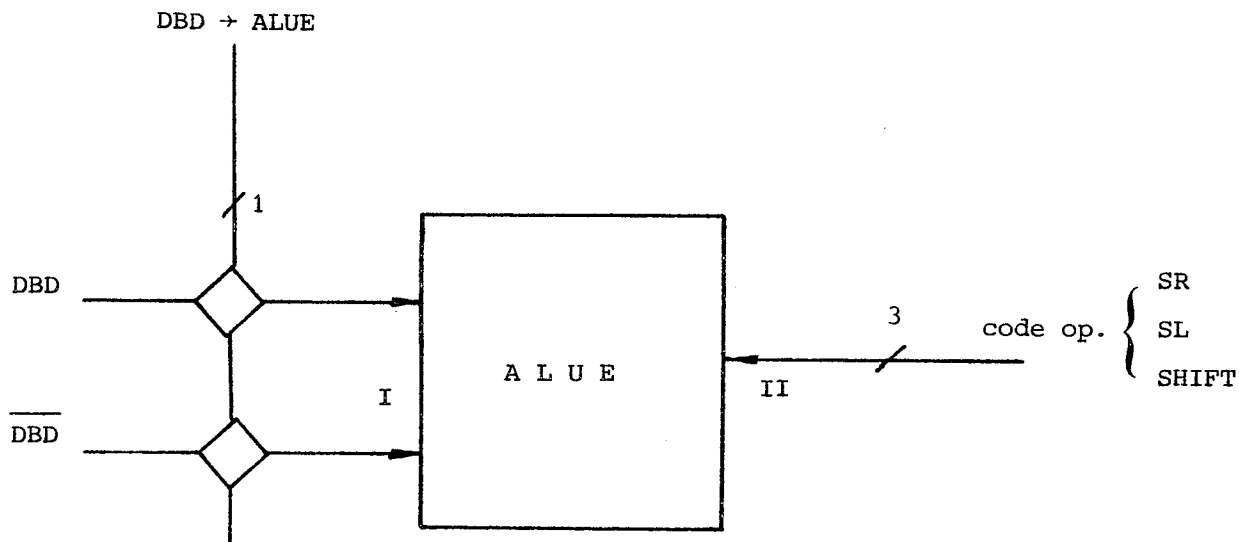


Figure 4. - Commandes de l'opérateur d'extension d'ALU : ALUE.

A l'entrée de l'ALUE est connecté le bus DBD (direct et complémenté), le code-op est constitué de 3 commandes, permettant le décalage à gauche, à droite, la 3ème commande validant le décalage. Sur les 9 valeurs possibles du code-op, deux seuls sont utilisés.

Il est nécessaire de déterminer la fonction réalisée pour les 7 autres. La table V résume les différents fonctionnements possibles en cas d'erreur sur les commandes.

SR	SL	SHIFT	Fonctionnement	Sortie (i = 1 à 14)
0	0	1	Anormal	$R_{(i)} = 1$
0	1	1	Normal	$R_{(i)} = R_{(i+1)}$
1	0	1	Normal	$R_{(i)} = R_{(i-1)}$
1	1	1	Anormal	$R_{(i)} = R_{(i-1)} R_{(i+1)}$
x	x	0	Anormal	$R_{(i)} = R_{(i)}$

R_0 et R_{15} dépendent du sens de transfert et des lignes ALUE(0) IN et ALUE(15) IN
 $R_{-1} = ALUE(0) IN$ et $R_{16} = ALUE(15) IN$.

Tableau V. Fonctionnement de l'ALUE du MC 68000 en cas d'erreur sur la génération des commandes.

Il est aisé pour cet opérateur d'admettre des pannes multiples. On se limitera par contre à la phase de génération correcte de chaque signal de commande.

En effet, il n'y aurait pas stabilisation de chaque étage de l'opérateur si les signaux SR, SL et SHIFT étaient simultanément valides.

En outre, la multiplicité d'erreurs est limitée comme il suit,

- S'il y a erreur concernant l'un des blocs, il n'y en a pas concernant les autres.
- S'il y a une erreur de sélection sur l'origine, il n'y en a pas sur les commandes de l'opérateur.
- Si la commande de connexion du bus DBD ne marche pas dans un sens (sur T3) elle ne marche pas dans l'autre (T1).

EXEMPLE :

Décalage (32 bits) de D_i à gauche (SL)

On notera l'opération SHIFT L (Ri). On ne s'occupera que de la vérification des poids forts, les poids faibles étant traités par l'ALU (cf. § III.1 opérateur arithmétique et logique ALU).

- a) vérifier la bonne origine du bus DBD (cf. § IV),
- b) Vérifier la bonne commande du bloc d'entrée :
 - . $R \neq FF$. (pas de changement ==> pas de sortie sur le bus).
- c) vérifier les bonnes commandes du bloc ALUE :
 - . $R \neq 4$ Résultats possibles donnés par la table V.
- d) vérifier la bonne instruction :
 - $R \neq$ tous les résultats possibles si une autre instruction modifiant D_i était générée.

- e) vérifier la bonne destination du résultat R_i vers D_i :
(cf. § IV).

III.4. Opérateur d'extension de signe (SIGNEX) :

Cet opérateur permet d'étendre le signe d'une donnée se trouvant sur les bus ABL , \overline{ABL} et DBL , \overline{DBL} . Les commandes de cet opérateur sont représentées à la figure 5.

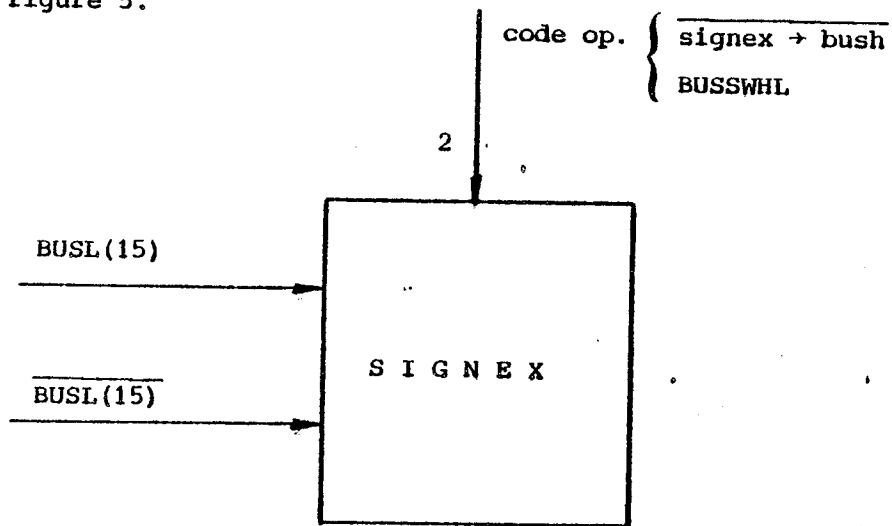


Figure 5. - Commandes de l'opérateur d'extension de signe du MC 68000.

Le MC 68000 possède en fait 2 opérateurs identiques, l'un opérant sur ABL , \overline{ABL} et effectuant l'extension sur ABH , \overline{ABH} , l'autre sur DBL , \overline{DBL} et effectuant l'extension sur DBH , \overline{DBH} .

L'opérateur ne possède qu'une seule commande en fait, c'est la commande Signex + bus h, mais la présence de la commande de switch des bus (BUSSWHL) est nécessaire car le résultat de l'opération est complètement différent

selon la position de cette commande. Le fonctionnement de cet opérateur est présenté à la table VI.

BUSL(15)	$\overline{\text{BUSL(15)}}$	SIGNEX	BUSSWHL	Fonctionnement	BUSL(i) i=0 à 15	$\overline{\text{BUSL(i)}}$ i=0 à 15	BUSH(i) i=0 à 15	$\overline{\text{BUSH(i)}}$ i=0 à 15
ϕ	$\overline{\phi}$	1	0	incorrect	inchangé	inchangé	inchangé	inchangé
0	1	0	0	Correct	inchangé	inchangé	0	inchangé
1	0	0	0	Correct	inchangé	inchangé	inchangé	0
0	1	0	1	incorr.	* 0	* inchangé	* 0	$\overline{\text{BUSL(i)}}$
1	0	0	1	incorr.	* inchangé	* 0	* BUSL(i)	* 0

*Possibilité de conflits entre le bus direct et le bus complémenté : ceux-ci n'ayant plus des valeurs complémentées l'une de l'autre.

Tableau VI. Fonctionnement de l'opérateur SIGNEX du MC 68000 en cas de pannes des commandes.

Il est aisé pour cette opérateur d'admettre des pannes de commandes en même temps que des pannes sur la sélection origine du bus.

EXEMPLE :

Extension de signe du registre D_i par l'intermédiaire du bus DBL :

- a) vérifier la bonne origine sur le bus DBL (cf. § IV).
- b) vérifier le bon fonctionnement des commandes :
R ≠ 4 Résultats possibles donnés par la table VI.

c) vérifier la bonne instruction :

R \neq tous les résultats possibles si une autre instruction modifiait D_1 .

d) vérifier la bonne destination du résultat R vers D_1 (cf. § IV).

III.5 Opérateur encodeur de priorité (PREN) :

Ce circuit est utilisé (essentiellement pour le masque de l'instruction MOVEM) pour reconnaître le premier bit du bus ABL qui est au niveau logique "1" (en partant des poids faibles).

Les commandes de cet opérateur sont représentées à la figure 6.

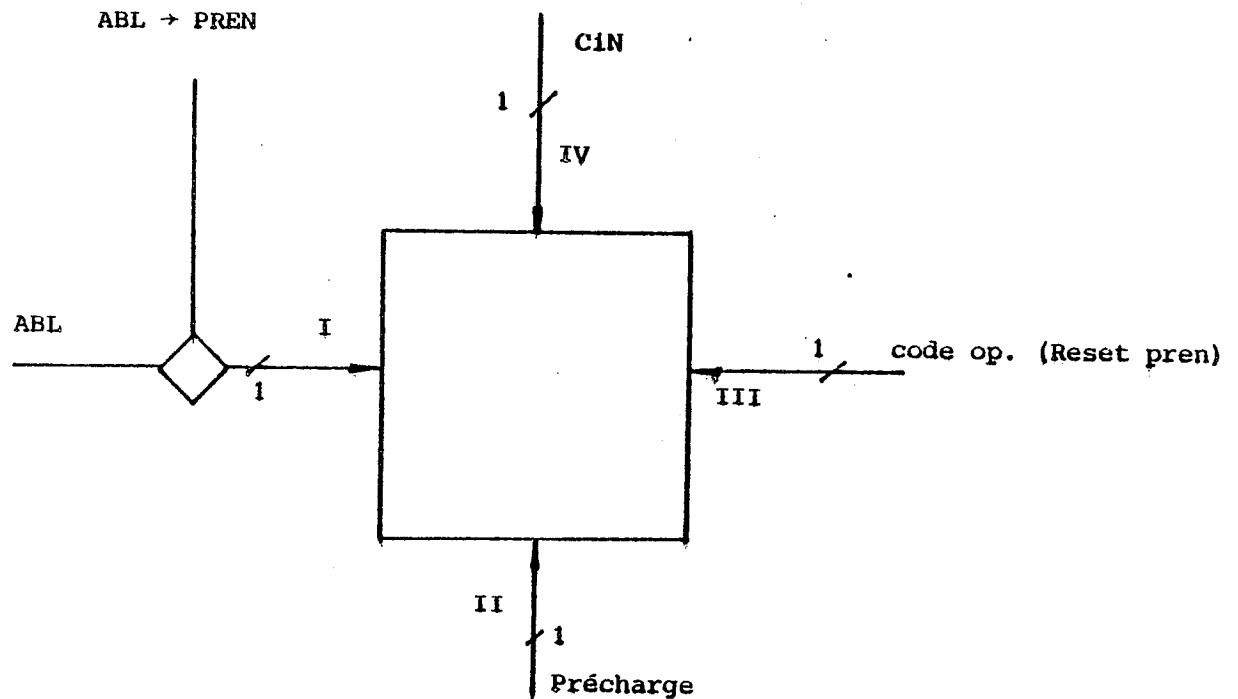


Figure 6. - Commandes de l'opérateur PREN du MC 68000.

Bloc I : Entrée du masque de MOVEM :

La commande doit être active sur la phase T1.

Si la commande ABL → PREN est inactive, l'opérateur reste alors dans l'état laissé précédemment, il y aura alors sortie du registre 15 (sélectionné par défaut) puis fin d'opération. Si la commande ABL → PREN est active en permanence, les données transférées depuis ou vers les registres correspondant au masque, transitant par le bus ABL, vont alors perturber la valeur du masque en venant se substituer à lui. Selon les valeurs, il peut y avoir boucle infinie sur la sortie d'un registre dont la valeur correspond à sa propre sélection en sortie.

Par exemple : le bit 0 du masque ayant la valeur "1" correspondant à la sélection du registre R0, si c'est une sortie qui est demandée (registre vers mémoire) et que le registre R0 contient la valeur "1" sur le bit 0. Il y aura bouclage sur la sortie de R0.

La commande peut être décalée ; qu'elle soit active en T2 ou en T3, le résultat est le même : sortie du registre 15 puis du masque complet.

Si la commande est active en T4 : sortie du registre 15 puis fin d'opération.

On limitera les hypothèses aux :

- Pannes de la commande d'entrée du masque dans l'opérateur ne sont pas admises en plus des pannes de sélection des registres sur les bus.

Bloc II : Précharge de l'opérateur :

La précharge doit être valide pendant chaque phase T1. Elle peut alors être active sur une autre phase, ou active en permanence, ou inactive.

Si elle est active en permanence, le fait que la sortie du registre boucle sur l'entrée aura comme conséquence : de faire sauter un ou plusieurs bits du masque.

En effet, la présence simultanée de Précharge et Reset-Pren; va faire basculer sur T3 un certain nombre de points du registre de PREN. Des simulations électriques seraient nécessaires afin de déterminer le nombre de points pouvant basculer.

Si la précharge est inactive, la valeur de sortie de PREN est indépendante de la valeur du registre de PREN. Certains MOS ont leurs grilles flottantes, la valeur de sortie du PREN dépend alors uniquement de la charge des capacités de grilles.

N'importe quel registre peut-être sélectionné, plusieurs entrées de l'encodeur peuvent être validés simultanément provoquant ainsi la sortie du ET logique des valeurs binaires qui correspondent aux différents numéros de registres sélectionnés. L'état stable étant la sortie d'un numéro de registre. (capacité de grille déchargée).

Si la phase de précharge est décalée, il y a alors décalage de la sortie du registre (la phase de prise en compte du compte-rendu étant fixé).

Si la précharge est active en T2 : le décalage va provoquer la sortie du registre 15 (sélectionné par défaut) puis la sortie normale des autres registres correspondants au masque sauf celui correspondant au dernier bit à '1' du masque.

Si la précharge est active en T3 : le comportement est identique à celui obtenu en cas de précharge active en permanence (certains bits du masque sont sautés).

Si la précharge est active en T4 : le décalage va provoquer la sortie du registre 15 sélectionnée par défaut puis la sortie normale des autres registres correspondant au masque y compris celui qui correspond au dernier bit à '1' du masque.

Bloc III : code opération :

La commande unique Resetpren doit être active sur T3. Plusieurs cas de panne peuvent être pris en compte : commande inactive, active en permanence ou décalée de sa phase normale d'exécution.

Si la commande est inactive, dans ce cas l'opérateur va boucler sur la sortie du numéro du registre correspondant au premier bit à '1' du masque (sauf si tous les bits sont à zéro = sortie du registre 15 uniquement).

Si la commande est active en permanence, le fait que la sortie du registre reboucle sur l'entrée va avoir comme conséquence de 'sauter' un ou plusieurs bits de masque. En effet, la présence simultanée de la Précharge et de Reset pren va faire basculer sur T1 un certain nombre de points du registre de PREN. Des simulations électriques seraient nécessaires afin de déterminer le nombre de points pouvant basculer.

Si la phase d'activation de la commande reset pren est décalée, le comportement sera le suivant :

- Si la commande reset pren est active sur T1 : le comportement est le même que si elle était active en permanence.
- Si la commande reset pren est active sur T2 ou sur T4 : le comportement reste correct, seule la consommation en courant risque d'augmenter légèrement.

Bloc IV : retenue entrante :

En fonctionnement normal, la retenue entrante d'un pas est égale à la retenue

sortante du pas précédent. Pour le 1er pas la retenue entrante est connectée à la masse, aucune commande de la partie contrôle n'influe sur son fonctionnement. Aucune panne n'est donc à prendre en compte vis-à-vis de cette entrée.

La multiplicité d'erreurs est limitée de la manière suivante :

- S'il y a une erreur concernant l'un des blocs I, II, III, il n'y en a pas concernant les 2 autres, à moins qu'il y ait génération de commandes correspondant à une autre instruction du répertoire (i.e. n'utilisant pas le PREN).
- S'il y a une erreur de sélection sur l'origine sur le bus AB, il n'y en a pas sur les commandes de l'opérateur.
- L'utilisation de l'opérateur impliquant un transfert depuis ou vers les registres figurant dans le masque, les pannes de sélection (source ou destination) de ces registres ne sont pas admises en plus des pannes de l'opérateur.

EXEMPLE :

Utilisation de l'instruction :

MOVEM.

- a) vérifier la bonne origine sur ABD.
(cf. § IV. Hypothèses de pannes de la gestion des transferts).
- b) vérifier la bonne commande du bloc I :
 - R ≠ sortie unique du registre 15.
 - R ≠ sortie des registres correspondant à leur contenu.
 - R ≠ sortie du masque précédé de la sortie du registre 15.

c) vérifier la bonne commande du bloc II :

- R ≠ plusieurs bits du masque ont été sautés.
- R ≠ valeur du masque.
- R ≠ sortie du masque sauf le dernier bit à '1', précédé de la sortie du registre 15.
- R ≠ sortie du masque précédé de la sortie du registre 15.

d) vérifier la bonne commande du bloc III :

- R ≠ sortie infinie du registre correspondant au premier bit à '1' du masque.
- R ≠ plusieurs bits du masque ont été sautés.

e) vérifier la bonne instruction :

R ≠ tous les résultats possibles si une autre instruction modifiant la ou (les) destination(s) était générée.

f) vérification de la bonne destination du résultat.

(cf. § IV. Hypothèses de pannes de la gestion des transferts).

III.6. Opérateur FOOL-IT :

Ce circuit est utilisé dans le cas où les 8 bits poids faibles du bus ABD doivent être écrits dans un registre données sans changer le contenu des 8 bits poids forts de ce registre.

Cet opérateur affectant uniquement les rangements dans un registre, il sera étudié au paragraphe IV : hypothèses de pannes de la gestion des transferts.

III.7. Opérateur de manipulations de bits : (DCR) :

Ce circuit est utilisé pour les instructions de manipulations de bits telles

que BSET, BCLR, BCHG. Il effectue la mise à "1" d'un bit du bus $\overline{\text{DBD}}$ et "0" sur DBD parmi 16 dans le cas d'un mot, parmi 32 dans le cas d'un mot long.

Les commandes de cet opérateur sont représentées à la figure 7.

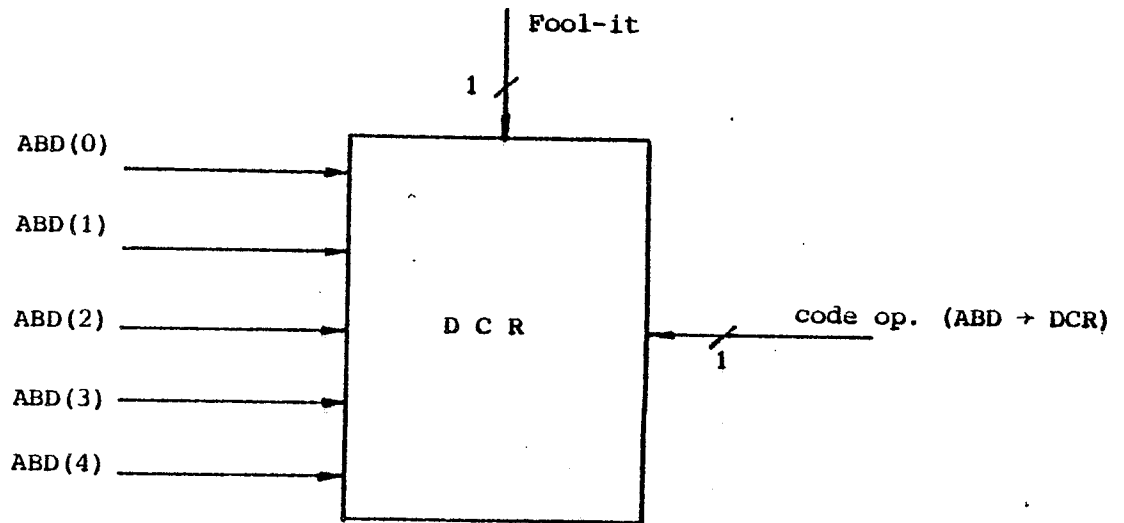


Figure 7. - Commandes de l'opérateur DCR du MC 68000.

Le fonctionnement en cas de pannes sur les commandes, impose de prendre en compte la valeur du bit ABD4.

Le tableau VII donne le fonctionnement de l'opérateur en cas de panne sur les commandes, code-op et fool-it.

ABD → DCR	Fool-it	ABD4	Fonctionnement	Le n° du bit modifié* est donné par
0	0	0	Anormal	** Ancienne valeur - R(0).R(1).R(2).R(3) (mod 16).
0	0	1	Anormal	** Ancienne valeur - R(0)R(1)R(2) R(3)R(4) (mod 32).
0	1	0	Anormal	** Ancienne valeur - R(0)R(1)R(2) (mod 8)
0	1	1	Anormal	** Ancienne valeur - R(0)R(1)R(2)R(4) (mod 8 poids forts)
1	0	0	Correct	ABD(0) ABD(1) ABD(2) ABD(3) (mod 16)
1	0	1	Correct	ABD(0) ABD(1) ABD(2) ABD(3) ABD(4) (mod 32)
1	1	0	Correct	ABD(0) ABD(1) ABD(2) (mod 8)
1	1	1	Anormal	ABD(0) ABD(1) ABD(2) ABD(4) (mod 8 poids forts)

* modifié → mis à "1" sur \overline{DBD} et mis à "0" sur DBD.

** Ancienne valeur = valeur stockée dans le registre de DCR lors de la précédente utilisation.

Tableau VII. Fonctions réalisées par le DCR du MC 68000 en cas d'erreur sur la génération des commandes.

La multiplicité des pannes sera limitée pour cet opérateur aux conditions suivantes :

- Les deux opérateurs DCR et ALU étant utilisés conjointement, s'il y a une erreur sur les commandes d'un des opérateurs, il n'y en a pas sur l'autre.

- S'il y a une erreur de sélection d'origine sur l'un des bus (trois origines sont nécessaires pour l'opération complète) il n'y en a pas sur les autres.
- Les erreurs de sélection d'origine sur les bus ne sont pas admises en plus des erreurs sur les commandes des opérateurs (DCR et ALU).
- Les hypothèses de pannes concernant l'opérateur ALU sont celles décrites au paragraphe III.1.

EXEMPLE :

BCHG.L D_i , D_j

D_i est le numéro du bit à changer dans D_j .

On notera RP le résultat partiel à la sortie du DCR.

L'opération exécutée par l'ALU est un ou exclusif entre D_j et le résultat du DCR.

- a) vérifier la bonne origine sur ABD (cf. § IV).
- b) vérifier les bonnes commandes du DCR.
RP \neq 7 résultats possibles donnés par la table VII.
- c) vérifier la bonne origine du bus DBD (sortie du DCR) et la bonne origine du bus ABD (registre D_j) (cf. § IV).
- d) vérifier le fonctionnement correct de l'ALU (cf. § III.1.).
- e) vérifier la bonne instruction :
R \neq tous les résultats possibles si une autre instruction modifiant D_j était générée.
- f) vérifier la bonne destination du résultat R vers D_j (cf. § IV).

Remarque :

Pour pouvoir s'assurer que la valeur du bus ABD entre bien dans le DCR, et que ce n'est pas l'ancienne valeur du registre qui est utilisée, il est nécessaire de lancer 2 opérations successives ayant chacune un numéro de bit différents (dont le codage sur ABD est différent sur les bits 0, 1 et 2 au moins) et de vérifier que ces 2 opérations se font correctement toutes deux.

III.8. Opérateur de multiplexage d'octets (MUX I/O) :

Ce circuit est utilisé pour rendre les entrées/sorties d'octets indépendantes de la parité de l'adresse. En effet, la mémoire adressée par le MC 68000, l'est au niveau du mot (Tous les mots ayant des adresses paires). L'adressage d'octets se fait au moyen des lignes \overline{UDS} (upper data strobe) et \overline{LDS} (lower data strobe) permettant de distinguer l'octet poids forts (adresse paire) de l'octet poids faible (adresse impaire). Le MUX I/O permet la copie de l'octet valide (poids faible ou poids fort selon la parité de l'adresse) sur l'octet inutilisé.

Les commandes de cet opérateur sont présentées à la figure 8.

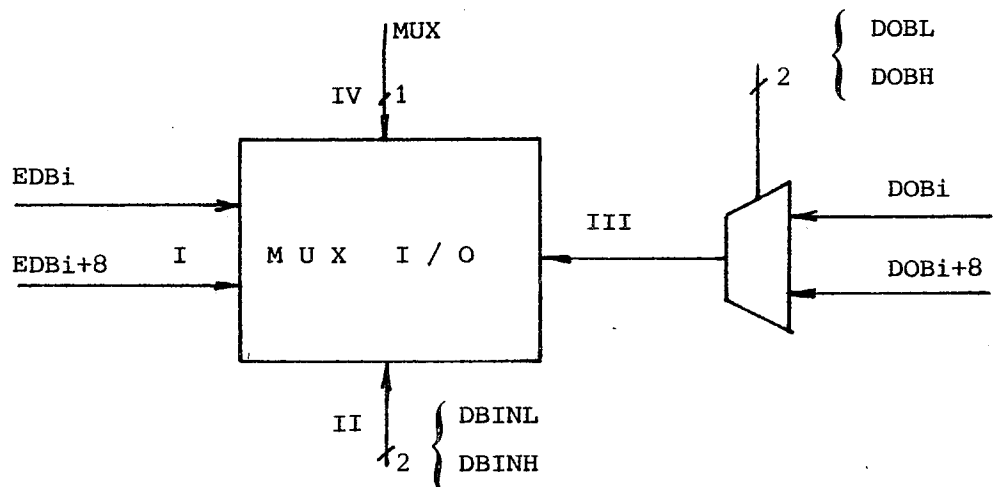


Figure 8. - Commandes de l'opérateur MUX I/O du MC 68000.

Les lignes EDB correspondent au bus données externes reliant l'opérateur MUX I/O aux plots de données.

Les commandes DBINL et DBINH permettent de choisir l'octet ou de sélectionner le mot complet en entrée.

Les commandes DOBL et DOBH permettent de choisir l'octet ou de sélectionner le mot complet en sortie.

La commande MUX effectue le multiplexage des lignes.

Bloc I : Entrée du bus de données externe :

Ce bus entre directement dans l'opérateur, aucune commande n'influe sur cette entrée, aucune panne des commandes n'est donc à prendre en compte.

Bloc II : commandes d'entrées :

Pour l'entrée d'un octet, seule la commande DBINL doit être active si la commande DBINH l'est, il n'y a pas création d'erreur, seul l'octet poids faible transitera par le bus (présence du fool-it).

Si les commandes sont actives alors qu'elles ne devraient pas (sortie sur les plots), il n'y a pas création d'erreur, la validation d'un transfert du registre sur le bus ne se faisant qu'après une lecture, le buffer sera donc rechargé avec la bonne valeur.

Si les commandes sont inactives, alors qu'elles devraient l'être (soit DBINL soit DBINH et DBINH), dans ce cas, le registre DBIN conservera la valeur acquise précédemment.

Bloc III : commandes de sorties :

Lors de la sortie d'un octet : la commande DOBL doit être active seule, si les 2 commandes DOBL et DOBH sont actives simultanément, la valeur sortie correspond au ET logique de l'octet poids faible et de l'octet poids forts du registre DOB (l'octet poids fort contenant la valeur correspondant à l'octet poids forts du mot précédemment sortie sur les plots).

Si la commande DOBH est active seule, c'est l'octet poids fort du registre DOB qui sort sur les plots (valeur correspondant à l'octet poids fort du mot précédemment).

Si aucune commande n'est active, le bus EDB est 'en l'air'. La sortie est indépendante de la valeur du registre DOB.

Lors de la sortie d'un mot : les commandes DOBL et DOBH doivent être actives simultanément.

Si une des commandes ou les 2 sont inactives, l'octet (ou les octets) correspondant du bus EDB est 'en l'air', la sortie de l'octet (ou du mot) est indépendante de la valeur du registre DOB.

Lors d'une lecture : les commandes sont inactives. Si une seule des commandes est active, l'octet correspondant reçoit en entrée le ET logique de la valeur présente en entrée sur les plots et de l'octet du registre DOB correspondant à la commande active.

Si les 2 commandes sont actives simultanément, lors de la lecture d'un mot, le mot entré correspond alors au ET logique du mot présent sur les plots et du registre DOB.

Si les 2 commandes sont actives simultanément, lors de la lecture d'un octet, l'octet entrée correspond alors au ET logique de l'octet présent

sur les plots d'entrées, de l'octet poids faible du registre DOB, et de l'octet poids fort du registre DOB.

Bloc IV : Commande de multiplexage :

La commande doit être active lors de l'entrée ou de la sortie d'un octet, inactive lors de l'entrée ou de la sortie d'un mot.

Lors de l'entrée d'un octet : si elle est inactive, il n'y a aucune perturbation pour la lecture d'un octet d'adresse impaire, par contre lors de la lecture d'un octet d'adresse paire le bus est en haute impédance, le registre DBIN n'est pas modifié.

Lors de la sortie d'un octet : si elle est inactive il n'y a aucune perturbation pour l'écriture d'un octet d'adresse impaire, par contre lors de l'écriture d'un octet d'adresse paire, l'octet correspondant du bus EDB est 'en l'air' la sortie est indépendante du contenu du registre DOB.

Lors de l'entrée d'un mot, si la commande est active, le registre DBIN reçoit sur chaque octet le ET logique des 2 octets du bus EDB.

Lors de la sortie d'un mot, si la commande est active le bus EDB reçoit le ET logique des 2 octets du registre DOB.

Les hypothèses sont limitées comme suit :

- Si il y a une panne concernant l'un des blocs, il n'y en a pas concernant les autres.
- Les pannes de transferts sur les bus ne sont pas admises en plus des pannes de l'opérateur.

EXEMPLE :

1 - Sortie d'un résultat octet :

MOVE.B Di, @paire

- a) vérifier la bonne origine Di sur le bus.
(cf. § IV. Hypothèses de pannes de la gestion des transferts).
- b) vérifier la bonne commande du bloc 1.
- pas d'erreur possible.
- c) vérifier la bonne commande du bloc 2.
- pas d'erreur possible.
- d) vérifier la bonne commande du bloc 3.
- $R_i \neq ET(DOB(i), DOB(i+8))$
- $R_i \neq DOB(i+8)$
- $R_i = Di$.
- e) vérifier la bonne commande du bloc 4.
 $R_i = Di$
- f) vérifier la bonne instruction :
 $R \neq$ tous les résultats possibles si une autre instruction modifiant la mémoire d'adresse @ était générée.

2 - Entrée d'un mot :

MOVE.W @, Di.

- a) vérifier la bonne commande du bloc 1.
- pas d'erreur possible.

b) vérifier la bonne commande du bloc 2.

$R_i \neq$ ce qu'il serait si un des octets ou les deux conservaient la valeur précédemment acquise.

c) vérifier la bonne commande du bloc 3.

$R_i \neq$ ET logique (EDB_i, DOB_i) ($i = 0$ à 16)

$R_i \neq$ ET logique (EDB_i, DOB_i) ($i = 0$ à 8) EDB_i ($i = 8$ à 16)

$R_i \neq$ ET logique (EDB_i, DOB_i) ($i = 8$ à 16) EDB_i ($i = 0$ à 8).

d) vérifier la bonne commande du bloc 4.

$R_i \neq$ ET logique ($EDB_i, EDB_i + 8$).

e) vérifier la bonne instruction.

$R \neq$ tous les résultats possibles si une autre instruction modifiant D_i était générée.

f) vérification de la destination du résultat.

(cf. § IV. Hypothèses de pannes de la gestion des transferts).

IV. - HYPOTHESES DE PANNES DE LA GESTION DES TRANSFERTS

La gestion des transferts se divise en deux parties, d'une part la gestion du bus et d'autre part la sélection des sources et destinations impliquées dans le transfert.

Pour les microprocesseurs N-MOS LSI, des hypothèses simples résultaient de l'implémentation : équipotentielle portant le ET des différentes entrées et bus préchargés à la valeur UN. Ceci était dû au fait que les microprocesseurs N-MOS LSI utilisaient des bus unifilaires connectés à des points mémoire asymétriques comme le montre la figure 9.

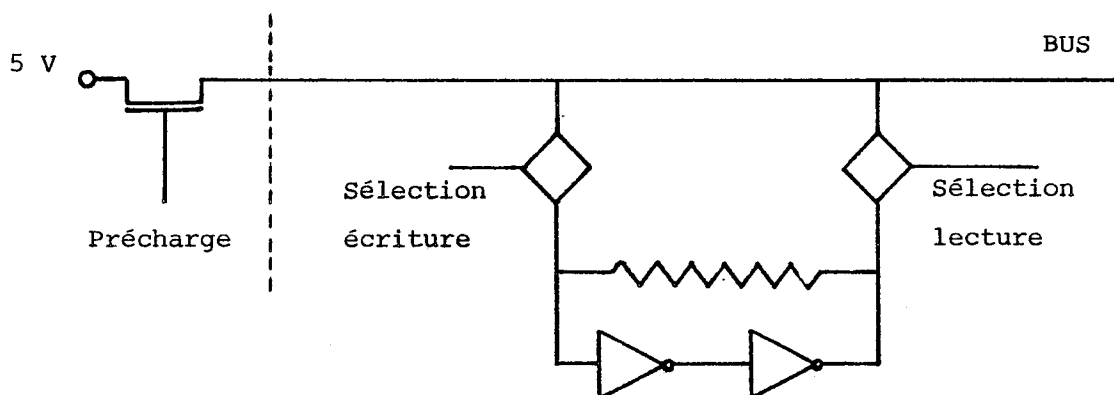


Figure 9. - Connexion d'un point asymétrique à un bus unifilaire.

Pour les microprocesseurs VLSI, le concept de bus bifilaire ou bus différentiel est couramment employé, les points mémoires peuvent alors être symétriques permettant un gain de surface important utilisé dans le MC 68000 est présenté à la figure 10.

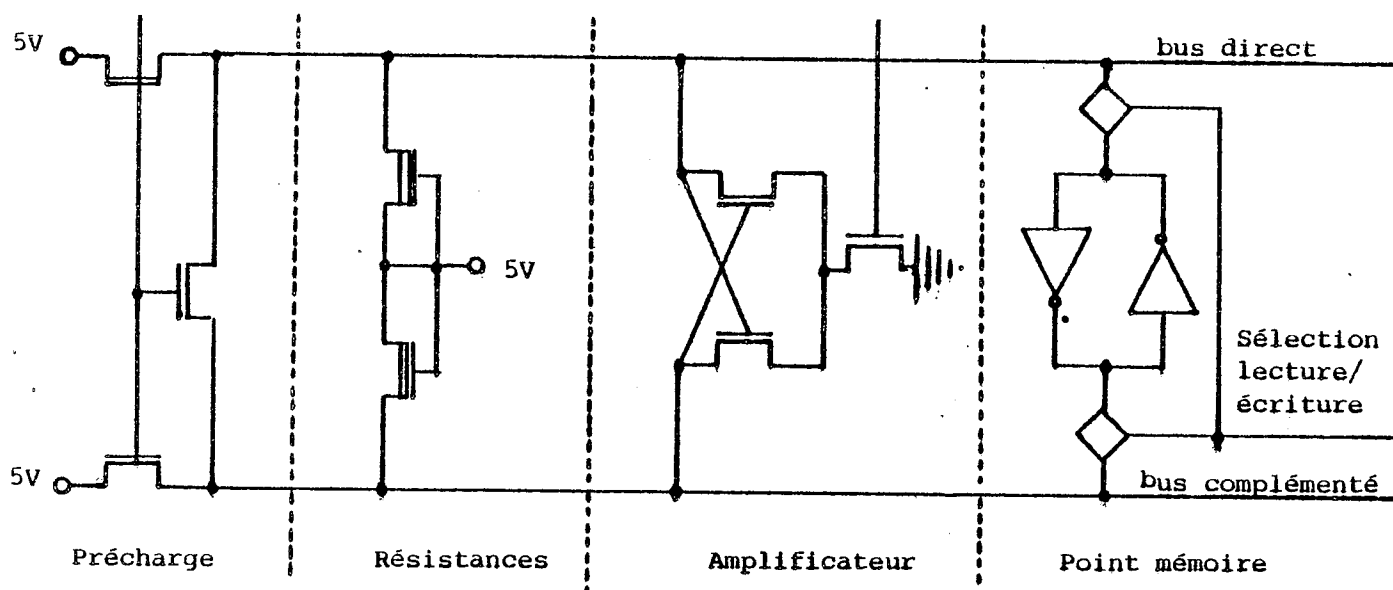


Figure 10. - Connexion d'un point symétrique sur un bus différentiel et circuit de gestion du bus du MC 68000.

Pour ce type de bus, des simulations électriques ont alors été nécessaires afin de déterminer les conséquences des pannes considérées. En effet, avec les nouvelles techniques de conception des circuits intégrés, les conséquences des pannes considérées sont à priori imprévisibles.

Les paragraphes suivants décrivent les résultats obtenus au cours des simulations électriques ainsi que les hypothèses de fonctionnement, qui en découlent en cas de panne.

Le fonctionnement des bus internes du MC 68000 a déjà été décrit au Chapitre VI § IV. La figure 11 présente le chronogramme de fonctionnement d'un transfert, et donne les valeurs des tensions du bus, des points mémoire source et destination. La figure 12 présente les valeurs des tensions lorsqu'aucun transfert n'est exécuté sur ce bus (bus inutilisé).

Fonctionnement correct :

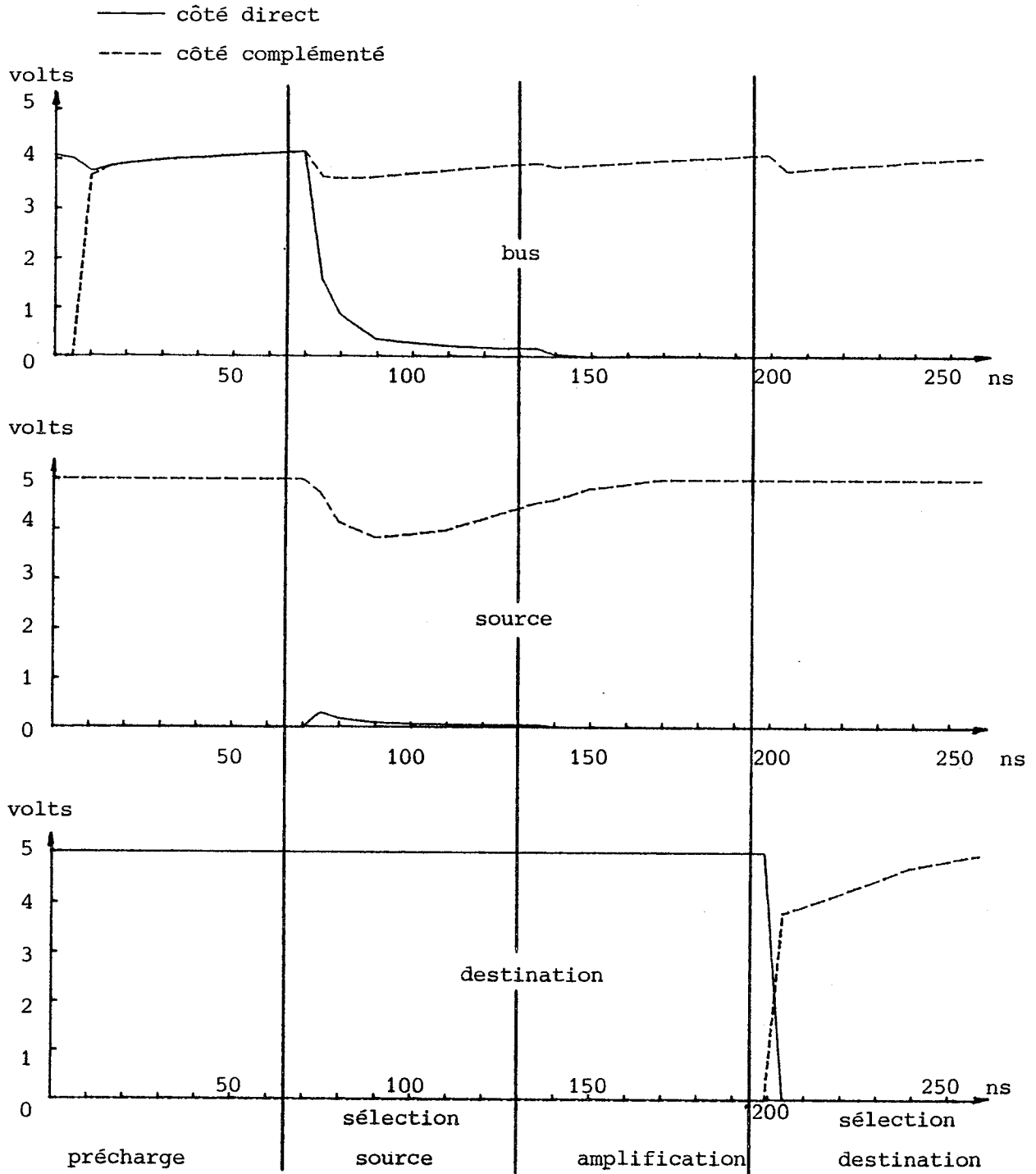


Figure 11. - Fonctionnement du bus lors d'un transfert.

Pas de transfert : (fonctionnement normal)

—— côté direct
----- côté complémenté

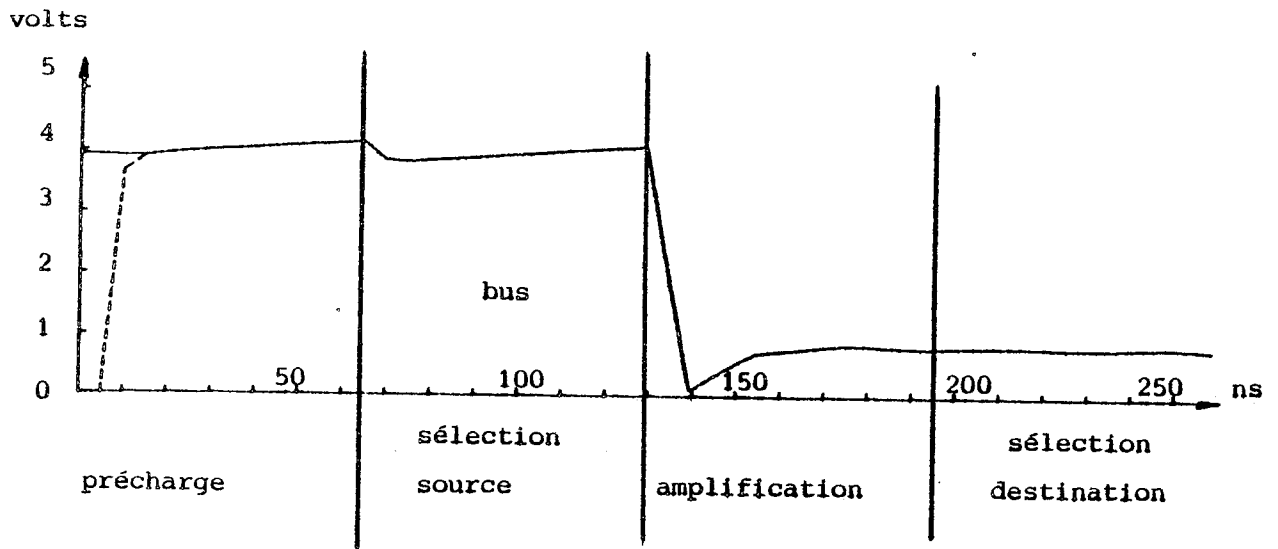


Figure 12. - Fonctionnement du bus sans transfert.

IV.1. Pannes de gestion du bus :

IV.1.1. Pannes de précharge :

Deux cas sont donc à observer pour les pannes de précharge. Le premier cas concerne un bus non préchargé, c'est-à-dire lorsque la commande de précharge est collée à zéro. Le second cas concerne un bus préchargé en permanence, c'est-à-dire lorsque la commande de précharge est collée à un.

IV.1.1.1. Précharge inexistante :

Il est clair que cette panne est transparente lorsque deux ou plusieurs bus sont connectés ensemble et qu'une seule des précharges ne fonctionne pas correctement.

Les conséquences de cette panne sont différentes selon le fait qu'un transfert a eu lieu ou non sur ce bus au cycle précédent.

a) - Il n'y a pas eu de transfert au cycle précédent. :

Dans ce cas, comme le montre la figure 12, le cycle précédent n'a vu qu'une précharge sur T4 puis la connexion de l'ampli sur T2, T3, ce qui a stabilisé le bus à environ 0.82 V sur les 2 lignes.

La figure 13, présente le chronogramme des tensions des points source et destination ainsi que les valeurs du bus pendant ce transfert sans précharge. Il en résulte donc que :

- . Le registre source n'est pas modifié.
- . Le bus prend la valeur du registre source.
- . Le(s) registre(s) destination(s) prend (prennent) la valeur du bus (registre source).

Tout se passe donc correctement, même en présence de plusieurs destinations.

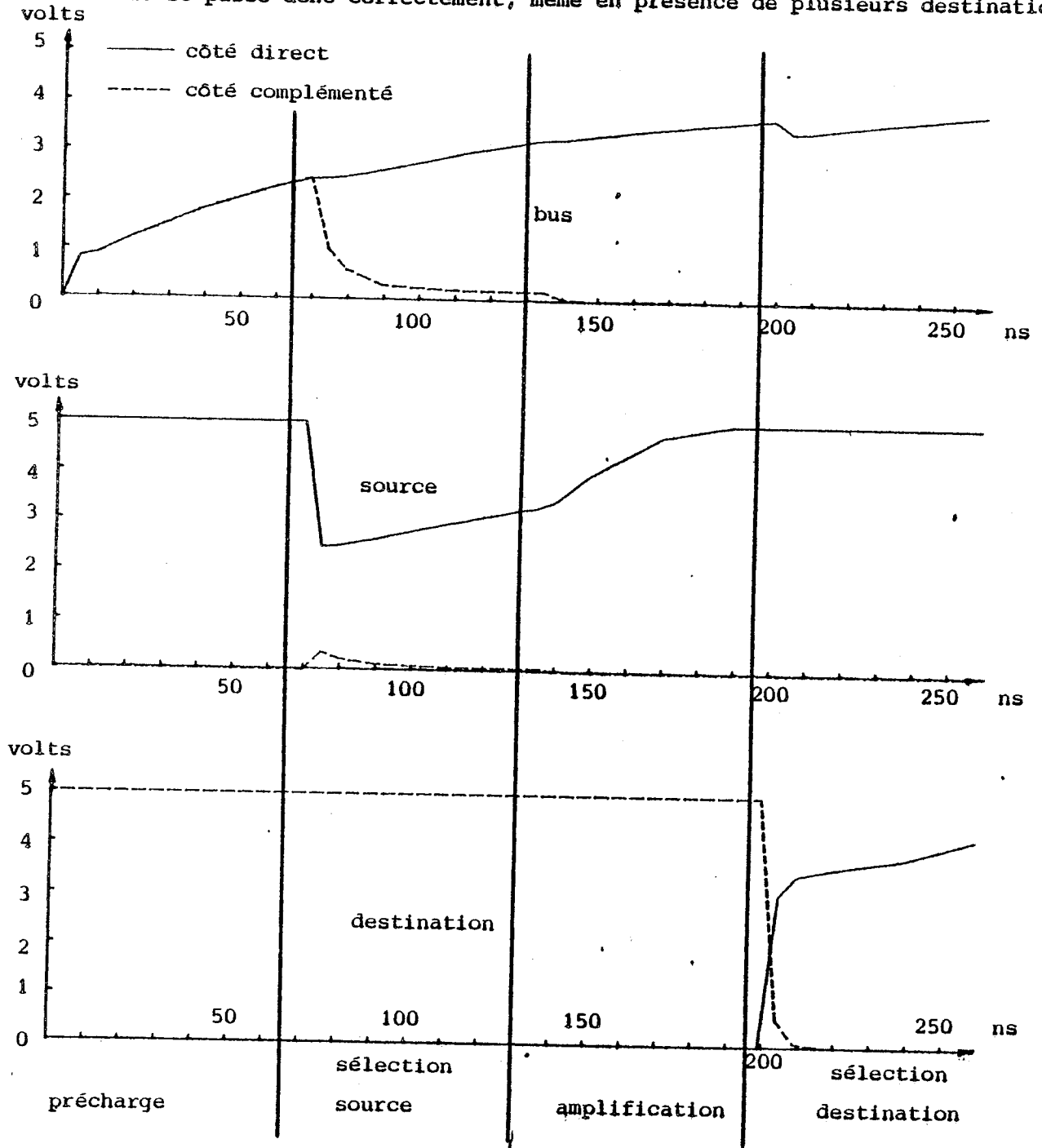


Figure 13. - Chronogramme d'un transfert sans précharge lorsqu'il n'y a pas eu de transfert au cycle précédent.

b) - Il y a eu un transfert au cycle précédent :

Dans ce cas, comme le montre la figure 11, le cycle de transfert précédent se termine sur un bus chargé aux valeurs du transfert précédent. On voit donc, que le bus qui se trouve au niveau '0', monte légèrement pendant la phase de précharge, ceci vient des résistances de bus. Mais, dès que la source est connectée, le bus reprend les valeurs du cycle précédent et les impose au registre source. Puis l'ampli est connecté, il maintient le bus dans cet état, et l'impose au registre destination, quelque soit le nombre de destinations.

La figure 14, présente le chronogramme des tensions des points source et destination ainsi que les valeurs du bus pendant ce transfert.

Il en résulte donc que :

- . Le registre source prend la valeur de la source du cycle précédent.
- . Le bus conserve la valeur du transfert précédent.
- . Le(s) registre(s) destination(s) prend (prennent) la valeur du bus (valeur du transfert précédent).

Tout se passe donc comme si le registre source avait été connecté en destination au cycle précédent.

Les résultats obtenus sont les mêmes pour toutes les plages de fréquence du MC 68000 de 4 à 8 MHz.

IV.1.1.2. Précharge permanente :

Il est clair que cette panne est maintenue même lorsque deux ou plusieurs bus sont connectés ensemble.

Précharge collée à '0' - (Transfert au cycle précédent).

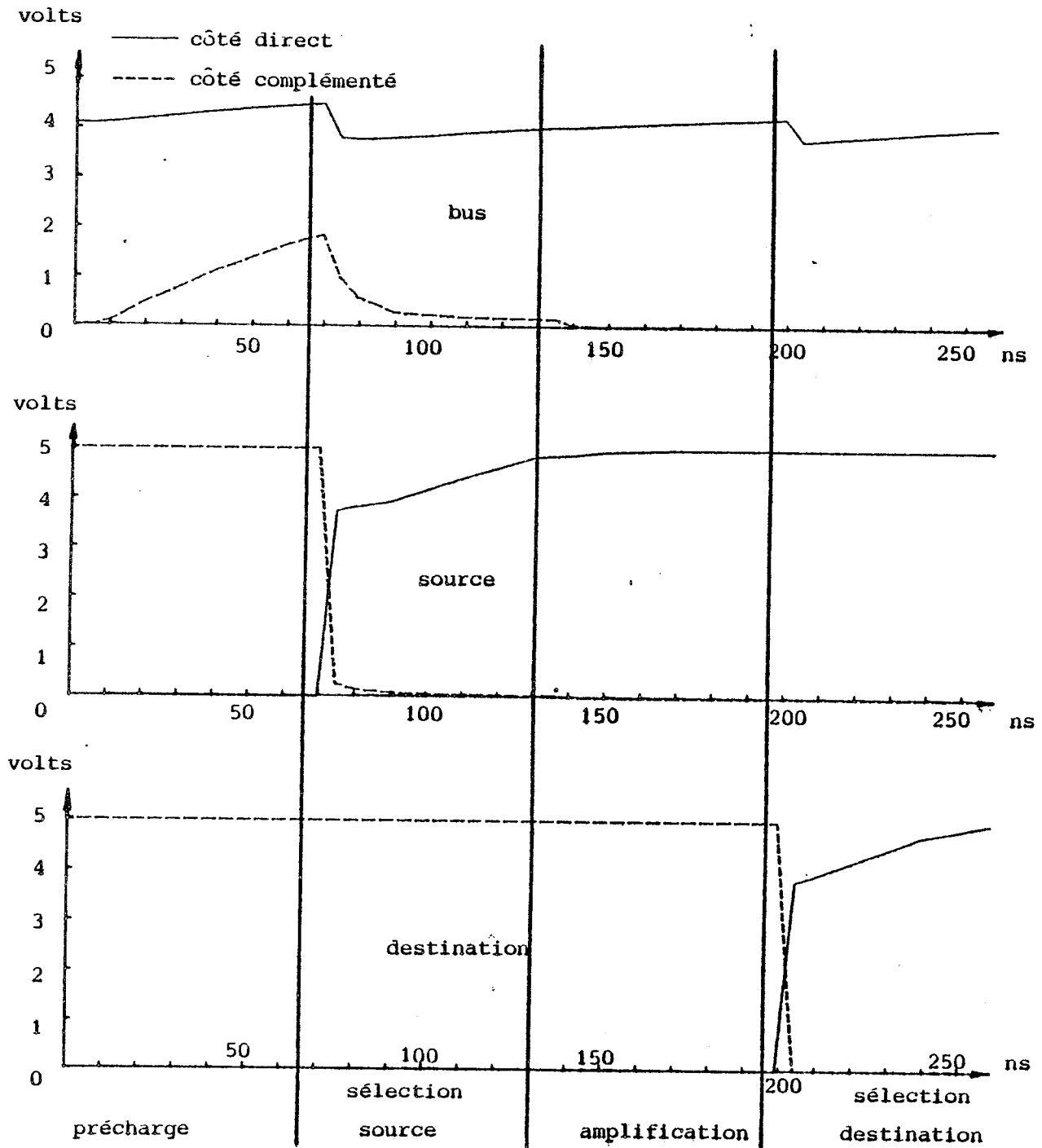


Figure 14. - Chronogramme du bus en cas de panne de précharge avec transfert au cycle précédent.

Pour cette panne trois cas sont à distinguer.

a) - Il y a une seule destination :

Dans ce cas, comme le montre la figure 15, le registre source n'est pas modifié. Le registre destination n'est pas modifié non plus.

Il en résulte donc que :

- . Le registre source n'est pas modifié.
- . Le bus reste préchargé (2.45 V).
- . Le registre destination n'est pas modifié.

Tout se passe donc comme si aucun transfert n'avait été commandé.

b) Il y a plusieurs destinations, autant à '1' qu'à '0' :

Dans ce cas, il y a plusieurs destinations, mais le nombre de registres destinations à '1' est égal au nombre de registres destinations à zéro. La figure 16 donne le chronogramme des tensions du bus et des registres destinations, deux ici, ayant donc des valeurs complémentaires. On s'aperçoit donc que les registres sont légèrement perturbés mais reprennent leurs valeurs après déconnection. Il en résulte donc :

- . Le registre source n'est pas modifié.
- . Le bus reste préchargé (2.45 V).
- . Les registres destinations conservent leurs valeurs.

Tout se passe donc comme si aucun transfert n'avait été commandé.

Précharge collée à '1' :

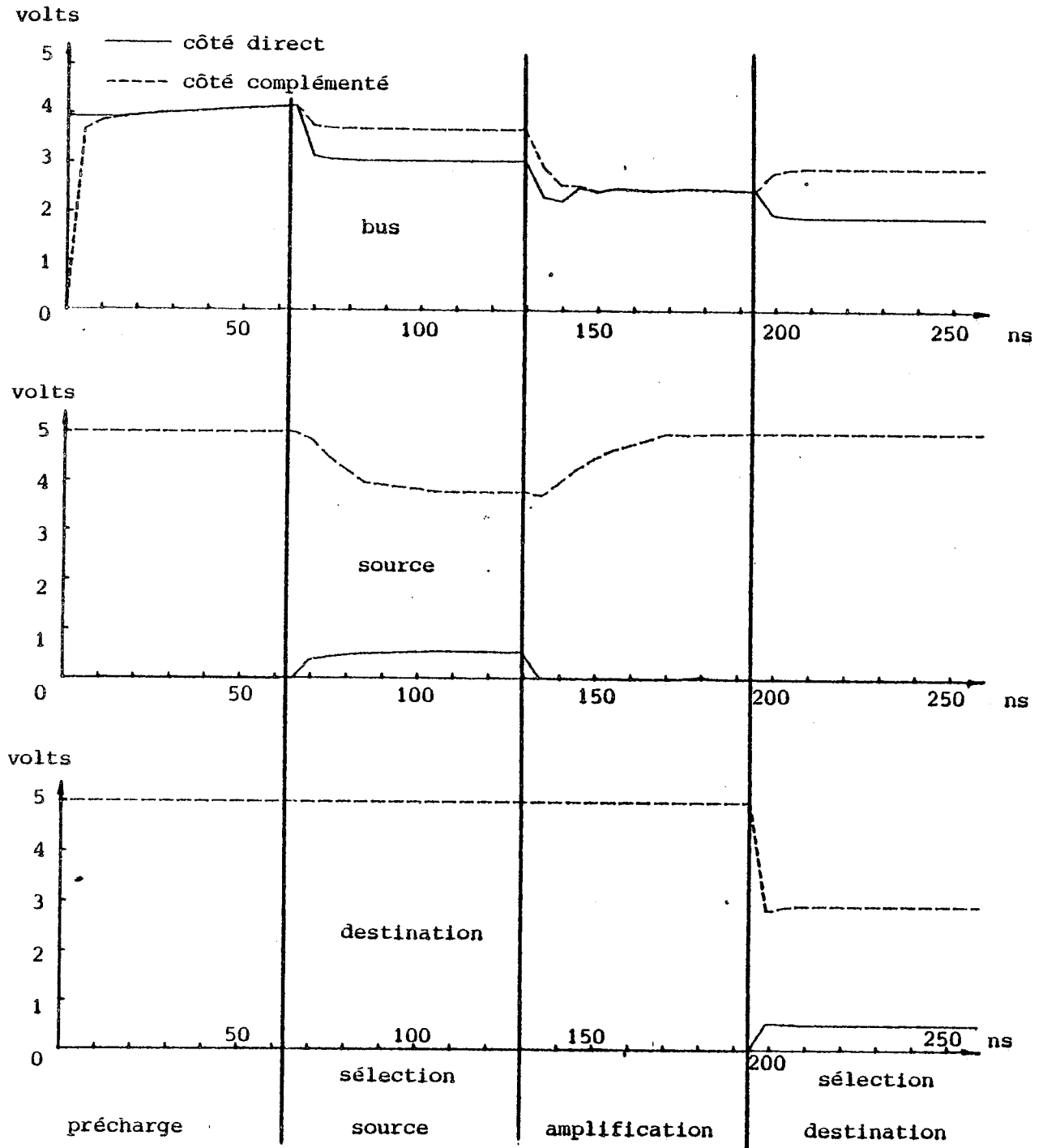


Figure 15. - Chronogramme du bus en cas de précharge permanente.

Précharge collée à '1' :

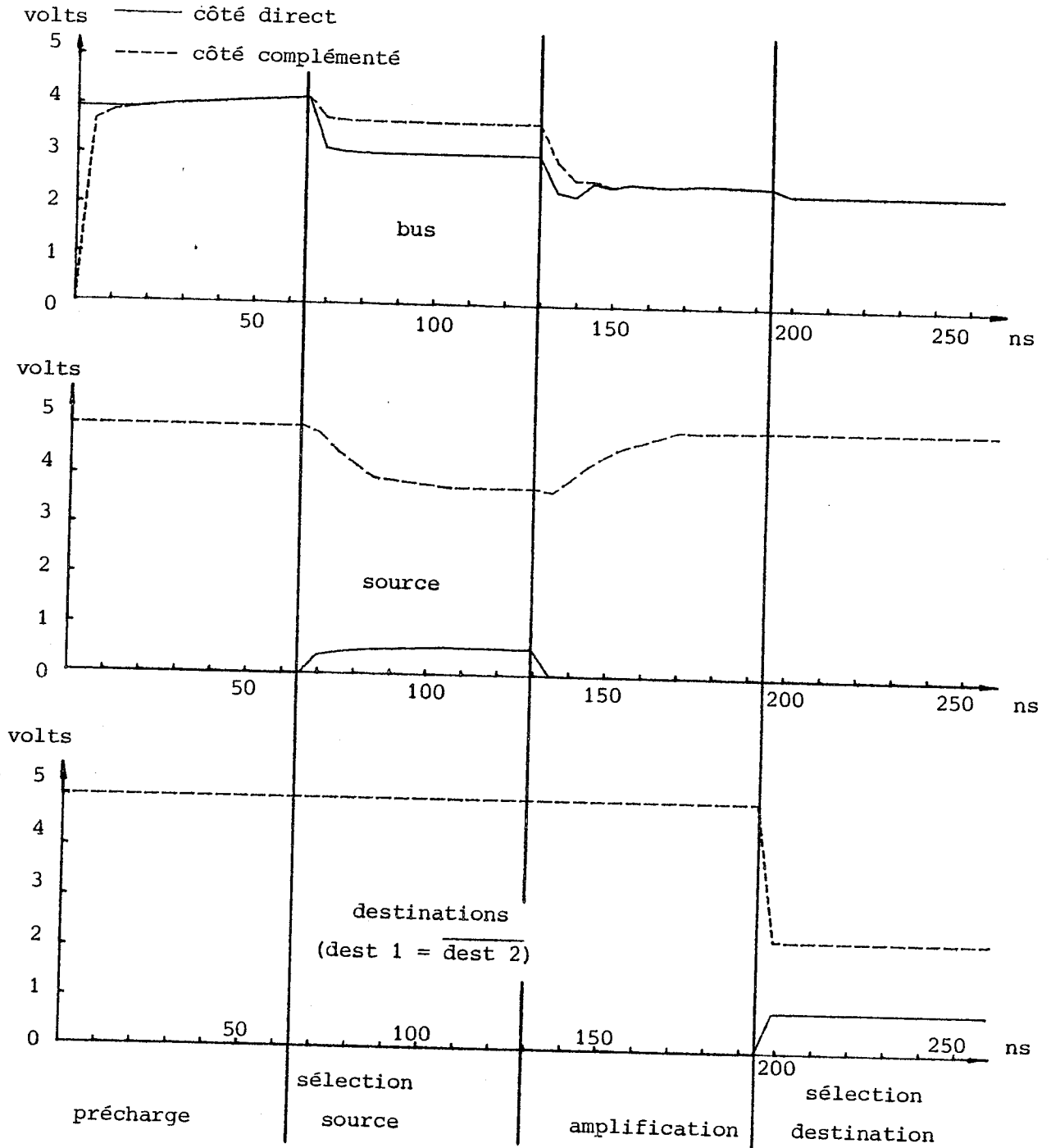


Figure 18. - Chronogramme du bus en cas de précharge permanente, 2 destinations complémentaires.

c) - Il y a plusieurs destinations, pas le même nombre à '1' qu'à '0' :

Dans ce cas, il y a plusieurs destinations, mais le nombre de registres destinations à '1' n'est pas égal au nombre de registres destinations à zéro. La figure 17 donne le chronogramme des tensions du bus et des registres destinations.

On s'aperçoit donc que c'est la majorité des registres qui l'emporte et qui s'impose au bus et aux autres registres.

Il en résulte donc que :

- . Le registre source n'est pas modifié.
- . Le bus reste préchargé (2.45 V).
- . Les registres destinations se positionnent à la valeur de la majorité de ces registres.

Ce type de résultat et tout à fait nouveau et particulier, il modifie et précise les hypothèses faites dans [THA-78] comme quoi les fonctions obtenues en cas de connexions multiples se traduisaient par des ET ou des OU des registres connectés.

IV.1.2. Pannes d'amplification :

Comme pour les pannes de précharge, on distingue pour les pannes d'amplification, le cas où la commande est collée à zéro du cas où celle-ci est collée à un.

Ici, le mauvais fonctionnement va se traduire uniquement par une forte consommation en courant du circuit.

Précharge collée à '1' :

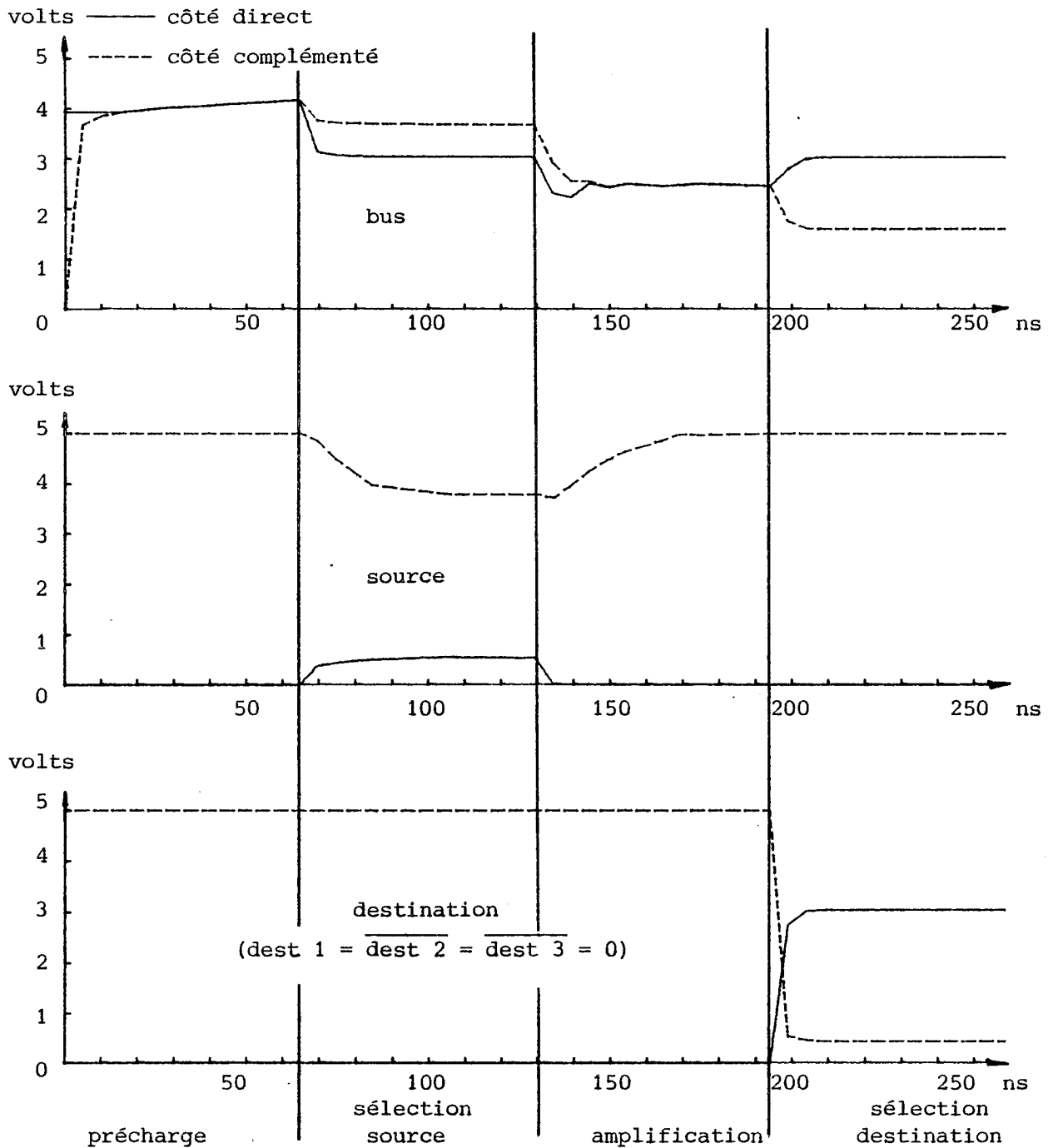


Figure 17. - Chronogramme du bus en cas de précharge permanente, 3 destinations.

IV.1.2.1. Amplification inexistante :

La figure 18 donne le chronogramme des tensions du bus et des registres source et destination.

Il en résulte donc que :

- . Le registre source n'est pas modifié.
- . Le bus prend la valeur du registre source.
- . Le registre destination prend la valeur du bus (valeur du registre source) quel que soit le nombre de destinations.

IV.1.2.2. Amplification permanente :

Les figures 19 et 20 donnent le chronogramme des tensions du bus et des registres source et destination, qu'il y ait ou non transfert au cycle précédent.

Il en résulte donc que :

- . Le registre source n'est pas modifié.
- . Le bus prend la valeur du registre source.
- . Le registre destination prend la valeur du bus (valeur du registre source) quel que soit le nombre de destinations.

IV.1.3. Pannes de switches :

On ne s'intéresse ici qu'au fonctionnement en panne c'est-à-dire commande collée à 'zéro' lorsqu'elle doit être active ou commande collée à 'un' lorsqu'elle doit être inactive.

Les conséquences de ces pannes sont simples car elles se ramènent à des pannes sur les registres sources.

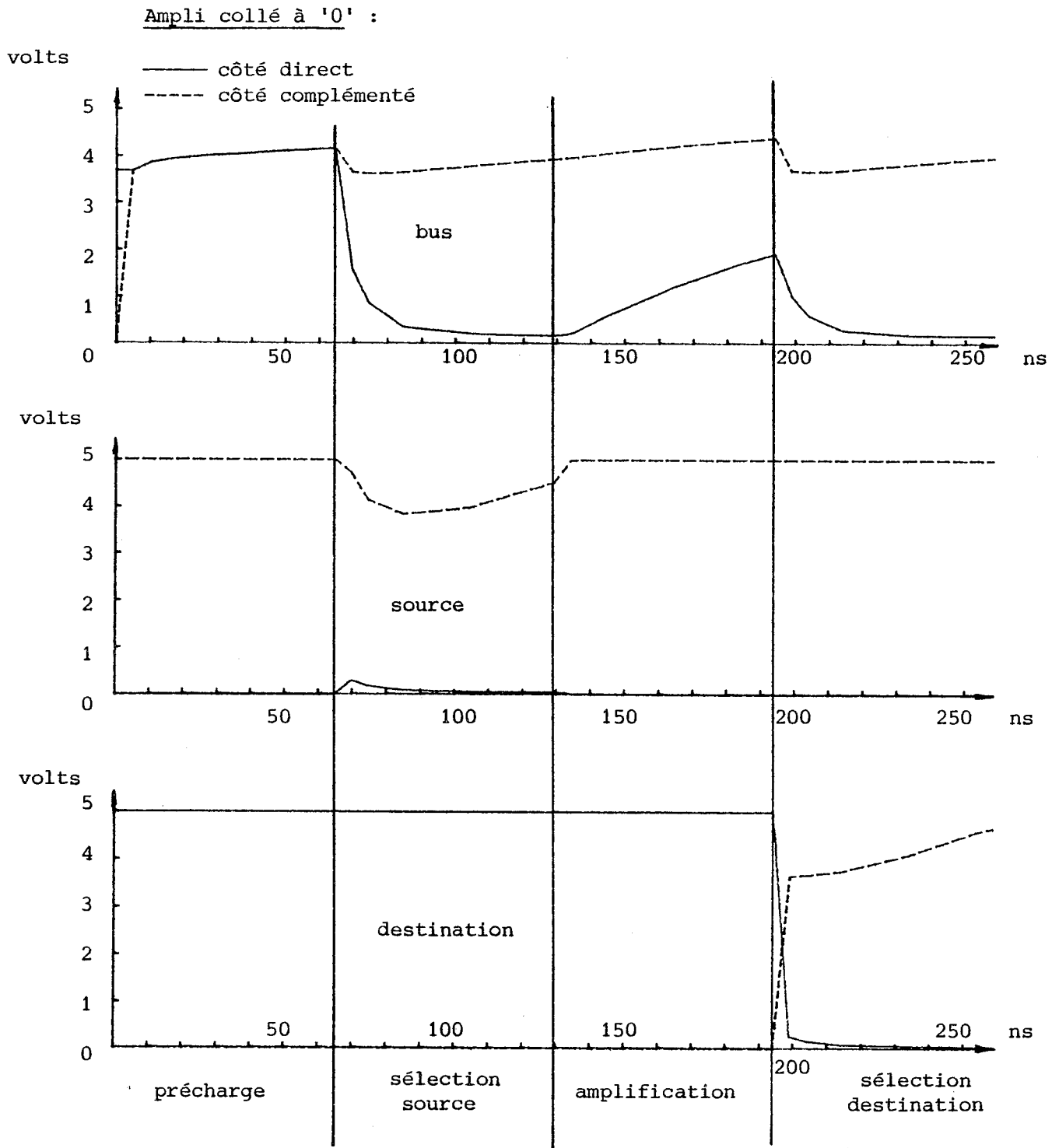


Figure 18. - Chronogramme du bus en cas d'amplification inexistante.

Ampli collé à '1' : (transfert cycle précédent)

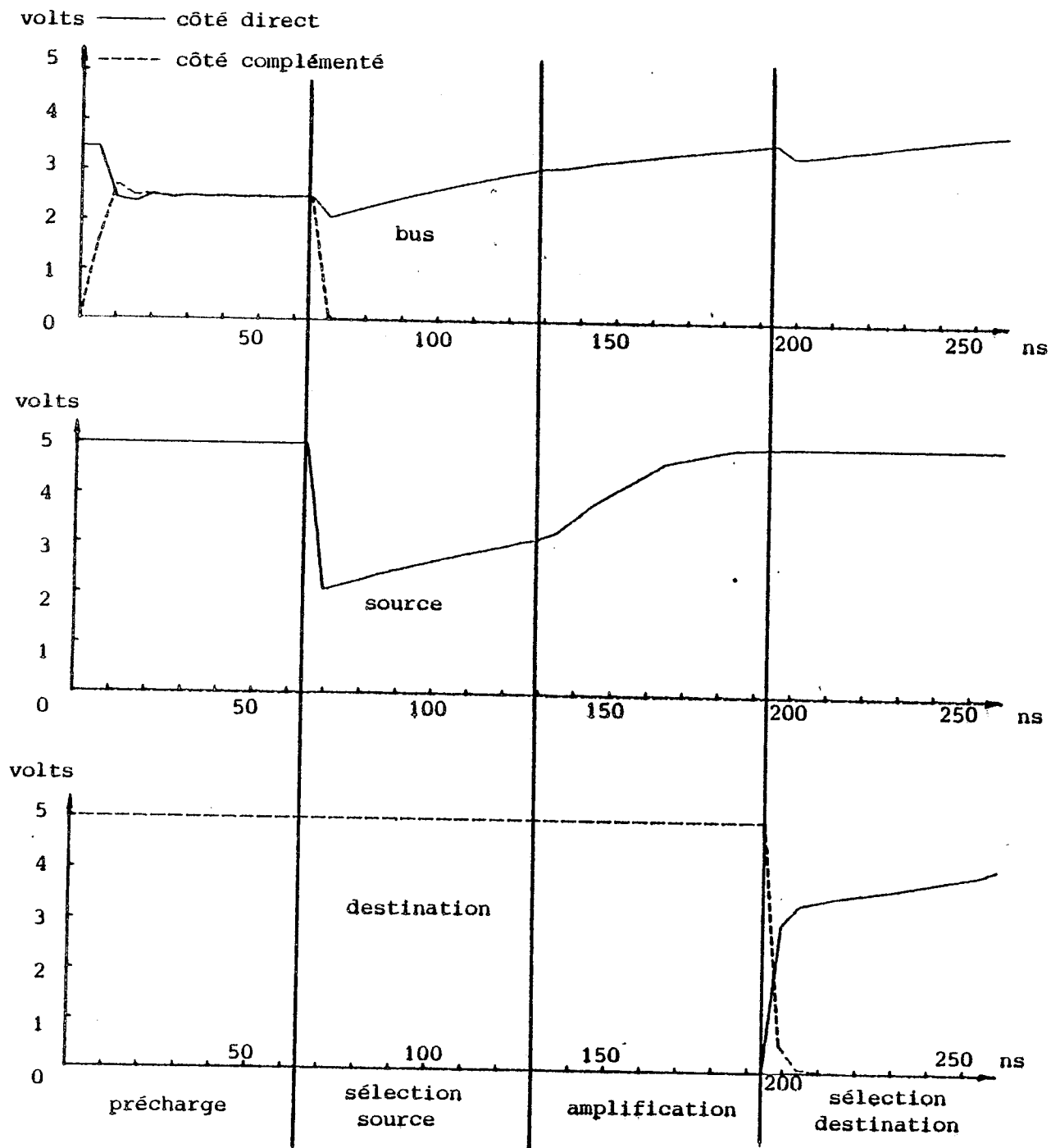


Figure 19. - Chronogramme du bus en cas d'amplification permanente, avec transfert au cycle précédent.

Ampli collé à '1' : (pas transfert cycle précédent).

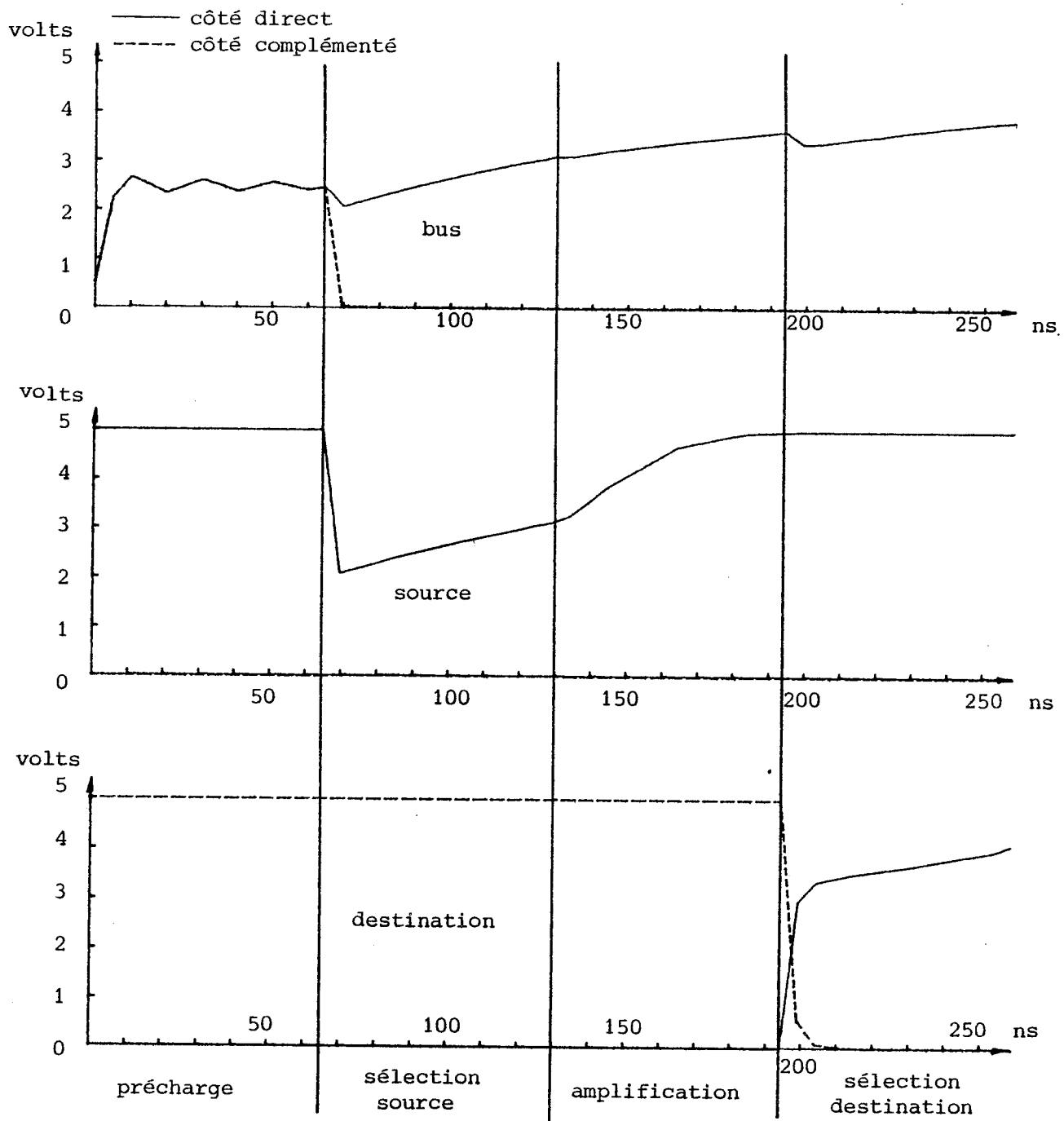


Figure 20. - Chronogramme du bus en cas d'amplification permanente, sans transfert au cycle précédent.

a) - Commande de switch collée à zéro :

Dans ce cas, les bus restent séparés, le registre source est connecté sur un bus, le ou au moins un registre destination se trouve connecté sur un autre bus où aucune source n'est connectée. Ce cas se ramène donc au cas où aucune source n'est connectée au bus (cf. § IV.2.1.) les registres destinations connectées sur la même portion de bus que le registre source, reçoivent correctement sa valeur.

b) - Commande de switch collée à un :

Dans ce cas, si un des 2 bus connectés ne participe à aucun transfert, le fonctionnement reste correct.

Par contre si les 2 bus connectés sont utilisés pour effectuer des transferts, ou se trouve avec 2 sources connectées ensemble. Ce cas se ramène donc, vis-à-vis des registres sources et destinations, au cas de connexions multiples de registres sources (cf. § IV.2.2.).

IV.1.4. Pannes de Fool-it :

Comme pour les pannes de switches, on ne s'intéresse ici qu'au fonctionnement en panne c'est-à-dire commande de fool-it collée à 'zéro' lorsqu'elle doit être active (opération sur octet) ou commande collée à 'un' lorsqu'elle doit être inactive (opération sur les mots).

a) - Commande de fool-it collée à zéro :

Dans ce cas, le 'fool-it' se comporte comme un amplificateur normal. La conséquence de cette panne est que la destination reçoit les 16 bits complets en provenance de la source.

Il en résulte donc que :

- . Le registre source n'est pas modifié.
- . Le bus prend la valeur du registre source.
- . Le registre destination reçoit les 16 bits (au lieu des 8 bits poids faibles uniquement) du bus (valeur du registre source).

Le fonctionnement reste le même si plusieurs bus sont connectés.

b) - Commande de fool-it collée à 'un' :

Deux cas sont ici à distinguer, selon le fait que le transfert s'effectue uniquement sur le bus ABD ou si un, ou les deux autres bus, lui sont connectés.

- Si le transfert s'effectue uniquement sur le bus ABD, seuls les 8 bits poids faibles du registre source sont réécrits dans le registre destination. Les 8 bits poids forts du registre destination ne sont pas modifiés.

Il en résulte donc :

- . Le registre source n'est pas modifié.
 - . Les 8 bits poids faibles du bus prennent la valeur du registre source.
 - . Les 8 bits poids forts du bus restent préchargés.
 - . Les 8 bits poids faibles du registre destination prennent la valeur du bus (valeur du registre source).
 - . Les 8 bits poids forts du registre destination ne sont pas modifiés.
-
- Si le transfert s'effectue alors que le bus ABD est connecté à un autre bus, les 8 bits poids faibles du registre source seront transférés normalement, mais les 8 bits poids forts du bus verront à la fois précharge et amplification simultanément donc on se ramène

au cas de fonctionnement avec précharge permanente, pendant le cycle d'écriture du registre destination.

Il en résulte donc que :

- . Le registre source n'est pas modifié.
- . Les 8 bits poids faibles du bus prennent la valeur du registre source (8 bits poids faibles).
- . Les 8 bits poids forts du bus restent préchargés (2.45 V).
- . Les 8 bits poids faibles du registre destination prennent la valeur des 8 bits poids faibles du bus (valeur du registre source).
- . Les 8 bits poids forts du registre destination conservent leur valeur ou se positionnent à la valeur de la majorité de ces registres selon le nombre respectif de points à '1' et de points à '0'.
(cf. § IV.3.1.2. b et c).

IV.2. Pannes de sélection de registres :

Au niveau de la sélection des registres on peut différencier 3 types de pannes, selon que le registre ou l'opérateur est utilisé comme source, destination, ou n'est pas du tout concerné dans le transfert.

En effet, si la commande de sélection du registre source est collée à zéro, aucune source n'est sélectionnée. Si la commande de sélection du registre source est collée à un, pendant le transfert considéré, aucune erreur ne se produit puisque la connexion correspond à l'utilisation correcte.

Si la commande de sélection du registre destination est collée à zéro, il n'y a donc pas de destination au transfert ; le registre destination n'est donc pas modifié. Si la commande de sélection du registre destination est collée à un, aucune erreur ne se produit puisque la connexion en destination correspond à l'utilisation correcte.

Si la commande de sélection d'un registre non concerné dans le transfert est collée à zéro, il n'intervient donc pas dans le transfert, on est dans le cas de fonctionnement correct. Si cette commande est collée à un, il va perturber le fonctionnement.

On se ramène ainsi, aux deux cas suivants de fonctionnement erroné : il n'y a pas de source au transfert, ou il y en a plusieurs. En effet, s'il n'y en a qu'une, quelle qu'elle soit, tous les registres destinations prendront sa valeur.

Les deux cas vont être passés en revue l'un après l'autre, afin de déterminer les conséquences des pannes de sélection.

A part le cas correct, présenté à la figure 11 où une source est sélectionnée, on peut distinguer 2 cas, soit aucune source n'est sélectionnée soit plusieurs le sont.

IV.2.1. Aucune source n'est sélectionnée :

Comme le montre la figure 21, les bus restent préchargés pendant T1 (4.1 V) puis les amplis tirent les lignes au seuil du MOS (0.82 V) pendant T2. Ensuite, on peut distinguer 3 cas dépendant du nombre de destinations connectées au bus.

a) - Une seule destination :

Cette destination est légèrement modifiée (cf. figure 21), mais elle permet de déséquilibrer le bus qui prend alors sa valeur. Il en résulte donc que :

. Le registre destination n'est pas modifié.

b) - Plusieurs destinations, même nombre de '1' que de '0' :

Dans ce cas, comme le montre la figure 22 pour 2 destinations dont les valeurs

sont le complément l'une de l'autre, les points se stabilisent à 0.803 V.
Une fois déconnectés ils se mettent dans un état métastable 0.759 V.

Il en résulte donc que :

. Les registres destinations sont dans un état métastable (0.759 V).

c) - Plusieurs destinations, nombre de '1' différent du nombre de '0' :

Dans ce cas, comme le montre la figure 23, c'est la majorité qui l'emporte
et qui s'impose aux autres points.

Il en résulte donc que :

. Les registres destinations prennent la valeur de la majorité.

Aucune source connectée :

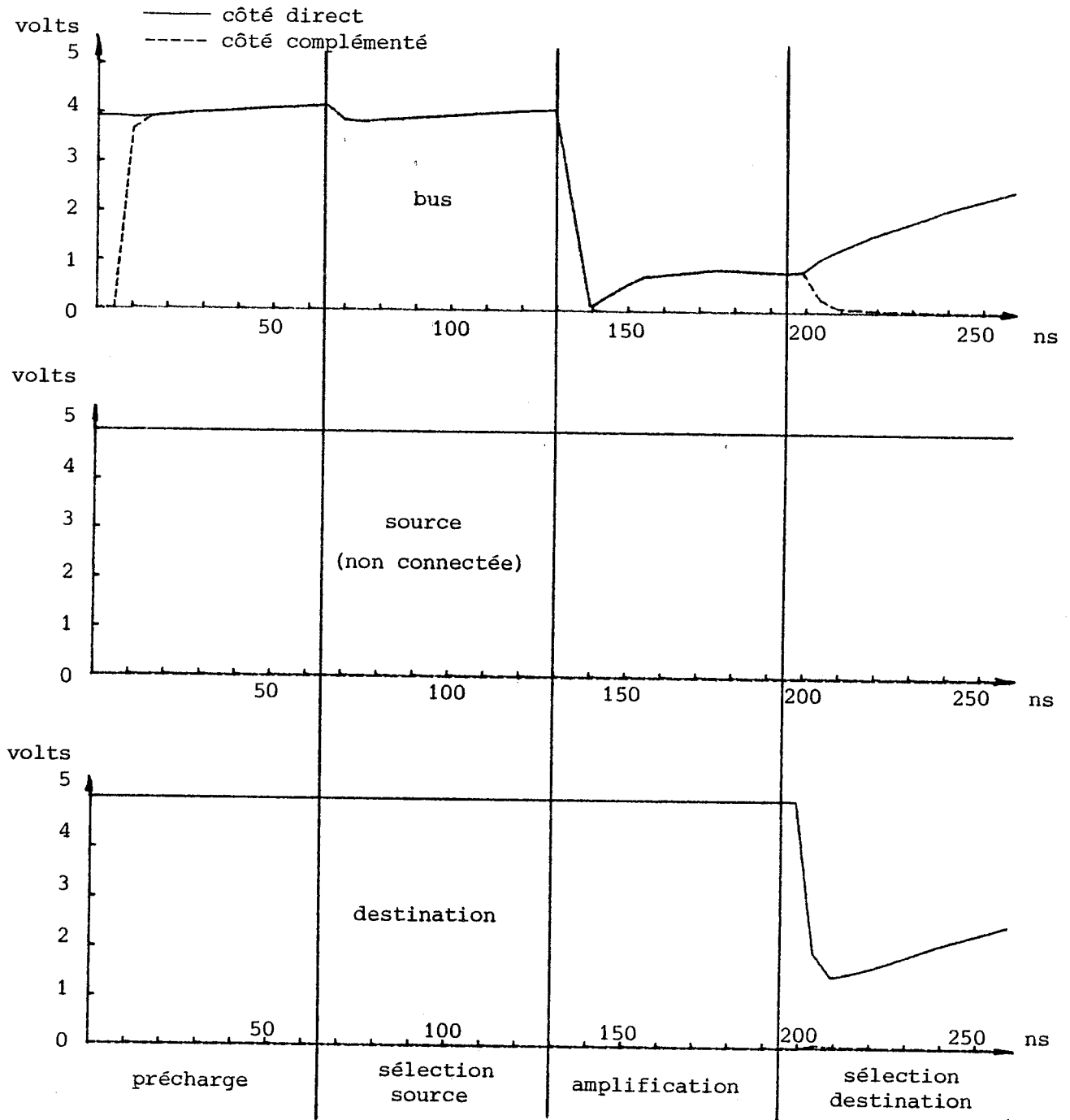


Figure 21. - Chronogramme du bus en cas de non connexion du registre source, 1 destination.

Aucune source connectée :

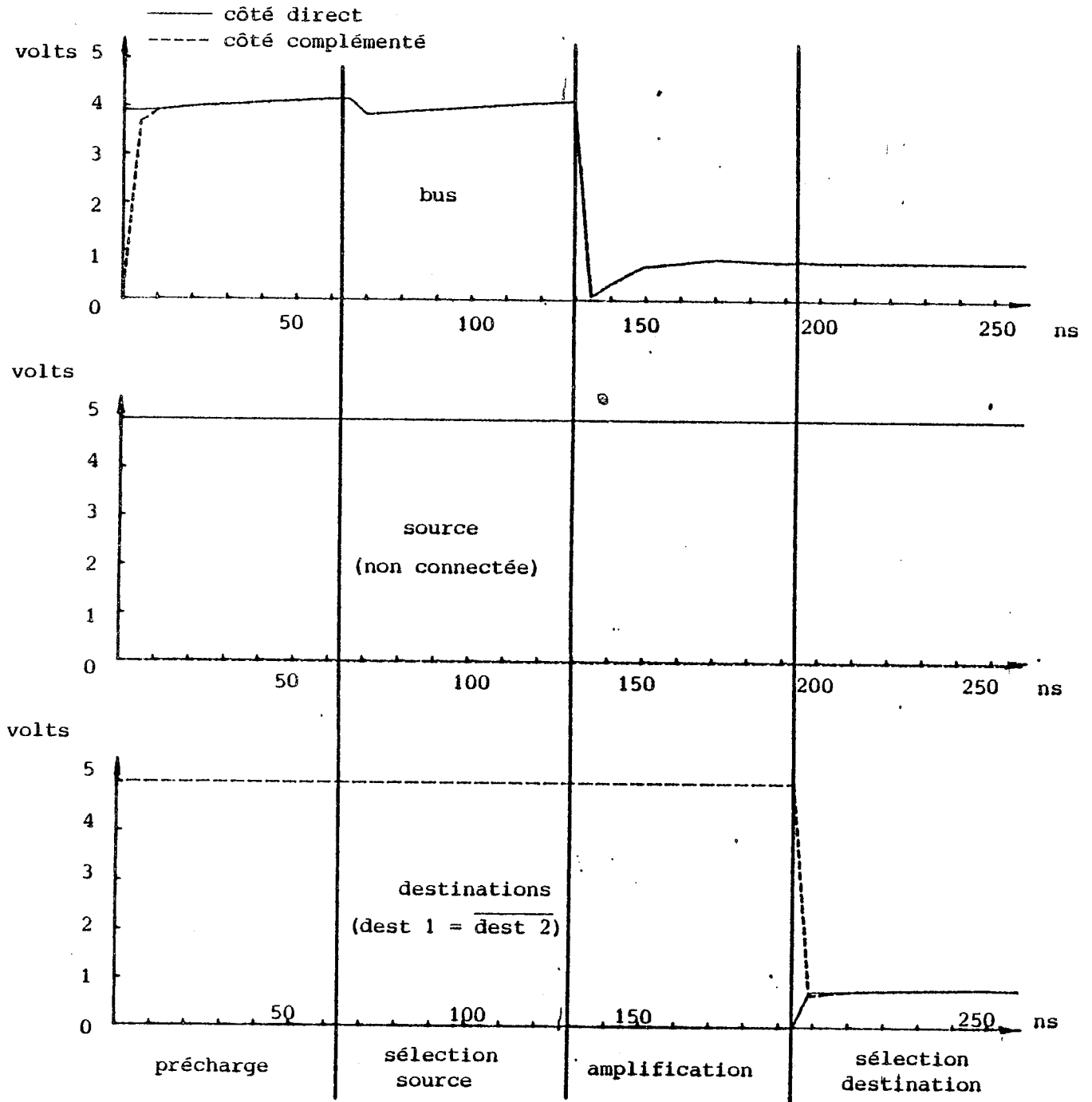


Figure 22. - Chronogramme du bus en cas de non connexion du registre source, 2 destinations complémentaires.

IV.2.2. Plusieurs sources sont sélectionnées :

Lorsque plusieurs sources sont sélectionnées simultanément, le nombre de point à '1' et à '0' permet de diviser en deux cas de figure :

a) - Le nombre de point à '1' est égal au nombre de point à '0' :

Dans ce cas, comme le montre la figure 24 sur T4, T1, T2 où 2 points complètent l'un de l'autre sont sélectionnés, le bus et les points se trouvent dans un état stabilisé à environ 1 V pour le bus et 0.95 V pour les points mémoire. Ceux-ci une fois désélectionnés se mettent dans un état métastable (0.759 V). Le bus lui rejoint la valeur de 0.84 V ; on retombe alors pour le(s) registre(s) destination(s) dans le cas où aucune source n'a été sélectionnée.

Il en résulte donc :

- . Les registres sources sont dans un état métastable (0.759 V).
- . Le bus se stabilise à 1. V sur T1 puis 0.82 V sur T2.
- . Le(s) registre(s) destination(s) deviennent selon le cas
 - Une seule destination n'est pas modifiée.
 - autant de '1' que de '0', les destinations sont dans un état métastable (0.759 V).
 - nombre de '1' différent du nombre de '0', la majorité s'impose à toutes les destinations.

b) - Le nombre de points à '1' est différent du nombre de point à '0' :

Dans ce cas, comme le montre la figure 25, sur T4, T1, la majorité s'impose sur le bus ainsi qu'à tous les registres connectés en source. On retombe ensuite, pour la (les) destination(s) dans le cas de fonctionnement correct car le bus a une valeur logique correcte.

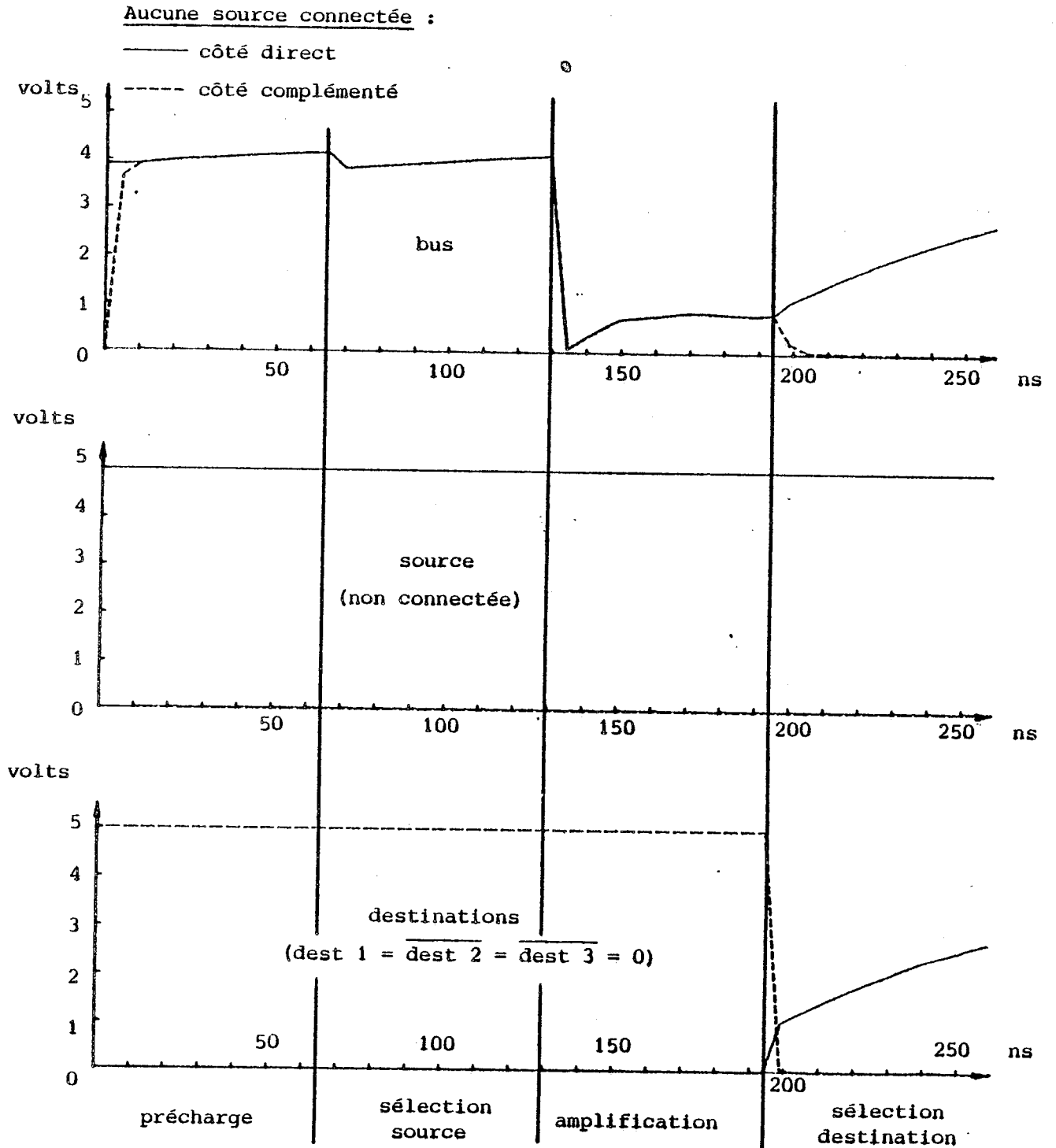


Figure 23. - Chronogramme du bus en cas de non connexion du registre source, 3 destinations.

Plusieurs sources connectées :

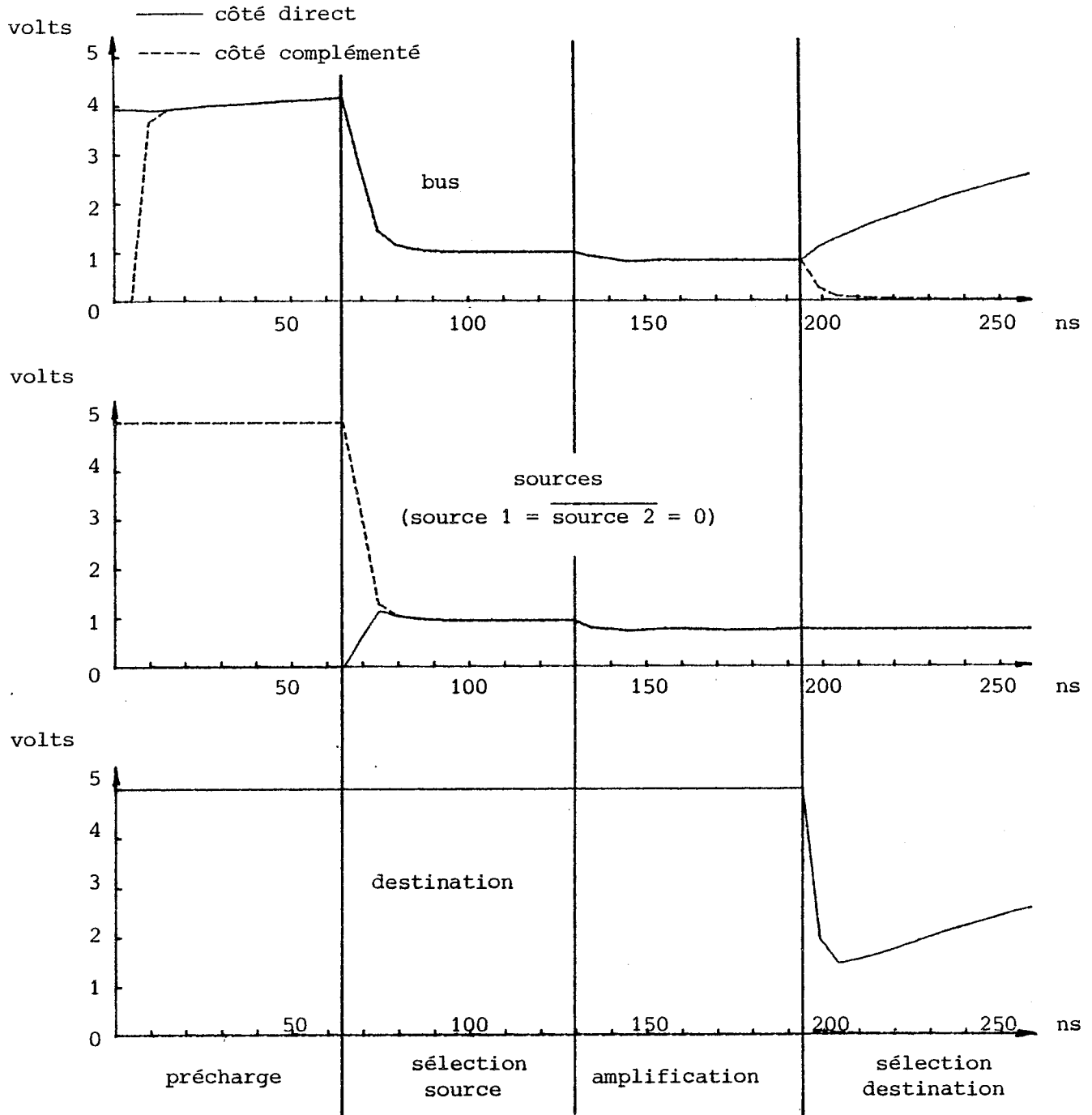


Figure 24. - Chronogramme du bus lorsque 2 sources complémentaires sont connectées simultanément.

Plusieurs sources connectées :

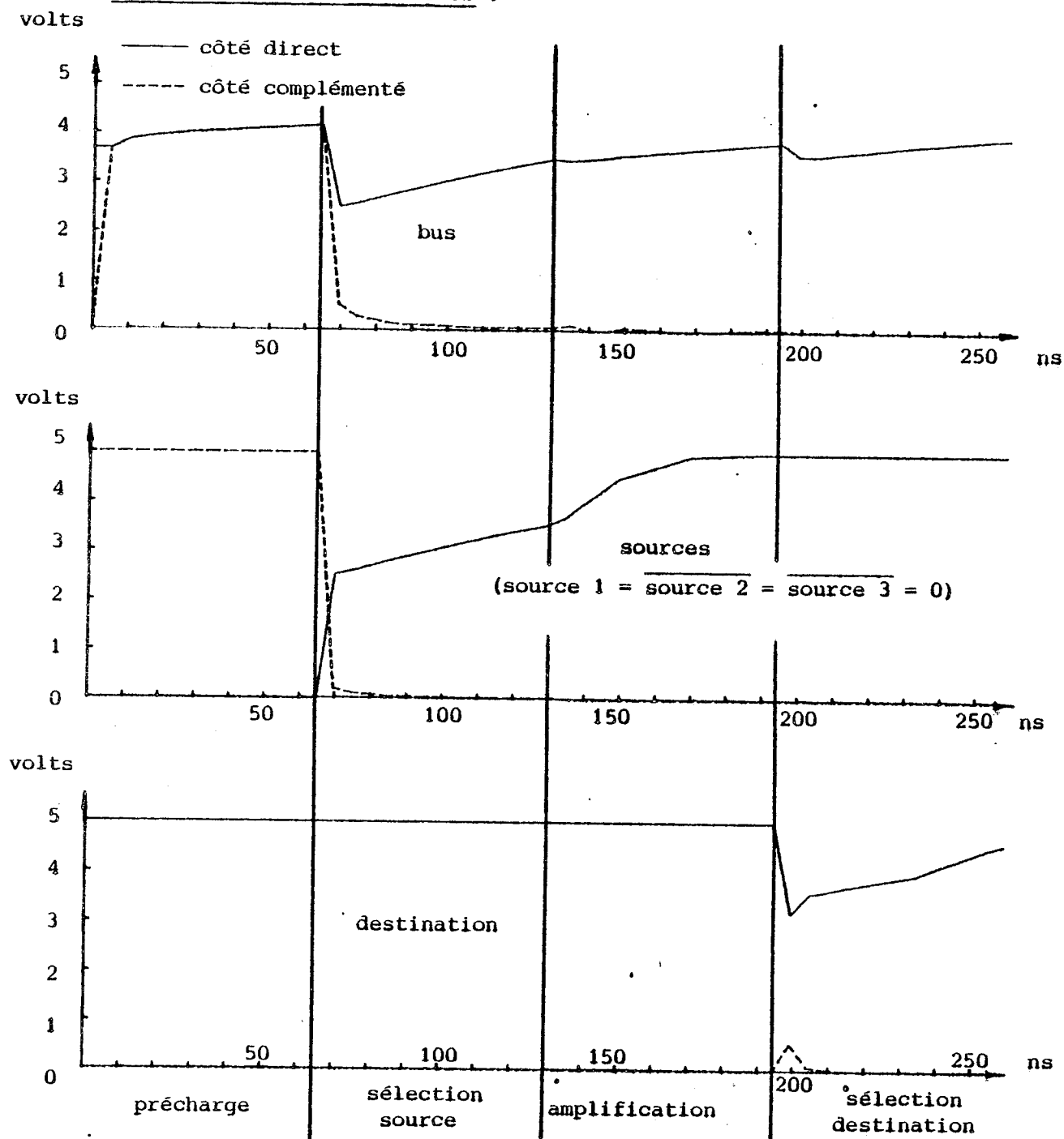


Figure 25. - Chronogramme du bus lorsque 3 sources sont connectées simultanément.

Il en résulte donc :

- . Les registres sources prennent la valeur de la majorité.
- . Le bus prend la valeur de la majorité des registres sources.
- . Le(s) registre(s) destination(s) prend (prennent) la valeur du bus (majorité des sources).

IV.3. Hypothèses retenues pour les pannes de transfert :

Le tableau VIII regroupe les conséquences des pannes de la gestion des transferts. La première colonne précise les pannes considérées, les colonnes suivantes donnent les valeurs respectivement des registres sources, du bus, des registres destinations ; enfin, les dernières colonnes correspondent aux connexions particulières de certaines destinations.

Ainsi le PREN est connecté sur T1, les AU et ALU sont connectées sur T1 T2, de plus certains registres ne sont connectés que sur un des bus, comme présentés dans le test analytique de la partie opérative du MC 68000 (cf, Chap. VI) : AOB reçoit le complément de \overline{AB} et de \overline{DB} , DOB reçoit le complément de \overline{AB} et de \overline{DB} ALUB reçoit \overline{AB} ou \overline{DB} etc...

Dans la colonne des pannes considérées, NB1 (registre) représente le nombre de registres qui sont à '1' et NB0 (registre) le nombre de registres qui sont à '0'. Lorsque $NB1 \neq NB0$, majorité dans les colonnes 2 à 9, représente la valeur des points correspondant à la majorité des registres connectés.

Remarque :

En ce qui concerne l'état métastable du point mémoire, ce résultat, tout à fait théorique est basé sur la symétrie exacte de chaque transistor, les tailles sont celles que 'devraient' avoir les transistors. Dans la réalité un

Pannes considérées	Source	Bus	Dest	PREN	AU	ALU	AOB DOB	ALU B
<u>Pannes de gestion du bus</u>								
• <u>Précharge collée à zéro</u>	NM SP	S SP	S SP	S SP	S SP	S SP	S SP	S SP
• Pas de transfert au cycle précédent								
• Transfert au cycle précédent								
• <u>Précharge collée à un</u>								
- NB1 (dest) = NBO (dest)	NM	P	NM	UN	Z	UN	Z	UN
- NB1 (dest) ≠ NBO (dest)	NM	P	MD	UN	Z	UN	Z	UN
• <u>Ampli collé à zéro</u>	NM	S	S	S	S	S	S	S
• <u>Ampli collé à un</u>	NM	S	S	S	S	S	S	S
• <u>Switch collé à zéro</u>								
- côté bus source	NM	S	S	S	S	S	S	S
- côté bus destination								
1 seule destination								
- NB1 (dest) = NBO (dest)	-	P	NM	UN	UN	Z	UN	Z
- NB1 (dest) ≠ NBO (dest)	-	P	M	UN	UN	Z	UN	Z
- NB1 (source) ≠ NBO (source)	-	P	MD	UN	UN	Z	MD	MD
• <u>Switch collé à un</u>								
- NB1 (source) = NBO (source)	M	SM	M	Z	UN	Z	UN	Z
- NB1 (dest) = NBO (dest)	M	SM	MD	Z	UN	Z	MD	MD
- NB1 (dest) ≠ NBO (dest)	MS	MS	MS	MS	MS	MS	MS	MS
- NB1 (source) ≠ NBO (source)								
• <u>Fool-it collé à zéro</u>								
- 8 bits poids faibles	NM	S	S	S	S	S	S	S
- 8 bits poids forts	NM	S	S	S	S	S	S	S

Tableau VIII. Hypothèses retenues pour les pannes de gestion des transferts du MC 68000.
(début)

<p>• <u>Fool-it collé à un</u> - ABD Seul 8 bits poids faibles 8 bits poids forts</p>	NM NM	S P	S UN	S Z	S UN	S Z	S UN	S UN
<p>- ABD + ABL ou ABD + ABL + ABH 8 bits poids faibles 8 bits poids forts : NB1 (Dest) = NB0 (dest) NB1 (Dest) ≠ NB0 (dest)</p>	NM NM NM	S P P	S UN UN	S Z Z	S UN UN	S Z Z	S UN UN	S UN UN
<u>Pannes de sélection des registres</u>								
<p>• <u>Aucune source</u> - Une seule destination - Plusieurs destinations : NB1 (dest) = NB0 (dest) NB1 (dest) ≠ NB0 (dest)</p>	- - -	P P P	UN UN UN	Z UN UN	UN Z Z	Z UN MD	UN Z MD	UN Z MD
<p>• <u>Plusieurs Sources</u> - NB1 (source) = NB0 (source) NB1 (dest) = NB0 (dest) NB1 (dest) ≠ NB0 (dest)</p>	M M	SM SM	Z Z	UN UN	Z Z	UN MD	UN MD	Z MD
<p>- NB1 (source) ≠ NB0 (source)</p>	MS	MS	MS	MS	MS	MS	MS	MS
<p>MS = Source P = Préchargé M = Métastable Z = "Zéro" MD = Majorité (destination) MS = majorité (source) NM = Non modifié SP = source précédente SM = Seuil du MOS UN = "UN"</p>								

Tableau VIII. Hypothèses retenues pour les pannes de gestion des transferts du MC 68000.

(fin)

facteur d'erreur dû à la fabrication est toléré, par exemple : $0,1 \mu\text{m}$ sur les longueurs et sur les largeurs. Un léger décalage des masques, à la fabrication, rendra alors les MOS légèrement désymétriques et le point reprendra une valeur stable. Ainsi les hypothèses correspondantes seront que le point bascule à 'un' ou à 'zéro' : les 2 valeurs sont possibles et sont à prendre en compte.

La restriction de la multiplicité de pannes est la suivante :

- . D'une part, les pannes de gestion du bus (ampli, précharge, switch etc...) en plus des pannes de sélection de registres (aucune source, plusieurs sources, etc...) sont considérées en dehors des hypothèses.
- . D'autre part, un état qui correspond aussi bien à l'état métastable qu'à la majorité telle qu'obtenue en simulation est définie ci-dessous.

Définition :

On appellera mélange de registres la valeur obtenue lorsque plusieurs (de deux à tous les registres connectables) registres sont connectés simultanément via le bus. Ainsi on pourra obtenir des mélanges de registres connectés en source, notés mélange (sources) et des mélanges de registres destinations, notés mélange (destinations).

La valeur logique d'un mélange est définie comme suit :

- Si ET (R_i, R_j) qualifie le ET logique entre R_i et R_j ,
- Si OU (R_i, R_j) qualifie le OU logique entre R_i et R_j ,
- Si $R_i [k]$ qualifie le k ème bit du registre R_i ,
- Si maj ($R_i[k]$) représente la valeur du k ème bit de la majorité des registres R_i considérés.

Alors, mélange $(R_i; i = 1 \text{ à } n)$ est donné par :

Pour $k : = 0 \text{ à } 15$

Si $\sum_{i=1,n} (R_i [k] = '1') > \sum_{i=1,n} (R_i [k] = '0')$

Alors mélange $(R_i [k]; i = 1, n) := '1'$ (* = maj_{i=1,n} (Ri [k])*)

Sinon si $\sum_{i=1,n} (R_i [k] = '1') < \sum_{i=1,n} (R_i [k] = '0')$

Alors mélange $(R_i [k]; i=1,n) := '0'$ (* = maj_{i=1,n} (Ri[k])*)

Sinon (* $\sum_{i=1,n} (R_i [k] = '1') = \sum_{i=1,n} (R_i [k] = '0')$)

Selon

fabrication :

Soit mélange $(R_i [k]; i=1,n) := '1'$ (* = OU (Ri[k])*)
i=1,n

Soit mélange $(R_i [k]; i=1,n) := '0'$ (* = ET (Ri[k])*)
i=1,n

Ce qui peut s'exprimer par :

Pour $k : \text{à } \text{à } 15$

Si $\sum_{i=1,n} (R_i [k] = '1') > \sum_{i=1,n} (R_i [k] = '0')$

Alors mélange $(R_i [k]; i=1,n) := \text{OU}(R_i [k])$
i=1,n

Sinon Si $\sum_{i=1,n} (R_i [k] = '1') < \sum_{i=1,n} (R_i [k] = '0')$

Alors mélange $(R_i [k]; i=1,n) := \text{ET} (R_i [k])$
i=1,n

Sinon $(\sum_{i=1,n} (Ri[k] = '1') = \sum_{i=1,n} (Ri[k] = '0'))^*$

Selon fabrication

Soit mélange $(Ri [k]; i=1,n) :=$ OU $(Ri[k])$
 $i=1,n$

Soit mélange $(Ri[k]; i=1,n) :=$ ET $(Ri [k])$
 $i=1,n$

Soit R un registre devant vérifier $R \neq$ mélange $(Ri; i=1,n)$. Deux cas sont à distinguer selon le fait que R appartient ou non à l'ensemble des Ri :

a) $R \notin Ri ; i = 1 \text{ à } n :$

$R \neq$ mélange $(Ri ; i = 1 \text{ à } n) \Leftrightarrow$

(1) $\forall i \neq j \exists k ; R[k] \neq Ri[k]. Rj[k]$

(2) $\forall i \neq j \exists k' ; R[k'] \neq Ri[k'] + Rj[k']$

(3) $\forall n > 2 \exists k'' ; R[k''] \neq \text{maj} (Ri; i=1,n)$

Soit :

(1) $\Leftrightarrow \forall i \exists k ; \left\{ \begin{array}{l} R[k] = 0 \text{ et } Ri[k] = 1 \\ \text{ou } R[k] = 1 \text{ et } Ri[k] = 0 \end{array} \right.$

(2) $\Leftrightarrow \forall i \exists k' ; \left\{ \begin{array}{l} R[k'] = 0 \text{ et } Ri[k'] = 1 \\ \text{ou } R[k'] = 1 \text{ et } Ri[k'] = 0 \end{array} \right.$

(3) $\Leftrightarrow \forall i \exists k'' ; \left\{ \begin{array}{l} R[k''] = 0 \text{ et } \left\{ \begin{array}{l} Ri[k''] = 1 \\ \text{ou } \exists j! Rj[k''] = 0 \text{ et } Ri[k''] = 1 \forall i \neq j \end{array} \right. \\ \text{ou } R[k''] = 1 \text{ et } \left\{ \begin{array}{l} Ri[k''] = 0 \\ \text{ou } \exists j! ; Rj[k''] = 1 \text{ et } Ri[k''] = 0 \forall i \neq j. \end{array} \right. \end{array} \right.$

b) $R \notin Ri ; i = 1, \text{ à } n :$

$R \neq$ mélange $(Ri ; i = 1 \text{ à } n) \Leftrightarrow$

(1') $\left\{ \begin{array}{l} \forall i \exists k ; R[k] \neq R[k]. Ri[k] \\ \text{et (1)} \end{array} \right.$

$$(2') \left\{ \begin{array}{l} \forall i \exists k ; R[k'] \neq R[k']. Ri[k'] \\ \text{et (2)} \end{array} \right.$$

$$(3') \left\{ \begin{array}{l} \forall n > 1 \exists k'' ; R[k''] \neq \text{maj} (R[k''], Ri ; i=1,n) \\ \text{et (3)} \end{array} \right.$$

$$(1') \Leftrightarrow \forall i \exists k ; R[k] = 1 \text{ et } Ri[k] = 0$$

$$(2') \Leftrightarrow \forall i \exists k ; R[k'] = 0 \text{ et } Ri[k'] = 1$$

$$(3') \Leftrightarrow \forall i \exists k'' ; R[k''] = 1 \text{ et } Ri[k''] = 0 \\ \text{ou } R[k''] = 0 \text{ et } Ri[k''] = 1.$$

Une condition suffisante pour ces 2 cas (1), (2), (3) et (1'), (2'), (3')

$\exists k, k' ; \forall i \left\{ \begin{array}{l} R[k] = 1 \text{ et } Ri[k] = 0 \\ \text{et } R[k'] = 0 \text{ et } Ri[k'] = 1. \end{array} \right.$

IV.4. Erreurs possibles lors d'un transfert élémentaire :

On appelle transfert élémentaire, le transfert d'un registre vers un autre ou vers plusieurs registres à travers un bus. On rappelle que les transferts vers opérateurs sont particuliers et seront étudiés plus loin.

D'après le tableau VIII du § IV.3., il est possible de tirer les cas de manifestation de la panne et de son influence sur les registres connectés en source ou en destination.

A la suite d'un transfert, un registre connecté en source sur le bus peut se trouver dans l'un des états suivants :

- . Source : = non modifiée
- . Source : = source du cycle précédent
- . Source : = mélange (sources).

où mélange est la valeur définie au paragraphe précédent.

A la suite d'un transfert, un registre, connecté en destination du bus, peut se trouver dans l'un des états suivants :

- . Destination : = non modifiée.
- . Destination : = mélange (sources).
- . Destination : = mélange (destinations).
- . Destination : = 1 source quelconque.
- . Destination : = Source du cycle précédent.
- . Destination : = Concaténation des 8 bits poids faibles de la source et de ses propres 8 bits poids forts.
- . Destination : = Concaténation des 8 bits poids faibles de la source et des 8 bits poids forts du mélange (destinations).

S'assurer qu'aucune panne n'a eu lieu pendant un transfert élémentaire, c'est :

- . Vérifier qu'aucun des registres connectables en source sur le bus a été perturbé.
- . Vérifier qu'aucun des registres connectables en destination sur le bus à part la (ou les) destination(s) correcte(s), n'a été perturbé.
- . Vérifier que le ou les registres correspondant à la bonne destination du transfert, ont bien été affectés à la valeur de l'origine correcte du transfert.

a) - Non perturbation des registres connectables en source :

Vérifier qu'aucun des registres connectables en source sur le bus n'a été perturbé par le transfert, c'est vérifier que la valeur finale est différente de toutes les valeurs que peut prendre un registre connecté en source dans

tous les cas de pannes de transfert.

Si RS désigne l'ensemble des registres connectables en source, alors les conditions de non perturbation de ces registres s'expriment par :

- (a) $R_{Si} \neq$ origine du cycle précédent $\forall i$
- (b) $R_{Si} \neq$ mélange (sources) $\forall i$

Comme aucun des registres connectables en source ne doit être modifié, et ceci y compris l'origine correcte du transfert, les conditions (a) et (b) doivent être vérifiées à la fois avant et après le transfert.

- b) - Non perturbation des registres connectables en destination ne devant pas intervenir dans le transfert :

Vérifier qu'aucun des registres, connectables en destination sur le bus et ne devant pas intervenir dans le transfert, n'a été perturbé par le transfert, c'est vérifier que la valeur finale est différente de toutes les valeurs que peut prendre un registre connecté en destination dans tous les cas de pannes de transfert.

Si RD désigne l'ensemble des registres connectables en destination et ne devant pas intervenir dans le transfert, alors les conditions de non perturbation de ces registres s'expriment par :

- (c) $R_{Di} \neq$ 1 source quelconque $\forall i$
- (d) $R_{Di} \neq$ mélange (sources) $\forall i$
- (e) $R_{Di} \neq$ mélange (destinations) $\forall i$
- (f) $R_{Di} \neq$ concatenation des 8 bits poids faibles de la source et de ses propres 8 bits poids forts. $\forall i$
- (g) $R_{Di} \neq$ concatenation des 8 bits poids faibles de la source et des 8 bits poids forts du mélange (destinations) $\forall i$.

Comme aucun des registres connectables en destination mais ne participant pas au transfert, ne doit être modifié, les conditions (c) à (g) doivent être vérifiées à la fois avant et après le transfert.

c) - Affectation de la (des) destination(s) correcte(s) :

Vérifier que le (ou les) registre(s) correspondant(s) à la bonne destination du transfert, ont bien été affecté(s) à la valeur de l'origine correcte du transfert, c'est vérifier que la valeur finale est différente de toutes les valeurs que peuvent prendre ces registres en cas de pannes durant le transfert.

Si RDC désigne l'ensemble des registres correspondants à la destination correcte du transfert, alors les conditions de non perturbation de ces registres s'expriment par :

- (h) $RDC_i \neq$ non modifié $\forall i$
- (i) $RDC_i \neq$ 1 source incorrecte $\forall i$
- (j) $RDC_i \neq$ mélange (sources) $\forall i$
- (k) $RDC_i \neq$ mélange (destinations) $\forall i$
- (l) $RDC_i \neq$ concaténation des 8 bits poids faibles de la source et de ses propres 8 bits poids forts $\forall i$.
- (m) $RDC_i \neq$ concaténation des 8 bits poids faibles de la source et des 8 bits poids forts du mélange (destinations) $\forall i$.

Comme chacun des registres correspondant à la destination correcte du transfert doit recevoir la valeur du registre origine, les conditions (h) à (m) se reportent donc ainsi sur la valeur de l'origine avant le transfert, soit ORG l'origine correcte.

- (h') $ORG \neq$ RDCi initial $\forall i$.
- (i') $ORG \neq$ RSi - {ORG} $\forall i$.
- (j') $ORG \neq$ mélange (sources).

- (k') ORG ≠ mélange (destinations).
- (l') ORG ≠ concaténation de ses propres 8 bits poids faibles et des 8 bits poids forts de RDCi Vi.
- (m') ORG ≠ concaténation de ses propres 8 bits poids faibles et des 8 bits poids forts du mélange (destinations).

Le tableau IX résume les conditions nécessaires et suffisantes pour la vérification de pannes durant un transfert élémentaire. ORG désigne l'origine correcte du transfert, RS tous les registres connectables en source sur le bus, RD tous les registres connectables en destination sur le bus et ne devant pas intervenir durant le transfert, RDC le ou les registres correspondant(s) à la destination correcte du transfert.

ORG	RS	RD	RDC
a	a	c	h
b	b	d	i
h'		e	j
i'		f	k
j' <=> b		g	l
k'			m
l'			
m'			

Tableau IX. Conditions nécessaires et suffisantes à la vérification des pannes durant un transfert élémentaire.

Il va de soit que si un registre peut être à la fois connecté en source et en destination il devra vérifier à la fois les conditions correspondant aux registres connectables en source et les conditions correspondant aux registres

connectables en destination. Si ces conditions sont incompatibles, alors plusieurs étapes de test seront nécessaires pour ces registres.

Une condition suffisante est donnée par :

$$(1) \left\{ \begin{array}{l} \exists k, k' ; \forall i \left\{ \begin{array}{l} \text{ORG}[k] = 1 \text{ et} \\ \text{et } \text{ORG}[k'] = 0 \text{ et} \end{array} \right. \left\{ \begin{array}{l} \text{RSi}[k] = 0 \text{ (sauf ORG)} \\ \text{RDi}[k] = 0 \\ \text{RDCi}[k] = 0 \\ \text{RSi}[k'] = 1 \text{ (sauf ORG)} \\ \text{RDi}[k'] = 1 \\ \text{RDCi}[k'] = 1. \end{array} \right. \\ \text{et } \left\{ \begin{array}{l} k \in \{0,7\}, k' \in \{8, 15\} \\ \text{ou } k' \in \{0,7\}, k \in \{8, 15\} \end{array} \right\} \Leftrightarrow k \text{ et } k' \text{ n'appartiennent pas au même octet.} \end{array} \right.$$

Les vérifiants suivants sont à effectuer après le transfert :

$\exists k, k'$ les mêmes que définis avant le transfert ;

$$(1') \left\{ \begin{array}{l} \forall i \left\{ \begin{array}{l} \text{ORG}[k] = 1 \\ \text{et } \text{RDCi}[k] = 1 \end{array} \right. \text{ et } \left\{ \begin{array}{l} \text{RSi}[k] = 0 \text{ (sauf ORG)} \\ \text{RDi}[k] = 0 \end{array} \right. \\ \text{et } \left\{ \begin{array}{l} \text{ORG}[k'] = 0 \\ \text{et } \text{RDCi}[k'] = 0 \end{array} \right. \text{ et } \left\{ \begin{array}{l} \text{RSi}[k'] = 1 \text{ (sauf ORG)} \\ \text{RDi}[k'] = 1. \end{array} \right. \end{array} \right.$$

Cette condition unique peut nécessiter pour elle seule plusieurs étapes successives pour sa mise en oeuvre.

Donc si les conditions (1) sont vérifiées avant le transfert, (1') après le transfert, on peut garantir qu'il n'y a pas eu de pannes durant le transfert élémentaire considéré.

IV.5. Erreurs possibles lors d'un transfert vers un opérateur :

Il est nécessaire de poser des conditions particulières pour les opérateurs, en effet, ceux-ci ne se comportent pas de la même manière qu'un registre connecté en destination. D'une part certains opérateurs, PREN, AU, ALU ne sont pas connectés sur la même phase que les registres destinations. D'autre part, les conditions de test de ces opérateurs, connectés sur la même phase que les registres destinations, ne seront pas forcément les mêmes selon le fait qu'ils sont connectés seuls, ou que des registres destinations sont simultanément connectés. En effet l'opérateur connecté seul n'a aucune influence sur les valeurs présentes sur le bus, dans ce cas, il suivra les valeurs du bus en fin de phase T2. Les opérateurs vont donc être passés en revue afin de déterminer leurs conditions particulières vis-à-vis des transferts.

IV.5.1. Opérateur PREN :

D'après le tableau VIII du § IV.3., il est possible de tirer les cas de manifestation de pannes et de leur influence sur l'opérateur PREN.

A la suite d'un transfert, un registre connecté en source et un registre connecté en destination peuvent se trouver dans l'un quelconque des états obtenus lors d'un transfert élémentaire.

A la suite d'un transfert, l'opérateur PREN, qui ne peut être connectable qu'en destination peut se trouver dans l'un des états suivants :

- PREN : = non modifié.
- PREN : = ZERO
- PREN : = UN
- PREN : = Mélange (sources)

- PREN : = source quelconque
- PREN : = source du cycle précédent
- PREN : = concaténation des 8 bits poids faibles de la source et des 8 bits poids forts de UN.

S'assurer qu'aucune panne n'a eu lieu pendant le transfert vers l'opérateur PREN, c'est :

- Vérifier qu'aucun des registres connectables en source sur le bus n'a été perturbé.
- Vérifier qu'aucun des registres connectables en destination sur le bus n'a été perturbé.
- Vérifier que l'opérateur PREN, correspondant à la destination correcte du transfert, a bien été affecté à la valeur de l'origine correcte du transfert.

a) - Non perturbation des registres connectables en source :

Les mêmes conditions que lors d'un transfert élémentaire sont à vérifier à la fois avant et après le transfert :

- (a) RSi \neq origine du cycle précédent $\forall i$
- (b) RSi \neq mélange (sources) $\forall i$

b) - Non perturbation des registres connectables en destination (ne devant pas intervenir dans le transfert) :

Les mêmes conditions que lors d'un transfert élémentaire sont à vérifier à la fois avant et après le transfert :

- (c) RDi \neq 1 source quelconque $\forall i$
- (d) RDi \neq mélange (sources) $\forall i$
- (e) RDI \neq mélange (destinations) $\forall i$

- (f) $RDi \neq$ concaténation des 8 bits poids faibles de la source et de ses propres 8 bits poids forts $\forall i$
- (g) $RDi \neq$ concaténation des 8 bits poids faibles de la source et des 8 bits poids forts du mélange (destinations) $\forall i$.

c) - Affectation de la destination correcte : opérateur PREN :

Vérifier que l'opération PREN, correspondant à la bonne destination du transfert, a bien été affectée à la valeur de l'origine correcte du transfert, c'est vérifier que la valeur finale soit différente de toutes les valeurs que peut prendre cet opérateur en cas de pannes durant le transfert .

Si OPER désigne l'opérateur PREN, correspondant à la destination correcte du transfert, alors les conditions de non perturbation de l'opérateur s'expriment par :

- (n) OPER \neq non modifié
- (o) OPER \neq UN
- (p) OPER \neq ZERO
- (q) OPER \neq mélange (sources)
- (r) OPER \neq 1 source incorrecte.
- (s) OPER \neq 1 concaténation des 8 bits poids faible de la source et des 8 bits poids forts de UN.

Comme l'opérateur PREN doit recevoir la valeur du registre origine, les conditions (n) à (s) se reportent donc aussi sur la valeur de l'origine avant le transfert, soit ORG l'origine correcte, on obtient alors :

- (n') ORG \neq OPER initial (ZERO)
- (o') ORG \neq UN
- (p') ORG \neq ZERO
- (q') ORG \neq mélange (sources)

(r') $ORG \neq RSi - \{ORG\} \quad \forall i$

(s') $ORG \neq$ Concaténation des 8 bits poids faibles de ORG et des 8 bits poids forts de UN .

Le tableau X résume les conditions nécessaires et suffisantes pour la vérification de pannes durant le transfert vers l'opérateur $PREN$. Les notations sont les mêmes que celles du tableau IX, et $PREN$ désigne l'opérateur $pren$.

ORG	RS	RD	OPER = PREN
a	a	c	n
b	b	d	o
$n' = p'$		e	p
o'		f	q
p'		g	r
$q' = b$			s
r'			
s'			

Tableau X. Conditions nécessaires et suffisantes à la vérification des pannes durant un transfert vers l'opérateur $PREN$.

La condition (1) est suffisante ici, mais du fait de la concaténation possible de l'octet poids forts à UN ou imposera alors l'octet pour les bits k et k' , d'autre part il est impossible d'initialiser le $PREN$ à UN avant utilisation. On obtient ainsi :

$$(2) \left\{ \begin{array}{l} - k, k' ; \forall i \left\{ \begin{array}{l} k \in [0, 7] \\ k \in [8, 15] \end{array} \right. \\ \text{et} \end{array} \right.$$

$$(2) \left\{ \begin{array}{l} \text{ORG}[k] = 1 \\ \\ \text{ORG}[k'] = 0 \end{array} \right. \text{ et } \left\{ \begin{array}{l} \text{RSi}[k] = 0 \text{ sauf ORG} \\ \text{RDi}[k] = 0 \\ \text{PREN}[k] = 0 \\ \\ \text{RSi}[k'] = 1 \\ \text{RDi}[k'] = 1 \\ (\text{PREN}[k'] = 0) \end{array} \right.$$

De même la lecture du PREN n'étant pas possible, le résultat du décodage étant envoyé en partie contrôle, l'utilisation de l'adressage post incrémenté ou prédécrémenté permettra de vérifier le nombre de bits à '1' reçu par le PREN. On rappelle que pren est utilisé pour décodage du masque de l'instruction de transfert multiple (MOVEM). Il faudra donc placer des valeurs particulières dans les registres correspondant au codage des bits k et k', afin de vérifier de k a été transféré et k' non.

IV.5.2. Opérateur AU :

D'après le tableau VIII du § IV.3., il est possible de tirer les cas de manifestation de pannes et de leur influence sur l'opérateur AU (AUH + AUL).

A la suite d'un transfert, un registre connecté en source et un registre connecté en destination peuvent se trouver dans l'un quelconque des états obtenus lors d'un transfert élémentaire.

A la suite d'un transfert, l'opérateur AU, qui ne peut être connectable qu'en destination, peut se trouver dans l'un des états suivants :

- au : = ZERO
- au : = UN
- au : = mélange (sources)
- au : = source quelconque.

au' : = source du cycle précédent

au : = concaténation des 8 bits poids faibles du registres source
et des 8 bits poids forts de ZERO.

S'assurer qu'aucune panne n'a eu lieu pendant un transfert vers l'opérateur
AU, c'est :

- vérifier qu'aucun des registres connectables en source sur le
bus n'a été perturbé.
- vérifier qu'aucun des registres connectables en destination sur
le bus n'a été perturbé.
- vérifier que l'opérateur AU, correspondant à la destination
correcte du transfert, a bien été affecté à la valeur de l'origine
correcte du transfert.

a) - Non perturbation des registres connectables en source :

Les mêmes conditions (a) et (b) que lors d'un transfert élémentaire sont
à vérifier à la fois avant et après le transfert.

b) - Non perturbation des registres connectables en destination (ne
devant pas intervenir dans le transfert) :

Les mêmes conditions, (c) à (g), que lors d'un transfert élémentaire sont
à vérifier à la fois avant et après le transfert.

c) - Affectations des registres destinations correctes :

Il est possible qu'il y ait simultanément transfert vers l'opérateur AU et
vers un ou plusieurs registres destination. Dans ce cas, les mêmes condi-
tions (h) à (m), que lors d'un transfert élémentaire sont à vérifier à la fois

avant et après le transfert pour ces registres destinations correctes. Ceux-ci impliquent pour l'origine les conditions (h') à (m').

d) - Affectation correcte de l'opérateur AU :

Vérifier que l'opérateur AU, correspondant à la bonne destination du transfert, a bien été affecté à la valeur de l'origine correcte du transfert, c'est vérifier que la valeur finale est différente de toutes les valeurs que peut prendre cet opérateur en cas de panne durant le transfert. Si OPER désigne l'opérateur AU, correspondant à la destination correcte du transfert, alors les conditions de non perturbation de l'opérateur s'expriment par (cet opérateur étant toujours connecté et obligatoirement modifié) :

- (o) OPER \neq UN
- (p) OPER \neq ZERO
- (q) OPER \neq mélange (sources)
- (r) OPER \neq source incorrecte
- (t) OPER \neq concaténation des 8 bits poids faibles de la source et des 8 bits poids forts de ZERO.

Comme l'opérateur AU doit recevoir la valeur du registre origine les conditions (o), (p), (q), (r) et (t) se reportent donc aussi sur la valeur de l'origine avant le transfert ; soit ORG l'origine correcte, on obtient alors les conditions (o'), (p'), (q') et (r') déjà vues pour l'opérateur pren. Il vient s'ajouter à ces conditions :

- (t') ORG \neq concaténation des 8 bits poids faibles de la source et des 8 bits poids forts de ZERO.

Le tableau XI résume les conditions nécessaire et suffisantes pour la vérification de pannes durant le transfert vers l'opérateur AU. Les notations sont les mêmes que celles du tableau IX et AU désigne l'opérateur AU.

La première partie résume les conditions lorsque l'opérateur est la destination unique, la seconde partie présente les conditions lorsque des registres destinations interviennent aussi dans le transfert.

ORG	RS	RD	OPER = AU (seul)
a b o' p' q' <=> b r' t'	a b	c d e f g	o p q r t

ORG	RS	RD	OPER = AU	RDC
a b h' i' j' <=> b k' l' m' o' p' q' <=> b r' t'	a b	c d e f g	o p q r t	h i j k l m

Tableau XI. Conditions nécessaires et suffisantes à la vérification des pannes durant le transfert vers l'opérateur AU seul ou avec des registres destinations.

La condition (1) est suffisante ici, mais du fait de la concaténation possible de l'octet poids forts à ZERO, on imposera alors l'octet pour les bits k et k' de la condition. On obtient ainsi :

$$(3) \left\{ \begin{array}{l} \exists k, k' ; \forall i \quad \begin{array}{l} k \in [8, 15] \\ k \in [0, 7] \end{array} \\ \text{ORG}[k] = 1 \quad \text{et} \quad \begin{cases} \text{RSi}[k] = 0 \\ \text{RDi}[k] = 0 \\ \text{RDCi}[k] = 0 \end{cases} \\ \text{ORG}[k] = 0 \quad \text{et} \quad \begin{cases} \text{RSi}[k'] = 1 \\ \text{RDi}[k'] = 1 \\ \text{RDCi}[k'] = 1 \end{cases} \end{array} \right.$$

Les vérifications à effectuer après le transfert sont données par (1') où RDCi sera testé par le résultat de l'opération dans les registres AU-Latch.

IV.5.3. Opérateur ALU :

D'après le tableau VIII du § IV.3, il est possible de tirer les cas de manifestation de pannes et de leur influence sur l'opérateur alu. On s'aperçoit que l'on retombe sur les mêmes cas de fonctionnement erroné que pour le pren, et que, comme pour l'au, l'opérateur Alu peut être destination unique, ou simultanément des registres destinations peuvent figurer au transfert.

Le tableau XII résume les conditions nécessaires et suffisantes pour la vérification des pannes durant le transfert vers l'opérateur ALU qui ne peut être connectable qu'en destination. Les notations sont les mêmes que celles des tableaux précédents et ALU désigne l'opérateur ALU.

La première partie résume les conditions lorsque l'opérateur est la destination unique, la seconde partie présente les conditions lorsque des registres destinations interviennent aussi dans le transfert.

ORG	RS	RD	OPER = ALU	
a b n' o' p' q' <=> b r' s'	a b	c d e f g	n o p q r s	
ORG	RS	RD	OPER = ALU	RDC
a b h' i' j' <=> b k' l' m' n' o' p' q' <=> b r' s'	a b	c d e f g	n o p q r s	h i j k l m

Tableau XII. Conditions nécessaires et suffisantes à la vérifications des pannes durant le transfert vers l'opérateur ALU seul ou avec des registres destinations.

La condition (1) est suffisante ici, mais du fait de la concaténation possible de l'octet poids forts avec UN, on imposera alors l'octet pour les bits k et k' (idem pour pren). On obtient ainsi la condition (2).

Les vérifications à effectuer après le transfert sont données (1') où le RDCi sera testé par le résultat de l'opération contenu dans le registre ALU-Latch.

IV.5.4. Opérateur ALUE :

Cet opérateur a exactement le même comportement vis-à-vis des pannes de transfert, qu'un registre. En effet, il peut être connecté en source sur le bus (sortie du résultat) ou en destination d'un transfert (acquisition de l'opérande).

Le tableau XIII résume les conditions nécessaires et suffisantes pour la vérification des pannes de transfert vers l'opérateur ALUE. Les mêmes notations que celles des tableaux précédents sont reprises :

ORG	RS	RD	ALUE et/ou RDC
a	a	c	h
b	b	d	i
h'		e	j
i'		f	k
j' <=> b		g	l
k'			m
l'			
m'			

Tableau XIII. Conditions nécessaires et suffisantes à la vérification des pannes durant un transfert vers l'opérateur ALUE.

Donc si la condition (1) où $RDCi = ALUE$ est vérifiée avant le transfert lors de l'acquisition de l'opérande dans ALUE, on peut garantir qu'il n'y a pas eu de pannes de transfert si la condition (1') où $RDCi = ALUE$ est vérifiée après le transfert. Puis si la condition (1) où $ORG = ALUE$, est vérifiée avant le transfert correspondant à la sortie du résultat on peut garantir qu'il n'ya pas eu de pannes de transfert si la condition (1') où $ORG = ALUE$ est vérifiée après ce transfert.

IV.5.5. Opérateur DCR :

L'opérateur DCR est particulier en ce sens que d'une part il ne peut être qu'une destination de 5 bits, bit 0 à bit 4, sur l'un des bus (bus adresses) et que d'autre part il ne peut être qu'une source de 16 bits sur l'autre bus (bus données), comme le montre la figure 14 au Chapitre VI.

D'après le tableau VIII du § IV.3., il est possible de tirer les cas de manifestation de pannes et de leur influence sur l'opérateur DCR.

A la suite d'un transfert, un registre connecté en source et un registre connecté en destination peuvent se trouver dans l'un quelconque des états obtenus lors d'un transfert élémentaire.

A la suite d'un transfert vers le DCR, l'opérateur peut se trouver dans l'un des états suivants :

dcr : = non modifié

dcr : = ZERO

dcr : = UN

dcr : = 1 source incorrecte (bit 0 à bit 4).

dcr : = mélange (sources) (bit 0 à bit 4).

dcr : = mélange (destination) (bit 0 à bit 4).

S'assurer qu'aucune panne n'a eu lieu pendant le transfert vers l'opérateur DCR, c'est :

- . vérifier qu'aucun des registres connectables en source sur le bus n'a été perturbé.
- . Vérifier qu'aucun des registres connectables en destination (n'intervenant pas dans le transfert) sur le bus n'a été perturbé.
- . Vérifier que l'opérateur d_{cr} , correspondant à la destination correcte du transfert, a bien été affecté à la valeur de l'origine correcte du transfert.

a) - Non perturbation des registres connectables en source :

Les mêmes conditions, (a) et (b) que lors d'un transfert élémentaire sont à vérifier à la fois avant et après le transfert.

b) - Non perturbation des registres connectables en destination (ne devant pas intervenir dans le transfert) :

Les mêmes conditions, (c) à (g) que lors d'un transfert élémentaire sont à vérifier à la fois avant et après le transfert.

c) - Affectation correcte de l'opérateur DCR :

Vérifier que l'opérateur d_{cr} , correspondant à la bonne destination du transfert, a bien été affecté à la valeur de l'origine correcte du transfert, c'est vérifier que la valeur finale est différente de toutes les valeurs que peut prendre cet opérateur en cas de panne durant le transfert. Si OPER désigne l'opérateur d_{cr} , correspondant à la destination correcte du transfert, alors les conditions de non perturbation de l'opérateur s'expriment par :

- (n) OPER \neq non modifié
- (o) OPER \neq UN
- (p) OPER \neq ZERO
- (q) OPER \neq mélange (sources)
- (r) OPER \neq 1 source incorrecte
- (u) OPER \neq mélange (destinations).

Comme l'opérateur doit recevoir la valeur du registre origine les conditions (n) à (u) se reportent donc aussi sur la valeur de l'origine (et sur les 5 premiers bits seulement) avant le transfert, soit ORG l'origine correcte, on obtient alors :

- (n'') ORG [0:4] \neq OPER [0:4] initial
- (o'') ORG [0:4] \neq UN
- (p'') ORG [0:4] \neq ZERO
- (q'') ORG [0:4] \neq mélange (sources) [0:4]
- (r'') ORG [0:4] \neq RSI - {ORG} [0:4] yi
- (u'') ORG [0:4] \neq mélange (destinations) [0:4]

Le tableau XIV résume les conditions nécessaires et suffisantes pour la vérification de pannes durant le transfert vers l'opérateur dcr.

Les notations sont les mêmes que celles des tableaux précédents et DCR désigne l'opérateur dcr.

ORG	RS	RD	OPER = DCR
a	a	c	n
b	b	d	o
n"		e	p
o"		f	q
p"		g	r
q"			u
r"			
u"			

Tableau XIV. Conditions nécessaires et suffisantes à la vérification des pannes durant un transfert vers l'opérateur dcr.

La condition (1) est encore suffisante ici, mais l'opérateur, n'ayant que cinq bits connectés sur le bus, impose une condition supplémentaire. On obtient alors :

$$(4) \left\{ \begin{array}{l} \exists k, k', k'', \forall i \\ \begin{array}{l} \text{ORG}[k] = 1 \\ \text{ORG}[k'] = 0 \end{array} \quad \text{et} \quad \left\{ \begin{array}{l} \text{RSi}[k] = 0 \\ \text{RDi}[k] = 0 \\ \text{DCR}[k] = 0 \\ \text{RSi}[k'] = 1 \\ \text{RDi}[k'] = 1 \\ \text{DCR}[k'] = 1 \end{array} \right. \\ \\ \text{et} \left\{ \begin{array}{l} k, k' \in [0,4] ; k'' \in [0,4], k \in [8,15] \\ \text{ORG}[k''] = 0 \\ \text{ou} \\ k' \in [0,4] ; k'' \in [0,4], k \in [8,15] \end{array} \right. \quad \left\{ \begin{array}{l} \text{RSi}[k''] = 1 \\ \text{RDi}[k''] = 1 \\ \text{DCRi}[k''] = 1 \end{array} \right. \end{array} \right.$$

$$(4) \quad \text{ORG [k'']} = 1 \quad \text{et} \quad \left\{ \begin{array}{l} \text{RSi [k'']} = 0 \\ \text{RDi [k'']} = 0 \\ \text{DCR [k'']} = 0 \end{array} \right.$$

Les vérifications à effectuer après le transfert sont données par (1') ou le RDCi sera testé par le résultat de l'opérateur contenu dans le registre ALU-latch.

En effet, la sortie de l'opérateur DCR est immédiatement envoyée vers l'ALU pour la manipulation du bit correspondant (instructions BCLR, BSET, BCHG).

REMARQUE :

En source sur le bus donnée l'opérateur se comporte comme un registre source, mais sans modification possible de cette source.

IV.5.6. Opérateur SIGNEX :

L'opérateur SIGNEX est particulier en ce sens que d'une part il ne peut être qu'une destination sur 1 bit (le bit 15) sur les bus ABL, DBL et que d'autre part, il ne peut être qu'une source de 16 bits sur les bus ABH et DBH, comme le montre la figure 21 au chapitre VI. De plus, cet opérateur ne nécessite aucun cycle opératoire, il effectue l'extension de signe simultanément au transfert.

A la suite d'un transfert via l'opérateur signex, un registre source sur le bus L peut se trouver dans l'un des états suivants :

- . Source (L) = non modifiée
- . Source (L) : = source du cycle précédent
- . Source (L) : = mélange des sources.
- . Source (L) : = signe (source).

un registre source sur le bus H peut se trouver dans l'un des états suivants :

- . Source (H) = non modifiée
- . Source (H) = signè (source).

un registre destination sur le bus L peut se trouver, en plus des états déjà obtenus à la suite d'un transfert, affecté de la valeur du signe de la source. Soit :

- . Destination (L) = signe (source).

un registre destination sur le bus H peut se trouver, dans l'un des états suivants :

- . Destination (H) = non modifiée
- . Destination (H) = signe (1 source (L) quelconque).
- . Destination (H) = signe (destination (L))
- . Destination (H) = signe (source (L) du cycle précédent)
- . Destination (H) = signe (mélange (source (L)))
- . Destination (H) = signe (mélange (destination (L)))
- . Destination (H) = mélange (destination(H)).

s'assurer qu'aucune panne n'a eu lieu pendant un transfert avec extension de signe, c'est :

- . Vérifier qu'aucun des registres connectés en source sur les bus L et H n'a été perturbé.
- . Vérifier qu'aucun des registres connectables en destination sur le bus L et le bus H, ne devant pas intervenir dans le transfert n'a été perturbé.
- . Vérifier que le ou les registres correspondant à la bonne destination du transfert sur le bus L, ont bien été affectés à la valeur de l'origine correcte du transfert.

. Vérifier que le ou les registres correspondant à la bonne destination de l'extension du signe sur le bus H, ont bien été affectés à la valeur du signe de l'origine correcte du transfert.

a) - Non perturbation des registres connectables en source sur le bus L :

Les conditions (a) et (b) définies précédemment sont reprises ici, à ces conditions s'ajoute la condition :

(v) $RS_i(L) \neq \text{signe (source)} \forall i$

Ces registres ne devant pas être modifiés par le transfert, ces conditions doivent être vérifiées aussi bien avant qu'après le transfert.

b) - Non perturbation des registres connectables en source sur le bus H :

La condition suivante, nécessaire et suffisante, doit être vérifiée à la fois avant et après le transfert :

(w) $RS_i(H) \neq \text{signe (source)} \forall i$

c) - Non perturbation des registres connectables en destination sur le bus L et ne devant pas intervenir dans le transfert :

Les conditions (c) à (g) définies précédemment sont reprises ici, à ces conditions s'ajoute la condition :

(x) $RD_i(L) \neq \text{signe (source)}$

Ces registres ne devant pas être modifiés par le transfert, ces conditions doivent être vérifiées aussi bien avant qu'après le transfert.

d) - Non perturbation des registres connectables en destination sur le bus H et ne devant pas intervenir dans le transfert :

(z1) $RD_i(H) \neq \text{signe (RST (L))} \forall i$

- (z2) $RDi (H) \neq \text{signe } (RDi (L)) \forall i$
- (z3) $RDi (H) \neq \text{signe } (ORG (L) \text{ du cycle précédent})$
- (z4) $RDi (H) \neq \text{signe } (\text{mélange } (RSi (L))) \forall i$
- (z5) $RDi (H) \neq \text{signe } (\text{mélange } (RDi (L))) \forall i$
- (z6) $RDi (H) \neq \text{mélange } (RDi (H)) \forall i.$

Ces registres ne devant pas être affecté par le transfert, les conditions doivent être vérifiées aussi bien avant qu'après le transfert.

e) - Affectation correcte (de la ou) des destination(s) du bus L :

Ces registres doivent être affectés de la valeur de l'origine correcte du transfert. Leur valeur doit donc être différente de toutes les valeurs qu'ils peuvent prendre en cas de mauvais fonctionnement. Les conditions (h) à (m) définies précédemment sont reprises ici, à ces conditions s'ajoute la condition :

$$(y) RDCi (L) \neq \text{signe } (ORG)$$

Ces registres devant recevoir la valeur de l'origine correcte, ces conditions (h) à (m) et (y) se reportent sur le registre origine correcte pour donner les conditions (h') à (m') et (y'), où (y') est définie par :

$$(y') ORG \neq \text{signe } (ORG).$$

Ces conditions doivent être vérifiées aussi bien avant qu'après le transfert.

f) - Affectation correcte (de la ou) des destination(s) de l'extension de signe sur le bus H :

Ces registres doivent être affectés à la valeur l'extension du signe de l'origine correcte du transfert. La valeur doit donc être différente de toutes les valeurs qu'ils peuvent prendre en cas de mauvais fonctionnement.

On obtient ainsi les conditions nécessaires et suffisantes suivantes :

- (z7) RDCi (H) \neq signe (RSi (L) - {ORG})
- (z8) RDCi (H) \neq signe (RDi (L)) $\forall i$
- (z9) RDCi (H) \neq signe (ORG (L) du cycle précédent)
- (z10) RDCi (H) \neq signe (mélange (RSi (L))) $\forall i$
- (z11) RDCi (H) \neq signe (mélange (RDi (L))) $\forall i$
- (z12) RDCi (H) \neq mélange (RDi (H)) $\forall i$
- (z13) RDCi (H) \neq non modifié.

Ces registres devant recevoir la valeur de l'origine correcte, ces conditions se reportent sur le registre origine correcte pour donner les conditions suivantes :

- (z7') signe (ORG) \neq signe (RSi - {ORG}) $\forall i$
- (z8') signe (ORG) \neq signe (RDi (L)) $\forall i$
- (z9') signe (ORG) \neq signe (ORG (L) du cycle précédent)
- (z10') signe (ORG) \neq signe (mélange (RSi (L))) $\forall i$
- (z11') signe (ORG) \neq signe (mélange (RDi (L))) $\forall i$
- (z12') signe (ORG) \neq mélange (RDi (H)) $\forall i$
- (z13') signe (ORG) \neq RDi (initial).

Ces conditions doivent être vérifiées aussi bien avant qu'après le transfert.

Le tableau XV résume les conditions nécessaires et suffisantes pour la vérification de pannes durant un transfert via l'opérateur d'extension de signe.

ORG (bus L)	RS (bus L)	RD (bus L)	RDC (bus L)	RS (bus H)	RD (bus H)	RDC (bus H)
a	a	c	h	w	z1	z7
b	b	d	i		z2	z8
v	v	e	j		z3	z9
$y' \Leftrightarrow v'$		f	k		z4	z10
z7'		g	l		z5	z11
z8'		x	m		z6	z12
z9'			y			z13
z10'						
z11'						
z12'						
z13'						

Tableau XV. Conditions nécessaires et suffisantes à la vérification des pannes durant un transfert via l'opérateur SIGNEX du MC 68000.

Une condition suffisante est donnée par la condition (1), à laquelle s'ajoute la condition :

$$k = 15 \text{ et } k' = 15$$

Ces 2 conditions sont incompatibles et par suite 2 séquences de test au moins seront nécessaires, elles sont données par :

$$(5) \left\{ \begin{array}{l} (1) \\ + \\ k = 15 \end{array} \right.$$

$$(6) \left\{ \begin{array}{l} (1) \\ + \\ k' = 15 \end{array} \right.$$

IV.5.7. Opérateurs non concernés par les pannes de transfert :

Deux opérateurs ne sont pas concernés par les pannes de transferts :
l'opérateur BCD et l'opérateur MUX I/O : En effet, aucune de ces 2 opérateurs n'est connecté sur un bus interne :

- . L'opérateur BCD est connecté directement entre le registre ALU-R et l'opérateur ALU.
- . L'opérateur MUX I/O, quant à lui, fait le lien entre les registres DBIN, DOB et le bus donnée externe (EDB).

Ainsi donc ces 2 opérateurs ne sont pas "touchés" par les pannes de transfert du bus et aucune condition n'est donc à poser pour ces 2 opérateurs.

Tous les opérateurs ont été passé en revue, il reste un dernier type de pannes de transfert à considérer.

IV.6. Erreurs possibles lorsqu'aucun transfert ne doit avoir lieu :

Il est en effet nécessaire de vérifier qu'aucun des registres n'est perturbé qu'il soit connecté en source et/ou en destination, lorsqu'aucun transfert ne doit avoir lieu. Les conditions obtenues dans les paragraphes précédents (§ IV.4. et § IV.5) sont toujours valables dans ce cas; mais, pour ce cas particulier, aucune source correcte n'existe et par suite, tout registre connectable en origine sur le bus doit vérifier les conditions (1) à (6) car, de plus, n'importe quel opérateur peut entrer en jeu. Ainsi, si n registres sont connectables en source sur le bus, n suites des 6 séquences sont à vérifier pour chaque registre connectable en source.

IV.7. Conclusion sur les hypothèses de pannes de la gestion des transferts :

Cette conclusion va en fait présenter une comparaison entre les résultats obtenus par THATTE & ABRAHAM dans [THA 78], ceux obtenus par COURTOIS dans COU 81 et ceux qui sont présentés ici.

Les hypothèses de THATTE & ABRAHAM concernant les microprocesseurs en général sont les suivantes :

- Si aucune source n'est sélectionnée, le bus peut se trouver à ZERO ou à UN.
- Si plusieurs sources sont sélectionnées alors, la valeur du bus correspond au OU câblé ou au ET câblé des sources connectées ensemble sur le bus, la fonction dépendant de la technologie utilisée.
- Les destinations connectées, bonnes et/ou mauvaises reçoivent la valeur du bus.

Les hypothèses de COURTOIS, concernant l'application au microprocesseur M 6800, sont les suivantes :

- Si aucune source n'est sélectionnée, le bus se trouve préchargé à la valeur UN.
- Si plus d'une source est sélectionnée alors, la valeur du bus correspond au ET câblé des sources connectées ensemble sur le bus.
- Les destinations connectées, bonnes et/ou mauvaise reçoivent la valeur du bus.

Les hypothèses ici, concernant l'application au microprocesseur MC 68000, sont les suivantes :

- Si aucune source, n'est sélectionnée, le bus reste préchargé à UN.
- Si plusieurs sources sont sélectionnées, alors la valeur des bus correspond au mélange des sources connectées ensemble sur le bus.

- Les destinations connectées bonnes et/ou mauvaise reçoivent soit la valeur du bus si cette valeur est une valeur logique, soit le mélange des destinations dans le cas contraire.

Les conditions suffisantes pour Thatte and Abraham sont données par :

- Transfert entre registres :

- 1) $ORG = ZERO$ $R_i = UN$ $\forall R_i \neq ORG$
- 2) $ORG = UN$ $R_i = ZERO$ $\forall R_i \neq ORG$

- Transfert vers opérateur :

R_1 (op1)		R_2 (op2)		R_3 (résultat)
OP1	op	OP2	+	OP3
OP4	op	OP2	+	x
OP4	op	OP5	+	x

- Pour chaque opération op.
- 1) $R_i = UN$ $\forall R_i \neq R_1, R_2, R_3$
 - 2) $R_i = ZERO$ $\forall R_i \neq R_1, R_2, R_3$

OP1	op	OP2	≠	OP3
OP4	op	OP2	≠	OP1 op OP2
OP4	op	OP5	≠	OP4 op OP2

Les conditions suffisantes pour COURTOIS sont données par :

a) bonne origine :

- 1) $A = UN$
 $R_i^* = 0$ $\forall R_i \neq A$ $R^* = \text{condensé multiplicatif de } R.$
- 2) $A \neq UN$

b) Bonne destination :

- 1) valeur finale $\neq UN$
- 2) valeur finale $\neq ET$ (valeur initiale, valeur finale).

c) Test de non perturbation :

$$R_i = UN \quad \forall i$$

$$BUS^* = 0$$

Les conditions suffisantes obtenues ici sont données par :

- Transfert entre registre :

$$\exists k, k' ; \begin{cases} \text{ORG}[k] = 1 & R_i[k] = 0 & \forall R_i \neq \text{ORG} \\ \text{ORG}[k'] = 0 & R_i[k'] = 1 & \forall R_i \neq \text{ORG}. \end{cases}$$

k et k' n'appartenant pas au même octet.

- Transfert vers opérateur :

Les conditions sont sensiblement les mêmes, mais la "place" de k et k' peut être imposée parmi les 15 bits, un k'' peut venir s'ajouter.

Remarque :

- On peut remarquer que :

Si les conditions annoncées par THATTE et ABRAHAM, pour les transferts entre registres sont suffisantes, pour le test du MC, les conditions associées aux transferts vers des opérateurs, ne le sont pas en ce sens qu'aucune hypothèse sur la valeur des opérandes et des résultats n'est donnée, si ce n'est que les 3 résultats doivent être différents.

Or ces trois résultats peuvent être différents, mais posséder un sous-groupe de bits qui sont égaux, et ceci provenant de la panne, ainsi rien ne permet de détecter cette panne.

Le M 6800 ne possédant pas de cycle sans transfert, aucune condition n'est à ajouter à celles obtenues par COURTOIS.

On peut remarquer que la condition (1) obtenue ici permet de tester les pannes

correspondant aux hypothèses de pannes de THATTE & ABRAHAM.

Ces conditions sont aussi suffisantes au test des pannes correspondant aux hypothèses de pannes de COURTOIS.

(Mais elles ne sont pas nécessaires : les conditions pour tester les transferts sur le MC 68000 sont plus complexes que les conditions pour tester le M 6800).

V. TEST DU SEQUENCEMENT DES INSTRUCTIONS :

Des classes de test sont nécessaires, non pas de manière à obtenir une façon hiérarchique de tester les différentes instructions, mais en fonction des conséquences qu'une erreur manifestée par une instruction peut avoir sur le séquençement d'un programme.

V.I. Définitions et classes de pannes (de test) :

La séparation, faite dans [COU 81] entre registres opératifs, instructions opératives, registres non opératifs et instructions non opératives, est reprise ici.

Les registres sont donc divisés en :

Registre opératif ::

Registre non destiné, en général, à contenir une valeur devant servir à l'élaboration d'une adresse instruction.

Registre non-opératif :

Registre destiné en général, à contenir une valeur devant servir à l'élaboration d'une adresse instruction.

Le tableau XVI présente les 2 classes de registres pour le microprocesseur MC 68000. Il est évident que cette répartition correspond à une utilisation standard des registres. Ainsi, par exemple, lorsqu'un registre données est utilisé comme index d'une instruction de branchement indexé : ce registre sera considéré comme non opératif, de même lorsqu'un registre adresses est utilisé comme index pour accéder à un élément de tableau (l'index représentant le n°. de l'élément dans le tableau). Dans ce cas particulier le registre adresses sera considéré comme opératif.

De même, les instructions sont divisées en :

Instruction opérative :

Instruction manipulant des données non destinées, en général, à être utilisées pour l'élaboration d'une adresse instruction.

Instruction non opérative :

Instruction manipulant des données destinées, en général, à être utilisées pour l'élaboration d'une adresse instruction, ou bien ne manipulant aucune donnée.

Le tableau XVII présente les 2 classes d'instructions pour le microprocesseur MC 68000.

	Registre opératifs	Registres non-opératifs
Connu du prog.	D0, D1, D2, D3, D4, D5 D6, D7,	A0, A1, A2, A3, A4, A5, A6 PC USP, SSP
Connu macro machine	FTU, PREN, DCR, ALUE ALU, ALUB DBIN, DOB DT	AOB AUR

Tableau XVI. Classes de registres du MC 68000 (Utilisation standard des registres).

Instructions opératives			Instructions non-opératives		
add adda addi addx clr cmp cmpm cmpi divs divu ext muls mulu neg negx sub subq subi subx tst	and andi eor eori or ori not scc	bchg bclr bset btest	adda cmpa suba movea	rtr rts nop reset rte stop	ori to SR andi to SR eori to SR move to SR move from SR move USP
		exg move moveq movem movep swap			
	asl asr lsl lsr rol ror roxl roxr	abcd pabcd sbcd	bcc dbcc bra bsr jmp jsr		
		tas			

Tableau XVII. Classes d'instructions du MC 68000.

On peut ainsi distinguer, d'après les classes de registres et d'instructions quatre classes de test (des hypothèses de pannes) selon le type d'actions réalisées par une instruction en cas de panne, telles que déjà introduites au Chapitre IV :

- Classe 0 : Une instruction opérative ne 'calcule' pas correctement : le résultat n'est pas le résultat juste, elle modifie la valeur de registres opératifs qu'elle ne devrait pas modifier.
- Classe I : une instruction non opérative modifie de manière intempestive des registres opératifs.

- Classe II : une instruction non opérative ne 'calcule' pas correctement, si elle doit calculer, ou bien elle modifie de manière intempestive des registres non opératifs qu'elle ne devrait pas modifier.

- Classe III : une instruction opérative modifie de manière intempestive des registres non opératifs.

Les conséquences particulières des pannes lors d'erreurs de gestion du code condition, de branchement à bon escient, font que deux classes supplémentaires seront considérées plus loin.

Contrairement au test analytique, pour lequel les différentes classes d'hypothèses de pannes correspondent à une qualité croissante du test, les classes définies ci-dessus caractérisent des types de pannes différents de par leurs conséquences sur le séquençement des programmes.

On rappelle en effet, que les valeurs des broches de sorties ne sont pas supposées être vérifiées, et par suite, des mécanismes tels que chien de garde et protection mémoire, sont nécessaires afin de détecter les erreurs se manifestant par un déséquençement des programmes, et ceci aussi bien pour un test en ligne que pour un test hors-ligne.

V.2. Procédures de test utiles pour les classes 0 et II, sur un cycle :

Pour ces classes, il est nécessaire de tester qu'une instruction, opérative ou non, 'calcule' correctement (le test de non-perturbation de registres opératifs ou non, sera abordé dans le cadre des procédures de tests utiles aux classes I et II). Tester qu'une instruction 'calcule' correctement implique, en plus du test de l'opérateur lui-même, de vérifier que l'origine

sur le bus est la bonne et que le résultat est bien dirigé vers la bonne destination.

Les paragraphes IV.5.1. à IV.5.7. décrivent les conditions nécessaires suffisantes au test d'un transfert vers un opérateur. Puis les conditions suffisantes sont données par opérateur. Ainsi selon l'opérateur les conditions (1), (2), (3), (4) ou (5) doivent être vérifiées avant le transfert afin de déterminer les pannes qui peuvent se produire durant ce transfert. Les conditions de test (1') et (2'), (3') et (4') ou (5') doivent alors être vérifiées par le latch de l'opérateur, le cas échéant.

Pour le test de la destination du résultat (cycle suivant) la valeur initiale du latch devra vérifier la condition (1), la destination finale du résultat vérifiant la condition (1') après le transfert.

Une remarque doit être faite au sujet d'instructions dont la seule manifestation est une modification du code-condition (telles que les instructions de test ou de comparaison). C'est alors le code-condition qui doit être observé. Ce sont toujours les origines de bus, commandes d'opérateur, etc.... qui sont testées, et non la gestion du code-condition. Il ne faut pas oublier en effet que dans de telles opérations, comparaison par exemple, un transfert est commandé depuis les 2 'valeurs' à comparer vers l'ALU, le résultat de l'opération n'est pas le rangement de la différence des 2 valeurs mais seulement le rangement des indicateurs correspondant à la différence.

V.3. Procédures de test utiles pour les classes I et III sur un cycle :

Ces classes doivent en effet tester qu'une instruction, opérative ou non, ne perturbe pas de registres, opératifs ou non, qu'elle ne doit pas perturber (même si elle utilise la valeur de ces registres). Les conditions

suffisantes (1), (2), (3), (4), (5) et (6) permettent d'effectuer le test de non perturbation des registres ne devant pas être perturbés. Mais, ici, le test ne s'effectuera pas seulement sur les registres ne devant pas être perturbés, le test s'effectuera aussi sur l'origine correcte et la destination correcte du transfert.

En effet, les conditions de majorité de registres connectés, fait que, pour les conditions suffisantes considérées, les effets des pannes ne vont pas seulement se traduire par une perturbation des registres ne devant pas être perturbés, mais aussi par une perturbation sur l'origine correcte et/ou la destination correcte, ces deux tests s'effectuant en une seule séquence.

Plusieurs étapes (de tests) peuvent être nécessaires si par exemple pour un registre connectable à plusieurs bus, des valeurs particulières indépendantes ne sont pas commandables simultanément sur tous les bus.

V.4. Multiplicité temporelle, erreurs sur plusieurs cycles :

Les paragraphes précédents ont examiné l'occurrence de mauvais séquençement durant un cycle de fonctionnement. Il faut naturellement examiner si les conditions de tests déterminées précédemment, restent applicables si des erreurs de séquençement se produisent durant plusieurs cycles, ce qui doit être l'hypothèse retenue. Dans ce cas, seules les valeurs de début d'exécution d'une instruction sont commandables et seules les valeurs de fin d'exécution d'une instruction sont observables.

Les notations suivantes vont être utilisées pour chaque cycle :

_____	caractérisera un registre non utilisé
ORG	caractérisera un registre utilisé comme origine sur un bus
DEST	caractérisera un registre utilisé comme destination sur un bus
-----	caractérisera la répétition de la séquence précédente un

nombre de fois quelconque.

Chaque instruction définit alors différentes séquences correspondant à l'utilisation de chaque registre.

La figure 26 donne l'exemple d'une utilisation possible d'un registre :

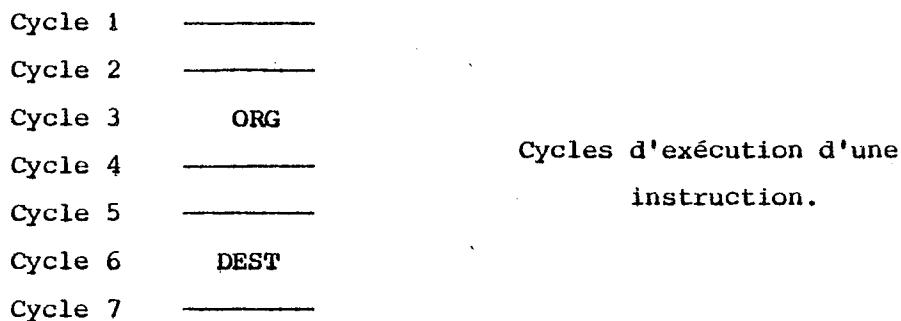


Figure 26. - Exemple de séquence d'utilisation d'un registre au cours d'une instruction.

Seule la valeur du cycle 1 est directement commandable et seule celle du cycle 7 est directement observable.

Durant chacun des 7 cycles, de mauvais séquencements de commandes peuvent apparaître, générant des pannes de transfert par exemple. Détecter les pannes se produisant durant un cycle implique de déterminer les opérandes nécessaires de façon à détecter la mauvaise séquence durant ce cycle, mais aussi avant et après ce cycle. Ainsi, dans l'exemple de séquence d'utilisation d'un registre de la figure 26, non seulement, les conditions de test utiles pour les classes 0, I, II et III doivent être vérifiées pour chaque cycle correspondant à l'utilisation particulière du registre, mais les conditions suivants sont à ajouter. La valeur chargée dans le registre au cycle 6 doit non seulement correspondre aux conditions d'un registre connecté

en destination, mais elle doit être différente de ce que peut devenir la valeur initiale (au cycle 3) en cas de perturbation durant les cycles 1, 2 et 5, 6, si le registre n'est pas connecté en destination. Elle doit également être différente de ce qu'elle peut devenir si elle est perturbée au cycle 7. En fonction de la complexité des instructions, la séquence de cycles peut être longue et impliquer des utilisations multiples d'un registre.

Les séquences de base de l'utilisation des registres, telles que déterminées dans [COU 81] sont reprises ici du fait de leur universalité. Les conditions de test correspondantes, déterminées dans [COU 81], sont suffisantes pour le microprocesseur étudié, seules les conditions (1) (1') (2) et (3) de COURTOIS sont à remplacer par les conditions (1), (2), (3), (4), (5) et (6) selon les opérateurs mis en jeu à chaque séquence.

Les huit séquences de base sont rappelées à la figure 27.

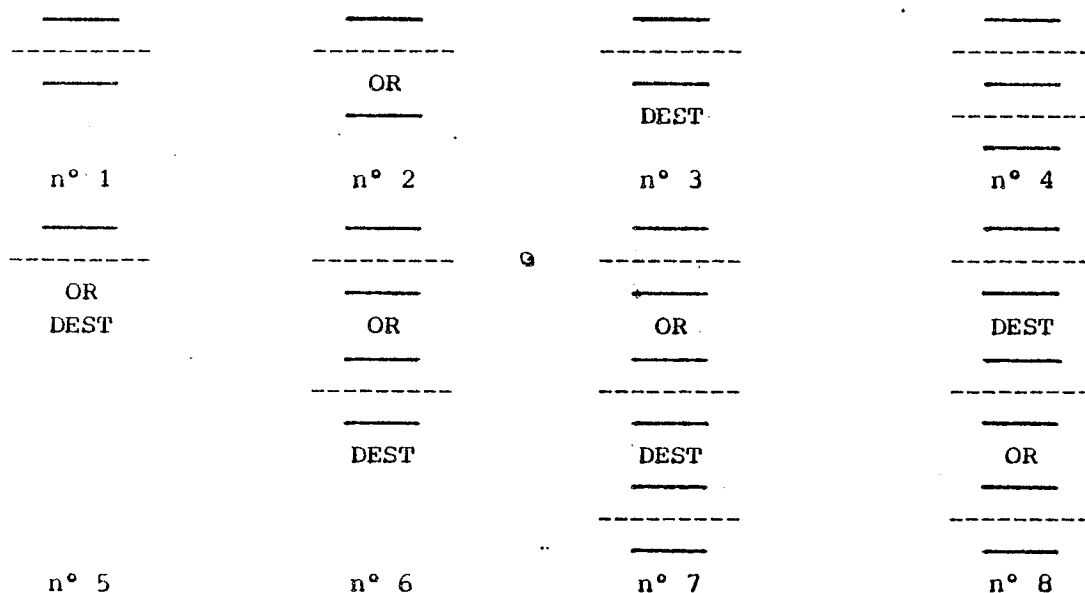


Figure 27. - Séquences de base d'utilisation d'un registre au cours d'une instruction.

La séquence n° 1 correspond à un registre non utilisé.

La séquence n° 2 correspond à l'utilisation d'un registre comme second opérande, ou opérande à résultat non rangé en registre, (opérande rangé en mémoire).

Les séquences n° 3 et n° 4 correspondent au chargement d'un registre, les opérations durant plus ou moins longtemps.

Les séquences n° 5, n° 6 et n° 7 correspondent à l'utilisation d'un registre comme premier opérande d'un calcul, et donc comme destination du résultat, l'opération durant plus ou moins longtemps.

La séquence n° 8 caractérise l'utilisation d'un registre cache ou tampon, servant à stocker un résultat intermédiaire, réutilisé par la suite.

V.5. Multiplicité temporelle et spatiale : test d'une instruction :

A une instruction (définie par son type et son mode d'adressage) correspond un ensemble de séquences, une pour chaque registre, ainsi que l'utilisation des opérateurs. En fonction de la classe de test considérée, les conditions de test sont à mettre en oeuvre et des valeurs sont à observer, concernant certains registres ou adresses mémoire.

Déterminer quels registres sont à observer ne présente pas de difficulté en fonction de la classe testée, du type de registre et de la séquence temporelle d'utilisation : un registre opératif utilisé selon la séquence de base n° 1 n'est pas à observer pour la classe III, etc....

Le tableau XVII résume les classes concernées en fonction du type de registre et des différentes séquences, en supposant qu'une instruction opérative ne concerne, pour la classe 0, que des registres opératifs (ce qui ne sera pas le cas pour des phases d'adressage).

Type de registre \ séquence	séquence							
	1	2	3	4	5	6	7	8
Opératifs	0, I	0	0	0	0	0	0	0, II
Non opératifs	II, III	II	II	II	II	II	II	

Tableau XVII. Classes de test concernées en fonction du type de registre et des différentes séquences.

L'interaction entre les différentes conditions (différentes séquences, conditions de test pour les opérateurs) peut amener à plusieurs étapes, i.e. plusieurs fois commander et observer des valeurs.

Le point le plus important est la commande/observation des valeurs et par conséquent les hypothèses de mauvais séquençement concernant une ou plusieurs instructions. L'hypothèse retenue ici, comme dans [COU 81], est la suivante :

- Un seul type d'instruction est affecté en cas de panne, mais toutes les instructions de ce type utilisant des modes d'adressage différents sont affectées.
- Un seul mode d'adressage est affecté en cas de panne, mais tous types d'instructions utilisant ce mode sont alors affectés.
- Il n'y a pas simultanément de mauvais séquençement pendant la phase d'adressage et pendant la phase d'opération.

Cette hypothèse permet de ne tester qu'un nombre réduit d'instructions possibles (pour différentes classes) en examinant en fait séparément les séquençements des phases d'adressage et des phases opération de la manière suivante (pour différentes classes) :

- i) Test des modes d'adressage, par exécution d'une instruction de type quelconque et mode m , en commandant/observant à l'aide d'instructions de types quelconques et de modes différents de m .
- ii) Test des types, par exécution d'une instruction de type t et de mode d'adressage quelconque, en commandant/observant à l'aide d'instructions de type différent de t et de mode quelconque.

La simultanéité de pannes à la fois pour un type et pour un mode, n'est pas admise en raison de la non-détection pouvant avoir lieu en l'absence de contraintes sur les utilisations des types et modes. En effet, si t_1 et m_1 sont les types et mode de l'instruction exécutée pour le test du mode m_1 , deux erreurs peuvent se masquer et ceci ne sera pas détecté si le mode m_1 est utilisé par la suite aux fins de commande/observation pour le test du type t_1 .

Le nombre de tests (par exemple pour une classe) est ainsi approximativement donné par la somme du nombre de types d'instructions (70 pour le MC 68000) et du nombre de modes d'adressage (12 pour le MC 68000) multiplié par 2 pour

les 2 tailles possibles mot ou double mot, alors que si l'on admettait qu'un mauvais séquençement puisse se produire pour uniquement un type donné associé à un mode d'adressage donné, le nombre de ces tests serait approximativement égal au produit du nombre de types par nombre de modes multiplié par 2 pour les 2 tailles possibles mot ou double mot (soit environ 1700 pour le MC 68000).

Une raison incitant à faire cette hypothèse restrictive est bien sûr, que ce sont les mêmes circuits, même séquence de microinstructions mis en oeuvre pour un mode d'adressage quel que soit le type de l'instruction, et les mêmes circuits, même séquences de microinstructions mis en oeuvre pour un type d'instruction quel que soit le mode d'adressage.

Pendant les phases d'adressage, les classes 0 et II sont concernées par le fait que les instructions, opératives ou non, doivent "calculer" avec les bons opérandes mémoire. Le test nécessaire est donc que l'opérande atteint est celui qui doit l'être. Les quatre classes sont concernées par le fait que pendant la phase d'adressage, les registres opératifs ou non, ne doivent pas être perturbés. Une fois encore on peut rappeler que seuls les adressages d'opérandes, et non ceux destinés à fournir une adresse d'instruction, sont considérés..

VI. HYPOTHESES ET TEST DE LA GESTION DU MOT D'ETAT :

Le mot d'état a déjà été présenté lors du chapitre II de la présentation du MC 68000.

Il peut être divisé en 3 unités différentes entre elles aussi bien de par leur utilisation, leur fonctionnement que par leur localisation à l'intérieur de la partie contrôle.

On distingue donc :

- Le registre code condition : ce registre est composé des cinq bascule N, V, Z, C, X, il est situé au-dessus de l'ALU, sa gestion est faite simultanément à la gestion de l'ALU.
- Le mot superviseur : dans ce mot, on placera les cinq bascules de l'état superviseur.
- Le masque d'interruption : ce registre est composé des 3 bascules du masque d'interruption permettant de déterminer si une nouvelle interruption, plus prioritaire que la précédente, arrive sur les plots d'interruption.
- Les bascules S et T : Les bascules S (mode superviseur) et T (mode trace).

Le test de la bonne gestion du mot d'état constituera la classe IV d'hypothèses de pannes. Le nombre de tests nécessaires pour cette classe serait très important si quelques particularités du circuit ne sont pas connues et si des hypothèses ne sont pas faites.

Ces 3 unités vont être passées en revue l'une après l'autre.

VI.1. Hypothèses et test de la gestion du code condition :

Comme il a été dit dans les paragraphes précédents, les classes 0, I, II, III ne testent pas la gestion du code condition, que les instructions soient ou non des instructions de manipulation directe de ce code condition. Ces classes testent uniquement la bonne source et la bonne destination dans le registre le code condition. A partir des informations plus ou moins élaborées délivrées par les opérateurs, la partie contrôle doit gérer l'état des indicateurs.

Ce circuit de gestion des indicateurs reçoit donc :

- des valeurs calculées par l'opérateur ALU, plus ou moins élaborées (13 lignes de prédicats de l'ALU).
- des valeurs du bus BC provenant de la partie opérative (via le FTU), lors des instructions de chargement du code condition, (5 lignes du bus BC).
- des signaux de contrôle, provenant de la partie contrôle aux fins de gestion (30 lignes).
- L'état précédent du code condition.

Les sorties de ce circuit, i.e. les indicateurs, sont utilisées pour :

- le calcul des conditions de branchement.
- le transfert vers le bus BC, correspondant soit à un changement de contexte (interruption, exception), soit à une instruction de rangement du code condition.
- l'exécution d'opérations par l'ALU.

Des hypothèses sont à prendre en compte en raison du nombre d'entrées du

circuit. En effet, bien que pour chaque cycle de chaque instruction, les valeurs provenant des opérateurs, ainsi que les valeurs provenant de la partie opérative soient connues et commandables, le nombre d'entrées est tellement important, qu'un test exhaustif concernant le séquençement de chaque instruction deviendrait prohibitif : 2^{13+5+5} pour chaque instruction du MC 68000 au supposant qu'un seul test soit suffisant pour chaque instruction (i.e. si chaque instruction ne se déroulait que sur un cycle).

De manière analogue au test du séquençement des instructions (cf. § V), les hypothèses suivantes sont faites, en ce qui concerne un mauvais séquençement des signaux de contrôle de la gestion du code condition :

- un seul type d'instruction est affecté mais toutes les instructions de ce type utilisant des modes d'adressage différents sont dans ce cas affectées.
- un seul mode d'adressage est affecté, mais tout type d'instructions utilisant ce mode est dans ce cas affecté.
- il n'y a pas simultanément mauvais séquençement durant la phase d'adressage et la phase d'opération.

Ces hypothèses permettent d'examiner séparément les phases d'adressage et d'opération : pendant les phases d'adressage, le code condition ne doit pas être modifié, et pendant la phase opération le code condition doit être correctement mis à jour. Malgré ces hypothèses le nombre de tests reste très élevé car, il est nécessaire de tester que la valeur finale du code condition est la bonne, pour toutes les valeurs possibles, de toutes les entrées du circuits, et ceci pendant chaque cycle, et ce :

- pour chaque mode d'adressage, par exécution d'une instruction de ce mode en commandant/observant les valeurs du code condition par des instructions d'un autre mode d'adressage.
- pour chaque type d'instruction, par exécution d'une instruction de ce type en commandant/observant les valeurs du code-condition par

des instructions d'un autre type.

Ce n'est qu'en connaissant plus finement la structure du circuit que ce nombre peut être réduit. Ainsi pour le MC 68000, la gestion d'un indicateur est indépendante de la gestion des autres, sauf pour un d'entre eux (X) : les circuits sont physiquement indépendants et la valeur finale d'un indicateur ne dépend ni de la valeur antérieure des autres indicateurs, ni de leur valeur courante sauf pour l'indicateur X (d'extension), qui, soit reçoit la même valeur que l'indicateur C (retenue) soit conserve sa valeur antérieure.

Sous réserve d'hypothèses de localité de pannes (analogues aux classes de pannes du test analytique), le nombre de tests peut être réduit puisque pour chaque indicateur, sauf X et C, les entrées possibles du circuit le gérant ne sont plus que sa valeur antérieure, la valeur de chargement provenant du bus BC, la valeur des lignes de compte rendu lui correspondant, comme le montre la figure 28.

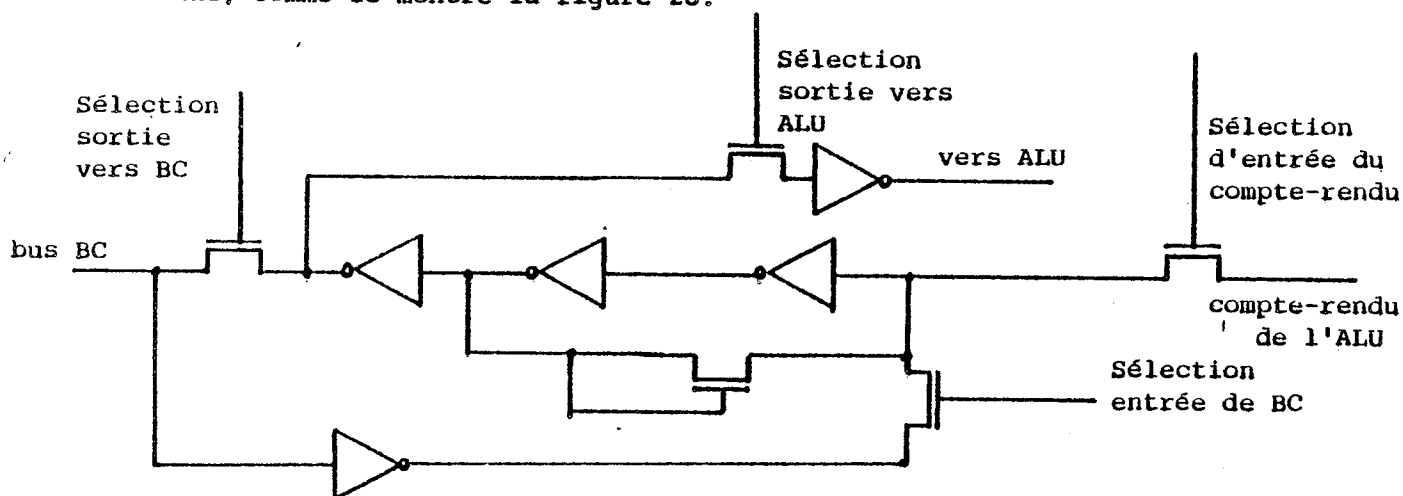


Figure 28. - Bascule du code condition.

En fait, il reste encore un grand nombre de possibilités et ceci dû au fait que le bus BC peut recevoir un grand nombre de valeurs. Le tableau XVIII regroupe les différentes valeurs que peut recevoir chaque indicateur,

en provenance du bus BC en cas de panne. La colonne (1) correspond à une utilisation normale des autres à des cas impossibles en fonctionnement sans panne.

INDICATEUR	BUS BC						
	1	2	3	4	5	6	7
X	FTU(0)	FC0	IRD(0)	1	1	IRD(9)	0
C	FTU(1)	FC1	IRD(1)	1	PSW(0)	IRD(A)	0
V	FTU(2)	FC2	IRD(2)	1	PSW(1)	IRD(B)	TRAP(0)
N	FTU(3)	R/W	IRD(3)	1	PSW(2)	IRD(9)IRD(A) .IRD(B)	TRAP(1)
Z	FTU(4)	NE	IRD(4)	0	1	1	TRAP(2)

Tableau XVIII. Valeurs complétement possibles du bus BC sur les lignes correspondant au code-condition.

La colonne (1) correspond à un usage normal, lorsque le code-condition doit être chargé par la partie opérative (MOVE TO CCR, MOVE TO SR, RTE, RTR, ORI ANDI, EORI TO CCR).

La colonne (2) correspond à une sauvegarde mémoire du mot d'état superviseur (en cas d'erreur d'adresses ou d'entrée de bus).

La colonne (3) correspond aux instructions où la donnée transite par le Pla IRD (BCC.B, MOVEQ) ou en cas d'erreur de bus d'adresse (mot opération).

La colonne (4) correspond à la constante '-16' chargée sur le bus BC multiplication, division.

La colonne (5) correspond à l'envoi de l'auto-vecteur d'interruption.

La colonne (6) correspond aux instructions où la donnée (3bits) transite sur le pla IRD (ADDQ, SUBQ, Décalages dynamiques).

La colonne (7) correspond à l'envoi d'un vecteur de démarrage d'exception ou d'interruption par le pla TRAP.

Il est évident que le bus reçoit en plus, la valeur du code-condition lorsque celui-ci doit être chargé en mémoire.

Le test du registre de code condition, va donc se diviser en 4 étapes qui sont les suivantes :

- 1ère Etape : vérifier que les instructions utilisant les bits du bus BC correspondants au code-condition, et ne devant pas modifier ce code-condition, ne le modifie pas.
Dans le cas où ces instructions effectuent un calcul sur l'ALU, les lignes de compte-rendu devront portées la valeur opposée à celle du bit correspondant du bus BC.
Par exemple, pour une instruction de multiplication correspondant à la colonne 4 du tableau XVIII:
Le résultat devra générer $C = 0$, $V = 0$, $N = 0$; $Z = 1$;
l'indicateur X ne devant pas être modifié, il sera initialisé à 0.
Cette vérification peut nécessiter plusieurs étapes.
- 2ème Etape : Vérifier que les instructions utilisant les bits du BC pour charger le code-condition, le charge effectivement, ce cas correspond à la colonne (1) du tableau XVIII. Aucune de ces instructions n'utilise l'ALU après avoir chargé le registre de code-condition.
- 3ème Etape : Vérifier que les instructions utilisant l'ALU, et ne devant pas modifier le registre de code-condition ne le modifient pas,

pour les deux valeurs de chaque bascule du code-condition.

- 4ème Etape : Vérifier que les instructions utilisant l'ALU, et devant mettre à jour le registre de code condition le mettent correctement à jour. Les entrées possibles à partir du bus BC n'étant pas toutes commandables. Ce test devra s'exécuter dans les mêmes conditions pour chaque instruction, c'est-à-dire, il ne doit pas y avoir d'utilisation du bus BC entre chaque vérification. La vérification étant faite sur les 2 valeurs initiales possibles de chaque bascule.

Chacune de ces étapes est divisée en deux, la première concernant la phase d'adressage (si il y en a une), la seconde concernant la phase d'opération.

Il est nécessaire de remarquer que certaines instructions à priori n'utilisant pas le bus BC se comportent comme tel, car un chargement de FTU est commandé alors qu'il n'est pas nécessaire, ces instructions seront donc rattachées à la première étape du test. Parmi ces instructions 2 risquent de perturber le fonctionnement, ce sont les instructions :

MOVE. $\left\{ \begin{matrix} W \\ L \end{matrix} \right\}$ M, xxxx.L où le FTU reçoit les 16 bits poids faible de l'adresse absolue longue (xxxx.L).

et Bcc.W déplacement où le FTU reçoit les 16 bits du déplacement (mot d'extension).

Les autres instructions ne risquent pas de créer d'erreurs supplémentaires car la valeur chargée dans le FTU correspond à celle qui se trouve déjà sur le PLA TRD (mot d'opération) ceci pour les instructions :

MOVE An to USP

LEA (An), An

Tous les transferts multiples (MOVEM quelque soit le mode d'adressage).

VI.2. Hypothèses et test de la gestion du mot superviseur :

On rappelle que l'on nomme ici, mot superviseur, les cinq bascules de mémorisation de l'état superviseur, c'est-à-dire FC_0 , FC_1 , FC_2 donnant le type de fonction (zone d'accès mémoire : programmes/données, utilisateur/superviseur, interruption), R/W précisant si l'accès est en lecture ou en écriture, NE précisant l'état d'exception (NE = 1 si interruption ou exception, NE = 0 pour l'exécution normale d'un programme).

Ce mot superviseur n'est utilisé qu'en cas d'erreur du bus ou d'erreur d'adresse, il est chargé dès que le microprocesseur a reconnu une telle erreur. Il est mémorisé jusqu'au chargement d'une nouvelle valeur, à la prochaine erreur de bus ou erreur d'adresse . Pendant le traitement de la procédure d'exception traitant l'erreur de bus et d'adresse, le mot superviseur est sauvegardé en mémoire (SP-14).

Pour l'initialiser, il est donc nécessaire de provoquer une erreur d'adresse ou de bus ainsi, l'état de ces bascules sera connu. Pour un test hors-ligne, il est envisageable de provoquer une erreur d'adresse et de vérifier le fonctionnement de ces bascules dans les différents états possibles, mais pour un test en-ligne l'erreur d'adresse sera immédiatement détectée, il n'est donc pas possible de tester en ligne ces 5 bascules.

Il est rappelé que FC_2 correspond à la valeur du bit S (les valeurs des FC peuvent être trouvées au § 3.6. du chapitre II).

VI.3. Hypothèses et test de la gestion des interruptions :

Les 3 bascules du masque d'interruption possèdent chacune deux entrées, une à partir du bus BC lors du chargement du mot d'état (MOVE TO SR, RTR, RTE), l'autre à partir de la logique de gestion des interruptions, déterminant la priorité des interruptions.

Ces 3 bascules possèdent chacune deux sorties, une sortie vers la logique d'interruption, et une double sortie sur le bus BC. Cette double sortie correspond soit aux bits 1, 2 et 3 du bus BC lors de la génération de l'autovecteur d'interruption, soit aux bits 8, 9 et A du bus BC pour les sauvegardes du mot d'état (MOVE FROM SR).

Il est rappelé que le registre d'état (SR) regroupe les cinq bascules du code-condition, les 3 bascules du masque d'interruption et les bascules S (mode superviseur) et T (mode trace).

L'entrée par la logique d'interruption ne pourra être testée que pour un test hors-ligne, pour lequel des demandes de priorité croissante puis décroissante seront faites afin de vérifier la bonne gestion des bascules. Pour un test en-ligne, cette vérification dépend uniquement de l'usage fait par l'utilisateur des interruptions.

En entrée, depuis le bus BC, les 3 bascules peuvent prendre les valeurs données dans le tableau XIX.

Masque	BUS BC			
	(1)	(2)	(3)	(4)
PSWI (0)	FTU (8)	0	IRD (8)	TRAP (7)
PSWI (1)	FTU (9)	0	IRD (9)	TRAP (8)
PSWI (2)	FTU (A)	0	IRD (A)	0

Tableau XIX. Valeurs complémentées possibles du bus BC sur les lignes correspondant au masque d'interruption.

La colonne (1) correspond à l'usage normal de chargement du masque par la partie opérative (MOVE TO SR).

La colonne (2) correspond au poids forts de la constante '-16' utilisée pour la multiplication et la division.

La colonne (3) correspond à la sauvegarde de IRD (complet) lors des cas de bus erreur et adresse erreur.

La colonne (4) correspond à l'envoi du vecteur de démarrage d'exception ou d'interruption par le pla TRAP.

Le test se fera donc en chargeant le masque avec une valeur initiale telle qu'elle soit modifiée si un cas d'erreur se produit : faire transiter une valeur sur le FTU (complément du masque) effectuer une multiplication ou une division. Les 2 derniers cas ne sont pas envisageables pour un test en-ligne. Pour un test hors-ligne, générer une erreur de bus ou d'adresse, le mot opération contenant le complément de la valeur du masque. (Idem avec une interruption).

La sortie, depuis le bus BC, est utilisée pour fournir au périphérique ayant provoqué l'interruption, le numéro de l'interruption acquitée, cette sortie ne pourra donc être testée qu'hors-ligne. Les conséquences d'une panne dans ce cas, peuvent être définies par le tableau XVIII, correspondant au code-condition, pour lequel dans ce cas la colonne (5) correspond au fonctionnement correct et normal, la colonne des indicateurs étant une entrée supplémentaire pour le bus BC.

VI.4. Hypothèses et test des bascules S et T :

Les bascules S et T possèdent 2 entrées et 2 sorties ; les entrées sont données, pour le bus BC par le tableau XX.

Bascules	(1)	(2)	(3)
S	FTU (D)	IRD (D)	'1'
T	FTU (F)	IRD (F)	'1'

Tableau XX. Valeurs complémentées possibles du bus BC sur les lignes correspondant aux bascules S et T.

La colonne (1) correspond à un usage normal, chargement des bits S et T par la partie opérative (MOVE TO SR, RTR, RTE).

La colonne (2) correspond au cas d'erreur de bus ou d'erreur d'adresse (sauvegarde du mot opération).

La colonne (3) correspond aux instructions où une donnée transite par le bus BC :

ADDQ, SUBQ, MOVEQ, BCC.B, multiplication et division, DBcc, décalages dynamiques, ou lorsqu'un vecteur d'interruption est placé sur BC venant du PLA TRAP.

La seconde entrée possible correspond à une remise à zéro pour le bit S et une mise à un pour le bit T.

Les sorties correspondent, soit à une sortie sur le bus BC, utilisée pour la sauvegarde du mot d'état (MOVE FROM SR), soit aux signaux état superviseur (PSWS) et mode trace (TPEND) servant à générer respectivement les sousprogrammes d'exception pour instructions privilégiées et de mode trace (signaux décodés au niveau du PLA A0 d'exception).

La même remarque que pour le mot superviseur la manipulation de ces bascules ne peut être envisagée que pour un test hors-ligne et non durant l'exécution d'une application.

Vu le faible nombre d'entrées, un test exhaustif pourra être envisagé hors-ligne. Il est clair qu'un mauvais fonctionnement en-ligne provoquera un arrêt immédiat, que ce soit pour une détection d'instructions privilégiées ou pour un basculement en mode trace.

VIII. TEST DU BRANCHEMENT A BON ESCIENT :

Une dernière classe, la classe V, concerne le test du branchement à bon escient, i.e. le bon fonctionnement des circuits de calcul de la condition de branchement.

Cette classe est différente des classes I et II pour lesquelles les instructions de branchement interviennent, mais uniquement pour la non-perturbation de registres opératifs et non-opératifs. La bonne exécution de ces instructions dépend de l'état des entrées arrivant sur le PLA de branchement en provenance du code-condition, des différents comptes-rendus d'opérateurs. Là aussi, une connaissance plus fine du circuit est nécessaire car le pla ne comporte pas moins de 21 entrées. En effet, ce PLA est divisé en deux parties, la première partie est utilisée pour les branchements dans le microprogramme des instructions ayant 1 alternative dans leur séquencement, la seconde partie décode la condition de branchement des instructions Bcc, DBcc et Scc pour lesquelles cette condition de branchement se trouve dans le mot opération.

Le test du branchement à bon escient va donc se diviser en deux parties, la première partie concernant les alternatives de séquence, la seconde la condition de branchement.

VIII.1. Test des alternatives de séquence :

On appelle alternative de séquence une microinstruction possédant deux microinstructions suivantes (la condition de passage à la suivante est définie par la microinstruction elle-même).

Un test exhaustif de toutes les alternatives de séquence sera mis en oeuvre, il sera composé des 25 instructions suivantes :

- (1) une instruction quelconque utilisant l'adressage indirect indexé plus déplacement, la taille de l'index étant le mot (W).
- (2) une instruction quelconque utilisant l'adressage indirect indexé plus déplacement, la taille de l'index étant le long mot (L).
- (3) une division signée ou non avec le diviseur nul ; cette instruction provoquera un TRAP, et ne sera donc envisagée que pour un test hors-ligne.
- (4) une division signée, le diviseur étant négatif, le dividende positif et de reste nul.
- (5) une division signée, le diviseur étant positif, le dividende négatif et de reste nul.
- (6) une division signée, le diviseur et le dividende étant positif et de reste nul.
- (7) une division signée, le diviseur étant négatif, le dividende positif et de reste non nul.
- (8) une division signée, le diviseur étant positif, le dividende négatif et de reste non nul.
- (9) une division signée, le diviseur et le dividende étant positifs et de reste non nul.
- (10) une division signée ou non provoquant un débordement ; la division ne sera pas exécutée la bascule V sera positionnée à zéro.

- (11) une multiplication non signée, le multiplicateur et le multiplicande étant positifs, chacun des deux contenant au moins 2 bits à '1' et 2 bits à '0'.
- (12) une multiplication signée, le multiplicateur étant positif, le multiplicande étant négatif, chacun des deux contenant au moins 2 bits à '1' et 2 bits à '0'.
- (13) une multiplication signée, le multiplicateur étant négatif, le multiplicande étant positif, chacun des deux contenant au moins 2 bits à '1' et 2 bits à '0'.
- (14) une multiplication signée, le multiplicateur et le multiplicande étant négatif, chacun des deux contenant au moins 2 bits à '1' et 2 bits à '0'.
- (15) une instruction DBcc, le registre Dn contenant une valeur positive, la condition étant toujours fausse, DBRA par exemple.
- (16) une instruction CHK, le registre Dn étant négatif, la borne étant positive, cette instruction provoquera une exception (TRAP) et ne sera donc envisagée que pour un test hors-ligne.
- (19) une instruction CHK, le registre Dn étant positif, la borne étant positive, le registre Dn étant inférieur à la borne.
- (20) une instruction de manipulation de bits, le bit à manipuler étant situé dans les poids faibles du registre (< 16).
- (21) une instruction de manipulation de bits, le bit à manipuler étant situé dans les poids forts du registre (> 16).

- (22) une instruction TRAPV avec V = '0'.
- (23) une instruction TRAPV avec V = '1', cette instruction provoquera une exception (TRAP) et ne sera donc envisagée que pour un test hors-ligne.
- (24) une instruction de transferts multiples (MOVEM) la taille de l'opération étant le mot (W).
- (25) une instruction de transferts multiples (MOVEM) la taille de l'opération étant le long mot (L).

Pour ce test exhaustif :

Les instructions (1) et (2) testent la ligne 1 du pla.

Les instructions (3), (4), (5), (6), (7), (8), (9) et (10) testent les lignes 2, 3, 4, 5, 6, 7, 8, 9 et 10 du pla.

Les instructions (11), (12), (13) et (14) testent les lignes 11, 12, 13, 14, 15 et 16 du pla.

L'instruction (15) teste la ligne 17 du pla.

Les instructions (16), (17), (18) et (19) teste la ligne 18 du pla.

Les instructions (20) et (21) testent la ligne 19 du pla.

Les instructions (22) et (23) testent la ligne 20 du pla.

Les instructions (24) et (25) testent les lignes 21 et 22 du pla.

La ligne 23 est inutilisée.

La ligne 25 sert à valider la seconde partie du pla, correspondant à la fois au décodage du type de condition qui se trouve dans le mot opération, et à l'évaluation de la condition.

VII.2. Test de la seconde partie du PLA de branchement :

Contrairement aux alternatives de séquences qui utilisaient, pour évaluer la condition, les valeurs du code-condition et celles de tous les comptes-rendus d'opérateur, la seconde partie du pla de branchement n'évalue la condition que sur le registre de code-condition.

La bonne exécution de ces instructions dépend de l'état des entrées arrivant sur la partie contrôle en provenance du code-condition. Pour cette classe V, chacune des instructions de branchement (en non une seule, représentant le type de branchement comme cela est suffisant pour les classes I et II) doit être exécutée, pour chaque valeur possible des indicateurs. Les seuls indicateurs utilisés sont les indicateurs C, N, V, Z et ainsi 2^4 tests sont suffisants.

Le même genre de procédure que celui développé dans [COU 81] pourra être utilisé. Pour chaque valeur de CNVZ, on peut classer tous les branchements possibles en deux types : l'un pour lequel la condition est vrai, l'autre pour lequel la condition est fausse. On peut avoir alors le programme suivant :

```
pour CNVZ = 0000 jusqu'à 1111 faire
début
    (* le code condition est positionné *)
    BC1          E1
    Allera       panne
E1 : BC2          E2
    Allera       panne
E2 : BC3          E3
    Allera       panne
} branchement de type 1
```

E ₈	: $\overline{BC1}$	panne	}	branchement de type 2.
	$\overline{BC2}$	panne		
	$\overline{BC3}$	panne		

E ₁₅	: $\overline{BC16}$	panne	}	
	Allera	non panne		

fin

où panne correspond à la procédure de traitement de panne et non panne
il n'existe pas de panne (passer à la configuration suivante de CNVZ).

Si ce programme est réalisé avec ;

- . DBRA Dn pour la boucle pour (Dn contenant la valeur de CNVZ)
- . MOVEQ Dn, 15 pour l'initialisation de Dn
- . MOVE Dn, CCR pour positionner le code-condition.

Ce programme peut être rapidement évalué à 2660 cycles pour les 256 bran-
chements et 16 positionnements de CNVZ.

IX. CONCLUSIONS :

L'évaluation de la cinquième étape de la méthodologie est donc couverte par le chapitre VI, pour la partie opérative et le chapitre VII, pour la partie contrôle.

Pour la partie contrôle, les hypothèses de pannes sur les modes d'adressage et sur les types d'opérations reproduisent parfaitement la réalisation physique des microprogrammes.

En effet, on peut séparer les microprogrammes du MC 68000 : ceux réalisant la phase d'adressage quel que soit le type d'opération, de ceux réalisant la fonction désirée, quel que soit le mode d'adressage des opérandes nécessaires à l'opération. Toutes les données ont donc été fournies pour programmer ces différentes procédures.

Une remarque importante peut être faite sur l'impact de la complexité de la machine sur la complexité du test.

On peut ainsi distinguer deux types de complexité, la complexité quantitative et la complexité qualitative.

La complexité quantitative est liée au facteur d'accroissement du nombre d'unités fonctionnelles, telles que les opérateurs, les registres etc... Ce nombre d'unités fonctionnelles est multiplié par cinq en passant du M 6800 au MC 68000, accroissant ainsi la complexité quantitative. L'augmentation de complexité quantitative se traduit par une augmentation du temps passé à l'exécution des programmes de test.

La complexité qualitative est liée à la réalisation de la fonction et plus particulièrement à la complexité des conséquences des pannes sur cette fonction. C'est le cas bien connu, de la différence de complexité du test

entre une porte standard et une porte non standard réalisant la même fonction (ou exclusif par exemple). Les hypothèses de pannes à prendre en compte sont complètement différentes. Comme on a pu le remarquer dans ce chapitre, il existe la même différence entre un bus unifilaire (standard) et un bus différentiel (non-standard). Dans le cas du bus différentiel, les hypothèses de pannes sont complètement différentes et une analyse complète du fonctionnement en présence de pannes, est nécessaire. Ainsi, même pour des hypothèses fonctionnelles (haut niveau), des simulations électriques (bas niveau) aient été nécessaires pour déterminer les conséquences des pannes. L'augmentation de la complexité qualitative se répercute ainsi plus sur l'effort nécessaire à déterminer les conséquences des pannes et à déterminer la façon de tester ces pannes que sur le temps passé à exécuter le programme de test. Ainsi, chaque fois qu'une fonction déjà existante sera réalisée d'une façon non standard l'étude complète sera à effectuer sur cette nouvelle réalisation.

On voit donc que la complexité d'une machine, telle qu'elle est définie, peut influencer de deux façons différentes sur la complexité du test : soit sur le temps passé à exécuter des programmes de test, soit sur l'effort à déterminer les conséquences de pannes.



REFERENCES

- [COU 81] COURTOIS B.
 "Test et LSI".
 Thèse de doctorat d'état, Grenoble, 12.06.81.
- [THA 78] THATTE S.M., ABRAHAM J.A.
 "A methodology for functional level testing of micro-
 processors".
 Proc. of the 8th Fault-Tolerant Computing Symposium,
 Toulouse, june 1978, France.

CHAPITRE VIII
FRÉQUENCES OPTIMALES
ORDONNANCEMENTS DES PROCÉDURES
Etape (vi)

CHAPITRE VIII - FREQUENCES OPTIMALES

ORDONNANCEMENT DES PROCEDURES

Etape (vi)

I	- INTRODUCTION	513
II	- FORMULATION DU PROBLEME	514
III	- EXPRESSION DES SOLUTIONS	516
	III.1. Solutions du problème I	516
	III.2. Solutions du problème II	517
IV	- TEMPS DE LATENCE ET TEMPS DE DETECTION	517
V	- CAS PARTICULIER : UN SEUL PROGRAMME DE TEST	518
	V.1. Interprétation géométrique	519
	V.2. Ordre optimal des procédures de test	521
	V.2.1. Rappel des hypothèses	522
	V.2.2. Ordre optimal pour une latence nulle	522
	V.2.3. Ordre optimal pour une latence non nulle	523
VI	- CONCLUSION	525
	REFERENCES	527

I. INTRODUCTION

Afin de couvrir les 6 étapes de la méthodologie, et ainsi que ce document constitue un tout, le présent chapitre reprend, dans le cas qui nous concerne, les résultats de [COU 82] et [COU 82-2]. Ces deux rapports traitent globalement du problème des fréquences optimales de relance des programmes de test d'une part, de l'ordonnancement des procédures de test d'un programme unique de test d'autre part ; tous les cas sont envisagés, programme de test interruptible, non interruptible etc....

Ici donc, on considère dans une première partie, l'évaluation des fréquences optimales de relance des procédures de test telles qu'elles ont été décrites aux chapitres VI et VII, chacune ayant sa propre période en fonction de la partie étudiée. Les formules tenant compte du temps de latence sont données. Puis le cas particulier, où toutes ces procédures sont groupées en un seul "programme" de test est abordé et le problème posé par l'ordonnancement optimal des différentes procédures les unes par rapport aux autres trouve une solution dans ce cas particulier. L'extension de ces résultats en tenant compte du temps de latence est aussi donnée.

II. FORMULATION DU PROBLEME

On ne s'intéressera ici qu'au problème posé par un microprocesseur unique et non un système complet. Il est rappelé que les formules complètes concernant un système peuvent être trouvées dans [COU 82-1] : On suppose donc que le microprocesseur sous test est doté de mécanismes de détection de pannes tels qu'ils ont été présentés au Chapitre III. Ces mécanismes incluent donc un phénomène périodique : le chien de garde.

Comme on l'a vu au Chapitre V, ces détecteurs ne détectent pas toutes les pannes, ou bien certaines pannes sont trop longues à détecter par ces mécanismes, c'est pourquoi des tests spécifiques tels que présentés aux Chapitres VI et VII, doivent être exécutés périodiquement afin de tester les pannes considérées. Ces tests entraînent eux aussi un certain temps de détection. Tous les phénomènes périodiques envisagés sont supposés entraîner un certain "coût" : temps d'exécution des programmes de test périodique, temps d'exécution de la routine de relance du chien de garde.

Les tests sont supposés parfaits et le temps de détection obtenu par les détecteurs est borné, c'est-à-dire, qu'il n'existe aucune panne du microprocesseur qui ne soit détectée en un temps fini :

Les paramètres qui seront considérés par la suite sont :

- . le temps de détection d'une panne affectant le système
- . le "coût" de la détection, mesuré par les "coûts" des différents tests.

En considérant des distributions de pannes exponentielles, et supposant que le temps d'exécution d'un test périodique est très inférieur à la période de ce test on obtient :

$$E [\text{Temps de détection}] = a_0 \varphi (T_0) + \sum_{i=1}^n a_1 \frac{T_i}{2}$$

L'expression du coût est donnée par :

$$\text{Coût} = \sum_{i=0}^n \frac{t_i}{T_i}$$

Avec : T_0 = période du chien de garde.

$\psi(T_0)$ = espérance du temps de détection des détecteurs.

a_i = probabilité d'occurrence d'une panne sur la partie i du microprocesseur, étant donné qu'une panne survient dans le microprocesseur ($i = 0$ à n).

T_i = période du test associé à la partie i ($i = 1, n$).

T_0 = durée d'exécution de la routine de relance du chien de garde.

t_i = durée d'exécution du test de la partie i ($i = 1, n$).

Remarques :

- Le "coût" adopté est donc la proportion de temps, passée pour les problèmes de test par rapport au temps total disponible.
- Le temps de détection des pannes de certaines parties, pourrait dépendre de plusieurs périodes T_0, T_1, T_2, \dots si des pannes peuvent être détectées par plusieurs moyens : détecteurs et programmes de test périodiques.
- Les coefficients a_i sont déterminés à partir des distributions de pannes à l'intérieur du microprocesseur (périphérie, parties en logique aléatoire, parties à motifs réguliers, ...).

Deux problèmes apparaissent donc :

PbI déterminer $T^* = [T_0^*, T_1^*, T_2^*, \dots]$ tel que le temps de détection minimal, pour un "coût" donné.

PbII déterminer $T^* = [T_0^*, T_1^*, T_2^*, \dots]$ tel que le "coût" adpoté soit minimal, pour un temps de détection donné.

III. EXPRESSION DES SOLUTIONS [COU 82-1]

Dans le cas général, les expressions des solutions sont assez complexes, mais pour notre cas particulier où φ est linéaire en fonction de la durée du chien de garde pour les pannes de la périphérie.

Si $\varphi(T_0) = c T_0 + d$, on a successivement pour les problèmes I et II :

III.1. Solution du problème I :

La contrainte de coût est à considérer, elle s'exprime par :

$$K = \sum_{i=0}^n \frac{t_i}{T_i}$$

La formule générale nous donne :

$$K = \frac{1}{T_0^*} \left[t_0 + \frac{\sum_{i=1}^n \sqrt{a_i t_i}}{\sqrt{2 \frac{a_0}{t_0} \varphi'(T_0^*)}} \right]$$

D'où l'on tire aisément :

$$T_i^* = \sqrt{\frac{t_i}{a_i}} \sqrt{\sum_{i=0}^n a_i t_i} \quad i = 0, \dots, n$$

$$\text{Avec : } \begin{cases} \alpha_0 = \frac{2a_0 c}{K} \\ \alpha_i = \frac{a_i}{K} \quad i = 1, \dots, n \end{cases}$$

III.2. Solution du problème II :

La contrainte de temps est à considérer, elle s'exprime par :

$$b = a_0 \psi(T_0) + \sum_{i=1}^n \frac{a_i T_i}{2}$$

La formule générale nous donne :

$$b = a_0 \psi(T_0^*) + T_0^* \sqrt{\frac{a_0}{2t_0}} \psi'(T_0^*) \left(\sum_{i=1}^n \sqrt{a_i t_i} \right)$$

D'où l'on tire aisément :

$$T_i^* = \frac{\sqrt{\frac{t_i}{\alpha_i}}}{\sum_{i=0}^n \sqrt{\alpha_i t_i}} \quad i = 0, \dots, n$$

$$\text{Avec : } \begin{cases} \alpha_0 = \frac{a_0 C}{b - a_0 d} \\ \alpha_i = \frac{a_0 C}{2(b - a_0 d)} \quad i = 1, \dots, n \end{cases}$$

IV. TEMPS DE LATENCE ET TEMPS DE DETECTION

Comme on l'a vu, au Chapitre III, il existe généralement un temps de latence entre l'instant d'occurrence physique d'une panne et sa manifestation. Il est donc plus réaliste de considérer le temps de détection à partir de la première manifestation de la panne : des sorties fausses sont délivrées à l'extérieur. Si EL_i représente l'espérance de cette latence associée à la partie i , le temps

de détection de cette partie devient alors :

$$\frac{T_i}{2} - EL_i$$

Lorsque les pannes de cette partie sont détectées par des tests périodiques. De la même façon cette latence s'intègre dans l'expression $\varphi(T_0)$ pour donner $\theta(T_0)$. Les relations de caractérisation sont inchangées. La solution du problème I reste la même, c'est-à-dire :

$$T_i^* = \frac{\sqrt{\frac{t_i}{\alpha_i}}}{\sum_{i=0}^n \sqrt{\alpha_i t_i}} \quad \text{avec} \quad \begin{cases} \alpha_0 = \frac{2a_0 C}{K} \\ \alpha_i = \frac{a_i}{K} \quad i = 1, \dots, n \end{cases}$$

La solution du second problème devient :

$$T_i^* = \frac{\sqrt{\frac{t_i}{i}}}{\sum_{i=0}^n \sqrt{\alpha_i t_i - k_i}} \quad \text{avec} \quad \begin{cases} \alpha_0 = \frac{a_0 C}{b - a_0 (d - EL_0)} \\ \alpha_i = \frac{a_i}{2(b - a_0 (d - EL_0))} \quad i = 1, \dots, n \\ k_0 = 0 \\ K_i = \frac{a_i EL_i}{\sqrt{2(b - a_0 (d - EL_0))}} \quad i = 1, \dots, n \end{cases}$$

V. CAS PARTICULIER : UN SEUL PROGRAMME DE TEST

On ne s'intéressera dans la suite au problème à deux paramètres, T_0 et T'_1 , où T_0 est la période du chien de garde, et T'_1 correspond à la période de relance du programme de test unique. Ce programme regroupant toutes les

procédures courant le test de chaque parties : $t'_1 = \sum_{n=1}^n t_i$, décrits dans les paragraphes précédents.

V.1. Interprétation géométrique

Dans le cas où $\varphi(T_0)$ est linéaire, les problèmes I et II se résolvent aisément de façon géométrique que l'on tienne ou non compte de la latence des pannes.

Ainsi, en se cantonnant au problème à deux paramètres, les courbes représentatives des fonctions de minimisation des coûts et temps de détection sont respectivement un ensemble d'hyperboles équilatères de mêmes asymptotes et un ensemble de droites parallèles. Les relations de caractérisation donnant l'équation de l'asymptote de l'hyperbole. Dans le cas du problème I, une hyperbole est donnée (contrainte) et les courbes de temps de détection sont des droites comme le montre la figure 1.

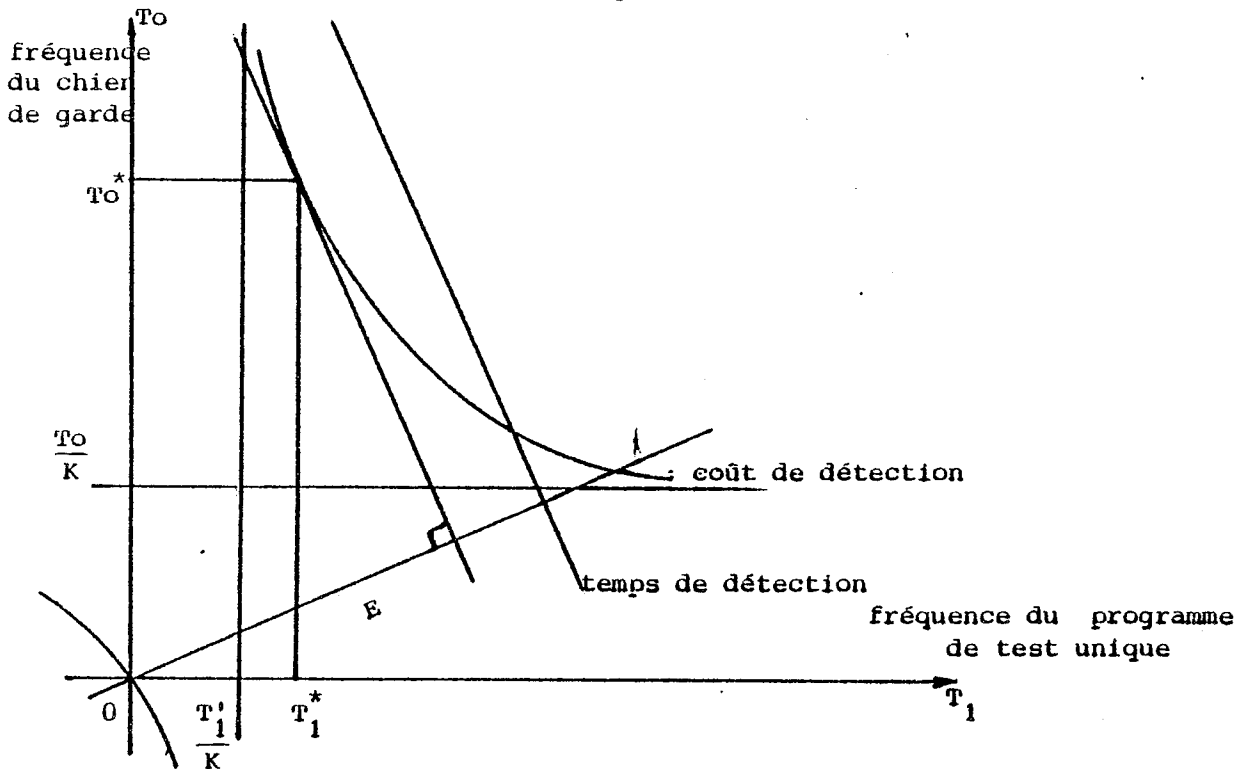


Figure 1. - Problème I : coût imposé.

La solution correspond naturellement à la droite tangente à l'hyperbole (relations de caractérisation).

Dans le cas du problème II, une droite est donnée (contrainte) et les courbes de coûts sont des hyperboles comme le montre la figure 2. La solution correspond aussi à la droite tangente à l'hyperbole (relations de caractérisation).

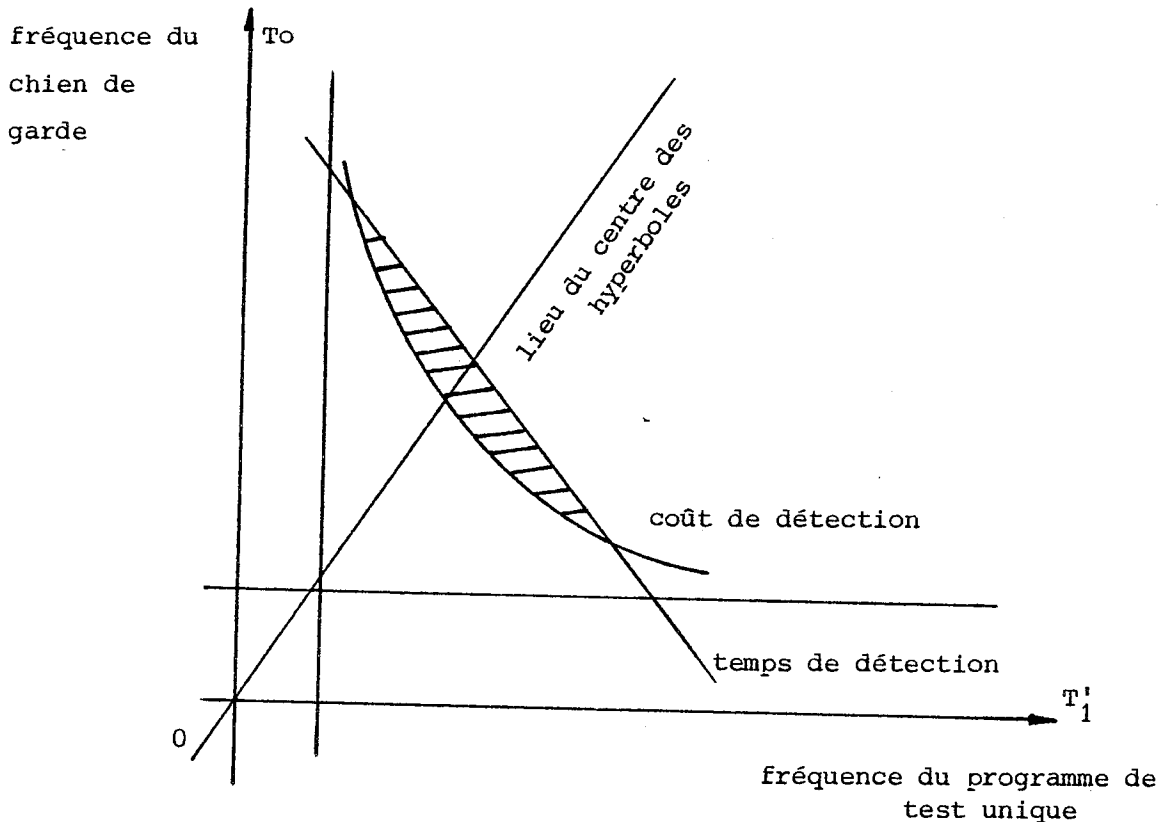


Figure 2. - Problème II : Temps de détection imposé.

Quand le nombre de paramètres est supérieur à 2, les droites et les hyperboles correspondent évidemment à des hyperplans et des quadriques de types hyperboloïde. La solution correspondante est donc donnée par l'hyperplan, de temps de détection, tangent à la quadrique de type hyperboloïde, des coûts.

V.2. Ordre optimal des procédures de test

On retombe ainsi dans le cadre des calculs développés dans [COU 82-2].

On va considérer que l'application s'exécute, et que selon la période T'_1 définie précédemment, la totalité des procédures de test s'exécute en une seule fois.

Si dans un premier temps on suppose que la latence des pannes est nulle et qu'une panne se produise pendant la phase d'application, l'application va produire une erreur, cette erreur sera détectée pendant la phase de test suivante. Dans ce cas, l'ordre des différentes procédures importe peu. Donc nous nous intéresserons au cas où les pannes se produisent durant l'exécution du programme de test, dans le cas d'une latence nulle, ou dans le cas où les pannes produisent des erreurs durant la phase d'exécution du programme de test.

Deux cas peuvent se produire :

- Soit la panne est détectée avant la prochaine phase d'application et dans ce cas aucun résultat erroné ne sera produit.
- Soit la panne n'est pas détectée avant le début de la phase d'application, elle sera détectée à la prochaine phase de test, mais va provoquer la sortie d'erreurs durant la phase d'application.

On considérera ainsi qu'une panne de la partie i , ne peut être détectée que si elle survient (latence nulle) ou si elle provoque une erreur (latence non nulle) avant le début de l'exécution de la procédure testant la partie i .

VI.2.1. Rappel des hypothèses

La distribution des pannes est considérée exponentielle, de plus, on suppose que 2 pannes ne peuvent survenir simultanément, la seconde se produira une fois la première détectée. Toutes les procédures de test sont regroupées en un seul programme de test correspondant à l'exécution consécutive des différentes procédures de test.

Ce programme de test est relancé avec une période T'_1 , son nombre d'instructions est égal à la somme des nombres d'instructions de chaque procédure.

$$\text{Soit } t'_1 = \sum_{i=1}^n t_i$$

VI.2.2. Ordre optimal pour une latence nulle

Alors si a_i , $i = 1$ à n désigne la probabilité d'occurrence d'une panne sur la partie i du microprocesseur, sachant qu'une panne survient dans le microprocesseur.

L'ordre optimal d'ordonnement des procédures dans le programme de test est donné par :

$$t_1, t_2, t_3, \dots, t_n$$
$$\text{pour } \frac{a_1}{t_1} < \frac{a_2}{t_2} < \frac{a_3}{t_3} \dots < \frac{a_n}{t_n}$$

Soit, si toutes les parties du microprocesseur ont la même probabilité d'occurrence d'une panne, alors les procédures sont placées dans l'ordre décroissant de la taille (temps d'exécution).

Ou encore, si toutes les procédures de test ont la même taille (temps

d'exécution) alors les procédures sont placées dans l'ordre croissant de la probabilité d'occurrence des pannes sur chaque partie.

VI.2.3. Ordre optimal pour latence non nulle

Dans ce cas, si a_i ($i = 1$ à n) désigne la probabilité d'occurrence d'une panne sur la partie i du microprocesseur, sachant qu'une panne survient dans ce microprocesseur, si EL_i désigne l'espérance de la latence de la panne i , alors l'ordre optimal du test est toujours donné par :

$$t_1, t_2, t_3, \dots, t_n$$

pour $\frac{a_1}{t_1} < \frac{a_2}{t_2} < \frac{a_3}{t_3} \dots < \frac{a_n}{t_n}$

En effet, soit A et B deux séquences distinctes par l'ordre des procédures t_i et t_j , comme le montre la figure 1 :

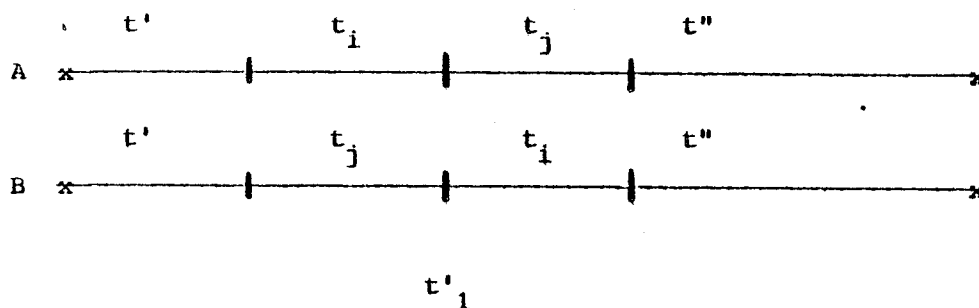


Figure 1. - Séquences de test A et B.

La probabilité de détecter une panne pendant la séquence A et la séquence B sont données respectivement par :

$$Pr[A] = a_i \frac{t' - EL_i}{t'_1} + a_j \frac{t' + t_i - EL_j}{t'_1}$$

$$\Pr[B] = a_j \frac{t' - EL_j}{t'_1} + a_i \frac{t'+t_j - EL_i}{t'_1}$$

donc A est meilleure que B si $\Pr[A] > \Pr[B]$, il s'en suit :

$$a_j t_i > a_i t_j.$$

soit $\frac{a_i}{t_i} < \frac{a_j}{t_j}$

On obtient donc le même résultat.

Donc, l'ordre optimal dépend uniquement de la probabilité d'occurrence de panne et de la longueur du programme de test respectivement à chaque partie du microprocesseur.

VI. CONCLUSION

Ce chapitre apporte donc une solution à la fréquence optimale recherchée lorsque les différentes procédures de test sont lancées avec leur période propre, ainsi que dans le particulier, usuel, où toutes ces procédures sont regroupées en un seul programme de test.

Dans ce cas, seule l'exécution non interruptible du programme de test est considérée, car dans le cas contraire, le problème devient beaucoup plus complexe, les résultats pouvant être trouvés dans [COU 82-2], et ce en considérant soit pour un coût minimum lorsque le temps de détection est imposé, ou pour un temps de détection minimum lorsque le coût est imposé. Le coût considéré correspond à la durée d'exécution des procédures de test, exprimée en nombre d'instructions.

REFERENCES

[COU 82-1]

COURTOIS B.

Performance modeling of partially self checking systems.

F*TCS 12. Santa Monica, June 82.

[COU 82-2]

COURTOIS B.

Ordering the procedures of a periodic test.

Rapport de Recherche IMAG, RR n° 312, September 82.

CHAPITRE IX
CIRCUITS
D'ANALYSE DE SIGNATURES
MACHINE AUTOMATIQUE DE TEST
(Hors Méthodologie)

CHAPITRE IX - CIRCUITS D'ANALYSE DE SIGNATURES.

MACHINE AUTOMATIQUE DE TEST

(Hors Méthodologie)

I	- INTRODUCTION	533
II	- TEST HORS-LIGNE'	534
III	- ANALYSE DE SIGNATURES	536
IV	- DESCRIPTION DES CIRCUITS D'ANALYSE DE SIGNATURES	538
	IV.1. Le circuit multiprojet	538
	IV.2. L'analyseur de signatures	538
	IV.2.1. Le CMP82 AS1	539
	IV.2.2. Le CMP83 AS1	541
V	- TEST DE FIN DE FABRICATION. TEST DE CONCEPTION	546
	V.1. Test visuel	546
	V.2. Test par contraste de potentiel	547
	V.3. Test à la machine à pointe	547
VI	- CONCLUSION	549
	REFERENCES	551

I. INTRODUCTION

Ce chapitre diffère de tous les autres en ce sens qu'il traite du test hors-ligne, en dehors donc de toute application. Il présente succinctement deux circuits conçus en vue de leur utilisation dans un équipement automatique de test. Tout d'abord le principe de cet équipement de test est présenté, puis l'évaluation de la probabilité de pannes est discutée, en effet, selon les hypothèses de pannes retenues, on obtient des écarts gigantesques entre les différents résultats.

Puis une présentation des cellules de base des circuits est donnée. Enfin, le test de fabrication et de vérification de conception est introduit, tel qu'il a été fait pour l'un des circuits d'analyse de signatures, le second étant en cours de fabrication.

II. TEST HORS-LIGNE

Le test envisagé est un test hors-ligne ex-situ, comme il a été présenté au chapitre I : le microprocesseur, hors de son site de fonctionnement, exécute les programmes spécifiques de test, déterminés pour le MC 68000 à partir des chapitres VI et VII. Il n'y a pas de problème de fréquence de relance, puisque le microprocesseur testé ne fait plus qu'exécuter les programmes de test. Le matériel nécessité par cet équipement automatique de test devient alors très restreint. Par exemple pour le MC 68000.

- Une dizaine de K-mots de ROM,
- le circuit réalisant le chien de garde,
- les circuits d'analyse de signatures, décrits ici (3 à 23 selon le polynôme choisi),

Sont nécessaires en plus du microprocesseur lui-même et de son horloge. La figure 1 présente le principe de cette machine de test.

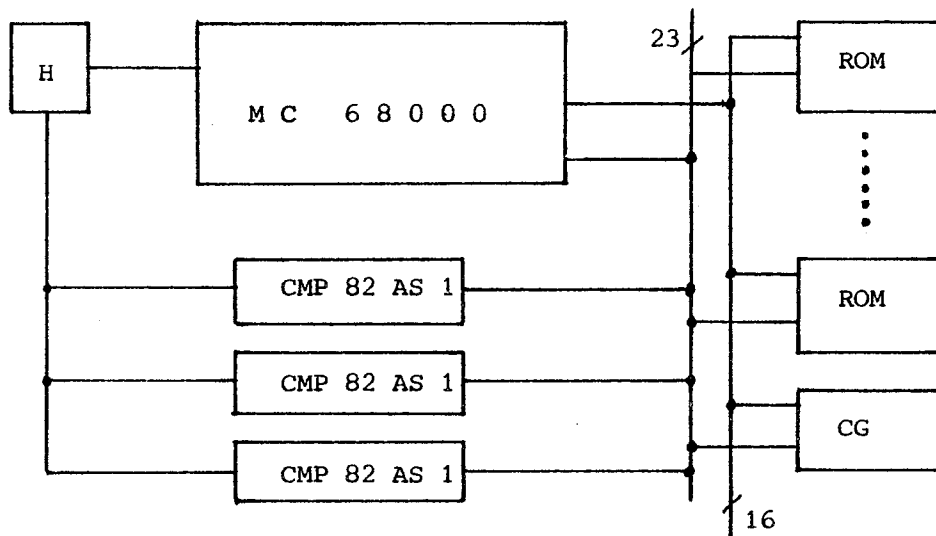


Figure 1. - Principe de la machine de test.

L'idée de réaliser un équipement automatique de test, est de COURTOIS dans COU 81 . En effet, pour des circuits tels que des mémoires ou des circuits périphériques un testeur conventionnel est nécessaire. Un microprocesseur, par contre, est suffisamment "intelligent" pour détecter lui-même ses pannes. C'est la raison pour laquelle l'équipement de test obtenu est si "léger".

Le principe de cette machine est très simple. Pour le MC 68000, par exemple, le détecteur de codes opérations invalides (interne), le détecteur d'adresse erreur (interne), le chien de garde (externe), permettent de s'assurer que les flots sur le bus données et le bus adresses transitent correctement aussi bien vers, que depuis le microprocesseur, en vérifiant qu'aucune panne ne survienne à la périphérie de la pastille du CPU, et ce moyennant un certain temps de latence tel qu'il a été présenté au chapitre III. Ces mêmes détecteurs permettent de s'assurer que les pannes de grandes fréquences affectant le compteur ordinal (Evenement E) sont aussi détectées (chapitre V).

Le séquençement d'une instruction à la suivante s'effectuant correctement, on peut alors exécuter les programmes spécifiques de test contenus dans les ROMs. Ces programmes spécifiques correspondent à la concaténation des procédures de test à mettre en oeuvre pour le test analytique de la partie opérative (chapitre VI), et pour le test fonctionnel de la partie contrôle (chapitre VII). Malgré cela, le problème des pannes de déséquençement de faible fréquence reste posé. C'est la raison pour laquelle ces circuits ont été développés. En effet, les flots de "données" transitant sur le bus adresses sont tellement énormes qu'une compaction s'impose. L'analyse de signatures est un moyen de remédier à ceci.

III. ANALYSE DE SIGNATURES

L'utilisation de l'analyse de signatures pour la compaction de flots de données est faite depuis longtemps, elle est particulièrement adéquate pour les systèmes à base de microprocesseurs [FRO 77], [NAD 78], [GOR 77]. Une nouvelle utilisation, avec entrées séries et parallèles est faite dans [KON 79], et a donné naissance à un autre circuit [BOR 82]. Le choix fait ici, est basé sur l'analyse hexadécimale utilisant le polynôme $X^{16} + 1$ en référence aux travaux de DAVID [DAV 80]. La possibilité de rajouter les autres polynômes ($X^{32} + 1$, $X^{64} + 1$, $X^{128} + 1$) se faisant à "peu de frais" (ajout d'une cinquantaine de transistors et deux plots pour les commandes de choix du polynôme et de sélections d'entrées/rebouclages), celle-ci a été adoptée.

Un point important est de connaître la probabilité d'omission d'une panne, c'est-à-dire la probabilité de déclarer bon un microprocesseur mauvais, dans le cas du test hors-ligne considéré ici. Cette probabilité peut grandement varier selon le type d'erreurs pris en compte.

Le tableau I donne les valeurs respectives des probabilités d'omissions de pannes, en utilisant les résultats de [SMI 80].

Polynôme	Erreurs simples	Erreurs multiples		
		Indépendantes	Multiplicité 8	Multiplicité 16
$X^{16} + 1$	0	1.5×10^{-5}	0.25	0.5
$X^{32} + 1$	0	2.3×10^{-10}	0.625	0.5
$X^{64} + 1$	0	5.4×10^{-20}	3.9×10^{-3}	6.2×10^{-2}
$X^{128} + 1$	0	2.9×10^{-39}	1.5×10^{-5}	3.9×10^{-3}

Tableau I. Probabilités d'omission des pannes en fonction de leur type

Il est clair que l'analyse de signatures proposée est très efficace pour des pannes équiprobables indépendantes [DAV 80] : ainsi, si l'on considère l'utilisation du polynôme $X^{128} + 1$ analysant les adresses d'instructions transitant sur le bus adresses du MC 68000, la probabilité d'omission de telles pannes multiples, serait d'une omission tous les 10^{38} microprocesseurs testés, c'est-à-dire d'une omission toutes les 10^{25} millions d'années, en supposant que le test d'un microprocesseur dure 1 seconde.

Mais, si l'on considère des pannes de multiplicité 8 [SMI 80] c'est-à-dire des pannes pouvant survenir à des intervalles multiples de 8, toujours pour l'application du même polynôme, on obtient une panne non détectée tous les 6.7×10^4 microprocesseurs testés : c'est-à-dire une omission toutes les 18 heures, en supposant toujours que le test d'un microprocesseur dure une seconde.

Ceci constitue donc un écart gigantesque entre les deux hypothèses. Des précautions sont néanmoins à prendre vis-à-vis de ces deux résultats. En effet, la considération d'erreurs répétitives tous les 8 bits a été faite par les auteurs dans le cas d'analyse de bus données, et particulièrement pour les sorties séries de microprocesseurs. Or, dans notre cas particulier où ne sont considérées que les adresses des instructions transitant sur le bus adresses, les 23 bits en parallèles, des pannes de multiplicité 8 simultanément sur tous les bits seraient pour le moins étonnantes ; malgré tout, si ce genre de pannes se produisait, on aurait alors à faire à des pannes de très grandes fréquences et donc détectées par d'autres moyens (détecteur de code-op invalide, détecteur d'adresse erreur et chien de garde) et dans ce cas, leur non détection par l'analyse de signatures n'aurait pas de conséquence. D'autre part, vis-à-vis des pannes équiprobables indépendantes, il est très difficile de dire que les conséquences de déséquilibrages du compteur ordinal vont se traduire par la création de pannes indépendantes.

IV. DESCRIPTION DES CIRCUITS D'ANALYSE DE SIGNATURES

IV.1. Le circuit multiprojet :

L'idée du circuit multiprojet [ANC 81] est de fabriquer collectivement plusieurs projets de très petites séries, pour l'enseignement ou pour la recherche, sur un ensemble de tranches de silicium, puis de les séparer et de les redistribuer afin d'en partager les frais de production. Chaque "wafer" comporte ainsi environ 6 "puces" différentes, chacune d'entre elles est répétée environ 20 fois. Chaque puce comporte de 1 à 10 projets, selon la taille de chaque projet.

Les circuits d'analyse de signatures présentés ici, ont été conçus à l'aide du langage universitaire de conception de circuits intégrés pour l'enseignement, [GUY 81]. Ce langage permet la description et la manipulation de masques de circuits intégrés. La description a été faite à partir de règles de dessin "en lambda" similaires à celles de [MEA 80].

IV.2. L'analyseur de signatures :

Le circuit est construit autour de 8 registres à décalage de 16 bits. Chaque entrée sur un registre correspond à un bit. Le nombre de registres (8) a été choisi du fait des largeurs des bus adresses de microprocesseurs qui sont en général multiples de 8 :

- 8048	8 bits
- M6800-Z80-8080	16 bits
- MC 68000	23 bits
- IAPx432	40 bits.

On utilisera alors, selon la largeur du bus, 1, 2, 3 ou 5 circuits d'analyse

pour le polynôme de base $x^{16} + 1$.

Deux commandes permettent en plus de faire un choix parmi 4 polynômes ; ces polynômes étant :

- . $x^{16} + 1$ → 8 bits d'entrées parallèles/8 bits de sorties parallèles
- . $x^{32} + 1$ → 4 bits d'entrées parallèles/4 bits de sorties parallèles
- . $x^{64} + 1$ → 2 bits d'entrées parallèles/2 bits de sorties parallèles
- . $x^{128} + 1$ → 1 bit d'entrée unique/1 bit de sortie unique.

IV.2.1. Le CMP 82 ASI :

Le circuit multiprojet 82 ASI a été conçu en 1982 pour une taille du "lambda" de 3 microns. C'est un circuit de 23 broches. Il comporte un peu plus de 1500 transistors, une seule alimentation unique de 5 volts est nécessaire . Il génère lui-même, à partir du signal d'horloge, deux phases sans recouvrement. Il est entièrement dynamique.

La figure 1 présente le plan de masse de ce circuit.

La cellule dynamique de base est une cellule classique de point de mémoire dynamique utilisant deux horloges sans recouvrement entre deux inverseurs. L'entrée se fait depuis l'entrée extérieure au point mémoire lorsque la commande ϕ_1 .Load est valide, le rebouclage se fait lorsque ϕ_1 .Mem est valide. Le premier inverseur a été transformé en une porte NAND dont la seconde entrée est le signal ϕ_1 .Reset permettant une remise à zéro simultanée de tous les points mémoire de chaque registre.

La figure 2 donne le schéma logique de cette cellule.

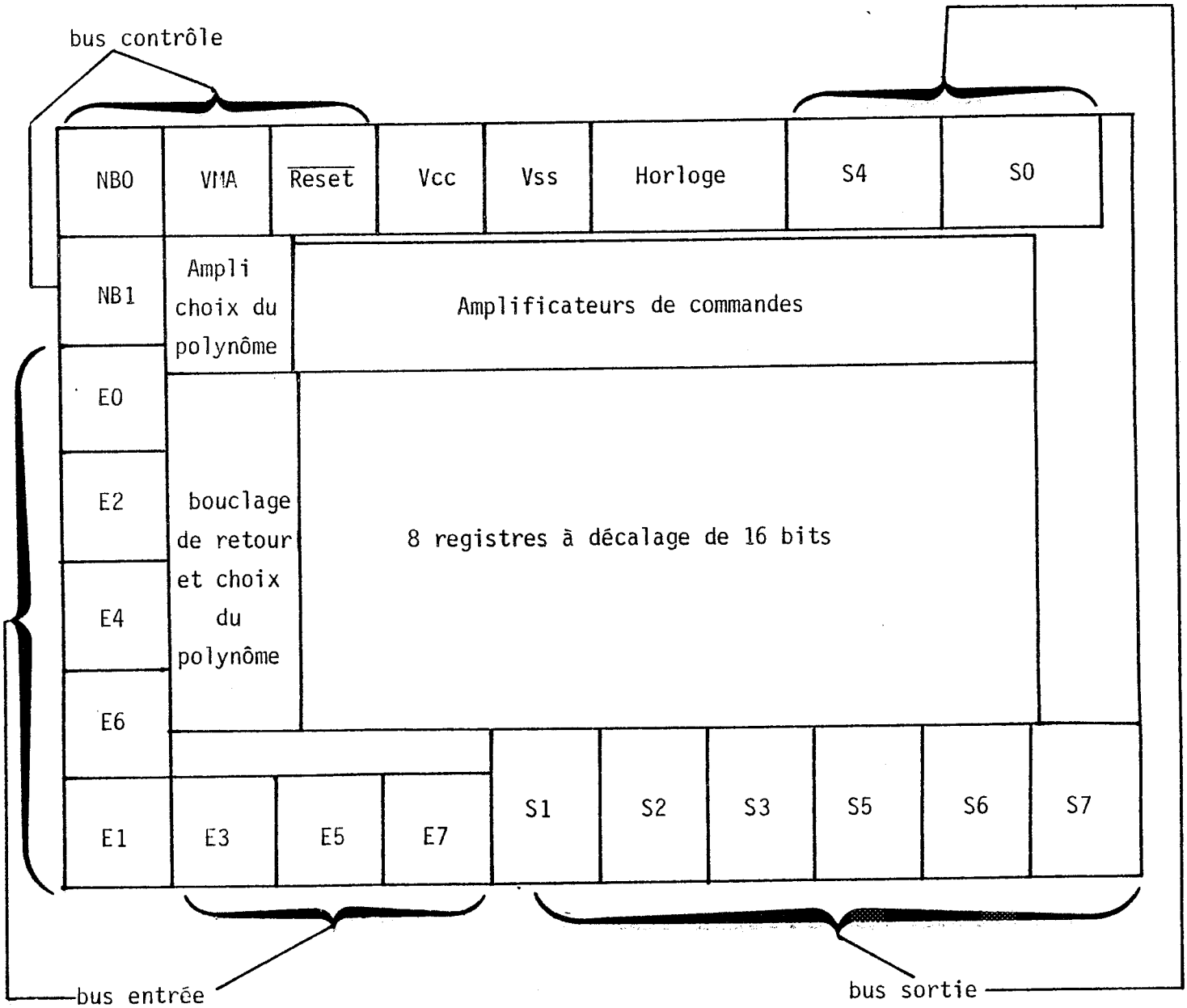


Figure 1. Plan de masse du CMP 82 AS 1.

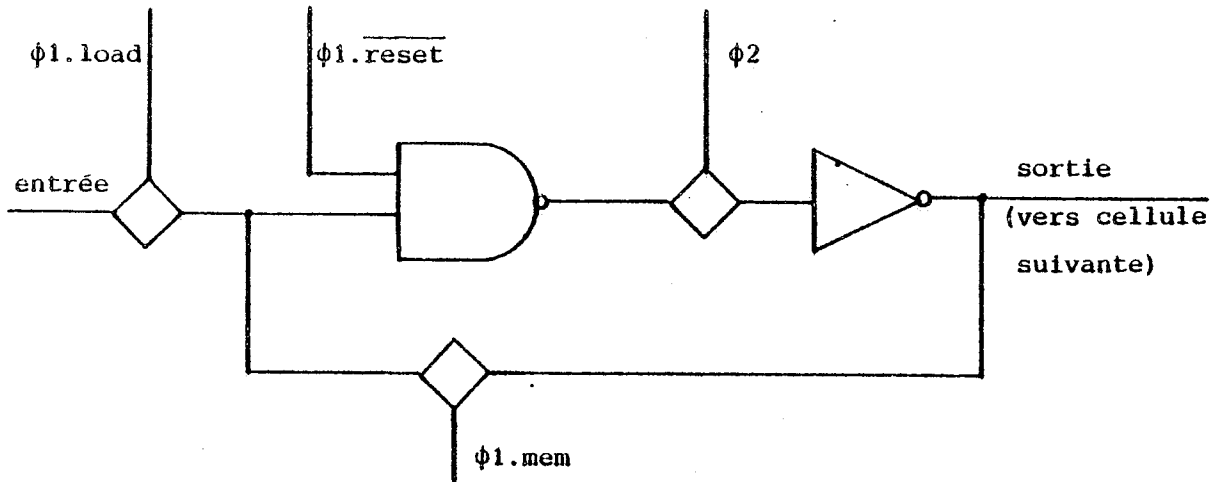


Figure 2. - Cellule dynamique de base.

La figure 3 présente les masques composants le circuit CMP82 AS1.

IV.2.2. Le CMP83 AS1 :

Le circuit multiprojet 83 AS1 a été conçu en 1983, pour une taille du "lambda" de 1,75 microns. Il est en cours de fabrication. C'est un circuit de 25 broches. Il comporte un peu plus de 2000 transistors, une seule alimentation unique de 5 volts est nécessaire. Il est entièrement statique et ne nécessite ainsi aucun système d'horlogerie.

La figure 4 présente le plan de masse de ce circuit.

Un nouvel outil a été utilisé pour sa conception, cet outil, LUBRICK, [SCH 83] permet la connexion automatique et le routage des liens.

La cellule statique de base est une cellule classique maître esclave. L'entrée se fait depuis l'entrée extérieure vers la bascule maître lorsque la commande load est valide, la bascule esclave fournit l'entrée de la bascule maître suivante. Puis le signal load redescend. L'état est mémorisé dans toutes les bascules maîtres du registre. La commande Mem

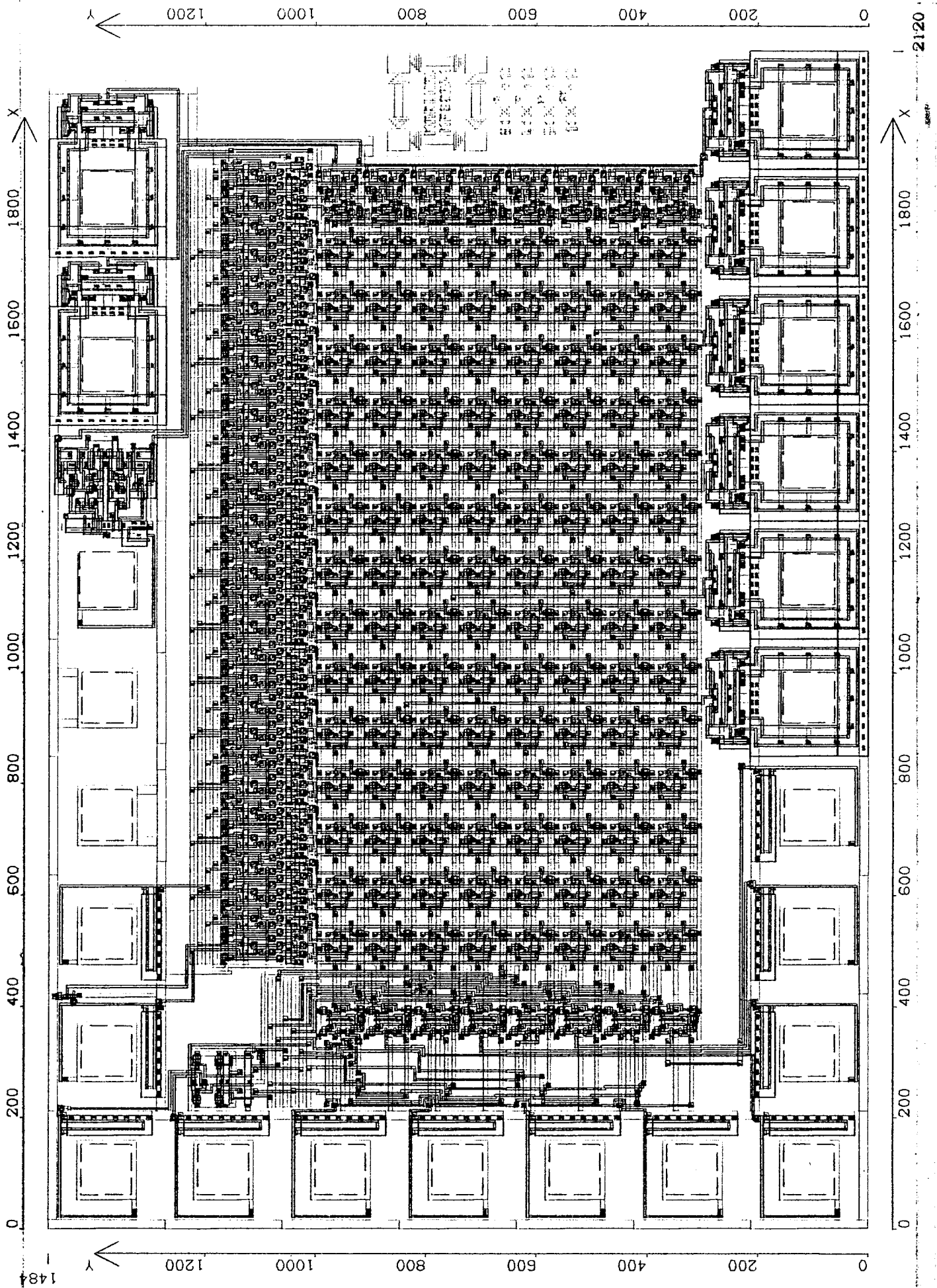


Figure 8 - Masques composant le CMP 82 AS 1.

permet le chargement des bascules esclaves à partir des bascules maîtres. Un ET, câblé à l'entrée de chaque bascule, commandé par le signal Reset permet la remise à zéro simultanée de toutes les bascules de chaque registre .

La figure 5 présente le schéma logique de cette cellule.

On peut remarquer, que ce circuit possède un plot appelé 'TEST' permettant de séparer ce circuit en deux, d'un côté le choix du polynôme de l'autre les registres à décalage. De plus chaque point mémoire possède 2 petits plots de test en aluminium.

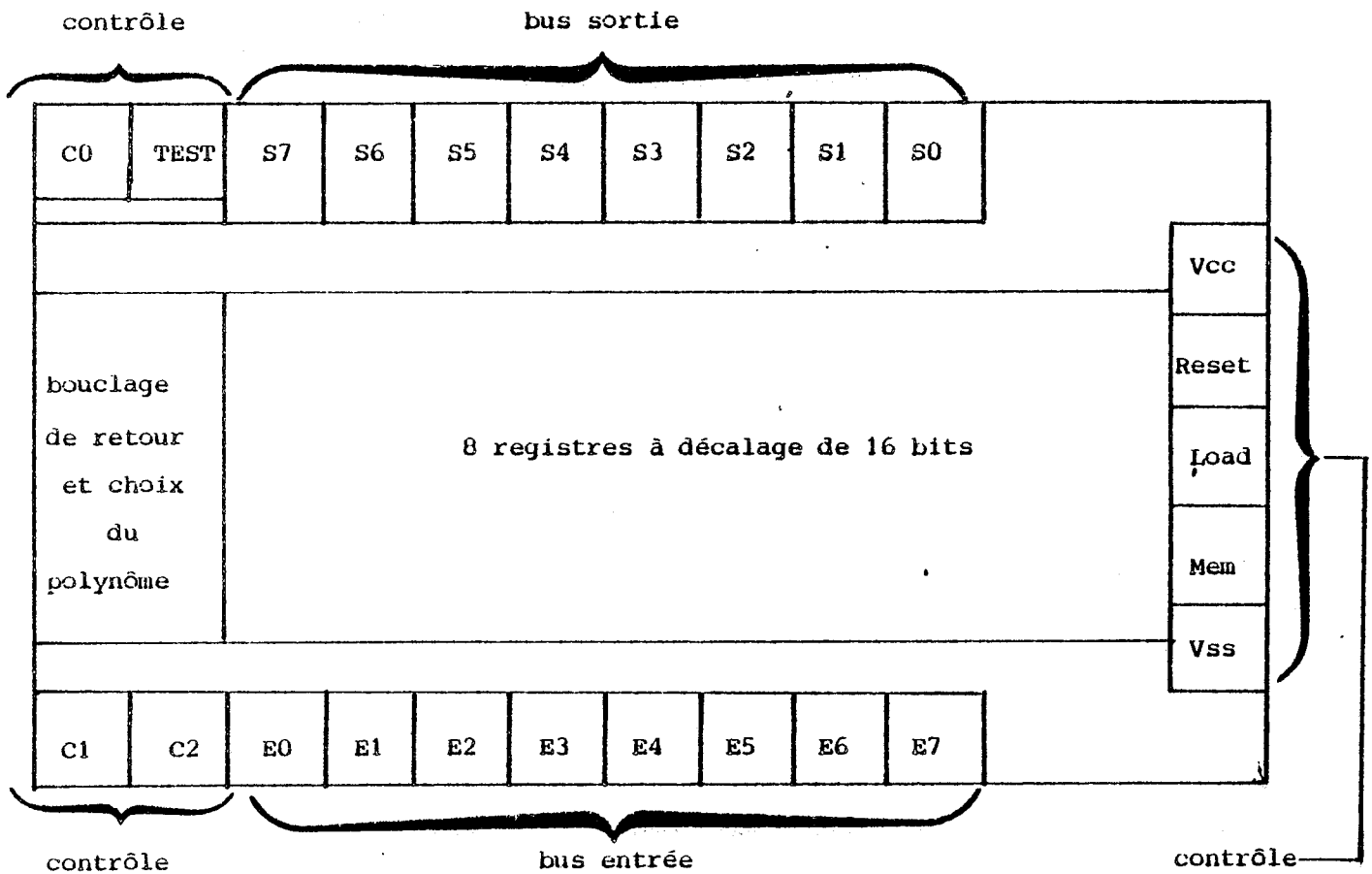


Figure 4. - Plan de masse du CMP83 AS1.

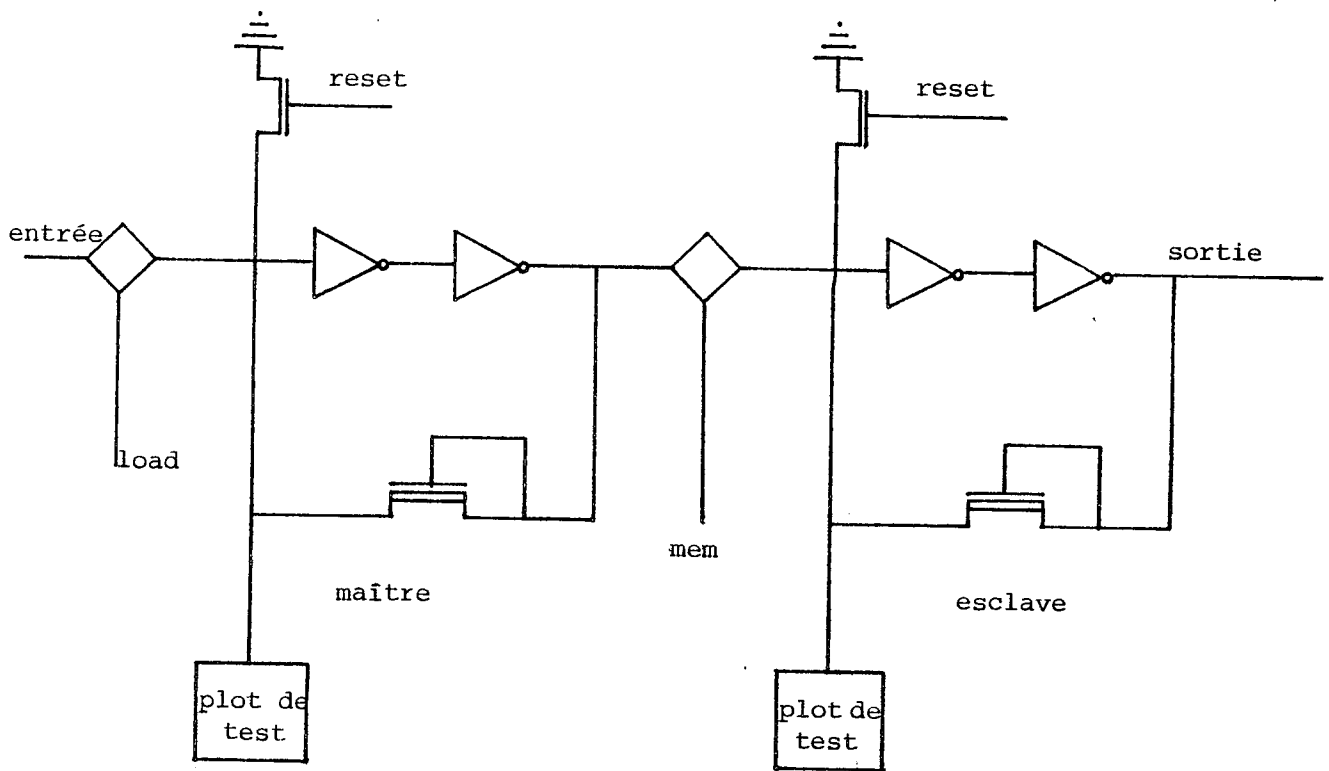
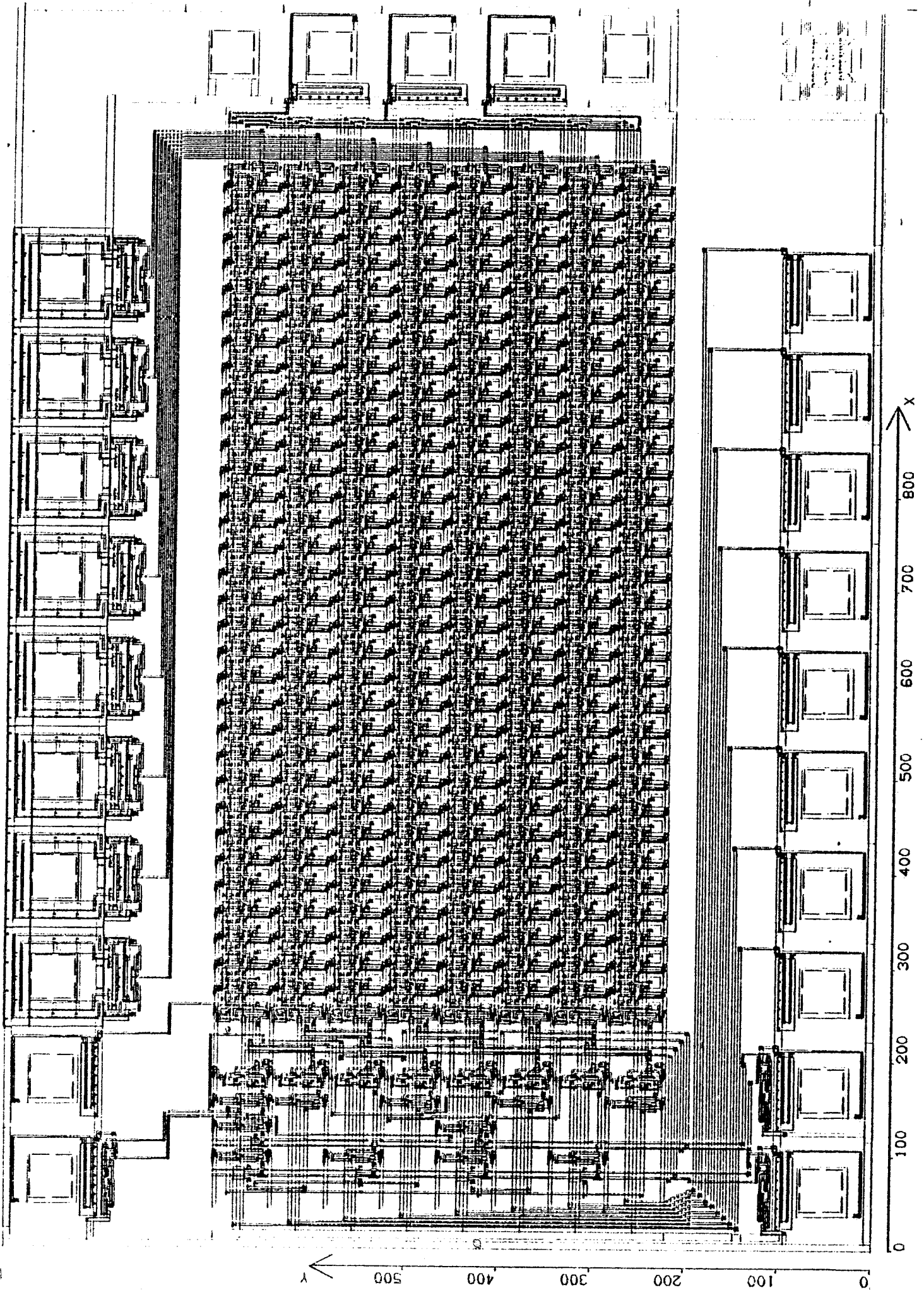


Figure 5. - Cellule maître esclave de base.

La figure 6 présente les masques composant le circuit CMP83 AS1.



V. TEST DE FIN DE FABRICATION - TEST DE CONCEPTION

Le CMP82 AS1 a été testé à la sortie de la chaîne de fabrication. Plusieurs tests ce sont succédés à ce niveau, pour vérifier à la fois la bonne exécution du procédé de fabrication et la bonne conception. Plusieurs étapes ont été nécessaires.

V.1. Test visuel :

Un test visuel, a été effectué avant la mise en boîtier et le brochage du boîtier. Ce test visuel exécuté sous microscope a permis de détecter certains défauts, tels que durée d'exposition en diffusion trop court (?) ou saleté sur le masque (?) ceci se traduit par le fait que les lignes de diffusion se rétrécissent plus ne forment plus que quelques "gouttes" de diffusion et de plus en plus petites jusqu'à une absence totale de diffusion, comme le montre la figure 7.

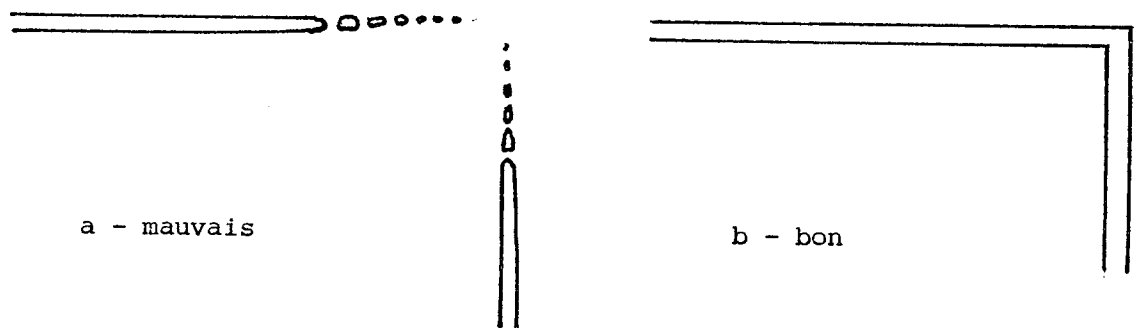


Figure 7. - Résultat d'un mauvais procédé de fabrication pour la diffusion.

De plus, sur certains circuits, des formations de cristallisation de surface sont apparues.

Un outil automatique a été développé afin d'effectuer ce test visuel [ZAR 82] ,pour le moment il ne fonctionne que sur les masques des circuits. Il effectue le suivi de contour, la vérification des gardes etc....

V.2. Test par contraste de potentiel :

Les circuits qui se sont révélés bons au test visuel, ont été découpés puis mis en boîtier et brochés. Ils ont subi alors un test en contraste de potentiel sur microscope électronique à balayage [BAI 82-1] [BAI 82-2] [COU 82].

Malheureusement, ne disposant pas pour le moment sur le microscope électronique d'un "beam blanking" permettant de faire de la stroboscopie, il a été assez difficile de trouver la cause du mauvais fonctionnement du circuit. En effet, s'il était possible de suivre le fonctionnement sur écran, le temps nécessaire pour prendre une photo est si long que le circuit se "charge" et les contrastes ne sont plus nets, seul l'aluminium renvoie un bon contraste, les diffusions et Sipoly se chargent vite et le niveau est difficilement prévisible.

Les amplificateurs de commande, à précharge commandé, les signaux d'horloge, marchaient très bien, mais au niveau du choix du polynôme, les entrées ne "passaient" pas.

Il n'a donc pas été possible de conclure quelle était la panne exacte, si ce n'est qu'elle était localisée au niveau du choix du polynôme.

V.3. Test à la machine à pointes :

Le circuit a ensuite été testé à la machine à pointes. Plusieurs étapes ont

été nécessaires :

- vérification de l'alimentation sur tout le circuit = correct
- vérification de la masse sur tout le circuit = correct
- vérification de l'horloge sur tout le circuit = correct
- vérification du fonctionnement des amplis = correct
- vérification du fonctionnement du choix du polynôme = incorrect.

En effet, les transistors de ce circuit ont été mal dimensionnés et la tension de sortie vaut 3 volts sur certaines lignes (si poly) au lieu des 5 volts attendus, comme ces lignes commandent les MOS de transferts à l'entrée du registre à décalage, aucun signal ne passait (dégradation successive à chaque porte de transfert).

C'est à cause de toutes ces raisons que le circuit réalisé cette année :

- est un circuit entièrement statique, qui pourra ainsi être entièrement testé au microscope électronique à balayage.
- est pourvu d'une ligne de test permettant de séparer le choix du polynôme de l'entrée des registres à décalages.
- est pourvu de 256 "plots de test" internes, qui sont de petits plots en aluminium, permettant à la fois de connaître le niveau en contraste de potentiel (le niveau de l'Alu est le meilleur) et de poser des pointes pour prendre des mesures ou forcer des valeurs avec la machine à pointes.

VI. CONCLUSION

Ce circuit d'analyse de signatures, conçu en premier lieu pour le test hors-ligne de microprocesseur, est en fait d'usage général. L'étude et la conception de ce circuit, ont été développées en vue de son utilisation dans un équipement automatique de test (Automatic Test Equipment), afin de détecter les pannes de faibles fréquences atteignant le compteur ordinal. Il apportera en plus des procédures développées pour le test en-ligne un gain de détection, puisqu'on ne peut pas garantir la détection des pannes de faibles fréquences.

En effet, que ce soit pour les pannes de la périphérie de la pastille, ou pour les pannes de grandes fréquences sur le compteur ordinal, d'autres détecteurs sont utilisés. La réalisation de ce circuit pouvait être faite, pour le moment, par l'utilisation de boîtiers MSI-LSI, mais nécessitait alors entre 10 et 20 boîtiers. La réalisation du circuit (statique ou dynamique) pourrait être envisagée sur une ligne industrielle.

Deux remarques sont à faire vis-à-vis de l'analyse de signatures elle-même et des polynômes choisis.

Il serait intéressant d'approfondir la connaissance du type d'erreurs générées par des déséquences du compteur ordinal dans le cas particulier considéré, c'est-à-dire par exemple, la réception en entrée de l'analyseur de signatures, au lieu des n bits corrects attendus, de $n+b_1$ bits ou de $n-b_2$ bits (où b_1 et b_2 sont les nombres de bits rajoutés ou supprimés dans la suite de bits corrects, par le déséquence du compteur ordinal), ou bien, la réception d'un flot composé de segments de bits corrects, mais dont l'ordre des segments a été modifié.

Il serait aussi intéressant de rechercher l'existence de polynômes maximisant la probabilité de détection de ce type d'erreurs particulier,

moyennant quoi, il serait possible d'approfondir avec certitude la probabilité de détection obtenue par l'utilisation de l'analyse de signatures dans cette application particulière.

REFERENCES

- [ANC 81] ANCEAU F.
"Le projet MPC français",
Journées CAO et VLSI, Université Paris 6 Juin 1981.
- [BAI 82-1] BAILLE G., LAURENT J.
"Présentation d'un outil d'aide à la mise au point de
circuits intégrés VLSI prototypes".
Rapport de recherche IMAG, RR n° 298, Mars 1982.
- [BAI 82-2] BAILLE G., COURTOIS B., LAURENT J.
"Analyse de défaillance de VLSI par microscopie élec-
tronique".
Rapport de recherche IMAG, RR n° 304, Juin 1982.
- [BOR 82] BORRIONE D., CAUBET B., COURTOIS B., GUYOT A., HUMBERT M.,
RONALD E.
"Un circuit intégré de génération et d'analyse de vecteurs
pour le test de cartes à microprocesseur".
Rapport IMAG, RR n° 293, Mars 1982.
- [COU 81] COURTOIS B.
"Debugging VLSI using SEM".
Rapport de recherche IMAG, RR n° 310, sept. 82.
- [DAV 80] DAVID R.
"Testing by feedback shift register".
IEE Trans. on Computer, vol C29 n° 7, july 1980.
- [PRO 77] FROHWERK R.A.
"Signature analysis : a new digital field service method".
Helwett Packard Journal, May 1977.

- [GOR 77] GORDON G. and NADIG H.
"Hexadecimal signatures identify troublespots in microprocessor systems".
Electronics, March 1977.
- [GUY 81] GUYOT A., JERRAYA A., RAYMOND J.
"Présentation du système LUCIE".
Disponible IMAG.
- [KON 79] KONEMANN B., MUCHA J., ZWIEHOFF G.
"Built-in test for complex digital integrated circuits".
ESSIRC 79.
- [MEA 80] MEAD C.A., CONWAY L.A.
"Introduction to VLSI systems".
Reading : Addison-Wesley.
- [NAD 79] NADIG H.
"Testing a microprocessor product using a signature analysis".
Cherry Hill Test Conference, November 1978.
- [SCH 83] SCHOELLKOPF J.P.
"LUBRICK : A silicon assembler and its application to data-path design for fisc".
Rapport de Recherche IMAG, RR n°. 363, Mars 1983.
- [SMI 80] SMITH J.E.
"Measures of the effectiveness of fault signature analysis".
IEEE Trans. on Computer, Vol. 29, n°. 6, 1980.
- [ZAR 82] ZARATE V.H., MARTINEZ J.M.
"Analyse Automatique de défaillances dans les circuits intégrés à l'aide du système ROMUALD d'Analyse d'image".
Rapport de projet d'ingénieur ENSIMAG, INPG, Juin 1982.

CHAPITRE X

CONCLUSION

CHAPITRE X - CONCLUSION

I - HYPOTHESES DE PANNES	558
II - MODELES UTILISES	561
III - RESULTATS OBTENUS	564
IV - TEST : TEMPS ET COMPLEXITE	568
V - AMELIORATIONS DE LA MACHINE	570
REFERENCES	573

Ce chapitre regroupe les conclusions concernant à la fois les hypothèses, modèles de pannes et résultats correspondant à chaque chapitre. Une comparaison est faite entre l'étude du M 6800 et celle du MC 68000, précisant les points communs et les points divergents,

Chaque paragraphe se terminant pas une conclusion sur ce que deviendrait cette étude s'il fallait l'appliquer à d'autres microprocesseurs.

Puis, le problème du temps de test et de l'impact de la complexité de la machine sur la complexité du test est abordé, les notions de complexité quantitative et de complexité qualitative sont rappelées.

Enfin, une gradation des diverses améliorations possibles de la machine est donnée, par le biais des résultats obtenus au cours de cette étude.

I. HYPOTHESES DE PANNES

Les hypothèses retenues pour les pannes de la périphérie de la pastille (étape i) sont les mêmes pour le MC 68000 que celles retenues pour M 6800. Elles se traduisent par des collages simples ou multiples des broches des bus données et adresses, les pannes des autres broches étant des cas d'espèces, dépendant de l'application particulière, telles que interruptions, interface M 6800 etc... Si la même étude est envisagée pour d'autres microprocesseurs on pourra ajouter d'autres hypothèses de pannes telles que les courts-circuits inter-broches, comme étudiés dans [STA 77].

Les hypothèses retenues pour l'évaluation de la fréquence de l'évènement E et de la latence d'erreur (étape ii) sont les mêmes que celles retenues pour le M 6800. Ces hypothèses sont valables quel que soit le microprocesseur pris en compte, elles pourront donc être utilisées directement si la méthodologie est appliquée à d'autres microprocesseurs.

Les hypothèses retenues pour les pannes du coeur créant E (étapes iii et iv) sont tout d'abord la génération d'un évènement E unique conduisant au modèle statique, puis le modèle dynamique est construit à partir des résultats obtenus avec le modèle statique, l'hypothèses étant la génération de E avec une certaine fréquence, telle que déterminée à l'étape précédente (étape ii). Ce sont les mêmes hypothèses que pour le M 6800, elles sont suffisamment globales pour être utilisées pour tout microprocesseur nécessitant cette étude.

Pour le coeur ne créant pas E (étape v) l'étude est divisée en deux parties représentant respectivement la partie opérative et la partie contrôle.

Pour la partie opérative, les trois classes d'hypothèses de pannes définies

pour le M 6800 sont conservées ; ces trois classes correspondent à trois qualités de test, en partant du défaut simple physique pour aboutir aux défauts multiples tels que courts-circuits etc... La classe de test correspondant à la partie opérative complète, est donnée par la plus faible classe utilisée. Ainsi, pour certains opérateurs une classe de forte qualité, obtenue par le biais d'un test exhaustif, est dérivée car l'obtention de cette classe est plus facile que pour une classe de qualité plus faible du même opérateur. Pour la partie contrôle, les hypothèses globales de mauvais séquençement des commandes ont été conservées. C'est ainsi les commandes, et non la façon de les générer, qui est observée. On remarque que des hypothèses particulières ont dû être développées pour les pannes de la gestion des transferts, le fonctionnement des bus différentiels du MC 68000 en présence de pannes conduisant à des conséquences particulières. Cette remarque sera reprise au niveau de la complexité du test (cf § IV).

Ainsi, si cette étude est nécessaire pour d'autres microprocesseurs, si les bus internes sont unifilaires, alors les hypothèses retenues pour le M 6800 seront intégralement adoptées, tandis que si les bus internes sont différentiels, alors les hypothèses retenues dans cette étude devront être envisagées.

Les hypothèses de panne retenues pour l'évaluation de la fréquence optimale de relance des procédures de test périodiques (étape vi) sont les mêmes que celles retenues pour le M 6800, puisque les résultats de [COU 82] sont repris intégralement. Ce sont les mêmes hypothèses qui sont reprises pour évaluer l'ordonnancement optimal des procédures les unes par rapport aux autres, lorsque le cas particulier d'un programme de test unique, regroupant toutes les procédures de test, est considéré. Ces hypothèses sont donc utilisables intégralement pour l'étude d'autres microprocesseurs.

On s'aperçoit donc que, exceptées quelques légères modifications des hypothèses dues à l'implémentation particulière du MC 68000, les hypothèses de pannes développées dans [COU 81] ont été intégralement conservées.

De plus, celles-ci restent valides pour d'autres microprocesseurs, si ceux-ci correspondent à la technologie considérée, c'est-à-dire NMOS ou H-MOS. Pour la méthodologie globale utilisée ici, les hypothèses retenues sont donc plus fortes que celles retenues en général pour le test de microprocesseurs.

II. MODELES UTILISES

Le modèle utilisé pour les pannes de la périphérie (étape i) s'est avéré adéquat, il reflète bien l'expérience.

L'insertion d'un nouveau détecteur n'a pas perturbé ce modèle, mais a renforcé la détection. Le modèle a été modifié en conséquence, toutes les formules ont été réévaluées pour tenir compte de ce nouveau détecteur, mais les résultats littéraux sont totalement équivalents à ceux obtenus pour le M 6800. Le modèle s'est ainsi révélé "extensible". Ainsi, si cette étude est reprise pour d'autres microprocesseurs, possédant d'autres types de détecteurs, si les hypothèses posées restent les mêmes, alors les formules littérales pourront être dégagées immédiatement sans avoir à réeffectuer les calculs.

Le modèle développé pour l'évaluation de la fréquence de l'évènement E et de la latence d'erreur (étape ii) a été repris intégralement, aucune modifications n'a été nécessaire. Les calculs développés restent valables pour tous les microprocesseurs quel que soit le type de leur partie contrôle et de leur partie opérative, par contre comme on a pu le remarquer, la largeur du chemin de donnée influe directement sur les valeurs obtenues pour les opérateurs. Cette remarque sera reprise au niveau des résultats (cf. § III).

Les modèles statique et dynamique développés pour les pannes du coeur créant E (étapes iii et iv) ont dû être adaptés à la configuration spéciale du MC 68000.

L'insertion d'un nouveau détecteur n'a pas perturbé ce modèle. Toutes les formules ayant été réévaluées pour tenir compte de ce détecteur, un paramètre supplémentaire a été ajouté : c'est le fait que la détection se fasse sur une adresse d'instruction ou non. En effet, si la détection

se produit sur une adresse d'instruction, c'est alors le contexte seul qui a induit la panne, tandis que si la détection se produit en dehors d'une adresse instruction, l'influence de la proportion de codes invalides et de leur répartition est en plus à prendre en compte. Cette remarque sera reprise au niveau des résultats (cf. § III). Ainsi le modèle a été modifié, mais les formules littérales obtenues sont de même nature que celles obtenues pour le modèle du M 6800, lorsque les détections, hors adresses instructions et sur adresses instructions, sont regroupées. Ce modèle conserve donc toute sa généralité et pourra ainsi être utilisé pour d'autres microprocesseurs.

Pour le coeur ne créant pas E (étape v) l'étude est divisée en deux parties représentant respectivement la partie opérative et la partie contrôle.

Pour la partie opérative, le même modèle que celui utilisé pour le M 6800 a été repris, ce modèle considère un test analytique, c'est-à-dire dépendant de la réalisation physique du circuit. Ce choix était dû principalement à la très forte répétitivité des blocs internes (les 16 étages sont en général identiques).

Pour la partie contrôle, le même modèle que celui utilisé pour le M 6800 a aussi été repris, mais dans ce cas, ce modèle est restreint à un test fonctionnel de la partie contrôle. Ce sont donc les commandes qui sont observées et non la façon de les générer.

Ces deux modèles correspondant à l'étape (v) sont tout à fait adéquats, leur usage sera donc repris in extenso lors de l'étude d'autres microprocesseurs.

Les modèles utilisés pour l'évaluation de la fréquence optimale de relance des procédures de test et pour l'évaluation de l'ordonnancement optimal de

ces procédures sont valables pour tout microprocesseur entrant dans les hypothèses. L'étude n'est ainsi plus à refaire, seuls les résultats numériques sont à évaluer pour d'autres machines.

III. RESULTATS OBTENUS

Les résultats obtenus sont très encourageants vis-à-vis du développement de l'application de cette méthodologie pour d'autres microprocesseurs.

Pour l'étape (i) de la méthodologie, pannes de la périphérie, les détecteurs COI et AE ont amélioré considérablement le temps de détection (division par 3) par rapport à ce qu'il serait en cas de détection unique par CG. Les résultats obtenus ne sont pas aussi bons que ceux obtenus pour le M 6800, mais il ne faut pas perdre de vue que ces résultats sont une caractéristique propre du microprocesseur. De plus, on s'aperçoit que la détection par COI est plus faible pour le MC 68000 que pour le M 6800, bien que pour ce dernier le pourcentage de codes invalides soit plus faible. En fait, la bonne proportion de détection par COI provient plus du profil des codes invalides et donc de leur répartition au sein des codes valides, que de leur nombre. Il est rappelé que les résultats donnés ici sont pessimistes, l'utilisation de la protection mémoire n'étant pas prise en compte puisque dépendante de l'application, et donc tout utilisateur aura des résultats meilleurs dès qu'il n'utilise pas tout l'espace mémoire.

Pour l'étape (ii) de la méthodologie, latence d'erreur et fréquence de génération de E, tous les résultats sont donnés aussi bien pour la partie opérative, que pour la partie contrôle. Ils permettent de choisir, selon la fréquence de génération de l'évènement E, le type de test à envisager pour chaque partie de la machine. Pour les fréquences grandes, les détecteurs "externes" suffiront à détecter la panne, mais pour les faibles fréquences un test périodique devra être exécuté, car le temps de détection serait trop important si les détecteurs "externes" étaient seuls considérés. Une remarque est à faire au sujet de l'opérateur de calcul d'adresses, vu la largeur du chemin d'adresses du MC 68000, un test périodique devra être envisagé pour cet opérateur, la

fréquence de génération de l'évènement E correspondant à environ 2h30 de calcul en temps réel (8 MHz) ce qui est une fréquence très faible comparativement aux résultats obtenus pour le M 6800, où pour le même genre d'opérateur la fréquence de génération de E correspondait à 0.06 secondes de calcul temps réel (2 MHz).

Pour les étapes (iii) - temps de détection consécutif à l'apparition d'un évènement E- et (iv) - prise en compte de la fréquence de l'évènement E - les résultats obtenus montrent que ce temps de détection est confiné dans des valeurs très acceptables. L'espérance du temps de détection est bornée par cinq fois la valeur de la fréquence de l'évènement E. De plus, les résultats numériques présentés sont pessimistes par rapport à ce qu'il serait si par exemple on devait considérer la détection par CG et/ou PM. Mais comme, il a été dit précédemment, la protection mémoire est dépendante de l'application considérée. Pour le chien de garde, l'impossibilité quasi-certaine d'insérer des points de relance de manière parfaitement périodique, dans des logiciels quelconque, et la grande difficulté d'évaluation mènent à une impossibilité à obtenir des résultats généraux. Ce sont les raisons pour lesquelles seuls les détecteurs COI et AE sont considérés ici, ainsi, les résultats généraux sont obtenus sans adjonction de matériel puisque les détecteurs considérés sont internes à la pastille. Une fois encore, il faut souligner la portée de tels résultats car ils sont nécessaires aussi bien au test en-ligne qu'au test hors-ligne, puisqu'un "programme de test" ne peut seul détecter l'apparition d'évènements E.

L'étape (v) - pannes du coeur ne créant l'évènement E - est divisée en deux parties : partie opérative et partie contrôle. En ce qui concerne la détection des différentes pannes de la Partie Opérative, les vecteurs de test sont fournis lorsqu'il n'y a pas d'ambiguïté entre le vecteur et l'instruction qui le générera. Pour certaines pannes, le lien entre le vecteur de test est l'instruction qui le générera n'est pas clair, il a

donc été nécessaire de préciser l'instruction qui permettra de propager la panne afin de la détecter. La totalité des procédures n'ayant pas été programmée, il n'est pas possible de donner une taille à la totalité des programmes qui recouvrent le test analytique de la partie opérative.

Seuls les quelques chiffres suivants peuvent être tirés : la partie ALU (opérateur + prédicats + acquisition + connecteur décimal) nécessite 144 vecteurs test, les autres opérateurs nécessitant 100 vecteurs de test supplémentaires. Les registres nécessitent à eux seuls 670 vecteurs de test tandis que la gestion des bus est couverte par 55 vecteurs de test.

Il est clair que plus le nombre d'opérateurs et de registres est important plus le nombre de vecteurs sera grand. Pour les registres par exemple, il est clair que le nombre de vecteurs est proportionnel au nombre de registres. Cette notion de complexité quantitative sera reprise au niveau du temps et de complexité du test (cf. § IV). On a pu remarquer aussi que la propagation des erreurs jusqu'à certains points observables était parfois très problématique et nécessitait l'utilisation de plusieurs instructions alors qu'une seule aurait été suffisante, ce point sera repris lors des considérations sur les améliorations de la machine (cf. § V).

En ce qui concerne les pannes de la partie contrôle, ce sont donc les commandes qui sont observées, et non la façon de les générer. Les résultats sont constitués par des conditions sur les valeurs des opérands et des différents registres, permettant de tester les différentes instructions, aussi bien pendant la phase d'adressage que pendant la phase d'opération. Pour chaque opérateur, registre, bus, des limitations de multiplicités des hypothèses ont été faites, les conditions déterminées sont ainsi valables dans tous les cas de pannes correspondants aux hypothèses considérées. Les résultats obtenus sont de mêmes natures que ceux correspondant au M 6800, ils n'ont pas été regroupés dans des tableaux de rappel comme

pour le M 6800, mais les conditions sont suffisamment claires pour chaque opérateur, chaque registre, chaque bus, pour pouvoir dériver les tableaux correspondants, selon l'instruction considérée. La bonne qualité des résultats provient de l'orthogonalité entre les phases d'adressage et phase d'opération. Pour d'autres microprocesseurs, si ces deux phases ne sont pas orthogonales, il serait alors nécessaire de modifier les hypothèses faites d'occurrence de pannes pour un mode d'adressages et/ou pour un type d'opérations.

Pour l'étape (vi) - évaluation des fréquences optimales ou d'évaluation de l'ordonnement optimal - les formules données sont valables pour tout type de microprocesseurs entrant dans les hypothèses, seuls les résultats numériques sont à évaluer dans ce cas, en tenant compte des résultats obtenus aux étapes précédentes. Les deux évaluations peuvent être utilisées conjointement. C'est le cas, par exemple, où deux procédures ont la même fréquence optimale d'après la première évaluation, ces 2 procédures pourront alors être groupées ensemble, l'ordonnement de l'une par rapport à l'autre étant donné par la seconde évaluation.

Ainsi les résultats obtenus sont très encourageants, ils sont valables pour une application quelconque s'exécutant sur le MC 68000. Les résultats considérés ici sont pessimistes, et tout utilisateur obtiendra de meilleurs résultats dès qu'il n'utilise pas tout l'espace mémoire, ou que cet espace possède certaines propriétés (zônes ROM, zônes RAM, zônes données, zônes programmes etc...).

Ils permettent une détection des pannes en un temps maximum déterminé sans utiliser de redondance massive du genre duplication/comparaison. Les résultats fournis ici, ne nécessitent donc que la réalisation d'un chien de garde, mais, comme il a été précédemment indiqué, si tout l'espace mémoire n'est pas utilisé, l'adjonction du circuit MMU réalisant la protection mémoire apportera un gain supplémentaire de détection non négligeable.

IV. TEST : TEMPS ET COMPLEXITE

Comme on a pu le remarquer, tout au long des différentes étapes de cette étude, l'impact de la complexité de la machine sur la complexité du test est très important.

Ainsi, deux types de complexité du test ont été différenciés : la complexité quantitative et complexité qualitative. Ces deux types de complexité sont directement liés à la complexité de la machine. En effet, l'augmentation d'intégration des circuits micro-électroniques est la conséquence de deux facteurs, d'une part la diminution de la largeur du canal qui fait que pour une même surface il est possible d'intégrer un plus grand nombre de transistors, d'autre part l'augmentation de la surface totale de la puce qui fait que pour une même largeur de canal il est aussi possible d'intégrer un plus grand nombre de transistors. Ainsi, pour le MC 68000, la largeur de canal a été divisé par deux (trois actuellement) par rapport à celle du M 6800, mais la surface de la puce a été multipliée par 2.5, le nombre de transistors a été ainsi multiplié par dix. Il n'est donc pas étonnant que le contenu fonctionnel (registres, opérateur, bus) ait été multiplié par cinq. Ceci permet d'introduire le paramètre de complexité quantitative. La complexité quantitative est ainsi lié au facteur d'accroissement du nombre d'unités fonctionnelles. Cette augmentation de complexité quantitative se traduit par une augmentation du temps de test.

En effet, il va de soi que, par exemple, une fois les vecteurs de test d'un registre, obtenus, ces vecteurs seront lancé n fois si n représente le nombre de registres testés par ces vecteurs. Donc si le nombre de registres de la machine est augmenté, le nombre de lancements des vecteurs de test est augmenté de la même valeur.

Cette augmentation du nombre de lancement correspond à une augmentation du temps de test, exprimé en nombre d'instructions exécutés pour ce test.

Le paramètre de complexité qualitative, par contre, est lié à la réalisation de la fonction et plus particulièrement à la complexité des conséquences des pannes sur cette fonction. C'est le cas bien connu, de la différence de complexité de test qui existe entre une porte standard et une porte non standard réalisant la même fonction, (ou exclusif par exemple). Pour la porte standard, toutes les pannes possibles se ramènent à des collages des entrées ou des sorties, tandis que pour la porte non standard, certaines pannes (coupure, court-circuit) modifient la fonction mais ne sont plus équivalentes à des collages des entrées ou des sorties. Alors l'analyse complète du fonctionnement en présence de pannes est nécessaire, c'est cette analyse complète qui a du être développée pour les pannes de gestion des transferts pour le MC 68000. L'augmentation de la complexité qualitative se répercute ainsi sur l'effort nécessaire à évaluer le test. Le temps de test sera aussi augmenté si des vecteurs sont ajoutés, mais cette augmentation n'est pas linéaire telle qu'elle l'était pour une augmentation de la complexité quantitative.

Ainsi donc, la complexité d'une machine, telle qu'elle est définie peut influencer de deux façons différentes sur la complexité du test : soit sur le temps passé à exécuter des programmes de test, soit sur l'effort à déterminer les conséquences des pannes et donc dériver les programmes de test.

V. AMELIORATIONS DE LA MACHINE

Des modifications du microprocesseurs peuvent être envisagées afin d'améliorer le test tel qu'il est présenté ici. Ces améliorations ne se traduisent pas forcément par un redessin complet de tout le circuit, mais pour certaines d'entre-elles par de légères modifications sur le dessin déjà existant. Ces améliorations sont données dans le sens d'une amélioration la qualité et du temps nécessaire au test.

Tout d'abord, on a pu remarquer l'avantage du fait que le MC 68000 possède des détecteurs internes (COI-AE). Cet avantage est contre-balancé par l'inconvénient que ces détecteurs peuvent aussi être atteint par les pannes. Deux cas sont alors possibles, soit l'un de ces détecteurs se déclenche alors qu'aucune panne ne devait être détectée par ce détecteur, ce cas correspond au fonctionnement correct puisque le détecteur lui-même est en panne et que l'exécution est arrêtée. Le second cas, où la panne du détecteur masque une seconde panne, est plus grave puisqu'aucune détection ne sera déclenchée.

L'amélioration à envisager à ce niveau est de rendre ces détecteurs internes autotestables. Les résultats de [NIC 83] pourront être utilisés pour rendre ces circuits autotestables. Ainsi la qualité du test obtenu sera améliorée.

Le temps nécessaire au test peut être amélioré de plusieurs façons différentes ; celles-ci sont présentées dans l'ordre d'un temps d'amélioration croissant du temps, mais aussi dans l'ordre croissant des modifications nécessaires.

La première amélioration à apporter, d'un coût totalement négligeable, porte sur la programmation des micro-instructions. En effet, comme on l'a vu au chapitre VI, certains chemins sont difficilement utilisables. Ainsi

l'observabilité peut être améliorée en modifiant les micro-instructions de façon à emprunter les chemins désirés. Le nombre d'instruction nécessaires au test serait alors grandement diminué.

Le coût négligeable provient de la structure du microprocesseur, en effet seul le contenu des commandes à l'intérieur de la ROM de microinstruction est à modifier.

La seconde amélioration porte aussi sur le contenu de la ROM, et correspond donc aussi à un coût négligeable pour les mêmes raisons. Cette amélioration consiste à utiliser certains profils invalides (rendus alors valides durant le test) pour programmer des instructions "spéciales test" permettant de tester rapidement et complètement un opérateur, un registre, un bus etc... Dans ce cas, une procédure complète de test se trouvera remplacée par une ou quelques instructions "spéciales test". Le coût apporté est négligeable vis-à-vis de la ROM qui est incomplètement utilisée pour le moment, la seule répercussion se situe au niveau de la modification des profils invalides, ceci étant réalisé par l'intermédiaire d'un PLA, la suppression de quelques transistors ne constituant pas une modification primordiale.

La troisième amélioration, n'est plus quant à elle de coût négligeable. En effet, cette amélioration est constituée par l'introduction de parties autotestables pour remplacer des parties de circuits trop difficilement testables, ou amenant un temps de détection trop long.

C'est alors non seulement un gain en temps de détection qu'en espace mémoire nécessaire aux programmes de test couvrant cette partie, qui est réalisé. De plus, le temps consacré à l'application est plus important puisque les fréquences de relances et tailles des programmes de test sont modifiées. On peut ainsi envisager l'ajout d'un nombre plus ou moins important de parties autotestables. Le coût de cette augmentation correspond au coût de l'augmentation de surface du circuit.

Cette augmentation de surface est alors comprise entre 0 % (aucun ajout) et 58 % (MC 68000 entièrement autotestable) telle que donnée dans [DIS 80].

Les coûts respectifs correspondants à certaines parties ont été évalués dans [NIC 83].

Toutes ces améliorations correspondent donc à un coût croissant du produit. Ceci pourrait donc donner naissance à toute une gamme de versions successives se distinguant entre elles par leur coût et leur temps de détection. Chaque utilisateur aurait alors le choix d'une gamme correspondant à ses besoins propres en temps de détection.

REFERENCES

- [COU 81] COURTOIS B.
 "Test et LSI"
 Thèse de doctorat d'état, Grenoble 1981.
- [COU 82] COURTOIS B.
 "Performance modelling of partially self checking
 systems".
 FTCS 12. Santa Monica, June 1982.
- [NIC 83] NICOLAIDIS M., COURTOIS B.
 "Design of self-checking systems based on analytical
 fault hypotheses".
 Rapport de Recherche IMAG, RR n°. 353, Mars 1983.
- [DIS 81] DISPARTE C.P.
 "A self checking VLSI microprocessor for electronic engine
 control".
 FTCS 11. Portland, June 1981.

TABLE DES MATIERES

=====

CHAPITRE I - INTRODUCTION

I	- CLASSIFICATION DES DIFFERENTS TYPES DE TEST -----	8
	I.1. Critère en ligne/hors ligne -----	9
	I.2. Critère continu/discontinu -----	9
	I.3. Critère in-situ/ex-situ -----	9
II	- METHODOLOGIE -----	10
	SOMMAIRE -----	15
	REFERENCES -----	17

CHAPITRE II - LE MC 68000 PRESENTATION DU MICROPROCESSEUR

I	- INTRODUCTION -----	23
II	- LES RESSOURCES -----	23
	1. Modèle de programmation -----	23
	2. Organisation des registres -----	25
	2.1. Registres données -----	25
	2.2. Registres adressés -----	26
	2.3. Le pointeur de pile système -----	26
III	- LES INSTRUCTIONS -----	27
	1. Format de l'instruction -----	27
	2. Jeu d'instructions -----	29
	2.1. Opérations arithmétiques entières -----	29
	2.2. Opérations logiques -----	30
	2.3. Décalages -----	31
	2.4. Manipulation de bits -----	31
	2.5. Mouvement de données -----	32

2.6. Opération BCD -----	32
2.7. Contrôle de programmes -----	32
2.8. Contrôle système -----	33
IV - L'ADRESSAGE -----	34
1. Le bus adresse -----	34
2. Modes d'adressage -----	34
3. Références programme/Référence donnée -----	35
4. Adresse effective -----	36
4.1. Mode registre direct -----	36
4.2. Mode d'adressage mémoire -----	37
4.3. Mode d'adressage spécial -----	40
4.4. Références implicites -----	43
4.5. Codage de l'adresse effective -----	43
5. Les catégories d'adressage -----	44
5.1. La catégorie données -----	44
5.2. La catégorie mémoire -----	44
5.3. La catégorie contrôle -----	44
5.4. La catégorie altérable -----	44
V - LES DONNEES -----	45
1. Types de données supportés -----	46
2. Organisation des données en mémoire -----	46
VI - LES SIGNAUX ET BUS -----	48
1. Le bus adresse -----	49
2. Le bus donnée -----	49
3. Le bus contrôle -----	49
3.1. Contrôle du bus asynchrone -----	49
3.2. Contrôle de l'arbitrage du bus -----	51
3.3. Contrôle des interruptions -----	51
3.4. Contrôle système -----	52
3.5. Contrôle de la périphérie M 68000 -----	53
3.6. Etat du processeur -----	53

VII - CONCLUSION -----	54
NOTE BIBLIOGRAPHIQUES -----	55
REFERENCES -----	57
<u>CHAPITRE III - PERIPHERIE DE LA PASTILLE - Etape (i)</u>	
I - INTRODUCTION -----	63
II - DEFINITION DU MODELE -----	63
1. Le modèle utilisé -----	63
2. Hypothèses et définition -----	64
3. Grandeurs caractéristiques du temps de détection -----	65
3.1. Densité du temps de détection -----	66
3.2. Grandeurs propres à chacun des détecteurs -----	66
3.3. Probabilité de détection -----	67
3.4. Résultats approchés -----	68
3.5. Expressions limites -----	69
III - RESULTATS EXPERIMENTAUX -----	70
1. Pannes affectant le bus données -----	70
2. Pannes affectant le bus adresses -----	75
3. Généralisation à une application quelconque -----	78
IV - CONCLUSION -----	80
ANNEXE -----	81
NOTES BIBLIOGRAPHIQUES -----	85
REFERENCES -----	87

CHAPITRE IV - LATENCE D'ERREUR. LATENCE ET FREQUENCE DE L'EVENEMENT E - Etape (iii)

I	- INTRODUCTION -----	93
II	- DEFINITION DES PROBABILITES DE TRANSFERT D'OBSERVABILITE -----	94
	II.1. Divergence et reconvergence -----	97
	II.2. Séquentialité -----	97
	II.3. Injection de pannes -----	97
III	- APPLICATION DYNAMIQUE AU MC 68000 -----	99
	III.1. Evènement considéré -----	99
	III.2. Partie opérative -----	100
	III.2.1. Modèle utilisé -----	100
	III.2.2. Propriétés dynamiques -----	100
	III.3. Partie contrôle -----	103
IV	- RESULTATS OBTENUS POUR LE MC 68000 -----	106
	IV.1. Partie opérative -----	106
	IV.1.1. Opérateurs -----	106
	IV.1.1.1. Opérateur arithmétique et logique (ALU) --	106
	IV.1.1.2. Opérateur décimal (DCB) -----	107
	IV.1.1.3. Extension de l'opérateur de calcul (ALUE) -	107
	IV.1.1.4. Opérateur de multiplexage (MUX I/O) -----	107
	IV.1.1.5. Opérateur de manipulation de bits (DCR) ---	108
	IV.1.1.6. Opérateur Fool-it -----	108
	IV.1.1.7. Opérateur encodeur de priorité (PREN) -----	108
	IV.1.1.8. Opérateur d'extension de signe (SIGNEX) ---	109
	IV.1.1.9. Opérateur de calcul d'adresses (AUH-AUL) --	109
	IV.1.2. Registres -----	134
	IV.1.2.1. Registres données (Dn) -----	134
	IV.1.2.2. Registres adresses (An) -----	135
	IV.1.2.3. Registres temporaires (DT et AT) -----	136

IV.1.2.4. Buffer d'adresse (AOB) -----	136
IV.1.2.5. Compteur ordinal (PC) -----	137
IV.1.2.6. Latch d'AU (AUR) -----	137
IV.1.2.7. Registre de FTU -----	137
IV.1.2.8. Registre de PREN -----	137
IV.1.2.9. Registre de DCR -----	138
IV.1.2.10. Registre DBIN -----	138
IV.1.2.11. Registre DOB -----	138
IV.1.2.12. Registre d'ALUE -----	139
IV.1.2.13. Latch d'ALU (ALUR) -----	139
IV.1.2.14. Registre ALUB -----	139
IV.1.3. Bus -----	140
IV.1.3.1. Bus ABD, DBD -----	140
IV.1.3.2. Bus ABL, DBL -----	140
IV.1.3.3. Bus ABH, DBH -----	140
IV.1.3.4. Bus EAB -----	141
IV.1.3.5. Bus EDB -----	141
IV.1.3.6. Bus BC -----	142
IV.2. Partie contrôle -----	145
IV.2.1. Classe 0 -----	146
IV.2.2. Classe I -----	146
IV.2.3. Classe II -----	147
IV.2.3.1. Sous-classe 1 -----	147
IV.2.3.2. Sous-classe 2 -----	148
IV.2.4. Classe III -----	150
IV.2.5. Classe IV -----	150
IV.2.6. Classe V -----	150
IV.2.7. Pannes à conséquences hors-classes -----	150
V - CONCLUSION -----	154
REFERENCES -----	155

CHAPITRE V - COEUR DE LA PASTILLE CREANT E - Etape (iii) - Etape (iv)

I	- INTRODUCTION	161
II	- ANALYSE STATIQUE MODELISATION	163
	1. Analyse Qualitative	163
	2. Définitions des modèles statiques	164
	2.1. Premier modèle	161
	2.1.1. Accès matriciel	165
	. Probabilités d'états	168
	. Distributions, densités, probabilités limites	169
	. Moments de détection	172
	2.1.2. Accès direct	175
	. Probabilités d'état, probabilités de détection	175
	. Moments de détection	177
	2.2. Second Modèle	178
	2.2.1. Accès matriciel	178
	. Probabilités d'états	178
	. Distributions, densités, probabilités limites	180
	. Moments de détection	182
	2.2.2. Accès direct	184
	. Probabilités d'états probabilités de détection	184
	. Moments de détection	185
	2.3. Récapitulation	186
III	- RESULTATS EXPERIMENTAUX	191
IV	- ANALYSE DYNAMIQUE MODELISATION	202
	1. Evènements E de faibles fréquences	202
	2. Evènements E de grandes fréquences. Définitions des modèles dynamiques	203
	2.1. Premier modèle	203
	2.1.1. Accès matriciel	205

. Probabilités d'états -----	205
2.1.2. Accès direct -----	205
. Probabilités d'états, probabilités de détection -----	207
. Moments de détection -----	209
2.2. Second modèle -----	211
2.2.1. Accès matriciel -----	211
. Probabilités d'états -----	211
. Distributions, densités et probabilités limites -----	218
. Moments de détection -----	221
2.2.2. Accès direct -----	224
. Probabilités d'états et probabilités de détection -----	224
. Moments de détection -----	225
2.3. Récapitulation et discussion - Caractéristique de la fréquence de E -----	226
2.3.1. Récapitulation -----	226
2.3.2. Valeur caractéristique de la fréquence de E -----	229
V - DYNAMISATION DES RESULTATS EXPERIMENTAUX -----	231
VI - GENERALISATION A UNE APPLICATION QUELCONQUE -----	235
1. Distribution et densité du temps de détection -----	236
1.1. Valeurs caractéristiques pour $f = 1$ -----	237
1.2. Valeurs caractéristiques pour $f \rightarrow 0$ -----	238
2. Moments du temps de détection en fonction de la fréquence -----	242
VII - CONCLUSION -----	245
NOTES BIBLIOGRAPHIQUES -----	247
REFERENCES -----	249

CHAPITRE VI - COEUR NE CREANT PAS E. PARTIE OPERATIVE - Etape (v)

I	- INTRODUCTION -----	257
	I.1. Stragégie du test -----	258
	I.2. Hypothèses de pannes envisagées pour le test analytique -----	258
	I.2.1. Classe 0 -----	259
	I.2.2. Classe I -----	259
	I.2.3. Classe II -----	259
	I.3. Dérivation des vecteurs de test -----	259
	I.3.1. Dérivation des vecteurs de test de classe 0 -----	260
	I.3.2. Dérivation des vecteurs de test de classe II limitée géographiquement -----	261
II	- PRESENTATION DE LA PARTIE OPERATIVE DU MC 68000 -----	263
III	- ETUDE ET TEST DES OPERATEURS DE LA PARTIE OPERATIVE -----	267
	III.1. Unité arithmétique et logique (ALU). Opérateur de calcul de données -----	267
	III.1.1. Présentation de l'opérateur -----	267
	III.1.2. Test de classe 0 de l'opérateur -----	268
	III.1.3. Test de classe II limitée géographiquement de l'opérateur -----	279
	III.2. Etude et test des prédicats de l'opérateur de calcul -----	282
	III.3. Acquisition des opérandes sur l'ALU -----	284
	III.3.1. Présentation de l'entrée A de l'ALU -----	284
	III.3.2. Test de l'entrée A de l'ALU -----	285
	III.3.3. Cas particulier du bit 7 : extension octet -----	286
	III.3.4. Décalage : cas particulier des bits 0, 7 et 15 -----	288
	III.3.5. Présentation de l'entrée D de l'ALU -----	293
	III.3.6. Test de l'entrée D de l'ALU -----	294
	III.4. Opérateur décimal (BCD) -----	295
	III.4.1. Présentation de l'opérateur -----	295
	III.4.2. Test de l'opérateur -----	296

III.5.	Extension de l'opérateur de calcul (ALU-E) -----	299
III.5.1.	Présentation de l'opérateur -----	299
III.5.2.	Test de l'opérateur ALU-E -----	300
III.5.3.	Cas particuliers des bits 0, 1 et 15 -----	301
III.6.	Opérateur de multiplexage : MUX I/O -----	303
III.6.1.	Présentation de l'opérateur -----	303
III.6.2.	Test de l'opérateur -----	304
III.7.	Opérateur registre décodeur (DCR) -----	305
III.7.1.	Présentation de l'opérateur -----	305
III.7.2.	Test de l'opérateur -----	309
III.7.3.	Acquisition et sortie de l'opérateur -----	310
III.8.	Opérateur Fool-it -----	311
III.8.1.	Présentation de l'opérateur -----	311
III.8.2.	Test de l'opérateur -----	314
III.9.	Opérateur PREN (Priority Register Encoder) -----	315
III.9.1.	Présentation de l'opérateur -----	315
III.9.2.	Test de l'opérateur -----	319
III.10.	Opérateur de calcul d'adresses (AUH-AUL) -----	320
III.10.1.	Présentation de l'opérateur -----	320
III.10.2.	Test de l'opérateur -----	323
III.11.	Opérateur d'extension de signe (SIGNEX) -----	326
III.11.1.	Présentation de l'opérateur -----	326
III.11.2.	Test de l'opérateur -----	326
IV	- ETUDE ET TEST DU BUS -----	329
IV.1.	Présentation du bus -----	329
IV.2.	Test de la précharge -----	332
IV.3.	Test des résistances -----	334
IV.4.	Test de l'amplificateur -----	335
IV.5.	Test des switches -----	335
IV.6.	Test des coupures de bus -----	336
V	- ETUDE ET TEST DES REGISTRES DE LA PARTIE OPERATIVE -----	337
V.1.	Test des registres organisés en RAM -----	338

V.1.1. Présentation du point de RAM -----	339
V.1.2. Test ^o du point de RAM -----	340
V.1.3. Cas particulier des registres DT et PC -----	343
V.2. Test des registres autres que ceux de la RAM -----	344
V.2.1. Test du registre AOB (Buffer de sortie adresse) -----	344
V.2.2. Test du registre AUR (Latch d'AU) -----	346
V.2.3. Test du registre AT (adresse temporaire) -----	348
V.2.4. Test du registre du PREN -----	351
V.2.5. Test du registre FTU (Field Translate Unit) -----	351
V.2.6. Test du registre DCR (decode register) -----	354
V.2.7. Test du registre DBIN (data input buffer) -----	354
V.2.8. Test du registre DOB (data output buffer) -----	356
V.2.9. Test du registre ALUE (ALU-Extender) -----	357
V.2.10. Test du registre ALU-R (latch d'ALU) -----	357
V.2.11. Test du registre ALU-B (buffer d'ALU) -----	359
VI - CONCLUSION -----	361
REFERENCES -----	363
 <u>CHAPITRE VII - COEUR NE CREANT PAS E. PARTIE CONTROLE - Etape (v)</u>	
I - INTRODUCTION -----	369
II - MODELE DE PANNE DE LA PARTIE CONTROLE -----	370
III - HYPOTHESES DE PANNES DE LA COMMANDE D'OPERATEURS -----	373
III.1. Opérateur arithmétique et logique (ALU) -----	373
III.2. Opérateur arithmétique (AUH, AUL) -----	383
III.3. Opérateur d'extension d'ALU (ALUE) -----	388
III.4. Opérateur d'extension de signe (SIGNEX) -----	391
III.5. Opérateur encodeur de priorité (PREN) -----	393

III.6. Opérateur Fool-it -----	398
III.7. Opérateur de manipulation de bits (DCR) -----	398
III.8. Opérateur de multiplexage d'octets (MUX I/O) -----	402
IV - HYPOTHESES DE PANNES DE LA GESTION DES TRANSFERTS -----	408
IV.1. Pannes de gestion du bus -----	412
IV.1.1. Pannes de précharge -----	412
IV.1.1.1. Précharge inexistante -----	412
IV.1.1.2. Précharge permanente -----	414
IV.1.2. Pannes d'amplification -----	419
IV.1.2.1. Amplification inexistante -----	421
IV.1.2.2. Amplification permanente -----	421
IV.1.3. Pannes de switches -----	421
IV.1.4. Pannes de Fool-it -----	425
IV.2. Pannes de sélection de registres -----	427
IV.2.1. Aucune source n'est sélectionnée -----	428
IV.2.2. Plusieurs sources sont sélectionnées -----	432
IV.3. Hypothèses retenues pour les pannes de transferts -----	436
IV.4. Erreurs possibles lors d'un transfert élémentaire -----	442
IV.5. Erreurs possibles lors d'un transfert vers un opérateur -----	448
IV.5.1. Opérateur PREN -----	448
IV.5.2. Opérateur AU -----	452
IV.5.3. Opérateur ALU -----	456
IV.5.4. Opérateur ALUE -----	458
IV.5.5. Opérateur DCR -----	459
IV.5.6. Opérateur SIGNEX -----	463
IV.5.7. Opérateurs non concernés par les pannes de transferts	469
IV.6. Erreurs possibles lorsqu'aucun transfert ne doit avoir lieu -	469
IV.7. Conclusion sur les hypothèses de pannes de la gestion des transferts -----	470
V - TEST DU SEQUENCMENT DES INSTRUCTIONS -----	474

V.1. Définitions et classes de pannes (de test) -----	474
V.2. Procédures de test utiles pour les classes 0 et II sur un cycle -----	477
V.3. Procédures de test utiles pour les classes I et III, sur un cycle -----	478
V.4. Multiplicité temporelle, erreurs sur plusieurs cycles -----	479
V.5. Multiplicité temporelle et spatiale, test d'une instruction --	482
VI - HYPOTHESES ET TEST DE LA GESTION DU MOT D'ETAT §-----	486
VI.1. Hypothèses et test de la gestion du code condition -----	487
VI.2. Hypothèses et test de la gestion du mot superviseur -----	493
VI.3. Hypothèses et test de la gestion des interruptions -----	494
VI.4. Hypothèses et test des bascules S et T -----	495
VII - TEST DU BRANCHEMENT A BON ESCIENT -----	498
VII.1. Test des alternatives de séquences -----	498
VII.2. Test de la seconde partie du PLA de branchement -----	502
IX - CONCLUSION -----	504
REFERENCES -----	507
 <u>CHAPITRE VIII - FREQUENCES OPTIMALES. ORDONNANCEMENT DES PROCEDURES - Etape (vi)</u>	
I - INTRODUCTION -----	513
II - FORMULATION DU PROBLEME -----	514
III - EXPRESSION DES SOLUTIONS -----	516
III.1. Solutions du problème I -----	516
III.2. Solutions du problème II -----	517
IV - TEMPS DE LATENCE ET TEMPS DE DETECTION -----	517

V	- CAS PARTICULIER : UN SEUL PROGRAMME DE TEST -----	518
	V.1. Interprétation géométrique -----	519
	V.2. Ordre optimal des procédures de test -----	521
	V.2.1. Rappel des hypothèses -----	522
	V.2.2. Ordre optimal pour une latence nulle -----	522
	V.2.3. Ordre optimal pour une latence non nulle -----	523
VI	- CONCLUSION -----	525
	REFERENCES -----	527

CHAPITRE IX - CIRCUITS D'ANALYSE DE SIGNATURES. MACHINE AUTOMATIQUE DE TEST
(Hors méthodologie)

I	- INTRODUCTION -----	533
II	- TEST HORS-LIGNE -----	534
III	- ANALYSE DE SIGNATURES -----	536
IV	- DESCRIPTION DES CIRCUITS D'ANALYSE DE SIGNATURES -----	538
	IV.1. Le circuit multiprojet -----	538
	IV.2. L'analyseur de signatures -----	538
	IV.2.1. Le CMP82 AS1 -----	539
	IV.2.2. Le CMP83 AS1 -----	541
V	- TEST DE FIN DE FABRICATION. TEST DE CONCEPTION -----	546
	V.1. Test visuel -----	546
	V.2. Test par contraste de potentiel -----	547
	V.3. Test à la machine à pointe -----	547
VI	- CONCLUSION -----	549
	REFERENCES -----	551

CHAPITRE X - CONCLUSION

I - HYPOTHESES DE PANNES -----	558
II - MODELES UTILISES -----	561
III - RESULTATS OBTENUS -----	564
IV - TEST : TEMPS ET COMPLEXITE -----	568
V - AMELIORATIONS DE LA MACHINE -----	570
REFERENCES -----	573

AUTORISATION DE SOUTENANCE

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

VU le rapport de présentation de Monsieur B. COURTOIS, Chargé de recherche

Monsieur Pierre MARCHAL

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

Fait à Grenoble, le 21 juin 1983

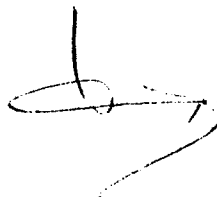
Le Président de l'I.N.P.-G

D. BLOCH

Président

de l'Institut National Polytechnique
de Grenoble

P.O. le Vice-Président,



RESUME

Le test en ligne du microprocesseur MC 68000 est abordé sans utiliser de redondance massive du type duplication/comparaison. La détection des pannes est réalisée par des détecteurs spéciaux : détecteurs de codes opérations invalides, d'adresses-erreurs, de protection mémoire ainsi qu'un chien de garde. Le gain apporté en circuits annexes est alors très important, mais ce gain est contrebalancé par un temps de détection non nul. Tous les modèles mathématiques nécessaires au long des six étapes de la méthodologie globale sont présentés. Les paramètres numériques de ces modèles sont évalués par simulations. Les résultats obtenus sont comparés à ceux obtenus pour le M 6800 . Les valeurs du temps de détection sont des valeurs pessimistes et sont valables pour tout utilisateur. Les développements vers diverses améliorations possibles de ce temps de détection, obtenues par le biais de modifications du circuit, sont présentés et discutés. Un test hors ligne peut être dérivé directement par le regroupement de toutes les procédures de test.

MOTS-CLES : Test - Test en ligne - Test hors ligne - Microprocesseur - Hypothèses de pannes - modélisation - Test analytique - Test fonctionnel - Temps de latence - Temps de détection.