Processeur base de données MAGE : aspect matériel

Philippe Navaux

THESE

présentée à

Institut National Polytechnique de Grenoble

pour obtenir le grade de
DOCTEUR INGENIEUR
Génie Informatique

par

Philippe O. A. NAVAUX

PROCESSEUR BASE DE DONNEES MAGE
ASPECT MATERIEL

soutenue le 27 novembre 1979 devant la commission d'examen

L. BOLLIET	Président
F. ANCEAU	
G. BERGER SABBATEL	
C. DELOBEL	Examinateurs
J. RACINE	
J. ROHMER	

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1977-1978

Président : M. Philippe TRAYNARD

Vice-présidents : M. René PAUTHENET
M. Georges LESPINARD

PROFESSEURS TITULAIRES

MM.	BENOIT Jean	Electronique - automatique
	BESSON Jean	Chimie minérale
	BLOCH Daniel	Physique du solide - cristallographie
	BONNETAIN Lucien	Génie chimique
	BONNIER Etienne	Métallurgie
*	BOUDOURIS Georges	Electronique - automatique
	BRISSONNEAU Pierre	Physique du solide - cristallographie
	BUYLE-BODIN Maurice	Electronique - automatique
	COUMES André	Electronique - automatique
	DURAND Francis	Métallurgie
	FELICI Noël	Electronique - automatique
	FOULARD Claude	Electronique - automatique
	LANCIA Roland	Electronique - automatique
	LONGEQUEUE Jean-Pierre	Physique nucléaire corpusculaire
	LESPINARD Georges	Mécanique
	MOREAU René	Mécanique
	PARIAUD Jean-Charles	Chimie - physique
	PAUTHENET René	Electronique - automatique
	PERRET René	Electronique - automatique
	POLOUJADOFF Michel	Electronique - automatique
	TRAYNARD Philippe	Chimie - physique
	VEILLON Gérard	Informatique fondamentale et appliquée

* en congé pour études

PROFESSEURS SANS CHAIRE

MM.	BLIMAN Samuël	Electronique - automatique
	BOUVARD Maurice	Génie mécanique
	COHEN Joseph	Electronique - automatique
	GUYOT Pierre	Métallurgie physique
	LACOUME Jean-Louis	Electronique - automatique
	JOUBERT Jean-Claude	Physique du solide - cristallographie

.../...

MM.	ROBERT André	Chimie appliquée et des matériaux
	ROBERT François	Analyse numérique
	ZADWORNY François	Electronique - automatique

MAITRES DE CONFERENCES

MM.	ANCEAU François	Informatique fondamentale et appliquée
	CHARTIER Germain	Electronique - automatique
	CHIAVERINA Jean	Biologie, biochimie, agronomie
	IVANES Marcel	Electronique - automatique
	LESIEUR Marcel	Mécanique
	MORET Roger	Physique nucléaire - corpusculaire
	PIAU Jean-Michel	Mécanique
	PIERRARD Jean-Marie	Mécanique
	SABONNADIERE Jean-Claude	Informatique fondamentale et appliquée
Mme	SAUCIER Gabrielle	Informatique fondamentale et appliquée
M.	SOHM Jean-Claude	Chimie Physique

CHERCHEURS DU C.N.R.S. (Directeur et Maîtres de Recherche)

M.	FRUCHART Robert	Directeur de Recherche
MM.	ANSARA Ibrahim	Maître de Recherche
	BRONOEL Guy	Maître de Recherche
	CARRE René	Maître de Recherche
	DAVID René	Maître de Recherche
	DRIOLE Jean	Maître de Recherche
	KLEITZ Michel	Maître de Recherche
	LANDAU Ioan-Doré	Maître de Recherche
	MATHIEU Jean-Claude	Maître de Recherche
	MERMET Jean	Maître de Recherche
	MUNIER Jacques	Maître de Recherche

Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique)

E.N.S.E.E.G.

MM.	BISCONDI Michel	Ecole des Mines St. Etienne (dépt. Métallurgie)
	BOOS Jean-Yves	Ecole des Mines St. Etienne (Métallurgie)
	DRIVER Julian	Ecole des Mines St. Etienne (Métallurgie)

.../...

MM.	KOBYLANSKI André	Ecole des Mines St. Etienne (Métallurgie)
	LE COZE Jean	Ecole des Mines St. Etienne (Métallurgie)
	LESBATS Pierre	Ecole des Mines St. Etienne (Métallurgie)
	LEVY Jacques	Ecole des Mines St. Etienne (Métallurgie)
	RIEU Jean	Ecole des Mines St. Etienne (Métallurgie)
	SAINFORT	C.E.N. Grenoble (Métallurgie)
	SOUQUET	U.S.M.G.
	CAILLET Marcel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	COULON Michel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	GUILHOT Bernard	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	LALAUZE René	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	LANCELOT Francis	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	SARRAZIN Pierre	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	SOUSTELLE Michel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	THEVENOT François	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	THOMAS Gérard	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	TOUZAIN Philippe	Ecole des Mines St. Etienne (Chim. Min. Ph.)
	TRAN MINH Canh	Ecole des Mines St. Etienne (Chim. Min. Ph.)

E.N.S.E.R.G.

MM.	BOREL	Centre d'études nucléaires de Grenoble
	KAMARINOS	Centre national recherche scientifique

E.N.S.E.G.P.

M.	BORNARD	Centre national recherche scientifique
Mme	CHERUY	Centre national recherche scientifique
MM.	DAVID	Centre national recherche scientifique
	DESCHIZEAUX	Centre national recherche scientifique

à ma femme

à mes enfants

à mon père

et à la mémoire de ma mère

La simplicité est autour de nous, c'est la "nature" simple et profonde.

Je tiens à remercier

Monsieur L.BOLLIET, Professeur à l'Université de Grenoble II, qui a bien voulu me faire l'honneur de présider le jury de cette thèse,

Monsieur F. ANCEAU, Maître de Conférence à l'Institut National Polytechnique de Grenoble, qui a bien voulu m'accepter au sein de l'Equipe de Recherche en Architecture d'Ordinateurs qu'il dirige, et pour m'avoir prodigué ses conseils tout au long de cette étude,

Monsieur BERGER SABBATEL, chercheur CNRS, qui s'est chargé de la partie logicielle de cette étude, pour les discusssions qu'on a eues et les observations qu'il a apportées à cet ouvrage,

Monsieur DELOBEL, Professeur à l'Université Scientifique et Médicale de Grenoble, qui a bien voulu faire partie de mon jury,

Monsieur RACINE, Chef de la Division Téléinformatique à la Société SAGEM, qui a apporté son soutien à ce projet,

Monsieur ROHMER, chercheur à l'IRIA, pour les observation et suggestions qu'il a apportées à cet ouvrage,

tous mes collegues, membres de l'équipe en Architecture d'ordinateurs avec qui j'ai eu des échanges fructueux.

Monsieur IGLESIAS et le service de reprographie du Laboratoire IMAG qui ont assuré le tirage de ce document.

SOMMAIRE

CHAPITRE VIII - IMPLÉMENTATION DU PROCESSEUR DISQUE : LE PROTOTYPE

AVANT-PROPOS

AVANT PROPOS

Pour une meilleure compréhension du sujet de cette thèse il est intéressant de présenter d'abord le cadre dans lequel s'est développé le projet du processeur base de données, PBD, avec ses antécédents.

Le projet MAGE (Matériel Adapté à la Gestion d'Entités) a démarré en janvier 1975 en collaboration avec la SAGEM (Société d'Applications Générales en Electricité et Mécanique). L'objectif était de développer un système matériel et logiciel de gestion de données adapté à des petits systèmes "clefs en main". Dans ce type de système, l'utilisateur dispose de fonctions adaptées à la réalisation de son problème, sans avoir couramment la possibilité de modifier la structure de base.

La première partie du projet MAGE a abouti à la définition du PBD. Elle a été menée par G. BERGER SABBATEL [BER1.78] et visait à obtenir une structure d'accès aux données adaptée aux petits systèmes. Cette étude a permis de définir une méthode d'accès inspirée de Socrate. D'après la définition de cette structure un compilateur a été écrit et une évaluation de la puissance de traitement nécessaire en termes de microprocesseurs a été effectué. Cette évaluation a permis d'établir quelles étaient les possibilités d'un microprocesseur pour exécuter la méthode d'accès aux données. Ces résultats ont permis de proposer une architecture à deux microprocesseurs.

A ce point, donc à partir de la définition des principaux blocs de l'architecture, commence le travail objet de cette thèse. Son but a été de définir et réaliser, la structure matérielle d'une machine sur laquelle, la méthode d'accès aux données pouvait tourner. Ce travail, sur l'aspect matériel du projet a commencé en octobre 1976 avec un préprojet de l'architecture, objet d'un rapport de DEA [NAV.77] , suivi par le projet définitif et la réalisation d'un prototype.

CHAPITRE I

INTRODUCTION

1. BASES DE DONNEES

2. PROCESSEURS BASE DE DONNEES

I. INTRODUCTION

I.1. BASES DE DONNEES

Dès le début de l'histoire des systèmes informatiques, les utilisateurs reconnurent qu'ils pouvaient faire le traitement des données à un plus faible coût et avec des meilleures performances que les systèmes précedemment utilisés. Ce travail étant anciennement réalisé par des appareils électro-mécaniques tels que les tabulatrices, trieuses, interclasseuses et autres. L'avènement du traitement de données par ordinateur a permis d'accomplir cette tâche grâce au stockage sequentiel des fichiers de données surbande magnétique.

Cependant, pour une meilleure utilisation du potentiel des ordinateurs dans le traitement des données, il était nécessaire de pouvoir accéder celles-ci plus directement, sans être obligé de suivre le séquencement des fichiers sur des bandes magnétiques. Ce problème a été partiellement résolu grâce aux progrès du matériel, spécialement avec le développement des unités de disque, ainsi que par les progrès du logiciel avec l'utilisation de l'accès indexé, et des autres méthodes d'adressage.

Entre les années 50 et 60 les méthodes d'adressage étaient appelées "systèmes de fichiers". Elles étaient utilisées comme un interface entre le programme d'application et les données stockées. Elles permettaient l'accès aux données sans prendre en compte les détails des unités périphériques. Ces "systèmes de fichiers" ont été un pas important dans la technologie du traitement des données, mais avec l'extension de leur usage, plusieurs limitations sont apparues. Les "systèmes de fichiers" créent une grande dépendance entre les données et les programmes qui les utilisent. Tant que peu de programmes

utilisent les fichiers, il n'y a pas de problèmes, mais lorsque le nombre d'utilisateurs augmente, de sérieux problèmes apparaissent : a) la manière dont les données sont structurées dans les fichiers pour une application n'est pas nécessairement adaptée à une autre application, b) si le fichier doit être restructuré pour être utilisé par plusieurs programmes d'application, tous les programmes devront être modifiés, c) si plusieurs copies du fichier sont faites, pour que chaque copie ne soit utilisée que par quelques applications, des problèmes pour maintenir la cohérence des copies apparaissent [YAO.78].

Les sytèmes de gestion de base de données (SGBD ou DBMS en anglais) sont apparus pour résoudre ces limitations. Les SGBD isolent les programmes d'application des structures des fichiers. En fonction des besoins d'une application, les formats des fichiers peuvent changer physiquement tandis que la définition logique des données restera la même. Les SGBD augmentent l'indépendance des données par rapport au programme d'application, de telle manière qu'ils soient insensibles aux changements logiques ou physiques dans l'organisation des fichiers.

Un SGBD typique possède deux langages d'interface, un langage de définition des données (LDD ou DDL data définition language) et un langage pour la manipulation des données (LMD ou DML data manipulation language). Le langage LDD, utilisé pour définir la structure logique des données, n'est employé que pour préparer la base de donnée avant son utilisation. Par contre, le LMD est employé pour l'utilisation de la base de donnée, il sert à spécifier les données à accéder. Il est important que les programmes d'application n'accèdent la base de donnée qu'à l'aide du LMD, cela assure que les application n'auront une vision que de la structure logique des données et non de leur organisation physique.

Les autres principaux services rendus par les SGBD incluent, des méthodes d'accès, des mécanismes de synchronisation qui permettent l'accès simultané aux données par plusieurs programmes ou utilisateurs, et des mécanismes de sécurité et d'intégrité qui assurent la validité des données et qui préviennent contre les accès non autorisés.

Actuellement les recherches dans le domaine des SGBD sont axées sur trois principaux créneaux : facilité d'emploi (usability), performance et intégrité des données. Nous pouvons résumer les directions actuelles de la recherche par la table 1.

Problèmes des années 80	Utilisabilité	Performance	Intégrité des données		
			Fiabilité	Confidentialité	Cohérence
Sémantiques des données	x				x
Conception de Bases de données	x	x	x	x	x
Langages d'accès aux données	x			x	
Systèmes de BD réparties	x	x	x	x	x
Traduction des données et des programmes		x			x
Machines BD		x	x		

Table I.1 - Directions de la recherche en BD [YAO.78]

Notre intérêt se portera sur la dernière ligne de cette table, les processeurs base de données.

I.2 PROCESSEURS BASE DE DONNEES

De nos jours, la possibilité de garder de grandes quantités de données dans des mémoires de masse, n'est plus en rapport avec la capacité de les traiter dans les unités centrales. De plus en plus les SGBD ont des problèmes pour traiter ces grandes masses d'informations.

Cependant, deux principaux facteurs ont modifié ces dernières années le panorama des bases de données créant des conditions favorables à l'avènement d'un nouvel élément : le Processeur Base de Données. Ces deux facteurs sont les nouvelles technologies et les systèmes distribués.

Dans les nouvelles technologie le point le plus important a été l'arrivée du microprocesseur sur le marché avec son coût faible par rapport à sa puissance de traitement. Cela a ouvert les portes de son utilisation dans les grands systèmes.
Le concept de PBD implémente avec des microprocesseurs, apparait comme une solution possible pour obtenir de meilleures performances dans les systèmes de base de données.

L'idée associée à ce concept, est que de simples opérations de manipulation d'informations dans une base de données, puissent être incluses dans un processeur séparé qui travaille en parallèe avec le système central. De cette façon on bénéficie d'une exécution parallèle qui conduit à des niveaux de performances supérieurs à ceux des systèmes traditionnels.

Le deuxième facteur, l'interconnection entre processeurs, est l'objet ces dernières années d'un des plus actifs créneaux de recherche en architecture d'ordinateur. Les systèmes qui en résultent sont appelés

de "systèmes distribués", "ordinateurs à fonctions distribuées" ou "réseaux d'ordinateurs" [AND.75]. Ce sujet sera l'objet d'un prochain chapitre.

Ce sont donc ces deux facteurs : systèmes distribués et nouvelles technologies qui ont eu, et ont encore, un sérieux impact sur l'architecture des SGBD et de ses fonctions [FRY.76]. C'est à partir de leur influence, que le concept de PBD a surgi.

Le projet MAGE (Matériel Adapté à la Gestion d'Entités) [BER1.78] [BER.77] [BER2.78] [SAG.76] est axé dans ces deux directions : organiser une machine pour des systèmes distribués et utiliser des microprocesseurs. MAGE est un processeur base de données typique qui a pour fonction, comme son propre nom l'indique, la gestion d'une base de données.

Le prochain chapitre, chapitre II, présentera plusieurs projets de PBDs actuellement en étude, développement ou construction. Après cet aperçu du cadre des recherches dans ce domaine, les chapitres III et IV aborderont le milieu extérieur auquel le PBD ira se relier. Les autres chapitres présenteront l'aspect matériel du projet PBD-MAGE.

CHAPITRE II

PROCESSEURS BASE DE DONNEES

1. INTRODUCTION

2. PROCESSEURS BASE DE DONNEES

   1- CASSM

   2- ROME AIR DEVELOPMENT CENTER

   3- RAP

   4- RARES

   5- BACK END MACHINE DE SPERRY UNIVAC

   6- DATA COMPUTER

   7- TREFLE

   8- INFOPLEX

   9- BACK END COMPUTER DES LABORATOIRES BELL

   10- DATA BASE COMPUTER

   11- PROJETS LEECH et CAFS

   12- PROJETS A PARIS VI et A RENNES

II. PROCESSEURS BASE DE DONNES

II.1 INTRODUCTION

Les Processeurs Base de données, PBD, connus aussi sous le nom de Machines Base de Données, MBD, (Data Base Machines DBM), sont des processeurs spécialisés dans la fonction de gestion des données [BAU.76] . Les progrès dans la technologie des semiconducteurs LSI spécialement des microprocesseurs, ont contribué à créer des conditions techniques et économiques pour la réalisation et la construction de ressources de traitement, dirigées vers des fonctions spécifiques de gestion des base de données.

Le principe d'architecture d'un PBD est de transférer une grande partie du traitement BD de l'unité centrale d'un système, vers un processeur qui soit intimement associé à la mémoire de masse dans laquelle la BD est stockée [AND 76] .

Donc un PBD peut être défini comme l'ensemble du matériel et du logiciel qui offre ou concentre des ressources spéciales de traitement, pour supporter la partie de la gestion des données dans un système de traitement.

Un PBD est différent d'un "back end computer" [CAN.74] dans la mesure où ce dernier est un ordinateur qui réalise une fonction de gestion de données, pendant que le PBD est une machine spécialisée dans la gestion d'une base de données. Un PBD est une machine dont le matériel a été conçu pour réaliser une fonction BD, alors qu'un "back end computer" consiste à utiliser un ordinateur auxiliaire pour traiter les fonctions BD. Il est clair d'après cela qu'un PBD peut être utilisé comme un "back end computer", mais que le contraire n'est pas vrai. Ce détail est important dans la mesure où les PBD peuvent évoluer à partir des systèmes actuels centralisés vers les futurs systèmes distribués. Ces

processeurs peuvent aider des systèmes centralisés surchargés en prenant en compte leur tâche de gestion des données. En d'autres mots, les PBD peuvent aussi bien être utilisés dans les systèmes centralisés comme unités périphériques intelligentes, ou comme processeurs fonctionnels dans un système fonctionnellement distribué.

Deux principales motivations ont amené à la recherche de solutions matérielles pour la gestion des BD. La première a été la complexité et la taille croissante des logiciels de gestion de BD. Cette croissance est due en partie à l'augmentation des besoins des utilisateurs qui a amené à passer du traitement en mode "batch" (traitement par lots) au traitement en temps partagé, avec les problèmes de concurrence et de multiaccès que cela implique.

La deuxième motivation concerne les ordinateurs eux-mêmes ; ils n'ont pas été conçus pour la gestion des BD. Ces ordinateurs du type Von Neumann sont bons pour la préparation et l'exécution de programmes de traitement numérique et pour des traitements simples de données [HSI.77]. La gestion des BD est plutôt caractérisée par le stockage, la mise à jour et la gestion de grandes BD et donc a besoin d'opérations rapides pour la recherche et la mise à jour des données.

Tout cela a amené le lancement de plusieurs travaux de recherches sur les PBD.

II.2 PROCESSEURS BASE DE DONNEES

Les recherches actuelles en PBD sont menées principalement dans des directions telles que les processeurs associatifs pour des

BD relationnels, des processeurs avec des cellules logiques par tête d'unité disque, des processeurs avec des primitives base de données, des PBD spécialisés dans des réseaux hétérogènes, des PBD réalisés à l'aide d'ensembles de microprocesseurs, et d'autres. [MAR.78]

II.2.1. CASSM (Context Adressed Segment Sequential Memory) [COP.73]

A l'Université de Florida le projet CASSM a développé le concept de système disque caractérisé par un adressage au contexte. Ce système est capable de réaliser des manipulations de données sur la mémoire secondaire sans l'intervention de l'ordinateur. Ce projet a été l'un des premiers orientés vers les BD. Il utilise une logique associée à chaque tête de piste et est donc conçu pour des disques à têtes fixes. C'est son point faible, car le coût des disques à têtes fixes avec un logique par piste le rend trop onéreux pour être industrialisé dans les prochaines années. Par son coût, sa limite économique se situe aux environs de 10^8 octets de données [KAN.78]. Ce PBD a été généralisé de manière à pouvoir aussi bien travailler avec des modèles de BD hiérarchisées, réseaux ou relationnelles. CASSM a été implémenté expérimentalement à l'aide de disques souples à têtes fixes (fig II.1)

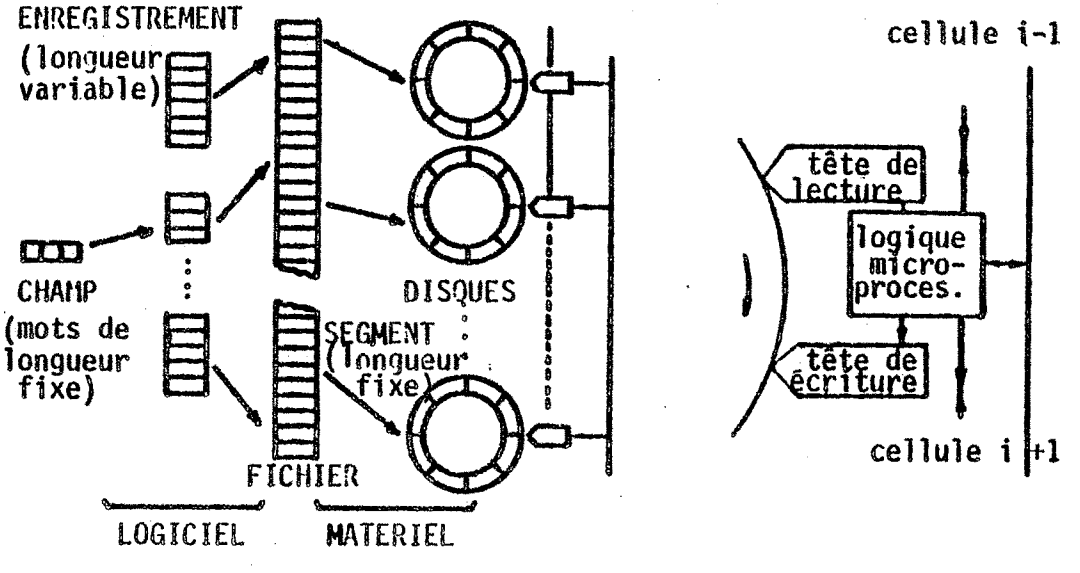


Fig. II.1 - Stockage d'enregistrements et une cellule du projet CASSM [LIP.78]

II.2.2 ROME AIR Development Center [FIO.73]

Une des opérations les plus communes réalisées par les SGBD, est l'utilisation de mots clés pour localiser les données. L'optimisation de cette opération permet donc d'avoir de meilleures performances. Une manière d'obtenir cette optimisation est l'utilisation de processeurs associatifs qui permettent une recherche associative sur des structures de listes inversées. Ces titres sont considérés comme le type d'organisation la plus intéressante pour des recherches d'enregistrement sur des systèmes conventionnels. Un système de ce type a été développé au "Rome Air Development Center". Il utilise un processeur associatif pour réaliser les opérations de recherche. La base de données est stockée dans des unités conventionnelles de mémoire secondaire et transférée page par page dans la mémoire associative, permettant des recherches de l'ordre de la microseconde, bien que le transfert de la page prenne de l'ordre d'une miliseconde.

II.2.3 RAP (Relational Associative Processor) [SCH.78] [OZK.77] [SCH.78] [SCH.76] [OZK.79]

A l'Université de Toronto, un autre projet, très semblable au CASSM, est développé avec des processeurs associatifs. Il a été projeté spécialement pour des BD relationnelles, avec un ensemble d'instructions en langage de haut niveau. Comme CASSM, il traite au niveau de la machine les opérations sur les données, ce qui permet d'économiser beaucoup de temps et ainsi de résoudre un des principaux problèmes des BD relationnelles. Son principe de fonctionnement est basé sur des cellules constituant les processeurs associatifs, où circulent les informations. Un prototype a été construit en utilisant des registres à décalage en technologie CCD.

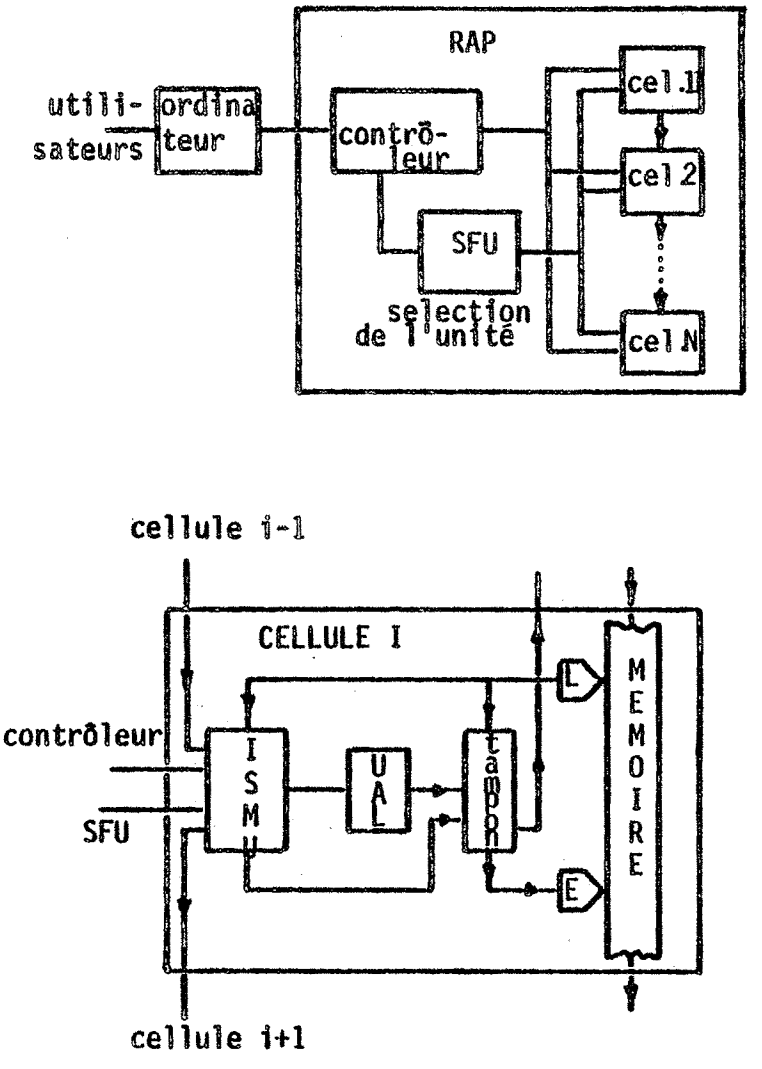


Fig. II.2 - Architecture et une cellule de RAP [OZK.79]

II.2.4 RARES (Rotating Associative Relational Store) [LIN.76]

Le projet RARES développé à l'Université d'Utah est lui aussi basé sur l'utilisation d'une mémoire associative, (CAM). Physiquement RARES est connecté à un ordinateur et à une mémoire tampon à travers

des canaux rapides. RARES utilise des disques ayant une tête par piste où sont stockés en "bandes" les enregistrements relationnels tels que SQUIRAL (Smart Query Interface for Relational Algebra). En raison du coût de leur matériel, le projet RARES ainsi que RAP et CASSM, ne sont économiquement utilisables que pour des configurations ayant au plus 10^8 octets.

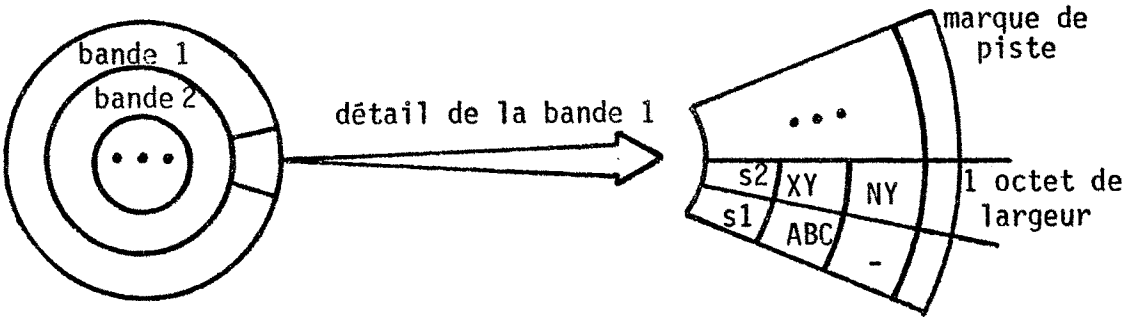


Fig. II.3 - Organisation des données dans le projet RARES [SMI.79]

II.2.5 Back end Machine de Sperry Univac [AND.76]

Le concept d'un processeur muni de primitives de base de donnée est églament un créneau de recherche. Sperry Univac a un projet d'une "back end machine" dans laquelle les fonctions de BD ont été inversées par microprogramme. Ce processeur peut être connecté directement à un ordinateur ou bien être utilisé dans un réseau.

II.2.6 Data Computer [MAR.75]

Dans les réseaux hétérogènes un processeur base de données spécialisé a été développé par Computer Corporation, sous le nom de Data Computer. Il est bâti à l'aide d'un DEC PDP-10 et est employé dans le réseau ARPA, offrant des services sur les structures de files inversées. Sa réalisation est logicielle.

II.2.7 TREFLE [MAS.78] [SCH.79]

En France le projet TREFLE développé à l'IRIA est une machine conçue spécialement pour des applications d'informatique documentaire et plus généralement la reconnaissance des structures et des contenus, la traduction des langages et la modification du format des fichiers. Un des aspects intéressants de ce projet, est le fait que le language d'interprétation de requête, permet de poser des questions avec un certain degré d'imprécision dans les mots et les phrases (manque, erreur ou excès de lettres dans les mots). La machine est microprogrammée dynamiquement et utilise des mémoires FIFOs pour l'entrée et la sortie des données. Une structure pipe-line pour les stations "filtres" (Fig. II.4) permet un traitement parallèle. Chaque station traitant une fonction spécifique par exemple, récupération de mots, récupération d'expression, récupérations de champs et autres. La caractéristique technique fondamentale de TREFLE est de traiter le flot de données en une seule lecture à la vitesse du flot provenant de la mémoire secondaire

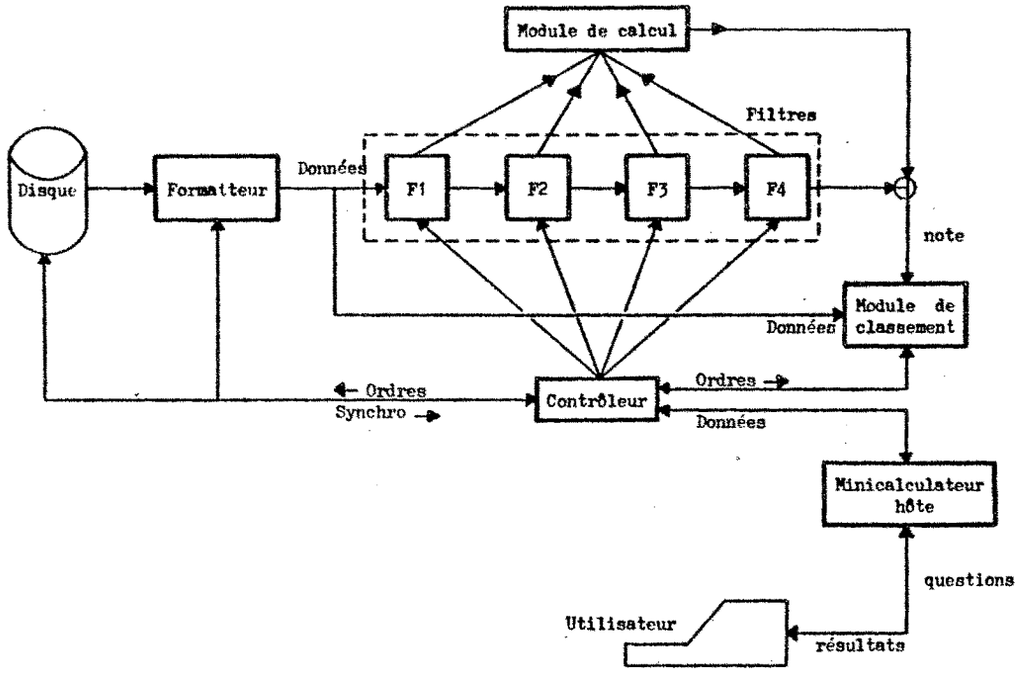


Fig. II.4 - Structure du processeur TREFLE [SCH.79]

II.2.8 INFOPLEX [MAD.75]

Dans les recherches sur les bases de données gérées par un complexe de microprocesseurs, nous trouvons le système INFOPLEX, développé au MIT. Ce système a un système de gestion des information très parallèle. INFOPLEX décompose fonctionnellement chaque requête de haut niveau à la base de donnée, en des segments exécutables en parallèle par des microprocesseurs. La mémoire et les microprocesseurs sont organisés d'une manière hiérarchique pour exploiter au mieux le parallélisme inhérent aux accès concurrents à une BD. Ce concept de décomposition des requêtes de la BD peut être appliqué également à un niveau plus élevé comme par exemple dans un réseau de miniordinateurs. (Fig II.5)

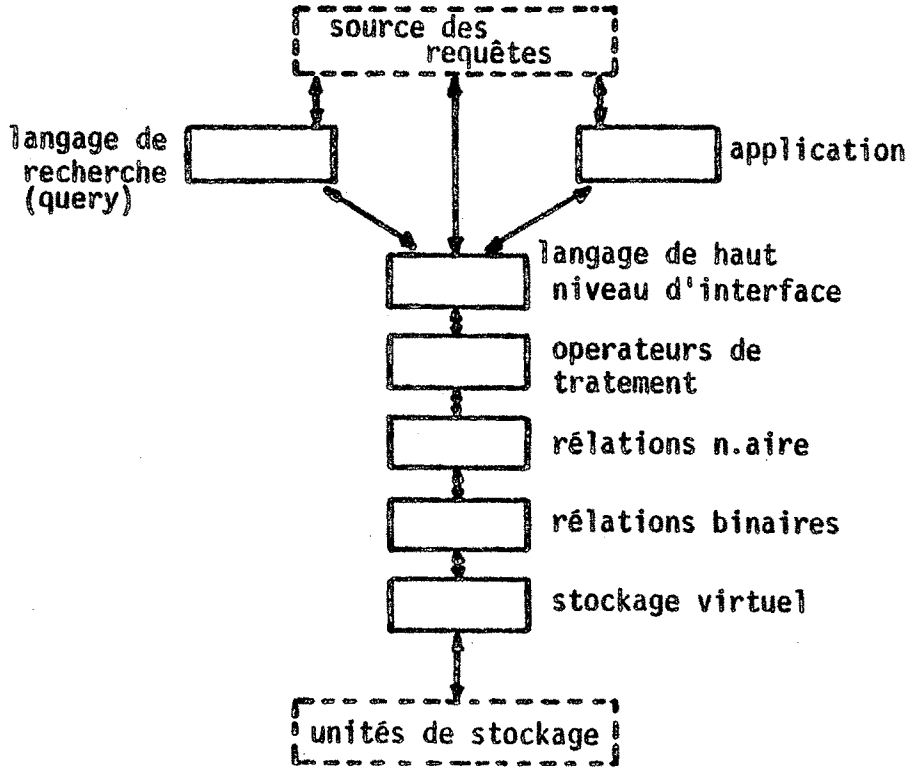


Fig.II.5 - Décomposition hiérarchique des fonctions dans INFOPLEX [MAD.75]

II.2.9 Back end computer des Laboratoires BELL [CAN.74]

Le premier projet qui ait fonctionné dans le cadre des recherches sur PBD, est le projet d'un "back-end", processeur prototype appeké X DMS, qui a été développé aux Laboratoires Bell. Ce système utilisait un Univac 1108 pour exécuter les programmes d'application, pendant qu'un Meta-4 de Digital Scientific Corp. réalisait la fonction "back-end" de BD. Le système logiciel utilisé était le DMS-1100 dérivé des spécification de CODASYL.

II.2.10 Data Base Computer, DBC [BAU.76] [MSI.77] [KAN.78]

Un autre projet qui utilise des systèmes associatifs est le projet DBC développé à l'Université de Ohio. Il est composé de parties fonctionnellement spécialisées qui réalisent une BD basé sur les attributs. Ce projet, à l'opposé de RAP, RARES ou CASSM, peut supporter des grandes BD et offre des outils matériels pour renforcer les aspects liés à la sécurité. Le DBC contient des parties spécialisées pour le stockage des informations du fichier indexé, pour le stockage de la BD et pour la sécurité. Les trois premiers sont des différentes formes de mémoires à adressage par le contenu (CAM) organisées par blocs. Cette machine utilise les possibilités des mémoires CAM non seulement pour aider à retrouver rapidement les données, mais aussi pour permettre des mises à jour rapides de la BD. Le DBC utilise actuellement des unités de disque à bras mobile pour stocker la BD dans lequel les cylindres sont individuellement adressés. Cela permet à ce système de gérer des très grandes BD. Les unités disques utilisées par DBC doivent permettre l'activation simultanée de toutes les têtes d'un cylindre. Les processeurs associatifs (TIPs) accèdent à chaque instant toutes les tête d'une unité de disque. Des circuits de comutation permettent de choisir l'ensemble de têtes qui seront activées. (Fig. II.6)

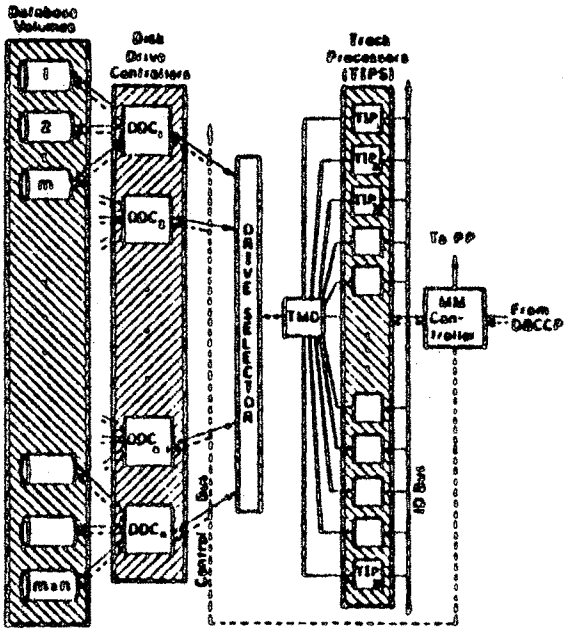


Fig II.6 - Organisation de la Mémoire de Masse dans DBC [KAN.78]

II.2.11 Projets LEECH et CAFS

Plusieurs autres projets existent. Par exemple en Angleterre, il y a le projet de la machine LEECH [MCG.76] construite à Glasgow basée sur un processeur de traitement de vecteurs pour traiter les relations. Un autre projet anglais est CAFS (Content-Addressed File Store) [BAB.79] développé par ICL à Stevenage qui utilise un processeur spécialisé de recherche, et des mémoires RAMs adressable au bit.

II.2.12 Projets à PARIS VI et à Rennes

En France quelques projets hors TREFLE et MAGE, à notre connaissance, s'intéressent aux aspects matériels de la recherche et du stockage des données. Un projet à l'Université PARIS VI [QUA.77]

est basé sur l'utilisation d'un opérateur câblé d'accès associatif à une mémoire secondaire. Une maquette a été réalisée en utilisant un ordinateur MICRAL.S pour le contrôle de l'opérateur câblé. La mémoire secondaire employée était une unité de disque souple.

A l'université de Rennes [UNG.77] un projet aborde l'étude d'un processeur associatif pour des bases de données relationnelles, réalisable à travers une grande mémoire associative réalisée à l'aide de circuits CCDS.

CHAPITRE III

SYSTEMES DISTRIBUES

1. INTRODUCTION

2. SYSTEMES DISTRIBUES

3. MECANISMES DE COMMUNICATION

4. PROCESSEUR BASE DE DONNEES ET SYSTEME DISTRIBUE

III. SYSTEMES DISTRIBUES

III.1 INTRODUCTION

Dans les ordinateurs actuels le ou les processeurs banalisés qui le composent sont obligé d'effectuer les tâches les plus diverses telles que : traitement de programmes, gestion d'une imprimante, gestion du système, gestion des données de la BD, etc...

Un des points les plus critiques des ordinateurs conventionnels est la modularité [JEN.75] . Normalment une configuration d'un ordinateur central peut difficilement évoluer vers des configurations plus performantes. Tout changement important de configuration oblige de profonds changements matériels et souvent du logiciel aussi.. Les multisystèmes, systèmes à plusieurs processeurs, sont une des techniques qui paraissent donner le plus de modularité et donc de pallier à ce problème. Jusqu'à récemment, les multisystèmes n'étaient pas développés compte tenu du coût élevé de leur matériel ; ils n'existaient qu'en tant que systèmes spéciaux. Les derniers progrès de la technologie, spécialement avec les microprocesseurs, ont transformé cet état de chose et les processeurs ne sont plus maintenant les ressources onéreuses. Le coût principal provient maintenant du logiciel et des périphériques. Cela a donné un essort aux recherches dans le domaine des processeurs distribués, des ordinateurs à fonctions distribuées et des réseaux d'ordinateurs. Les principaux bénéfices qu'apportent de telles architectures sont : le temps de réponse plus court, les possibilités de distribution géographique, l'augmentation de la disponibilité et la sûreté de fonctionnement, le partage des ressources et l'augmentation de la tolérance aux fautes, l'augmentation du débit, sans compter la modularité et d'autres propriétées [THU.78]

Le chapitre précédent a montré les principales recherches sur les PBD. Cependant un Processeur BD tout seul ne fait rien, il a besoin d'être relié, d'un côté à un ordinateur ou à un système distribué pour recevoir des ordres, et de l'autre côté à une mémoire secondaire pour stocker les données de la BD. Ce chapitre, chap.III, aura pour but de situer le PBD par rapport à son milieu extérieur et plus spécialement les systèmes distribués alors que le prochain chapitre, chp. IV, abordera les mémoires secondaires.

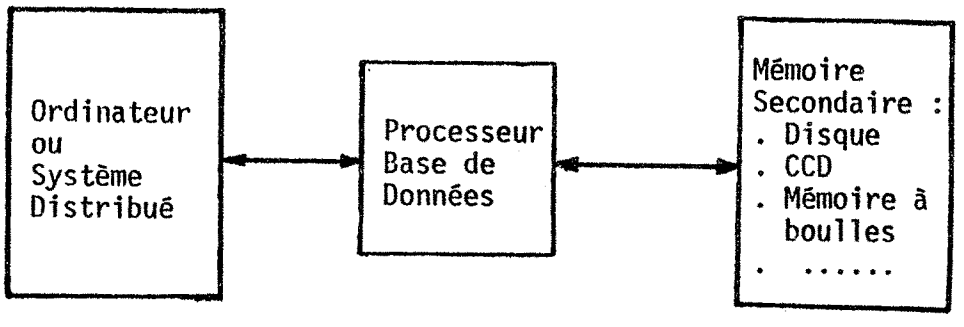


Fig III.1 - Connexions d'un PBD avec l'extérieur

III.2 Systèmes Distribués

Il y a plusieurs discussion et divergences à propos de la définition de systèmes distribués. Nous optons pour celle la plus largement acceptée. Un système distribué est défini comme un système où un traitement est distribué entre des processeurs qui sont physiquement interconnectés, mais dont le degré de couplage logiciel est variable [THU.78]

D'autre part par traitement distribué on entend un simple ensemble logique de fonctions de traitement qui sont implémentées dans un nombre d'unités physiques de telle manière que chaque unité n'effectue seulement qu'une partie du travail requis par le traitement total [BOO.76]

Dans un système distribué au moins quatre composants peuvent être distribués physiquement : les données, le traitement, le contrôle (système opérationnel par exemple) et le matériel. Les principales caractéristiques que doivent avoir un système pour être distribué sont [ENS.78] :

- la "multiplicité"de resources tant physiques que logiques.
- une "distribution physique" des ressources matérielles ainsi que logicielles du système. Ces ressources inter agissant au travers d'un réseau de communication.
- un "système opérationnel de haut niveau", de manière à intégrer et unifier le contrôle des ressources du système.
- un "système transparent" pour l'utilisateur qui n'a pas besoin de connaître les ressources utilisées.
- une "coopération autonome", chaque composant du système (ressource) coopère avec les autres composants, en maintenant son autonomie et donc le pouvoir de refuser un travail à tout instant.

Après ces quelques définitions et caractéristiques d'un système distribué, il est intéressant de voir rapidement ce qui le différencie d'un système centralisé.

Dans les ordinateurs conventionnels d'aujourd'hui, le traitement centralisé peut être vu comme une structure multicouches, dans laquelle le centre représente les tâches de l'utilisateur, la première couche les ressources du systèmes et la couche externe les unités périphériques [ANC.78] (Fig.III.2)

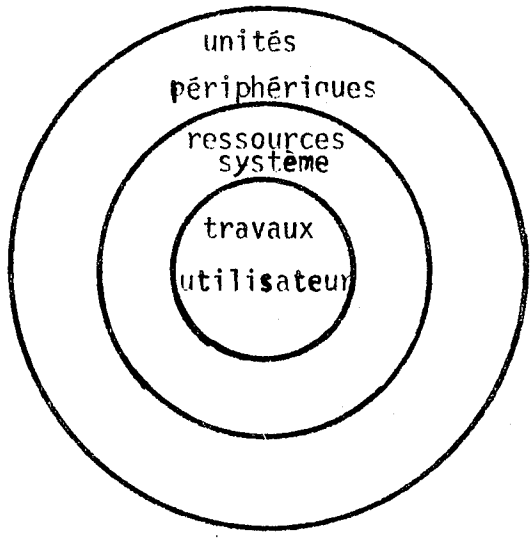


Fig III.2 - Système centralisé

En utilisant ce point de vue, un système décentralisé est le résultat de l'éclatement des couches systèmes. C'est la situation d'un certain nombre d'ordinateurs actuels où on retrouve des processeurs auxiliaires exécutant des tâches spécialisées telles que le contrôle de prériphériques, la gestion de base de données, FFT, etc (Fig III.3).

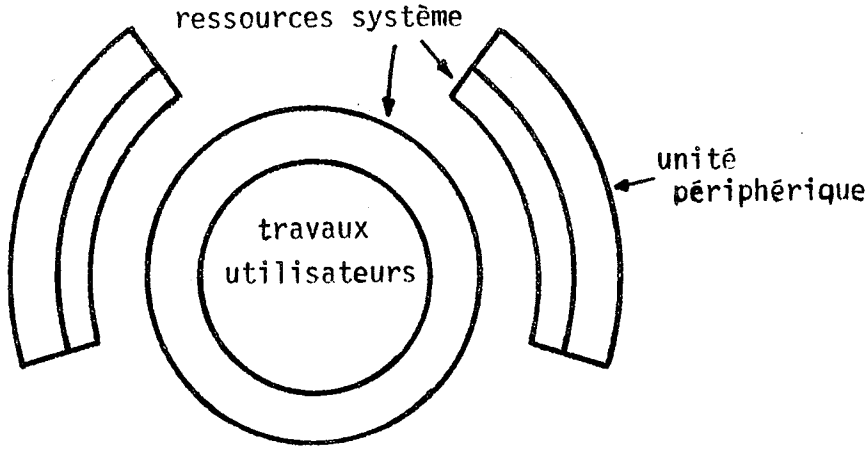


Fig. III.3 - Système décentralisé

Le résultat de l'éclatement de toutes les couches, y compris le noyau central, donnera un système distribué. Cet éclatement va

créer des modules spécialisés, dont quelques uns vont être par exemple, des modules de gestion d'unités périphériques avec les ressources systèmes nécessaires ; les modules de ce type sont appelés processeurs fonctionnels. L'autre famille de modules est constituée des processeurs de traitement ou utilisateurs, créés par l'éclatement des tâches utilisateur (Fig III.4)

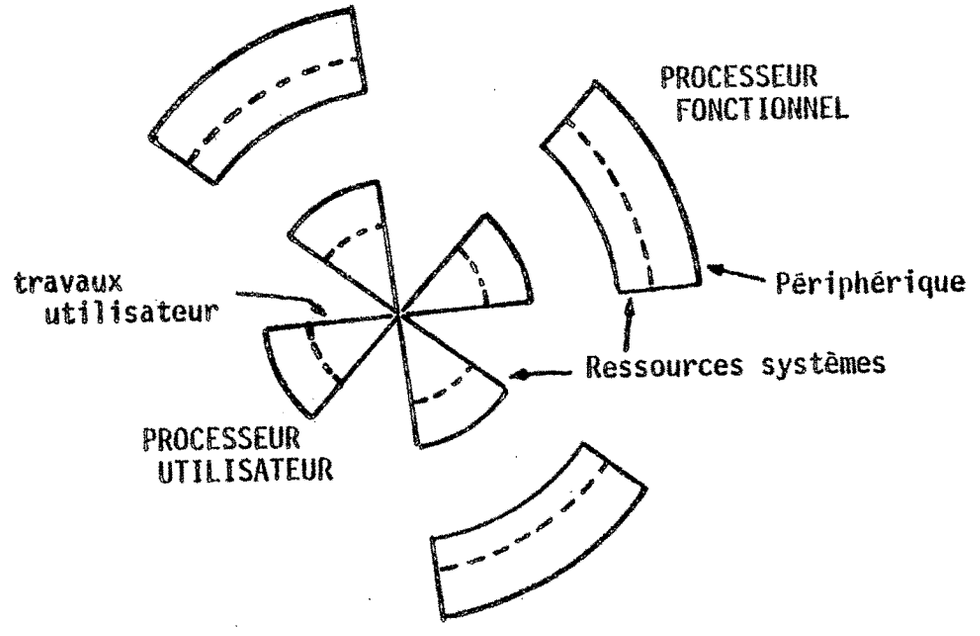


Fig III.4 - Système distribué

Le système CORAIL [POU.77] [ANC.77] représente bien ce dernier type de structure. Ce projet utilise une décomposition des processeurs en processeurs fonctionnels et processeurs utilisateurs. Les processeurs fonctionnels sont dédiés à des fonction spécialisées comme la gestion de la bibliothèque, le contrôle des unités périphériques, gestion de base de données, etc ; pendant que les processeurs de traitement (ou utilisateurs) exécutent des programmes utilisateur. Dans le système CORAIL, ces derniers processeurs sont alloués aux utilisateurs humains via des terminaux (Fig.III.5)

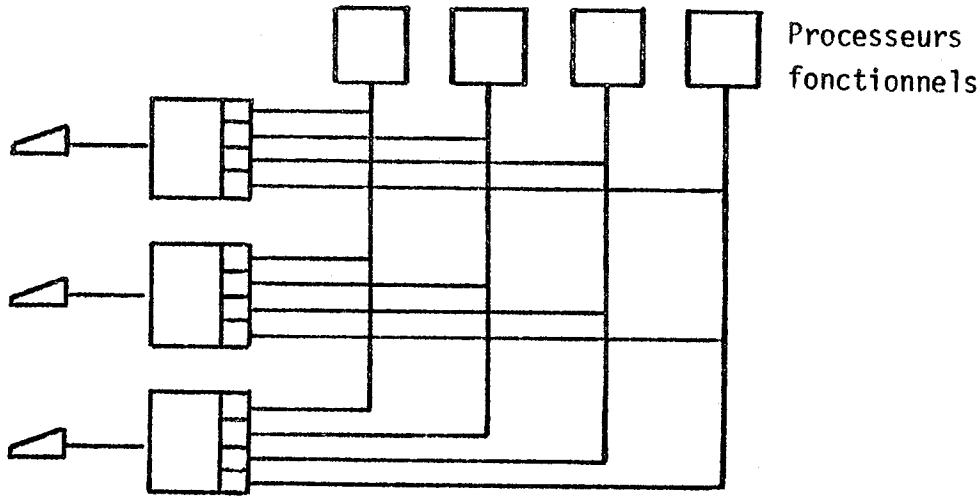


Fig.III.5 - Machine CORAIL [POU. 77]

III.3 Mécanismes de communication

Le grand problème lors de l'utilisation de divers processeurs coopérant pour réaliser une tâche, est la manière de les faire communiquer entre eux. Ce problème n'est pas seulement un problème de communication entre processeurs mais aussi, de communication avec les unités périphériques et avec le milieu extérieur. La détermination du meilleur mécanisme de communication entre deux unités quelconque est très lié à l'importance du flux dedonnées que devra véhiculer ce lien. En effet, si on étudie les flux de communication existant aux différants niveaux d'un système, on constate que ceux-ci sont plus grands près des organes périphériques et diminuent à mesure que l'on se rapproche de l'utilisateur [ANC.78]

Le graphique de la Fig. II.6 montre que plus le dialogue est d'un niveau élevé, plus bas est le flux de données. Par contre plus les ordres sont directs et moins intellingents, plus grand est le flux nécessaire au dialogue.

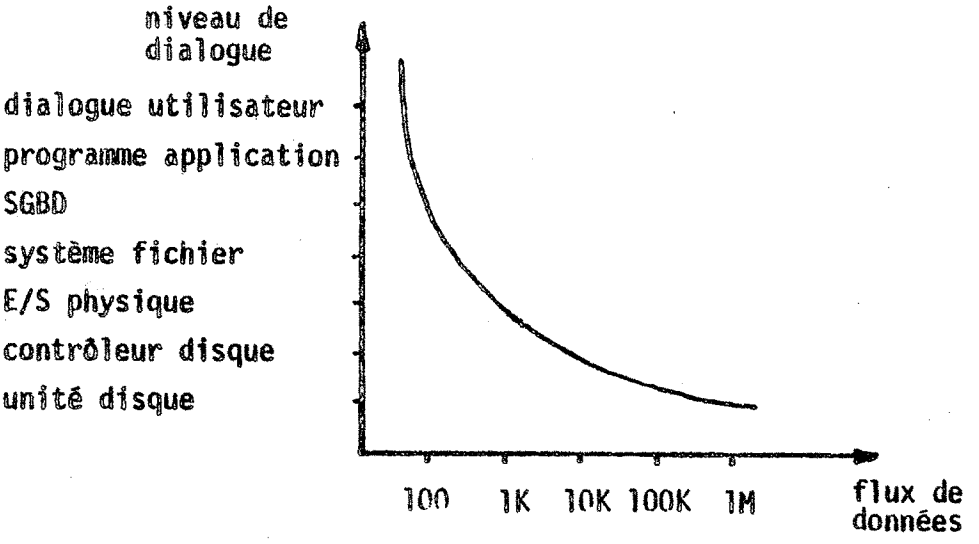


Fig.III.6 - Rapport entre le niveau de dialogue et le flux de données

Cela conduit, pour traiter une fonction "α" à avoir un degré d'intelligence "I" qui sera inversement proportionnel aux flux de données "ϕ"

$$\alpha = f\ (I.\phi)$$

Donc pour résoudre une certaine tâche à un certain niveau du système, plus l'intelligence est utilisée, moins le flux de données est nécessaire pour dialoguer avec cette tâche.

α réprésente le degré de dialogue nécessaire pour communiquer entre deux éléments d'un système. A partir de cette relation et avec la donnée du flux à un niveau déterminé, on peut déterminer quel mécanisme de communication doit être utilisé. Un résumé de ces mécanismes d'interconnection entre processeurs est présenté par Anderson et Jensen [AND.75] (voir Fig. III.7).

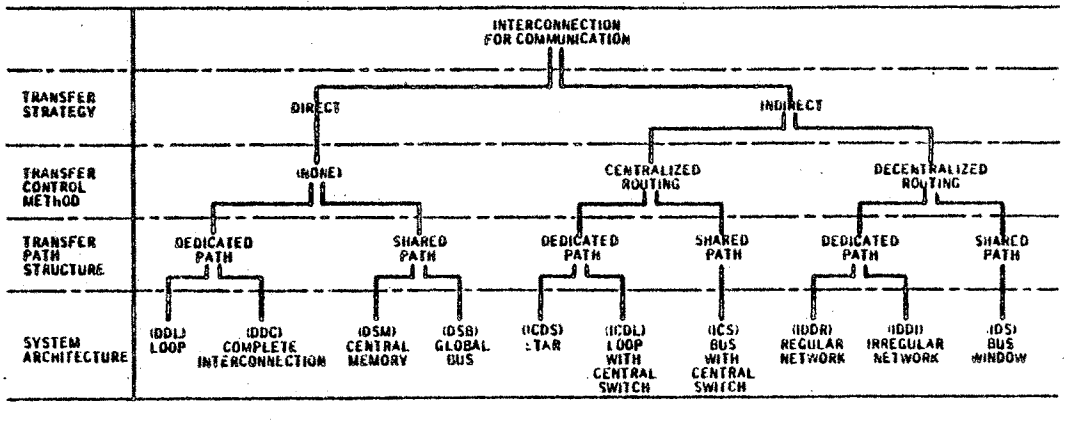


Fig.III.7 - Mécanismes d'interconnection [AND.75]

Deux principaux types d'interconnexion existent, les directes et les indirectes. La communication directe peut être réalisée à travers le partage d'un lien de connexion ou à travers des liens privés entre processeurs, pendant que l'indirecte utilise un commutateur pour aiguiller les données. La communication indirecte sera centralisée quand le commutateur est centralisé et décentralisée quand le commumateur l'est aussi. D'une manière résumée, ce sont là les types de mécanismes disponibles pour permettre de réaliser un lien entre deux unités.

III.4 Processeur Base de Donnée et Système Distribué

Un exemple, dans le cas de la machine CORAIL, est la manière la plus simple de situer le PBD par rapport à un système distribué. En reprenant la Fig.III.5, le PBD est représenté par l'un des processeurs fonctionnels. Généralement dans les configurations de la machine CORAIL, il y a au moins un bus commun à tous les processeurs du système et selon les différents débits demandés pour les échanges d'information, les processeurs fonctionnels peuvent disposer de bus privés ou partagés selon leur utilisation.

Les processeurs du système communiquent par échanges de messages. L'envoi d'un message par un processeur expéditeur provoque la génération d'un "message écho" par le processeur destinataire, qui avertit son interlocuteur de la bonne ou mauvais réception du message [ANC.77] [HAD.79]

Les processeurs du Système CORAIL étant totalement indépendants, le système d'exploitation est distribué sur les différents processeurs. Chaque processeur est donc capable de communiquer

avec les autres processeurs du système grâce au dispositif physique qui lui est attaché et également d'assurer son rôle dans le système d'exploitation.

Utilisant la décomposition proposée pour CORAIL [HAD.79], le PBD serait décomposé en trois modules : a) module d'émission réception de messages, b) un module de contrôle de la communication et c) le module d'application qui peut être le PBD en lui-même.

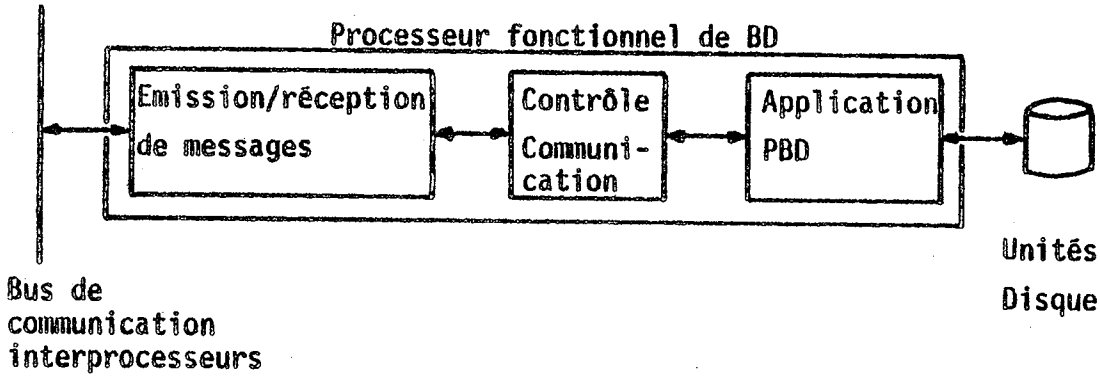


Fig. III.8 - Découpage du PBD dans un environnement de système distribué

Le module d'émission/réception de messages s'occupe du protocole physique de communication et le module de contrôle exécute la vérification des messages, la mise à jour des tables de communication etc...

Ces trois niveaux de modules représentent les couches du système d'exploitation : 1) la communication entre processeurs ; 2) l'organisation des travaux ; 3) la gestion des ressources et l'interprétation des requête utilisateur selon le type de processeur. La première couche possède un lien physique de communication avec les couches correspondantes des autres processeurs, pendant que les autres couches utilisent les liens virtuels de communication entre les couches de même niveau.

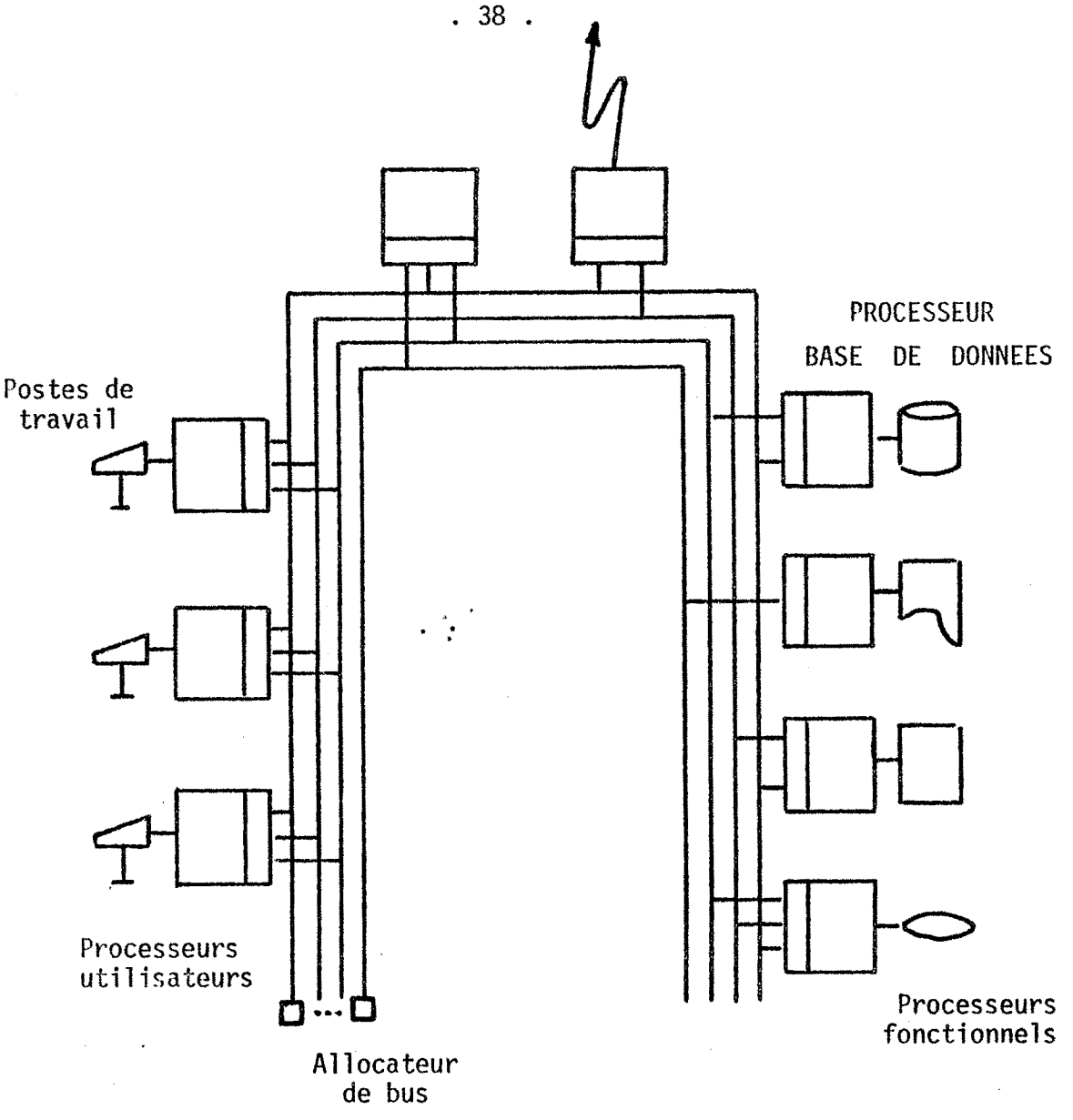


Fig.III.9 - Le PBD dans un système distribué du type CORAIL [HAD.79]

CHAPITRE IV

MEMOIRES SECONDAIRES

1- INTRODUCTION

2- UNITES DISQUE

3- TYPES D'UNITE DISQUE

4- CODE CORRECTEURS

5- PLUSIEURS BRAS

6- DEPLACEMENT DU BRAS

7- TEMPS DE ROTATION

8- MEMOIRES TAMPON

9- CACHE D'UNE PISTE

10- NORMES D'INTERFACE SMD

11- TYPES DE LIAISONS ENTRE CONTROLEUR DISQUE ET UNITES DISQUE

12- CONCLUSION

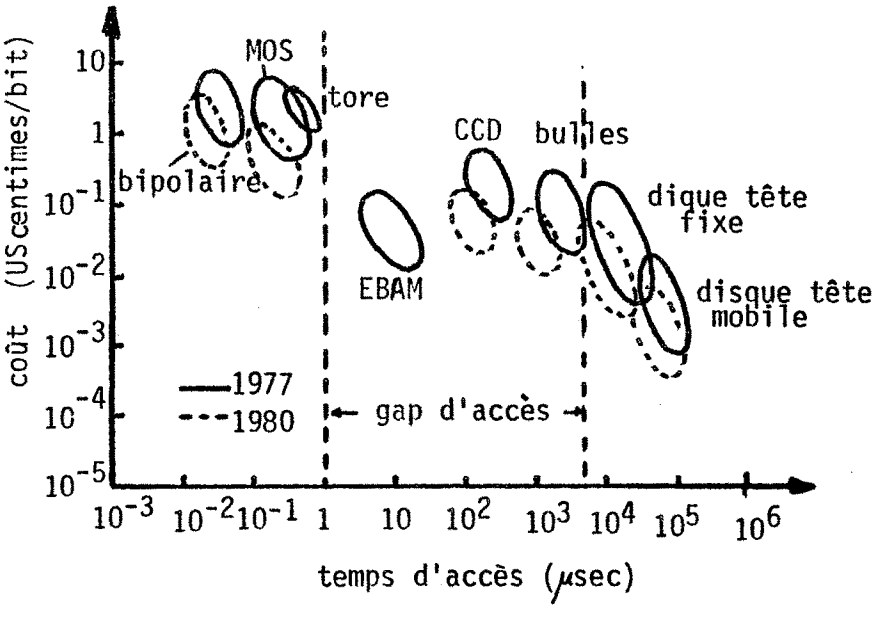
IV. MEMOIRES SECONDAIRES

IV.1 INTRODUCTION

Les derniers développements technologiques des mémoires ont créé de nouvelles possibilités pour le stockage des données. Ces nouvelles technologies sont en train d'influencer aussi bien les architecteures des mémoires principales que celles des mémoires secondaires. Tout doucement les mémoires CCD (charge coupled Devices) [TOO1.78] [BHA.79] ainsi que spécialement les mémoires à bulles magnétiques (MBM) [TOOO1.78] [YPM.75] prenent leur place à côté des unités disque dans le stockage des données. Il ne faut pas oublier aussi les mémoires adressables à faisceau d'électrons (EBAM), ces mémoires cependant ne sont pas encore au point et ont un intérêt économique à partir de 30 M bits de capacité. Cette restriction étant due au coût élevé de circuits auxiliaires nécessaires et à la fabrication des EBAMs.

Historiquement, il y a toujours eu un "gap" de performance/ coût entre les technologies de mémoires principales et les technologies économiques des mémoires de masse. Cependant, avec l' arrivée des mémoires CCD à bulles ce "gap" tend à disparaître. Cela est vu clairement dans la figure IV.1 qui présente la relation entre le coût par bit et le temps d'accès à l'information pour les différentes technologies connues.

Dans la même figure est aussi présentée l'évolution des technologie entre l'année de 77 et celle de 80 (estimation). On observe que le prix des mémoires CCD et bulles décroissent beaucoup plus rapidement que celui des unités de disque, de telle manière qu'ils deviendront rapidement de sérieux concurrents. L'utilisation de ces nouvelles technologies dépend donc principalement de deux facteurs : le prix du bit et l'évaluation de la performance en termes de temps d'accès.



Fif IV.1 - Evolution des diverses techonologies par rapport au prix du bit et du temps d'accès
[TOO.78] [CAS.78]

Le prix du bit de mémoire est dans la plupart des technologie lié au volume de la mémoire, car il y a des zones limites de performance/coût au delà desquelles il n'est plus intéressant d'utiliser une technologie. La figure IV.2 donne ces zones par technologie d'après des valeurs actuelles. On peut voir dans ces courbes que par exemple, les unités disquettes sont moins performantes que les CCDs ou MBMs en dessous de 2 M bits de capacité.

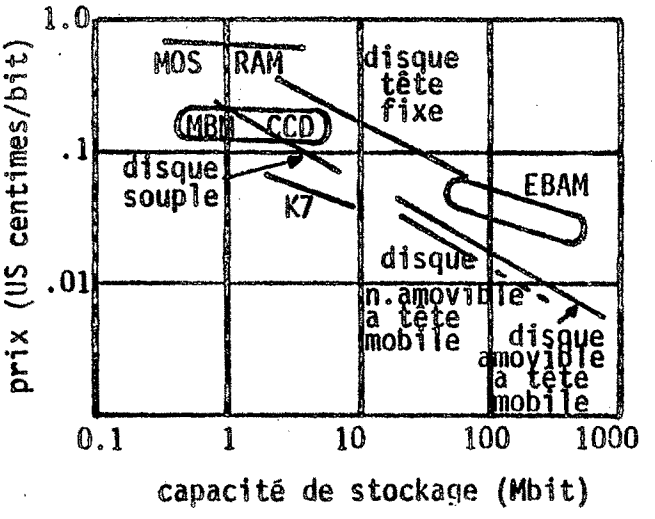


Fig IV.2- Prix du bit par rapport à la capacité de stockage [T00.78]

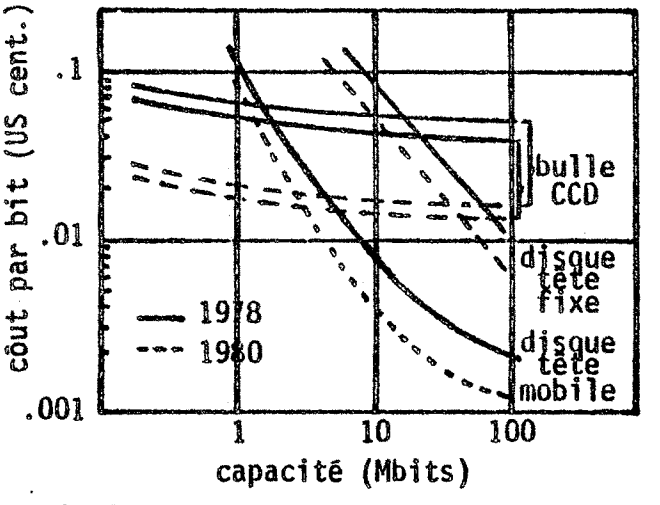


Fig IV.3 - Evolution des 3 technologies [T00.78]

Compte tenu de leur prix actuel, les principales application pour ces mémoires, actuellement, est leur utilisation comme des mémoires tampons (cache) entre les mémoires principales des ordinateurs et les unités disque. Les mémoires à bulles seront plutôt employées pour des applications pour lesquelles il y a un intérêt de maintenir les informations après la coupure de tension car elles sont non-volatiles. Elles remplaceront aussi des disques à têtes fixes de faible capacité. Les CCDs, de leur côté, seront emplyés dans des cas où le temps d'accès (latency time) doit être court, car il y a un facteur de 10 entre le temps d'accès des CCDs et celui des MBMs.

En comparant les mémoires à bulles avec les unités conventionnelles de stockage magnétique, les disques, deux points apparaissent. Le premier est que les unités à disque ont un prix initial relativement élevé par bit dû à toute la mécanique et l'électronique nécessaire. Donc le coût initial du bit est élevé, mais ils se dilue avec la croissance de la capacité de stockage de l'unité. Cela n'est pas le cas de MBMs qui ont aussi un coût initial, mais beaucoup plus faible, ce qui leur permet aujourd'hui de remplacer des unités de disque à bras mobile d'une capacité inférieure à 1 M bit. Cela est très faible même non significatif, mais doit être pris en compte car pour les unités à tête fixe ce seuil de remplacement arrive à 9 M bit (Fig IV.3).

Le deuxième point est que les mémoires à bulles sont une technologie nouvelle et qui est encore dans une courbe d'apprentissage tandis que la technologie des unités disque est vieille de 30 ans. Cette évolution amènera vers les années 80 les bulles à être compétitives jusqu'à 4M bits pour les unités de disque à bras mobile et jusqu'à 40 M bits pour les disques à têtes fixes. La figure IV.4 donne les courbes estimées du coût du bit par rapport à la capacité de stockage pour les années 80. Il ne faut pas oublier que le phénomène actuel est tout à fait comparable à celui qui est arrivé pour les mémoires à "tore" au moment de la venue des mémoires intégrées.

La conclusion logique est que dans un premier temps ces nouvelles technologies seront employées plutôt comme des mémoires auxiliaires telles que les "caches" des unités disque ou pour des cas spéciaux tels que les appareils portables. Les unités à disque continueront à garder une grande partie du marché des mémoires de masse, au moins jusqu'aux années 90, résistant à l'attaque des mémoires à semi-conducteurs. D'autre part, il est évident que la technologie

des unités à stockage magnétique (unités disque), elle aussi continuera à progresser. La figure IV.4 montre ce progrès tout au long des 15 dernières années. Un autre facteur qui plaide en faveur des unités disque à bras mobile est le fait qu'elles sont amovibles et portables, ce qui permet l'échange et le stockage des "piles de disques" ; cela n'est pas le cas des mémoires à bulles.

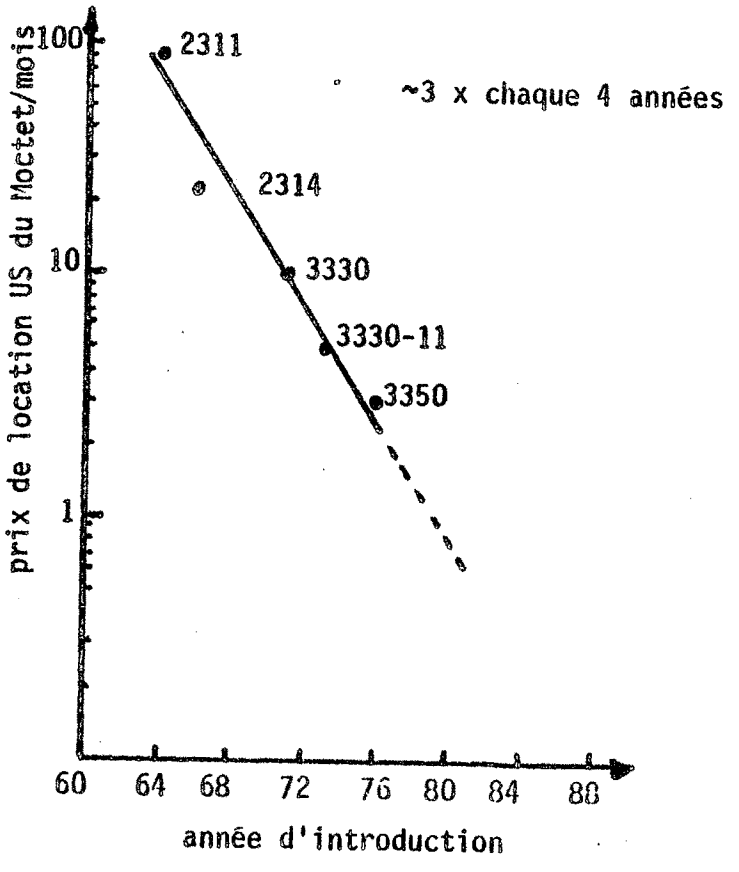


Fig IV.4 - Diminution du prix du bit dans les unités disque tout au long des dernières années [CAS.78]

Une projection vers le futur montre que des unités actuelles de 300 M octets (pour 77) on passera en 82 à des unités de 1000 M octets ou plus [SLA.78] . Les fréquences de transfert passeront pour la même époque de 10 MHz a environ 30 MHz et le prix passera de 300 FF le mégaoctet à moins de 200 FF le méga octet.

IV.2 UNITES A DISQUE

Depuis le commencement des ordinateurs a été ressenti le besoin d'accéder à de grands volumes de données rapidement. Ces besoins ont abouti vers 1950 à la création des unités à disque magnétique. Depuis leur création, ces unités n'ont pas beaucoup évolué dans leurs fonctions de base. Leur coeur est basé sur un nombre restreint de composant technologiques : a) le disque lui-même, sa surface magnétique qui est le milieu utilisé pour l'enregistrement des données; b) les transducteurs mégnétiques ou têtes, qui permettent l'enregistrement et la lecture des données de la surface du disque ; c) le bras qui déplace les têtes de manière à pouvoir accéder tous les cylindres ; d) les circuits de commutation qui permettent le partage par plusieurs têtes des mêmes circuits de lecture et d'écriture.

Un disque magnétique est une surface de métal circulaire avec une couche de particules ferro-magnétiques déposée sur un ou les deux côtés ; c'est cette couche qui constitue le milieu pour le stockage des données. Un certain nombre de ces disques magnétiques peuvent être empilés l'un sur l'autre de manière à avoir une "pile de disques " (disk pack). La surface de chaque disque est divisée en zones concentriques appelées pistes et ces dernières sont aussi divisées en blocs appelés secteurs. Les secteurs sont les unités d'adressage de stockage. Les données sont lues ou écrites sur les secteurs grâce à une tête de lecteur/écriture qui flotte sur un coussin d'air au-dessus de la superficie du disque en rotation.

De tous ces facteurs composant d'un disque on peut en déduire certains paramètres qui permettent de mesurer la performance des unités disque [HAU.75] : 1) le temps moyen d'accès ; 2) le prix du million d'octet stocké ; 3) le temps moyen de rotation (latency) ; 4) le temps de transfert des données ; 5) la capacité par actuateur (bras).

L'objectif des prochaines sections n'est pas d'aborder tous les détails et possibilités d'évolution des unités disque, mais plutôt de discuter quelques détails qui nous paraissent des plus intéressants dans les actuels changements de la technologie des unités disque et qui ont l'air d'apporter le plus d'améliorations de performances.

IV. 3 TYPES D'UNITES DISQUE

Il existe deux types communs d'unités disque : ceux à têtes mobiles et ceux à têtes fixes. Dans un disque à tête mobile les têtes de lecture/écriture sont attachées à un bras mobile formant une espèce de peigne. C'est ce bras mobile associé au peigne de têtes qui permet à l'unité disque d'accéder un certain cylindre. Un cylindre étant composé de toutes les pistes situées sous les têtes de lecture/écriture pour une position déterminée du bras. Quand certaines données d'un secteur doivent être accédées, l'unité positionne d'abord les têtes sur le cylindre qui contient la piste et donc le secteur désiré et après, active la tête propre à la piste où sont les données. Le temps employé dans la recherche du bon cylindre est appelé "temps de recherche" (seek time). Après l'arrivée sur la bonne piste il faut tenir compte du temps d'attente de rotation du disque pour que les têtes puissent être positionnées au début du bon secteur pour sa lecture ou son écriture. Ce temps est appelé le "délai rotationnel" (latency time). Une opération de lecture ou écriture peut être faite sur plusieurs secteurs, le temps utilisé pour cette opération à partir du premier secteur à transférer jusqu'au dernier est connu sous le nom de "temps de transfert" (transmission time). Donc le temps total nécessaire pour réaliser une opération d'entrée/sortie sur une unité disque est constitué de l'addition du temps de recherche, du délai rotationnel et du temps de transfert.

Un autre type d'unités disque est constitué par les disques à têtes fixes. Dans un disque à têtes fixes chaque piste de la surface du disque possède sa propre tête de lecture/écriture. Pour sélectionner une piste, au lieu de déplacer la tête jusqu'à la piste choisie, il suffit de commuter électroniquement la tête de la bonne piste. Le temps de recherche pour ces unités est donc quasiment nul. Le temps total pour une opération d'entrée-sortie est seulement l'addition du temps de transfert et du délai rotationnel. Cela représente une assez grande économie de temps par rapport aux unités de disque à têtes mobiles. Par contre, le prix en est beaucoup plus élevé vu la multiplicité des têtes et la faible capacité due au nombre relativement réduit de pistes.

IV. 4 CODES CORRECTEURS

L'augmentation de la densité d'enregistrement sur les surfaces du disque avec tous les problèmes qui l'entourent oblige de plus en plus à avoir des codes correcteurs (ECC, Error Correcting Codes) pour vérifier et corriger les données enregistrées. Jusqu'en 1970, tès peu avait été fait de ce côté, seul un code détecteur d'erreur (CRC; Cyclic Redundancy Check) était utilisé pour indiquer si l'information lue sur le secteur était juste ou non. Maintenant, il devient de plus en plus standard, d'associer un code correcteur à la fin de tout bloc de donnée (secteur). Le choix de ce code ECC doit être fait de manière judicieuse pour que sa longueur par rapport à la quantité total d'information, ne soit pas trop grande. Plus il y a de bits à corriger, plus long est le code correcteur. Les unités de disque d'AMPEX [T.AMP.77] ajoutent 7 octets de ECC pour 560 octets de données. Il faut obtenir un compromis entre le nombre d'erreurs à corriger et la taille du code ECC par rapport à l'information.

IV. 5 PLUSIEURS BRAS

Une des caractéristiques importantes que doit avoir toute unité de disque est la fidélité de positionnement. L'unité doit être capable de revenir à tout emplacement physique de sa surface d'enregistrement. Plusieurs variables sont concernées par cette fidélité [HAU.75] :

1) Toutes les têtes de toutes les unités doivent être placées à la même position relative vis-à-vis de l'axe. Cela permet d'échanger les disques d'une unité à l'autre.
2) L'alignement des "piles de disques" : les "piles de disques" (disk pack) doivent être construites de manière qu'à chaque fois qu'elles sont montées, elles gardent le même alignement pour les cylindres.
3) Dilatation thermique : c'est le cas de la mise en place de "piles de disques" dans des unités chaudes.
4) La vitesse d'une unité à l'autre doit être la même.
5) Les "gaps" magnétiques (entrefers) d'une tête doivent être alignés pour qu'il n'apparaisse pas de différence d'une unité à l'autre.

Tous ces facteurs conduisent à considérer l'utilisation des disque à têtes fixes comme la meilleure solution pour éviter ces problèmes. Cependant ces unités sont beaucoup plus chères que celles à tête mobile qui d'autre part, présentent l'intérêt de pouvoir échanger les disques d'une unité à l'autre. Il est vrai que ce dernier facteur n'est pas tellement important car l'expérience a montré que dans la plupart des installations, les unités à disques interchangeables sont utilisées comme fixes pour la plus grande partie de leur temps. Normalement dans des grandes installations, on trouve la plupart des unités employées comme fixes et seulement quelques unes comme interchangeable. Cela dit, avec la réduction du prix des disques à têtes fixes, il est bien probable qu'ils prendront dans le futur une partie plus appréciable du marché.

La mesure du temps moyen d'accès d'un disque est considérée comme une bonne mesure de ses performances. Cette valeur est définie par le temps que prend une unité pour positionner sa tête sur les données, à partir du moment de la réception de l'ordre. Des grandes vitesses d'accès peuvent être obtenues en utilisant une grande vitesse de déplacement du bras, ou plusieurs bras, ce qui permet la recherche entrelacée ou alors une combinaison de ces techniques avec celle des têtes fixes.

La figure IV.5 fournit la relation entre le nombre de têtes fixes ou mobiles et le nombre d'accès par seconde ; pour une vitesse de rotation de 3600 t/m (rpm) avec un temps de lecture de 2,67 ms et une file d'attente assez longue pour que les bras soient toujours actifs. Ces courbes bien qu'étant un peu idéales, permettent de dégager certaines observations. Elles sont idéalisées dans le fait que :

a) pour la courbe à têtes fixes, le nombre d'accès par seconde croît avec le nombre de têtes par surface et par cylindre, ce qui actuellement n'est pas réalisable compte tenu du coût que cela représenterait ;

b) pour les courbes à bras mobile, elles croissent avec le nombre de bras, ce qui aussi représente un coût trop élevé.

De toute manière les observations qu'on peut dégager sont que la courbe à tête fixe a toujours le plus haut taux d'accès et d' autre part une diminution du temps de recherche du cylindre (seek time) ne produit pas de grandes différences quand on approche du temps d'un tour de disque

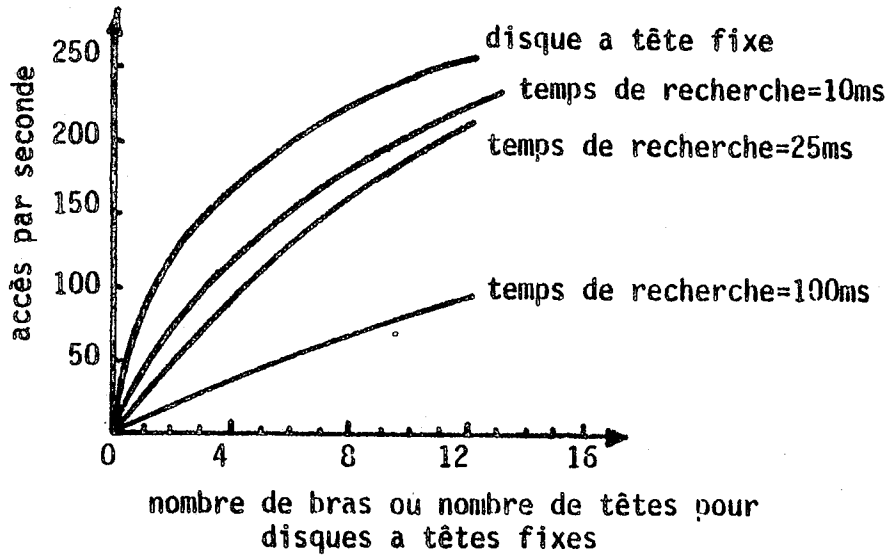


Fig IV.5 - Taux d'accès par rapport au nombre de têtes [HAU.75]

Cela permet de suggérer une combinaison hybride de têtes fixes et mobiles pour les disques. Les têtes fixes s'occuperaient des tables d'index par exemple, pendant que les têtes mobiles s'occuperaient des données stockées aléatoirement. Cette combinaison hybride a l'air d'être la plus performante, même du point de vue du coût. La figure IV.6 donne des courbes comparatives de performances, où l'on voit qu'un disque hybride avec un temps de recherche de 25 ms et 12 têtes fixes par actuateur, est plus performant q'un disque de 10 ms de temps de recherche mais sans des têtes fixes. Des unités de ce type existent déjà, l'unité disque ShugartSA 4000 à 1 tête mobile et de l'ordre de 8 têtes fixes.

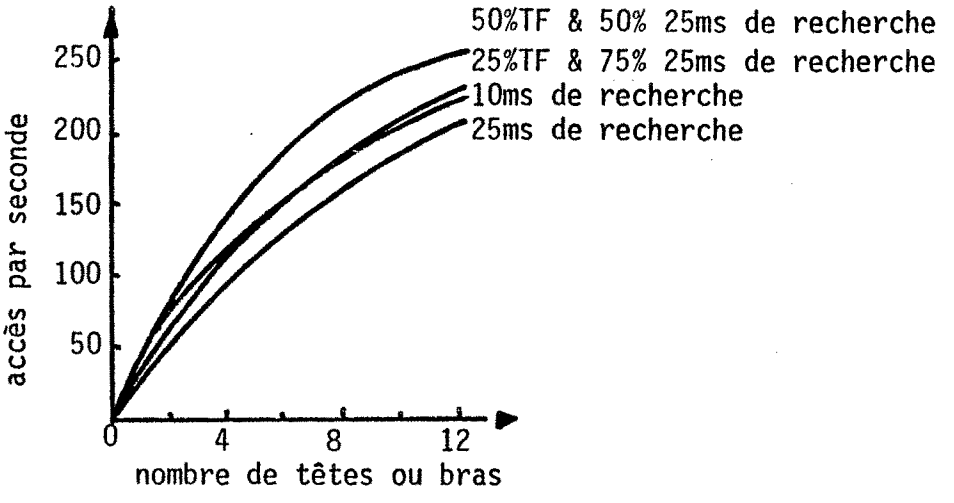


Fig IV.6 - Comparaison de performances entre disques avec têtes mobiles et hybrides [HAU.75]

IV.6 DEPLACEMENT DU BRAS

D'autres facteurs peuvent améliorer la performance d'unités de disque, particulièrement ceux causés par la modification de paramètres tels que l'accélération-décélération du bras, vitesse de déplacement du bras et distance de déplacement du bras.

D'une manière idéale le déplacement du bras peut être représenté par la courbe de la Fig IV.7 ou "a" représente l'accélération du bras et "b" la vitesse de déplacement du bras.

Cette courbe montre très bien que pour diminuer le temps de recherche cylindre, trois mesures principales peuvent être prises : avoir une accélération-décélération plus rapide, une vitesse de déplacement plus grande ou alors un déplacement de bras plus court. Augmenter l'accélération ou la vitesse pose des problèmes techniques et coûte cher. Une solution plus simple est de diminuer le déplacement du bras en augmentant le nombre de têtes par superficie. Si un bras a quatre têtes sur une même superficie d'un disque, son déplacement maximum sera quatre fois plus petit. Les courbes de la Fig. IV.8 présentent d'une manière idéale la relation entre le coût de l'augmentation du nombre de têtes, par rapport au gain sur le temps d'accès. Ces courbes permettent de conclure que l'utilisation d'une tête par surface n'est pas la meilleure solution. Au niveau actuel de la technologie des têtes, la solution optimale pour obtenir le meilleur rapport prix/performance, d'après ces courbes, serait l'utilisation de trois têtes par surface.

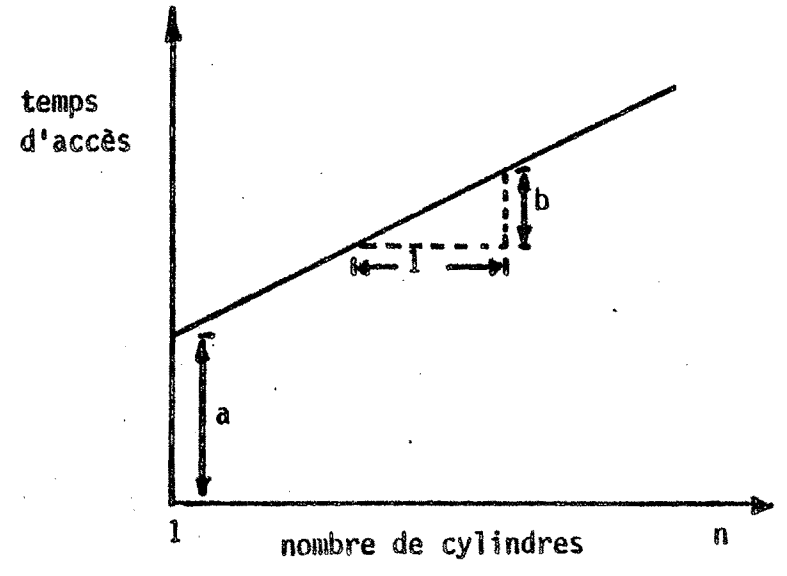


Fig. IV.7 - Temps de recherche (seek) par rapport au nombre de cylindres

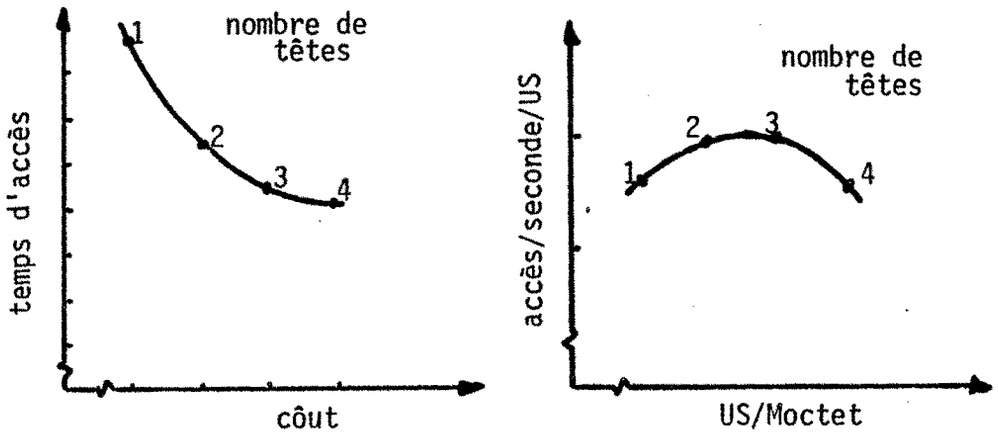


Fig IV.8 - Rapport entre le temps d'accès et le coût en considérant plusieurs têtes [HAU.75]

IV.7 TEMPS DE ROTATION

Un autre paramètre qui influence la performance d'un disque, est son temps de rotation (latency time). Comme il a été montré dans la section IV.2, le temps de réponse à une requête au disque "t_T" (temps total) est égal au temps de recherche cylinde "ta", plus le delay rotationnel "tr" pour arriver au bon secteur, plus le temps du transfert des données "td".

$$t_T = ta + tr + td$$

La figure IV.9 donne les courbes comparatives de la variation de ces paramètres en ne considérant pas le temps de transfert "td" négligeable. Ces courbes sont intéressantes, car elles montrent que passer d'un temps de recherche de 20 ms à 15 ms ce qui est assez compliqué et coûteux, donne à peu près le même résultat que passer de 2400 RPM à 3600 RPM ce qui est technologiquement plus simple.

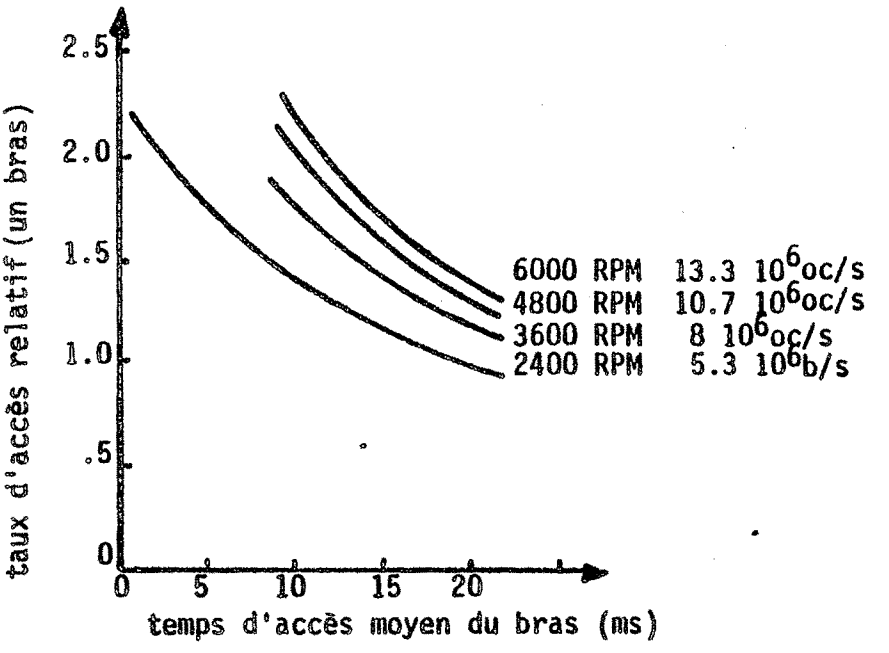


Fig IV.9 - Rapport entre le temps moyen de deplacement de bras et la frequence relative d'accès pour plusieurs vitesses de rotation [HAU.75].

IV.8 MEMOIRES TAMPON

Un autre facteur qui peut contribuer à l'amélioration des performances des unités disque sont les mémoires tampons. Ces mémoires aussi appelées buffers ne font pas partie intégrante des unités disque d'aujourd'hui, mais permettent d'obtenir, quand elles sont bien utilisées des taux de présence jusqu'à 96 % du temps, c'est-à-dire que l'ordinateur connecté à ce système de disque devra attendre seulement 4 % de ses accès au disque, pendant que tous les autres accès trouveront l'information déjà dans les mémoires tampons.

Une étude [SMI.78] assez intéressante sur la comparaison entre l'utilisation de mémoires buffers ou de plusieurs bras par unité disque, montre assez clairement que dans l'état actuel de la technologie, le coût d'une mémoire buffer, pouvant minimiser le temps d'attente d'un système, est trop grand. D'autre part, l'utilisation de plusieurs bras revient encore plus cher que les mémoires tampons et donc présente encore moins d'intérêt.

Cette étude montre que pour une bonne optimisation il faudrait avoir un minimum de trois buffers et que chacun devrait avoir une taille de 20 pistes. Si nous prenons les unités disque CDC 9766 [TCDC.77] de 300 M octets qui ont par piste 20 K octets cela ferait une taille pour chaque buffer d'environ 400 K octets soit au total 1200 K octets. Il est vrai qu'avec une telle taille la présence serait de 93 %, ce qui est très bien. Même si en prenant une taille de 10 pistes qui ne donnerait des taux de présence que de 87 %, cela ferait déjà 200 K octets et donc son coût actuel ne rend pas cette solution intéressante. Cela amène à abandonner provisoirement l'utilisation de grandes mémoires tampons pour augmenter le taux de présence des données.

On voit dans les courbes de la Fig. IV.10 qu'un net progrès apparaît si on dépasse la valeur de 20 pistes pour la taille totale des buffers. Cette valeur coïncide avec le nombre de pistes par cylindre que possèdent les unités disque 2314, lesquels ont été utilisées comme exemple pour les calculs des courbes. Cela montre l'influence assez forte du type d'unité disque utilisée pour les calculs. Même en considérant ce détail, les conclusions qu'on retire de ces courbes sont valables pour donner une idée d'ordre de grandeur pour les buffers.

L'étude en question [SMI.78] suggère même que le transfert des 20 pistes vers le buffer se fasse en une rotation, ce qui impliquerait d'avoir des unités de disque spéciales avec la possibilité d'activer toutes les têtes de lecture/écriture en parallèle. Il faut signaler que tous les calculs des courbes ont été faits avec l'algorithme de LRU (least recently used) pour l'optimisation du remplacement des buffers.

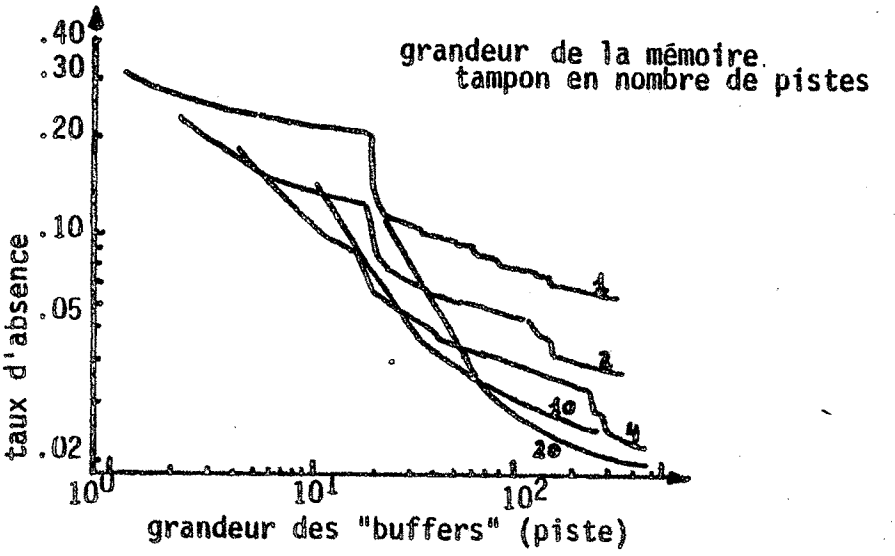


Fig.IV.10 - Rapport entre la taille total des buffers et le taux de présence des données [SMI.78]

IV. 9 CACHE D'UNE PISTE

La section précédente a montré qu'actuellement, au niveau de la technologie, il n'est pas intéressant d'utiliser des grandes mémoires tampons pour obtenir des taux de présence de l'ordre de 90 %. Cela n'empêche cependant pas d'utiliser une mémoire tampon, un cache, d'une piste seulement pour améliorer un peu les performances. La ligne IV.10 montre qu'un cache d'une piste donnerait une présence de l'ordre de 65 % ce qui est assez appréciable. L'argument (suivant une idée de J. Gastinel) est que si on travaille sur des programmes qui appellent plusieurs fois des données stockées séquentiellement sur une même piste, il y a intérêt lors de la première requête sur cette piste, de transférer toute la piste en une fois, au lieu de transférer secteur par secteur.

Les calculs ci-dessous déterminent à partir de combien d'appels à la même piste il y a intérêt à transférer la piste entière.

Soit les variables :

T = temps total de transfert
ta = temps de recherche cylindre (seek)
td = temps de transfert des données
R = temps d'une rotation entière du disque
tr = délai de rotation pour arriver au secteur (latency)
n = nombre de lectures/écritures
ne = nombre d'écriture

Pour un transfert quelconque, le temps total est :

$$T = ta + tr + td$$

Cependant statistiquement le délai rotationel pour arriver au bon secteur est égal à R/2 et d'autre part le temps de transfert est beaucoup plus petit que les autres variables, ce qui fait qu'il ne sera pas considéré. Par exemple, en considérant une piste divisée en 64 secteurs, le temps de transfert sera de R/64 ce qui permet bien de le négliger.

Appelons Tn le temps d'un transfert secteur normal, on obtient

$$Tn = ta + R/2 \quad \text{(transfert d'un secteur)}$$

Si le transfert est d'une piste entière la valeur de td sera le temps d'une rotation R. Appelant Tp ce type de transfert

$$Tp = ta + R/2 + R$$

En considérant que la lecture ou écriture puisse être commencée presque immédiatement à partir du premier secteur trouvé, la valeur du délai rotationnel tr peut être négligée

$$Tp = ta + R \quad \text{(transfert d'une piste)}$$

La modification d'une donnée sur un secteur occasionnera une lecture suivie d'une écriture. Pour Tn il y a juste un temps de recherche ta, car la lecture/écriture se fera sur la même piste et la valeur tr est égale à R/2.

$$Tn = ta + 2\ tr = ta + R$$

Cependant, pour une modification, le tr de l'écriture aura probablement la valeur de R car il faudra attendre un tour au minimum pour être de nouveau sur le même secteur.

$$Tn = ta + R/2 + R = ta + 3/2\ R$$

Il est vrai que cette valeur tr = R est un peu relative car le traitement des données avant l'écriture peut prendre un temps très grand et à ce moment on retombe sur R/2.

Utilisant la technique de piste, deux hypothèses peuvent être faites pour une opération de modification. Dans le premier cas on considère que si une modification a lieu, uniquement le secteur modifié sera réécrit, donc on aura un temps R pour la lecture de la piste et un temps R/2 pour le délai rotationnel d'écriture du secteur.

$$Tp1 = ta + R + T/2$$

La deuxième hypothèse considère que si une modification a lieu, la piste est réécrite automatiquement.

$$Tp2 = ta + R + R$$

Le premier R est le temps de lecture et le deuxième R est celui de l'écriture.
Prenant ces équations comme une moyenne et introduisant la variable du nombre d'écritures, les équations deviennent :

$$Tp1 = ta + R + ne\ \frac{R}{2}$$

$$\overline{Tn} = ta + \overline{n}\,\frac{R}{2}$$

Faisant l'égalité des deux formules

$$ta + \overline{n}\,\frac{R}{2} = ta + R + \overline{ne}\,\frac{R}{2}$$

$$\overline{n}_1 = 2 + \overline{ne} \quad \text{(cache piste avec écriture secteur).}$$

Ce résultat nous indique que si le nombre d'écritures est très bas, à partir en moyenne de 2 opérations sur une même piste, il y a intérêt à utiliser la technique du "cache de piste avec écriture du secteur".

La deuxième technique implique l'écriture de la piste entière du moment qu'il y a une modification

$$ta + \frac{R}{2} = ta + R + R$$

$$\overline{n2} = 2 + [\overline{ne}]_{0}^{1} \quad \text{(cache piste avec écriture piste)}$$

Le résultat montre que s'il n'y a pas d'écriture n sera égal à 2, mais du moment qu'il y a une écriture ou plus la valeur passe à 4.

La figure IV.11 permet de mieux observer le résultat de ces deux équations par rapport au pourcentage d'écriture.

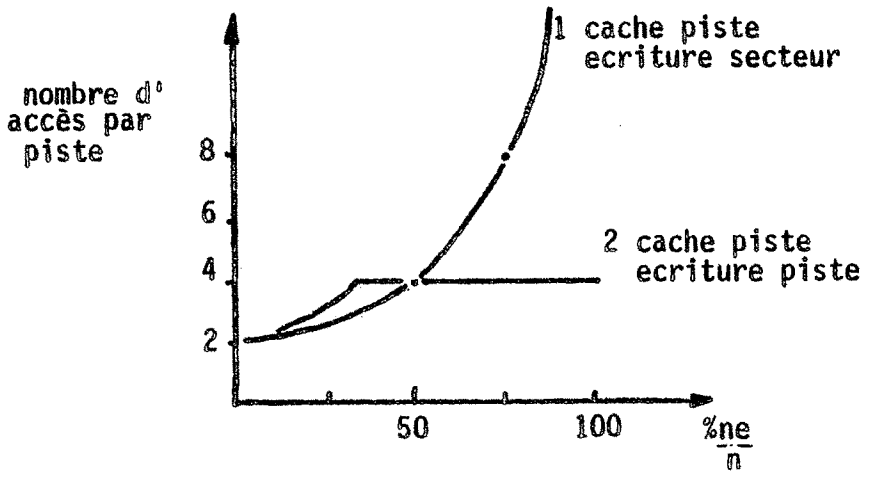


Fig. IV.11 - Rapport entre le pourcentage d'écriture et le temps d'accès

Ces deux courbes sont approximatives et permettent plutôt de donner une idée de l'ordre de grandeur (elles ne prennent pas en compte les probabilités). Elles délimitent les régions où il est plus intéressant d'utiliser une technique ou l'autre. La courbe 1, cache piste avec écriture secteur, montre qu'avec un nombre d'opérations sur une même piste, de l'ordre de 2 à 4 et un taux d'écriture bas, cette technique est la plus adaptée. Si le taux des opérations est élevé sur une même piste, ainsi que le pourcentage d'écritures, la deuxième technique de cache de piste avec écriture de piste est la plus conseillée (courbe 2). En dessous de deux événements en moyenne par piste, il est préférable d'utiliser la technique conventionnelle sans cache de piste.

Une autre méthode qui peut être employée, est la technique de réécrire le cache systématiquement, qu'il y ait eu écriture ou non. Cette méthode est intéressante pour des systèmes où le taux d'écriture est haut, de même que le nombre d'accès à la même piste.

La conclusion qu'un peut retirer de cette étude est que ces techniques ont de l'intérêt juste pour certains types de stockage, par exemple pour une exploitation séquentielle de l'information comme dans le cas de fichiers. Il montre aussi que l'utilisation de certaines méthodes de stockage de fichiers dans un disque peuvent amener à avoir des meilleurs temps de réponse des unités disque.

IV.10 NORMES D'INTERFACE SMD

Ces dernières années ont vu apparaître une norme de connexion physique pour les unités disque, la SMD (Storage Module Drive) [T.CD.77] [T.AMP.77]. Anciennement chaque constructeur utilisait son propre standard de connexion pour ses unités disque. La norme SMD était à l'origine tout comme un autre standard de connexion de fabricant (CDC) et, son adoption par tous les constructeurs, en a fait une sorte de standard.. Son nom même, SMD-Storage Module Drive, n'a aucun rapport avec un terme de connexion. SMD est un sigle utilisé par CDC pour désigner une famille d'unités disque modulaires. Cette norme ne peut cependant être considérée comme un standard mais plutôt comme un consensus entre fabricants, car il existe quelques légères variations dans les signaux utilisés d'un constructeur à l'autre. Ces différences apparaissent plutôt avec les options, mais d'une manière générale que ce soit au niveau physique du connecteur, au niveau du type de signal ou au niveau électrique, les connexions sont identiques.

Ces normes concernenent généralement les unités disque de 10 MHz de vitesse de transfert des données, utilisant comme technique d'enregistrement le MFM (Modified Frequency Modulation) et ayant une surface d'horloge préenregistrée sur la pile de disques.

C'est le cas de la famille 9760 de CDC [T.CDC.77] et 900 d'Ampex [T. AMP.77] . Cela n'empêche cependant pas que cette norme soit utilisée, à peu de choses près, dans d'autres unités disques, par exemple la famille 9730 [T.CDC1.77] .

La principale caractéristique de ces normes SMD, en dehors de la définition logique, électrique et physique des signaux a été d'établir l'utilisation de deux câbles de connexion pour une unité disque. Le premier câble, appelé câble A, est concerné par tous les signaux de contrôle du disque : il y en a environ 40. Les signaux véhiculent des ordres tels que les déplacements du bras, ainsi que les signaux d'erreurs tels qu'erreur de recherche cylindre. Le deuxième câble, câble B, est celui du transferts des données et donc des signaux rapides (10 MHz). Il possède environ 6 signaux selon l'option choisie, signaux du type donnée de lecture, donnée d'écriture, horloge d'écriture etc...

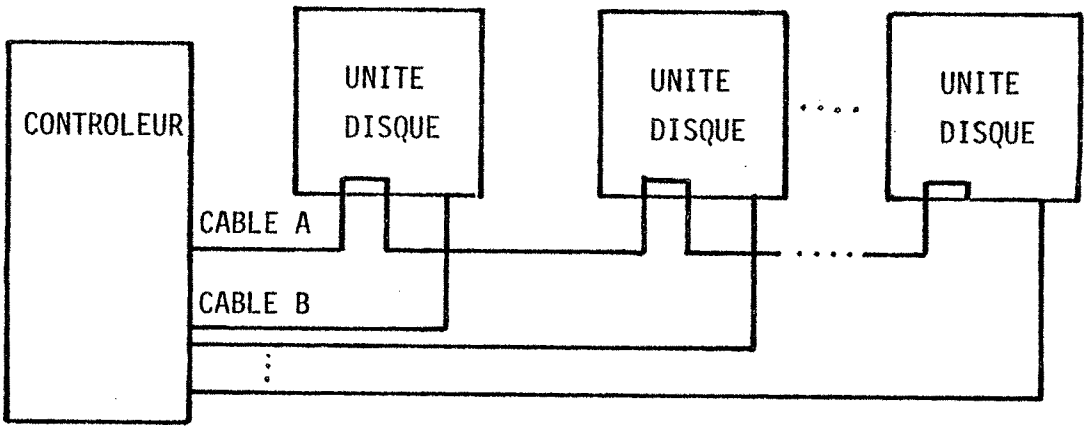
Le but de cette section est seulement de donner une idée des normes SMD et donc pour plus de détails techniques tels que la liste des signaux, il faut se rapporter à l'Annexe 1 ou aux documents des constructeurs [T.CDC.77] [T.NAV.78] [T. AMP.77]

IV.11 TYPES DE LIAISONS ENTRE CONTROLEUR DISQUE ET UNITES DISQUE

Il existe deux types possibles de liaison entre un contrôleur disque et plusieurs unités disque utilisant les normes SMD : système connecté en étoile et système connecté en "daisy chain".

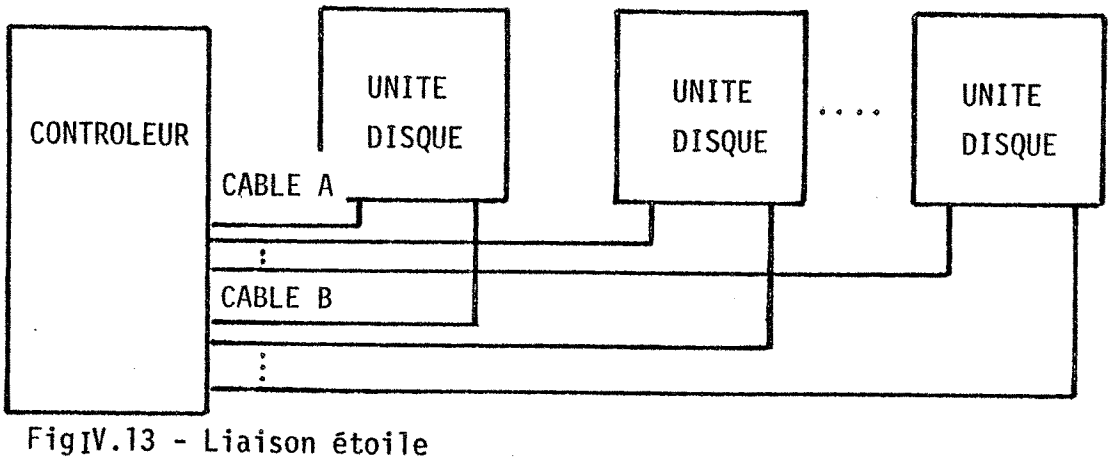
Dans la connexion "daisy chain" le câble des signaux de contrôle, câble A, parcourt successivement toutes les unités disque (Fig V.12)

Un câble B, (signaux de données) est connecté par unité disque. Cette structure est due en partie au fait que les signaux qui parcourent le câble B sont rapides (10 MHz) et donc il est déconseillé de les transmettre d'une unité à l'autre, pour éviter leur dégradation.



FigIV.12 - Liaison daisy chain

Dans la connexion étoile (star) (Fif.V.13) les unités disque sont reliées chacune au contrôleur à travers les câbles A et B. Par rapport à la connexion précédente la différence fondamentale est que le câble A ne parcourt pas successivement les unités disque mais qu'il y en a un pour chaque unité.



FigIV.13 - Liaison étoile

La comparaison de ces deux types de connexion montre que la "daisy chain" est plus économique car le contrôleur n'a besoin que d'une logique électronique concernant le câble A, pendant que la connexion en étoile nécessite la même électronique pour chaque câble A. Mais d'autre part, la conséquence de cette différence est que la connexion "daisy chain" ne pourra pas effectuer des travaux en parallèle. Cela veut dire, qu'il n'est pas possible, par exemple, de lancer une recherche cylindre (ou plusieurs) sur une unité disque, pendant qu'une opération de lecture/écriture s'effectue sur une autre unité disque. Le fait que les signaux du câble A et ceux du câble B sont nécessaires pour une opération de lecture/écriture, empêche la réalisation de travaux parallèles (seek overlap). En résumé la liaison étoile permet du travail parallèle pendant que la connexion "daisy chain" permet seulment un travail séquentiel.

Des contrôleurs avec des connexions du type hybride qui obtiendraient une plus grande performance sans une trop grande augmentation du prix sont toujours concevables. C'est le cas d'une structure "daisy chain" avec deux câbles A, un servant une partie des unités disque et l'autre les autres. Cela permet d'obtenir un certain degré de parallèlisme.

IV.12 CONCLUSION

Le point le plus important qui ressort de ce chapitre est que pour les prochaines années, cinq au moins, les unités disque à bras mobile continueront à exister et dominer une grande partie du marché des mémoires secondaires. Les fabricants d'unités de disque sont entrain de contre-attaquer l'arrivée des mémoires à bulles par le lancement de petites unités de disque de 8 pouces non amovibles en technologie "Winchester", avec des capacités de stockage du 10 M octet à 40 M octet et même plus. Ces disques ont des prix inférieurs à ceux des unités disques amovibles et viennent se pla-

cer dans la zone de petite capacité de stockage, justement celle recherchée par les mémoires à bulles.

Il aurait été possible de s'attarder encore plus sur divers aspects liés aux unités disque comme les règles d'ordre de prise en compte de requêtes (SSTF, SCAN, etc...), règles d'organisation de l'espace disque, etc... mais cela n'était pas le but de ce chapitre. Son but était plutôt de faire le tour de quelques aspects et techniques qui sont en train d'influencer les recherches actuelles et qui donc seront importants d'ici quelques années (5 années). C'est le cas des mémoires à bulles qui commencent lentement à être utilisées. La plupart de ces techniques ne sont pas encore économiquement viables. Deux aspects cependant sont pour un avenir immédiat, les codes correcteurs ECC et les mémoires tampons de l'ordre d'au moins une piste.

Un fait qui indique bien les futurs développements, est le choix fait par IBM d'utiliser dans leur ordinateur de moyenne échelle, le modèle 38 (Division de Systèmes de Grande Diffusion), une mémoire secondaire basée sur des unités disque non amovibles. L'utilisation de ces disques pour la mémoire secondaire est un pas vers le futur, le prochain pas serait le remplacement des disques de petites capacités par des mémoires à bulles, ou EBAM.

CHAPITRE V

PRESENTATION DU PROJET MAGE

1. INTRODUCTION
2. CARACTERISTIQUES ET MOTIVATIONS
3. METHODE D'ACCES AUX DONNEES
   1- STRUCTURE D'ACCES
   2- MECANISME DE DESIGNATION D'UNE REALISATION
   3- OPERATIONS SUR UNE REALISATION : MODES D'ACCES
   4- LANGUAGE D'ACCES AU PBD
   5- ADRESSAGE DES DONNEES PAR NOMS INTERNES
4. ARCHITECTURE LOGICIELLE
5. DEFINITION DE L'ARCHITECTURE MATERIELLE
   1- EVALUATION DU TEMPS MOYEN DE TRAITEMENT D'UNE REQUETE DE L'UTILISATEUR
   2- EVALUATION DU NOMBRE D'ACCES DISQUE PAR REQUETE DE L'UTILISATEUR
   3- SUGGESTION POUR L'ARCHITECTURE MATERIELLE

V. PRESENTATION DU PROJET MAGE

V.1 INTRODUCTION

Les précédents chapitres ont présenté le contexte dans lequel s'est développé le travail de recherche sur MAGE et plus spécialement son aspect matériel. Ce chapitre aura pour but de situer le projet du PBD-MAGE.

La différence fondamentale entre les sytèmes précédents tels que CASSM, RAP, RARES et autres (voir chapitre II), est que MAGE permet d'accéder à des informations structurées plus ou moins en ordre. Les autres sont des moyens d'accès associatifs directs à des données qui permettent la recherche de certains mots ou groupes de mots et l'établissement de relations sur ces données.

Comme il a été vu dans le chapitre II la plupart de ces projets de processeurs associatifs ont des coût trop élevés pour être industrialisés dans l'état actuel de la technologie. Par exemple, le fait d'avoir une tête par piste avec une électronique associée rend ces projets économiquement non viables.

Le but du projet MAGE était de réaliser un PBD destiné à de petits systèmes clefs en mains. D'autre part, il devait aboutir à un produit industriel ce qui imposait de travailler avec des technologies courantes sans trop essayer d'utiliser des technologie de pointe pas encore au point.
Le PB-MAGE possède néanmoins la propriété d'effectuer une recherche directe de formes sur les unités disques visant à accélérer l'accès aux données, mais ce n'est pas un système à mémoire associative comme RAP, RARES ... D'autre part, il possède une architecture à deux processeurs qui travaillent en parallèle exécutant la méthode d'accès aux disques, mais cela n'est pas vraiment un complexe de processeurs

comme INFOPLEX. On aurait plutôt tendance à encadrer ce projet comme un type de "back-end processor" vu qu'il nécessite un processeur intermédiaire pour traiter les requêtes utilisateur, les transformant dans un langage qui lui soit compréhensible.

Les principaux objectifs qui ont guidé ce projet étaient : de réusssire à développer une méthode d'adressage à une base de donnée en utilisant des microprocesseurs ; de montrer qu'il était possible de réaliser un PBD qui puisse travailler avec des petits systèmes ; et troisième objectif, ou plutôt conséquence, que le PBD soit adapté aux systèmes décentralisés et distribués.

Ce chapitre a pour but de présenter les travaux qui précèdèrent cette thèse et qui donc en ont été la base. Cela permettra de mieux comprendre l'évolution de ce projet.

La description de la méthode d'accès aux données sera abordée d'une manière succinte et après, les évaluation qui ont abouti à l'architecture matérelle de MAGE seront présentées. Ces travaux ont été réalisés, par G. Berger Sabbatel [BER.77] [BER1.78] [BER2.78].

V.2 CARACTERISTIQUES ET MOTIVATIONS

Les principales motivation de MAGE étaient :

a) permettre de libérer les unités centrales des ordinateurs d'une grande partie de leur traitement utilisé pour exécuter le système de base de données. Un PBD est économiquement intéressant dans la mesure où il augmente la puissance disponible des ordinateurs surchargés.

b) réaliser un processeur fonctionnel de puissance et de capacité suffisante pour les systèmes distribués.
c) minimiser la maintenance et les coûts des projets de systèmes. L'idée de processeurs spécialisés permet une diminution des coûts dans les projets de systèmes vue que l'addition d'un nouvel élément dans le système n'impliquera pas la modification des autres processeurs déjà connectés à celui-ci.

Spécifications globales

a) traiter plus de 100 M octets de mémoire secondaire.
b) permettre à d'autres processeurs de partager sa mémoire secondaire
c) être étudié indépendamment du processeur qui l'utilise
d) supporter un dialogue simultanément avec 32 processeurs utilisateurs.
e) avoir une bonne sécurité pour les informations.

La méthode d'accès a été définie d'après une étude de "benchmark". Cette étude a montré que les possibilités offertes par une base de donnée hiérarchique étaient suffisantes pour résoudre la plupart des problèmes de gestion de données pour des petits systèmes, par exemple des systèmes pour les laboratoires d'analyse médicales, ou bureautiques. La méthode d'accès a été basée sur SOCRATE [ABR.70], méthode d'accès aux BD conversationnelles. Ce système de BD a été développé à l'Université de Grenoble en 1970. La méthode de MAGE a dû prendre en compte des adaptations pour des petits systèmes. Ces modification sont dues en partie au fait qu'il serait implanté avec des micro-processeurs.

MAGE est spécialement inspiré de SOCRATE en ce qui concerne le langage de définition de structure (LDD), cependant il ne possède aucun des aspects conversationnels de SOCRATE et doit être utilisé par l'intermédiaires de programmes, de la même façon que n'importe quel système de gestion de fichiers classiques : il est considéré comme une méthode d'accès.

En ce qui concerne l'utilisateur, MAGE doit pouvoir être adapté à deux types de processeurs, soit un mini-ordinateur classique, vis-à-vis duquel il devra se comporter comme un périphérique intelligent, soit un réseau de processeurs constituant un système distribué.

V. 3 METHODE D'ACCES AUX DONNES

La méthode d'accès est du type hiérarchique et est basée sur la description d'une structure et d'un espace virtuel de pointeurs appelés "noms internes".

V.3.1 Structure d'accès

La structure définit l'organisation des données et les relations existant entre les éléments d'information. La base de données comporte un certain nombre d'éléments d'information (enregistrements) de divers formats. Les éléments de même format constituent des familles que l'on appelle "entités". Les éléments d'une entité prennent le nom de réalisations. Chaque réalisation peut être composée à son tour de plusieurs éléments accessibles individuellement qui peuvent être :

- d'autres entités (structure arborescente) ;
- une zone de données de longueur fixe, non décomposable appelée caractéristique simple ;

- un pointeur entre deux réalisation d'entités, appelé référence ;
- une clé qui permet l'accès associatif aux réalisation de l'entité.

Exemple :

```
entité 10.000 PERSONNE :
début ;
   caractéristique simple NOM 20 ;
   caractéristique simple ADRESSE 50 ;
   entité 5 VOITURE ;
   début ;
      caractéristique simple MARQUE 20 ;
      caractéristique simple NUMERO 20 ;
   fin ;
fin ;
```

Cette structure définit une entité PERSONNE qui peut avoir jusqu'à 10 000 réalisations. Chaque réalisation de PERSONNE comporte deux caractéristiques simples, le NOM pour lequel sont réservés 20 octets, et l'ADRESSE qui a un espace réservé de 50 octets. Chaque réalisation de PERSONNE peut contenir jusqu'à 5 réalisations de VOITURE. VOITURE est une nouvelle entité et donc descend dans la structure, elle peut avoir un maximum de 5 x 10 000 = 50 000 réalisations. Une réalisation de VOITURE va contenir à son tour deux caractéristiques simples : la MARQUE avec 20 octets réserves et le NUMERO avec aussi 20 octets réservés.

Le fichier structure est créé à la phase d'initialisation de la base de donnée et donc peut seulement être lu en exploitation. Normalement sa création sera réalisée au niveau de la définition de l'application. Il contiendra les informations nécessaires pour

calculer les noms internes, la largeur des zones de données, etc.. Lors de la définition de la structure, l'utilisateur devra spécifier le nombre maximum de réalisations que chaque entité peut avoir. Ce fichier est de petite taille, généralement il occupe moins de 4 K octets.

V.3.2. Mécanisme de désignation d'une réalisation

Une donnée est adressée par la spécification des réalisation d' entités qui la contiennent. Pour désigner une réalisation il existe quatre mécanismes principaux :

a) l'accès séquentiel -qui permet l'accès successif aux réalisations d'une même entité ;
b) l'accès direct -qui permet l'accès à une réalisation dont on connait le numéro d'entité ;
c) l'accès associatif -qui réalise l'accès à une réalisation dont on connaît la valeur d'une caractéristique clé ;
d) l'accès par référence qui permet d'accéder une réalisation par l'intermédiaire d'un pointeur contenu dans une autre réalisation.

V.3.3 Opérations sur une réalisation : mode d'accès

Les principales opérations qui peuvent être effectuées sur une réalisation sont :

- lecture
- écriture
- création
- suppression

V.3.4 Langage d'accès

MAGE est destiné à être utilisé par l'intermédiaire de programmes d'un autre processeur lié a l'utilisateur.
Le PDB nécessite d'un langage BD intermédiaire pour l'accéder. Le logiciel BD au niveau du processeur utilisateur assure son dialogue avec le PBD. C'est l'équivalent d'un superviseur d'E/S pour un périphérique classique (Fig.V.1).

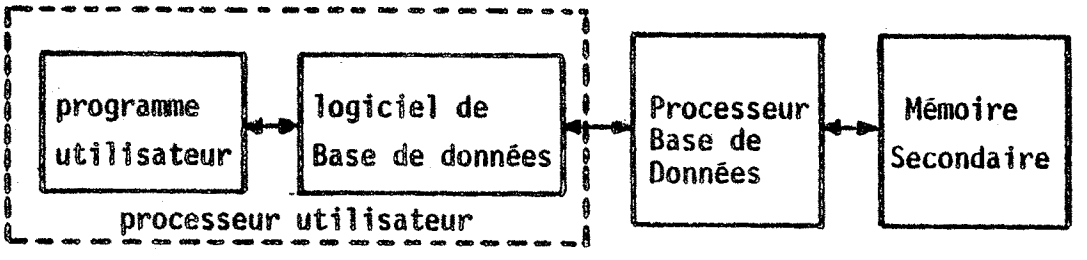


Fig V.1 - Le logiciel BD du processeur utilisateur

Les requêtes adressées à MAGE sont de longueur fixe et syntaxe simple pour être gérées et traitées facilement. A chaque utilisateur est associé un contexte au début d'un accès. Le contexte définit la position de l'utilisateur dans la structure. L'utilisateur peut à l'aide de requêtes du langage, effectuer des déplacements élémentaires du contexte dans la structure : descendre un niveau, remonter, passer à un caractéristique soeur, changer de réalisation... Une fois le contexte positionné sur la donnée voulue, l'utilisateur spécifie le type d'opération qu'il désire effectuer à travers un paramètre de la requête : le mode (voir V.3.3)

Exemple :

En reprenant l'exemple précédent, la lecture du numéro de la troisième voiture de la centième personne demandera trois requêtes au PBD de la part du processeur utilisateur :

- positionner sur la 100 ème PERSONNE ;
- positionner sur la 3 ème VOITURE de cette personne ;
- positionner sur le NUMERO de cette voiture et lire la donnée.

V.3.5 Adressage des données par noms internes

Dans tout système de gestion de données il est nécessaire d'établir un lien entre l'adressage logique employé par l'utilisateur et l'adressage physique du support de l'information. Dans le cas de MAGE cette relation est réalisée à travers un adressage appelé "nom interne" et d'un dictionnaire.

La section V.3.1 a montré que lors de l'initialisation de la base il faut spécifier le nombre maximum de réalisations que chaque entité peut avoir. Cela permet de statiquement affecter un numéro, le nom interne, à toute réalisation pouvant exister.

En reprenant l'exemple précédent, les noms internes seront affectés comme suit :

PERSONNE de 1 à 10 000
VOITURE de 10 001 à 60 000

Pour calculer le nom interne d'une réalisation, il suffit de connaître le nom interne (numéro) de la première réalisation de l'entité.

Exemple : calcul du "nom interne" de la voiture 3 de la personne 100

- nombre de réalisation de VOITURE précédentes
  99 personnes x 5 voitures possibles par personne = 495
- la troisième voiture
  495 + 3 = 498
- le nom interne de la première voiture est 10001 le nom interne cherché est : 10 000 + 498 = 10 498

Donc 10 498 est le nom interne de cette voiture.

Les informations nécessaires pour le calcul des noms internes sont contenues dans le fichier structure.

Le lien entre un nom interne et la zone de données correspondante, si elle existe, est assuré par un dictionnaire. Celui-ci est accédé par "hash coding" sur les noms internes et donc il contiendra une entrée pour toute réalisation existante. Cela permet d'éviter d'avoir une entrée statiquement allouée pour toutes les réalisations (pour tous les noms internes), ce qui occasionnerait une grande perte d'espace mémoire. Donc, le nom interne est un pointeur virtuel et le lien entre lui et l'adresse disque du bloc de données est assuré par le dictionnaire. Le nombre d'entrées du dictionnaire est égal au nombre maximum de réalisations effectives que l'on espère stocker dans la base. Il peut contenir jusqu'à 16 millions d'entrées et occupe environ 1/6 de l'espace disque (Fig.V.2)

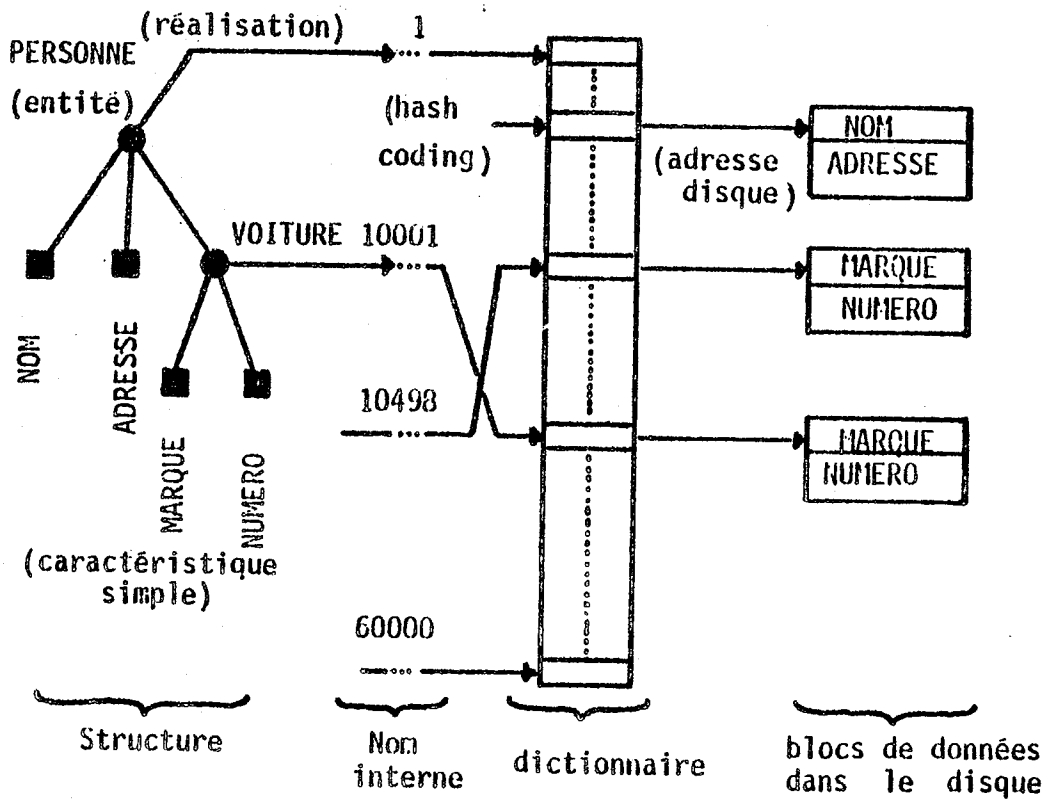


Fig V.2 - Méthode d'accès aux données

V.4 ARCHITECTURE LOGICIELLE

La définition de la méthode d'accès implique d'avoir un certain traitement à réaliser. La décomposition de cette activité en un nombre de grands modules fonctionnels crée des processus, ou plutôt des processeurs logiques, chacun exécutant une des différentes tâches de MAGE. Les processeurs logiques pourront être concrétisés soit par un processeur physique spécialisé, soit, plus vraisemblablement, par une des tâches d'un processeur physique ou groupe de processeurs physiques.

La figure V.3 donne le résultat de la décomposition en plusieurs processus montrant les liens entre eux.

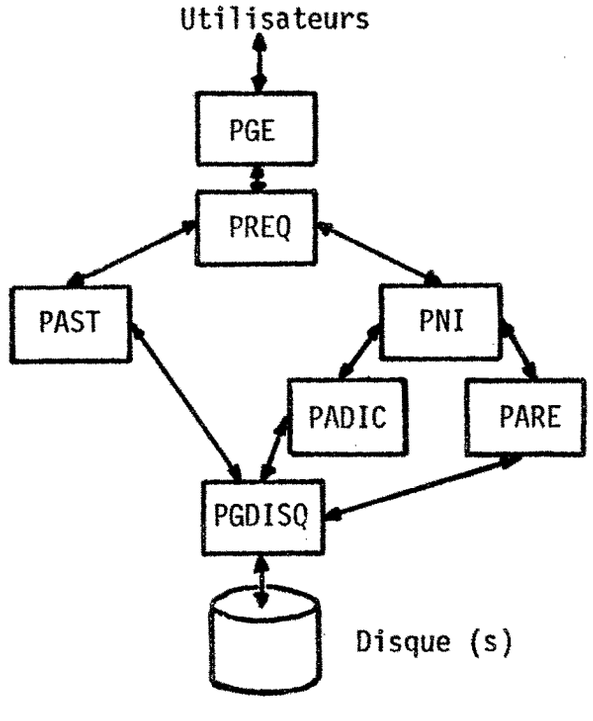


Fig. V.3 - Processeurs logiciels

Ces processeurs sont :

PGE - processus de gestion d'échanges. Il assure l'interface avec le ou les utilisateurs.

PREQ - processus de traitement de requêtes. Il effectue le calcul des noms internes, et commande l'accès aux réalisations.

PAST - processus d'accès à la structure. Il permet d'obtenir un enregistrement de structure à partir de son numéro. Il a la charge d'optimiser le nombre d'accès à la structure.

PNI - processus de noms internes. Il effectue l'accès à une réalisation connaissant son nom interne. L'accès est fait par l'intermédiaire de PADIC et PARE.

PADIC - processus d'accès au dictionnaire. Il effectue l'accès à une entrée dictionnaire connaissant son nom interne.

PARE - processus d'accès aux réalisations. Il effectue l'accès à une réalisation dont on connait la longueur, le nom interne et l'adresse disque.

PGDISQ - processus de gestion disque. Il effectue les échanges disque. Il convertit les adresses logiques fournies en adresses physiques, il gère les requêtes, il optimise les déplacements de bras. Ce processus doit être suffisament modulaire pour s'adapter à des configurations différentes de disques.

V.5 DEFINITION DE L'ARCHITECTURE MATERIELLE

Avec l'architecture logicielle et certaines évaluations, a été obtenue la première architecture matérielle pour le PBD-MAGE (suggestion). Ces évaluations ont permis de définir quelle architecture est la plus adaptée pour exécuter le plus efficacement la méthode d'accès MAGE, tout en restant simple et peu coûteuse. Elles déterminaient s'il fallait utiliser un ou plusieurs microprocesseurs, ou alors des circuits câblés, ou éventuellement un miniordinateur.

L'objectif du point de vue performance est d'obtenir un PBD avec un temps moyen d'exécution d'une requête utilisateur plus petit que la durée moyenne de l'accès disque d'une requête. En d'autres termes, le PBD doit être assez rapide pour obtenir une utilisation optimale du disque, l'unité disque n'ayant presque pas à attendre pour une nouvelle requête après la fin d'un échange. Pour une optimisation encore meilleure, il faudrait aussi avoir une file d'attente de requêtes au disque, permettant l'utilisation d'algorithmes d'optimisation du déplacement du bras du disque.

Pour choisir l'architecture, deux évaluations ont été réalisées : 1) une pour déterminer la puissance de traitement nécessaire, dans le cas de l'utilisation de microprocesseurs M 6800 [T.MOT.76] , de manière à savoir le temps pris pour traiter une requête de l' utilisateur ; 2) l'autre déterminant le nombre de requêtes disques nécessaires par requête utilisateur.

L'utilisation de microprocesseurs amène certaines contraintes différentes de celles des systèmes conventionnels sans microprocesseurs. En réalité le microprocesseur a un grand pouvoir de traitement par rapport à son coût, donc il n'est pas intéressant d'optimiser son taux d'utilisation, tout au moins pas beaucoup. Le principal objectif est d'obtenir la meilleure utilisation des unités disque qui sont les ressources du systèmes les plus coûteuses. Un autre objectif à prendre en compte est d' avoir un logiciel simple pour avoir une meilleure sécurité, une facilité de maintenance et un coût matériel plus bas, car les circuits mémoires (RAM) sont actuellement les boitiers électroniques les plus coûteux.

V.5.1 Evaluation du temps moyen de traitement d'une requête de l'utilisateur

Pour l'évaluation du temps de traitement il est nécessaire de faire un minimum d'hypothèses sur le matériel qui exécutera MAGE, spécialement le type de microprocesseur, pour permettre de calculer le temps des opérations. L'évaluation a été réalisée en sélectionnant les principales opérations qui seraient utilisées telles que : multiplication, division, opérations sur octets, transferts de données, certains algorithmes fréquents, etc... Les résultats ont été obtenus en prenant en compte le temps de traitement de chaque opération et la fréquence moyenne de son utilisation par rapport aux autres opérations. Il a été aussi nécessaire de faire une évaluation du temps pris par les autres opérations moins importantes.

Les résultats obtenus par catégorie d'opération sont [BER.78] :

- opérations divers	4250 cycles	- 34 %
- multiplication/division	3800 "	- 30,4 %
- transfert donnée de mémoire à mémoire	3800 "	- 30,4 %
- échange avec l'utilisateur	650 "	- 5,2 %
TOTAL	12500 cycles	

Cette valeur de 12 500 cycles machine du microprocesseur M6800 représente le temps moyen de traitement par celui-ci d'une requête utilisateur. Ce temps est égal à 12,5 ms si le microprocesseur tourne avec une horloge de 1 MHz.

V.5.2 Evaluation du nombre moyen d'accès disque par requête de l'utilisateur

Le nombre d'accès disque par requête utilisateur a été évalué d'après l'analyse des conditions moyennes d'utilisation de MAGE, c'est-à-dire la fréquence moyenne relative de chaque type de requête. Pour cela, a été utilisé comme application typique pour MAGE le système clefs en mains SCRIB (Saisie et Contrôle de Résultats et d'Informations Biologiques) [T.SAG.74] [SAG.76] Pour l'évaluation, quelques programmes de SCRIB ont été écrits en langage d'accès MAGE. Connaissant les conditions moyennes d'utilisation pour chaque type de requête, a été déterminé le nombre approximatif d'appels par type de requête. Ce nombre est fonction du "mode" d'accès et du "type" d'élément adresse.

Les résultats de cette évaluation ont établi que pour 100 requêtes utilisateur au PBD, 83 accès disque sont nécessaires. Ces accès sont distribués de la manière suivante :

- structure 10 accès
- dictionnaire 37,3 "
- données 35,8 "

Ces résultats tiennent compte de l'optimisation des accès disque et aussi du fait que certaines requêtes utilisateur ne nécessitent pas d'accès disque. L'évaluation a été réalisée en prenant en compte l'utilisation de 1 K octet de mémoire pour la structure, une entrée dictionnaire par contexte (10 octets) et une zone de 256 octets pour les données par contexte.

V.5.3 Suggestion pour l'architecture matérielle

En utilisant les deux évaluations précédentes, le choix de l'architecture matérielle a été fait suivant les critères de choix qui suivent.

Le temps moyen entre deux accès consécutifs au disque est fourni par le fabricant : 38 ms pour la famille de disques CDC 9760 [T.CDC.77] (temps moyen de recherche cyclindre + délai moyen de rotation). Utilisant un algorithme d'optimisation de déplacement du bras du disque, ce temps peut tomber à 29 ms [BER.78]

L'évaluation qui a été faite sur le nombre d'accès disque par requête a montré que pour 100 requêtes au PBD 83 accès disques étaient nécessaires, soit 0,83 accès par requête. En multipliant ce facteur par le précédent, on arrive à 0,83 x 29 ms = 24 ms pour le temps moyen parmis entre deux requêtes successives. C'est le temps moyen maximum permis pour le traitement par requête, si une bonne performance est souhaitée.

Cette valeur, 24 ms, comparée aux 12,5 ms de l'évaluation du temps de traitement par requête, donne un facteur de 52 %. Cela montre qu'un microprocesseur, dans le cas de M6800 tournant à 1 MHz peut s'occuper tout seul des tâches du PBD.

Cependant, si on prend en compte :

- l'imprécision de la méthode d'évaluation, qui est statistique ;
- que d'autres facteurs peuvent augmenter ce pourcentage, par exemple, les unités de disque avec un accès plus rapide;
- le coût des microprocesseurs, toujours en baisse par rapport à leur puissantce ;
- qu'il y a intérêt que le PBD traite les requêtes à un débit suffisant pour entretenir une file d'attente sur le disque permettant l'utilisation d'un algorithme d'optimisation du mouvement du bras ;
- et surtout, le fait que l'utilisation d'un seul 6800 occasionnerait des conflits entre les interruptions de l'utilisateur et du disque. Ces sources d'interruptions doivent être prises en compte le plus vite possible de manière à obtenir une utilisation maximum du

disque et pour éviter des attentes du (ou des) processeurs utilisateurs.

Tous ces facteurs ont amené à distribuer le travail entre deux microprocesseurs, un connecté à l'utilisateur et l'autre au disque, de manière à distribuer les sources d'interrruption. Un autre facteur qui renforce ce choix est l'aspect sécurité, vu qu'avec deux processeurs l'un peut toujours surveiller l'autre de manière à détecter une panne éventuelle et donc d'améliorer la sûreté.

L'architecture en bloc proposée (Fig. V.4) suggère que chaque microprocesseur ait sa propre mémoire de programme (ROM) et de travail (RAM). La communication entre les deux processeurs se fait par l'intermédiaire d'une mémoire commune. Cette mémoire regroupe les contexte utilisateurs avec les informations concernant les requêtes et leur gestion. Elle joue le rôle d'une boite aux lettres, permettant une communication asynchrone. Le processeur P1 aura la charge de communiquer avec l'utilisateur à travers le bloc interface. Cet interface sera différent en fonction du type de liaison avec le processeur utilisateur (et du type de processeur utilisateur lui-même).

Quant au transfert de données disques, les débits des disques obligent à avoir une mémoire tampon (buffer) comme moyen de communication avec le processeur P2.

En ce qui concerne le logiciel, le processeur P1 exécutera PGE, PREQ, pendant que P2 exécutera PAST, PADIC, PARE, PGDISQ. PNI et PRQ seraient partagés entre P1 et P2.

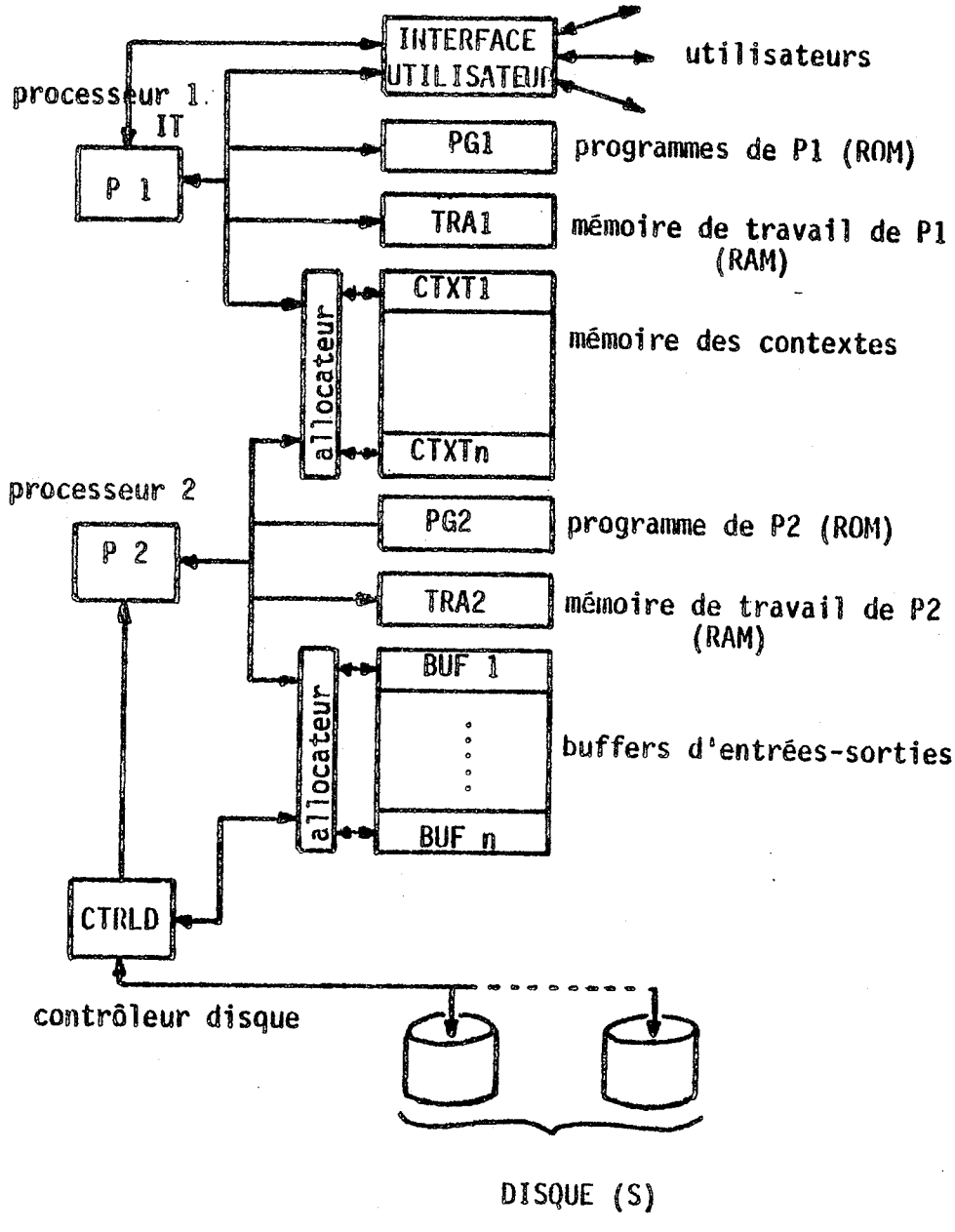


Fig. V.4 - Suggestion pour l'architecture matérielle

CHAPITRE VI

ARCHITECTURE DU PBD-MAGE

1. ARCHITECTURES NON NUMERIQUES
2. ARCHITECTURE DU PBD-MAGE
   - 1- INTRODUCTION
   - 2- PRESENTATION DE L'ARCHITECTURE RETENUE
   - 3- COMMUNICATION AVEC L'UTILISATEUR
   - 4- COMMUNICATION ENTRE LES PROCESSEURS DU PBD : LA MEMOIRE DES CONTEXTES
   - 5- MECANISME POUR L'ADRESSAGE DES CONTEXTES
   - 6- PROCESSEURS ET MEMOIRES
   - 7- COMMUNICATION ENTRE LE CONTROLEUR DISQUE ET LE PROCESSEUR P2 : LA MEMOIRE TAMPON
   - 8- CONTROLEUR DISQUE
3. CONSTRUCTION DE PROTOTYPES
   - 1- LE PROTOTYPE du PBD-MAGE
4. VIE D'UN PROJET

Ce chapitre fait un tour d'horizon des architectures liées aux PBD puis présente l'architecture du PBD.MAGE.

VI.1 ARCHITECTURES NON NUMERIQUES

Ces dernières années, la recherche et le développement de machines consacrées aux bases de données ont porté un intérêt croissant aux mémoires associatives, mémoires secondaires intelligentes et évidemment les systèmes de gestion de base de données (SGBD). Ces recherches ont montré que les architectures actuelles des ordinateurs, qui continuent à etre basées sur les idées fondamentales lancées par Von Newmann [NEU.66][BEL.71] , ne sont pas adaptées à des activités de recherche de données [LIP.76] [COP.73] [HSI.77].
On peut remarquer que les opérations numériques restent très élémentaires, les principaux progrès ayant eu lieu au niveau de l'adressage, de certaines opérations "systèmes" et des transferts de contrôle.

Le nom architecteur non-numérique [MUK.78][ZAK.77] [VRA.76] permet de différencier toutes les architectures différentes de celles des ordinateurs actuels qui ont des architectures basées sur des opérations numériques ; c'est la cas des opérateurs de recherche de données. Le terme non-numérique a été utilisé la première fois lors d'un "ACM workshop" en 74 et depuis lors est plutôt associé aux architectures des bases de données spécialisées. Ces architectures de BD utilisent les recherches en mémoires associatives et les mémoires secondaires intelligentes.

Le grand problème des actuels SGBD est le fait que toute donnée à manipuler doit être transférée de la mémoire secondaire vers la mémoire primaire pour pouvoir effectuer les opérations nécessaires.

Tout cela prend du temps. Les projets CASSM, RAP, RARES et d' autres (voir chap II) proposent de pallier à ce problème en associant à la mémoire secondaire l'intelligence nécessaire. "Il est plus facile de bouger David (l'intelligence) vers la montagne (données) que d'apporter la montagne à David" (T.C. Chen).

L'idée est d'avoir des processeurs associés aux mémoires secondaires qui exécuteraient un "adressage par le contenu". L'une des composantes des systèmes informatiques qui a été la plus délaissée depuis leur création, a été les mémoires secondaires. Leur architecture n'a pas changé beaucoup et un des principaux problèmes actuels des systèmes se trouve être le transfert des données entre la mémoire secondaire et la mémoire principale ; c'est le point d'étranglement. (Fig. VI.1)

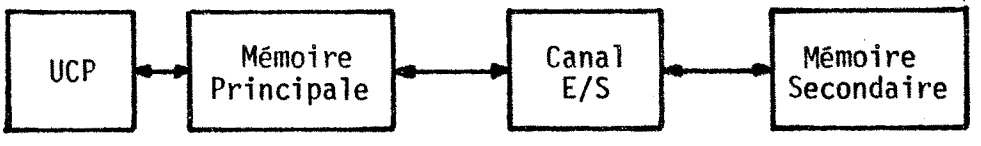


Fig VI.1 Architecture des actuels systèmes

Les systèmes à architectures non-numériques pourraient utiliser des "circuits logiques cellulaires" [SU.79] associés aux têtes des unités disque ou à tout autre mémoire secondaire circulante. Ces circuits spéciaux exécuteraient des opérations telles que la comparaison (association), le marquage de mots, quelques opérations numériques etc.. Le grand impact de ces architectures est la possibilité, avec des disques à têtes fixes de réaliser simultanément une recherche associative sur toutes les pistes de tous les cylindres en même temps. En une seule rotation du disque, toutes les données stockées sont balayées. Ces machines auraient donc un cycle égal à un tour du disque.

L'arrivée de telles machines amènerait beaucoupe de solutions aux problèmes actuels dans les recherches en BD relationnelles et en intelligence artificielle entre autres (Fig VI.2)

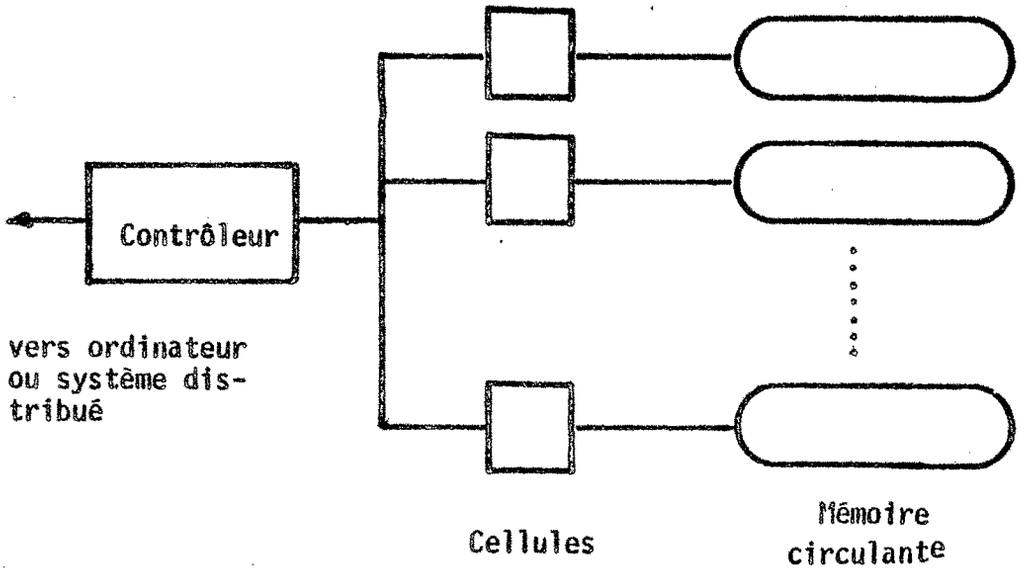


Fig VI.2 - Architecture d'une machine à logique cellulaire

VI.2 ARCHITECTURE DU PBD-MAGE

Le projet du PBD-MAGE ayant pour but de réaliser une machine industrialisable, ne possède pas de processeurs associatifs sur chaque tête de disque. Il a été vu dans le chap. II que ces PBD (CASSM, RAP, RARES) ne sont pas viables actuellement compte tenu de leur énorme coût. Cependant le PBD-MAGE a dans son contrôleur disques quelques fonctions de recherche de mots-clés basées sur le contenu. Cette opération permet de retrouver la tête de certains fichiers et donc d'accélérer la méthode d'accès aux données (voir chap. VII).

Ce chapitre a pour but de présenter l'architecture du PBD MAGE en discutant certains détails, car si dans son ensemble le PBD possède une certaine originalité, dans ses blocs on retrouvera l'utilisation de techniques bien connues mais adaptées aux besoins du projet.

Le chapitre précédent (chap. V) a donné un résumé du travail qui a abouti à une suggestion pour l'architecture du PBD-MAGE. A partir de cette suggestion a été menée une étude pour aboutir à l' architecture matérielle finale.

VI.2.1 Introduction

Un choix dans un projet est toujours difficile, car pour bien le faire, plusieurs variables doivent être prises en compte, ce qui n'est pas toujours possible. Souvent, au fur et à mesure du développement d'un projet, de nouvelles variables apparaissent. Par conséquent, un projet doit éviter d' être trop rigide et il doit plutôt évoluer tout au long de son développement.

Les critères de choix adoptés dans ce travail sont :

a) coût - c'est la variable la plus importante, encore plus dans un projet industrialisable.
b) fiabilité des circuits - ce critère , par exemple, oblige à éviter de faire le choix de circuits récemment lancés sur le marché avant qu'ils n'aient fait leurs preuves.
c) économie du nombre de composants et de liaisons - ce critère est lié au premier mais concerne plutôt l'implantations (lay out). La minimisation des liaisons entre composants, plaques, etc.., permet de minimiser le coût et d'augmenter la fiabilité. Moins de liaisons, moins d'erreurs de connexion, moins de points sensibles aux défauts.

d) modularité - préserver si possible une modularité au niveau des plaques et de l'implantation pour que le changement, par exemple d'un processeur, n'occasionne pas la modifications des autres plaques .

e) nouvelles technologies - pour qu'un projet soit valable et puisse avoir la plus grande vie possible, il est intéressant qu'il prenne en compte l'évolution technologique. Ce critère n'est pas en contradiction avec le critère "b" concernant la fiabilité des circuits, car cette prise en compte des nouveaux produits doit être faite judicieusement en tenant compte de la fiabilité de ceux-ci.

Il est désagréable de trouver des projets qui après leur développement aboutissent à des matériels complètement dépassés, compte tenu de la vitesse d'évolution des technologies. Cela est très fréquent avec tout ce qui se rapporte directement à la technologie des composants et des mémoires.

f) l'esthétique - pourquoi pas l'esthétique ? Le terme architecture et donc architecture des machines, est intimement lié à l'esthétique. C'est un critère un peu difficile à définir car il concerne un peu tous les autres critères et des petits critères tel que la clarté. Un projet, clair, simple, bien documenté est beaucoup plus facilement compréhensible, modifiable. L'esthétique serait le critère qui permettrait de lier les autres critères en créant une harmonie du projet final.

D'une manière générale, dans le projetdu PBD-MAGE, le matériel et les circuits utilisés ont été maintenus les plus standards et conventionnels possibles. Cela permet d'avoir une structure fiable avec le moins possible de problèmes ultérieurs. L'utilisation de techniques plus raffinées a été réservé aux points les plus critiques.

Deux postulats ont été pris comme base et non remis en cause :
1) l'utilisation du microprocesseur M 6800 lié à un choix industriel (ou de ses successeurs éventuels) ;
2) la suggestion de l'architecture matérielle en modules.

Pour que dans ce projet, le logiciel et le matériel puissent être menéss en parallèle, il fallait que les principales lignes du fonctionnement du PBD soient figées et éviter tout changement, car la structure d'accès aux données et le logiciel sont basés sur cette architecture. Quelques évolutions étaient possibles, mais les grandes lignes de l'architectures devaient être maintenues.

VI.2.2 Présentation de l'architecture retenue

L'architecture retenue [NAV.77] (Fig VI.3) est basée sur deux bus, l'un du processeur P1 et l'autre du processeur P2. Sur le premier bus est connecté le processeur P1, la mémoire de programme réalisée avec des circuits ROM, sa mémoire de travail avec des RAMS, ses circuits d'interface avec le milieu extérieur du PBD et la mémoire des contextes qui est le moyen de communication avec l'autre processeur.

Sur le deuxième bus, est connecté le processeur P2, la mémoire ROM de programmes, sa mémoire RAM de travail, la mémoire RAM des contextes qui est le lien avec le processeur P1, la mémoire tampon (buffer) qui est le moyen de transfert des données avec les unités disque et le contrôleur disque. Un circuit de contrôle permet d' arbitrer l'accès à la mémoire des contextes par les deux processeurs. A la plupartde ces blocs correspondra des cartes au niveau du prototype (voir VI.3)

Les deux blocs qui réalisent l'interface du PBD avec le milieu extérieur : le bloc d'interface utilisateur et le bloc contrôleur disque, doivent être très modulaires de manière à ce qu'ils puissent s'adapter respectivement aux changements d'utilisateur et aux changements du type d'unité disque.

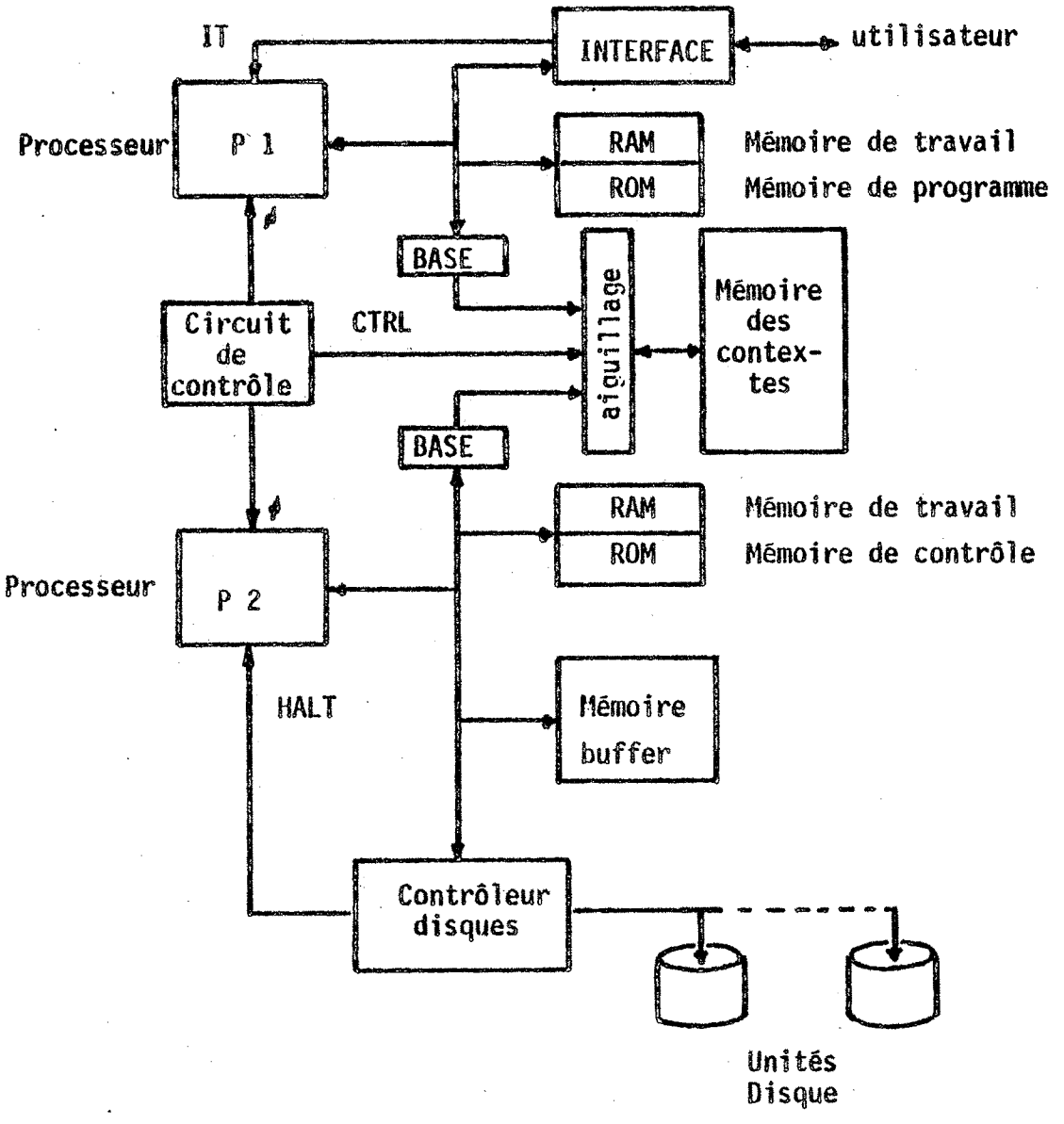


Fig VI.3 - Architecture du PBD-MAGE

VI.2.3 Communication avec l'utilisateur

C'est l'unique lient du PBD avec le milieu extérieur (ses utilisateurs). Ce lien est représenté par le bloc "interface utilisateur" (Fig VI.4). Par lui circuleront les requêtes et les réponses échangées entre le PBD et l'ordinateur où le réseau de processeurs d'un système distribué.

Cette unité de communication devra être adaptée aux besoins du système utilisateur du PBD, ce qui fait qu'elle ne peut pas être définie à priori. Ce bloc d'interface réalise la communication entre le processeur P1 et le milieu extérieur. Chaque fois qu'il y a un message de l'extérieur à recevoir, le microprocesseur sera réveillé par une interruption.

La définition de MAGE indique que la communication du PBD avec le milieu extérieur sera faite à travers un language de niveau moyen ou élevé et par conséquent le flux de données ne devra pas être grand. Une solution d'échange de messages, semble être la plus indiquée du point de vue économique. Le mécanisme devra être du type connexion directe avec partage du lien de connection. Ce partage du lien pourra se faire à travers une mémoire centrale ou un bus global. Pour l'utilisation dans les systèmes distribués du type Corail, la solution du bus global pourra être retenue.

Le mécanisme de communication utilisé sera celui du système sur lequel le PBD sera raccordé. Selon le processeur utilisateur, ce lien pourra être physiquement parallèle ou série (ligne série). Les protocoles de gestion du dialogue physique pourront être associés à l'interface utilisateur de telle sorte que le changement du système vers un autre type d'utilisateur implique seulement un changement de la plaque d'interface utilisateur.

Les programmes nécessaires à la gestion du dialogue seront en mémoire ROM dans l'interface. Cela donne une modularité de l'interface par rapport aux différents types de liaison.

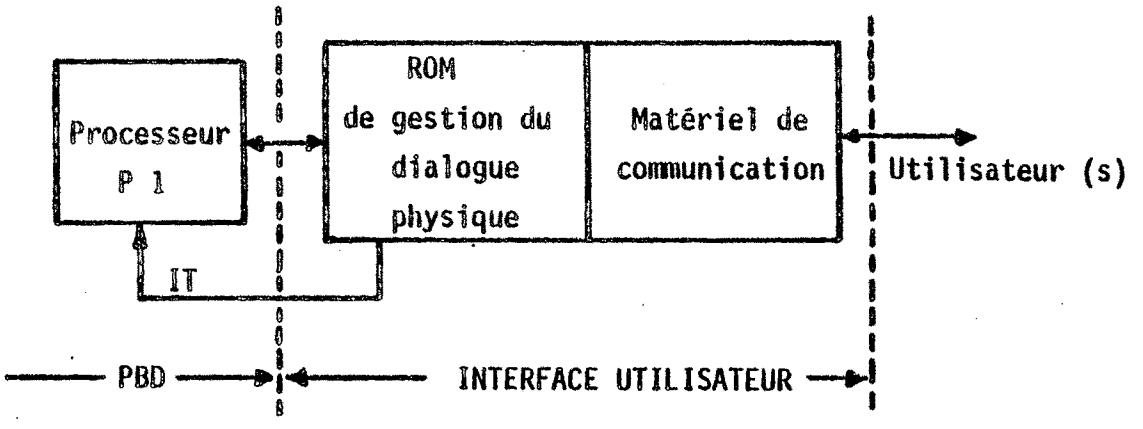


Fig VI.4 - interface du PBD avec l'extérieur

VI.2.4 Communication entre les processeurs du PBD : la mémoire des contextes

Le chapitre V a montré que le choix d'une structure a deux microprocesseurs était influencé par l'indépendance nécessaire que devait avoir un microprocesseur par rapport à l'autre. Chaque microprocesseur étant chargé d'une interruption, le processeur P1 est chargé des interruptions provenant de l'utilisateur et le processeur P2 de celles venant du disque. L'orthogonalité que doivent avoir ces deux processeurs, oblige à utiliser un mécanisme de communication entre eux qui puisse maintenir cette orthogonalité.

Deux solutions existaient, une communication direct par un lien direct entre les processeurs qui obligerait à utiliser une interruption pour annoncer à l'autre processeur l'envoi d'un message, ou bien une communication directe à travers une ressource partagée (chap. III.3) qui servirait de tampon entre ces deux processeurs

tout en leur laissant leur liberté. La première solution entraîne uen perte d'indépendance et donc d'orthogaonalité, tandis que la deuxième solution maintient l'orthogonalité des processeurs. Le choix s'est donc porté sur la deuxième solution : l'utilisation d'une interconnexion directe à travers le partage d'un lien de communication. Le lien de communication utilisé est une mémoire partagée entre ces deux processeurs.

Dans le sytème MAGE, chaque utilisateur travaillant sur la BD a besoin d'un ou plusieurs contextes. Ces contextes contiennent toute l'information nécessaire pour traiter une requête utilisateur. les contextes sont dans la mémoire partagée entre les processeurs P1 et P2 et constituent le moyen de communication et de synchornisation entre les processeurs. Cette mémoire est appelée "Mémoire des Contextes". Elle contient un maximum de 64 contextes, de 512 octets chacun (Fig VI.5), ce qui représente 32 K octets pour toute la mémoire des contextes dans sa version maximale.

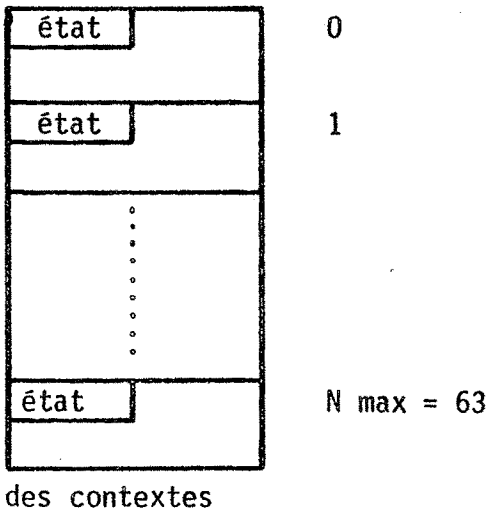


Fig VI.5 - Mémoire des contextes

Chaque contexte possède un octet qui indique son état, qui peut être [BER.78] :

- non affecté
- inactif (pas de requêtes en traitement sur ce contexte)
- en attente de traitement par P1
- en traitement par P1
- en attente de traitement par P2
- en traitement par P2
- en attente d'échange avec le disque
- en attente de communication avec l'extérieur (utilisateur)
- en attente sur exclusion mutuelle.

Ce système de communication est une sorte de file d'attente, où un processeur prépare les données et indique à travers l'octet d'état que le contexte est en attente de traitement par l'autre processeur. Quand un contexte traité par un processeur nécessité l'intervention de l'autre, il suffit que le premier place le code de l'opération souhaitée et les paramètres nécessaires dans le contexte et qu'il le mette en état d'attente de traitement par l'autre processeur. Quand un processeur finit de traiter un contexte, il ira balayer les états des divers contextes dans la mémoire, jusqu'à en trouver un en attente de son traitement. A cet instant il appelera le programme qui correspond au conde contenu dans le contexte et traitera les données, après avoir changé l'état du contexte.

A un instant donné, un contexte ne peut être traité que par un seul processeur, mais les deux processeurs peuvent accéder simultanément un contexte pour connaître son état. La solution matérielle pour permettre l'accès simultané à l'état d'un contexte , et le problème matériel du partage de la mémoire des contextes, est fonction du taux moyen d'accès à cette mémoire. L'évaluation des temps de traitement (chap V.5.1) donne un taux maximale d'accès aux contextes de 12 % pour chaque processeur. Ce taux de 12 % correspond à un cas particulier ; celui où un processeur n'ayant pas de travail, passe son temps à tester l'état des contextes.

Quand un processeur est occupé par le traitement d'une requête ce temps baisse fortement. Avec ce taux maximal d'accès, on constate que la valeurque peut atteindre le taux de conflit est de 1,44 %. Cette valeur montre que le conflit d'accès à la mémoire des contextes par lesprocesseurs est inférieur à 2 %, donc une valeur infime. Avec un tel taux, le problème de l'accès simultané se simplifie et, il eput être résolu par le retardement de l'horloge de l'un des microprocesseurs lors des conflits [NAV.77].

VI.2.5 Mécanisme d'adressage des contextes

Le logiciel de MAGE travaille sur plusieurs contextes (jusqu'à 64) avec le même programme. Tous les contextes se ressemblent et possèdent un format fixe pour les paramètres et données. Avec une solution logicielle pure, le mode d'adressage indexé du microprocesseur M 6800 permet d'accéder les données avec des adresses qui n'ont pas été fixées au moment de l'assemblage. Cependant, l'usage de ce mode d'adressage occasionne une perte de temps due aux nombreuses manipulations d'index, car un contexte fait 512 octets et l'adressage indexé du MC 6800 ne permet un déplacement que de 256 octets. La solution pour ce problème est l'utilisation d'un circuit matériel auxiliaire qui permettrait de traiter tous les contextes avec un même programme, sans entraîner de manipulation de l'index.

Pour résoudre ce problème, un registre de base externe, a été utilisé qui pointe vers le contexte en traitement. L'adressage des contextes est obtenu par la concaténation de l'adresse fournie par le microprocesseur avec le contenue du registre de base externe. Ce registre a une longueur de 6 bits pour permettre d'adresser 64 contextes. Les 6 bits du registre plus les 9 bits les moins significatifs de l'adresse du microprocesseur, composent l'adresse de la mémoire des contextes (Fig VI.6).

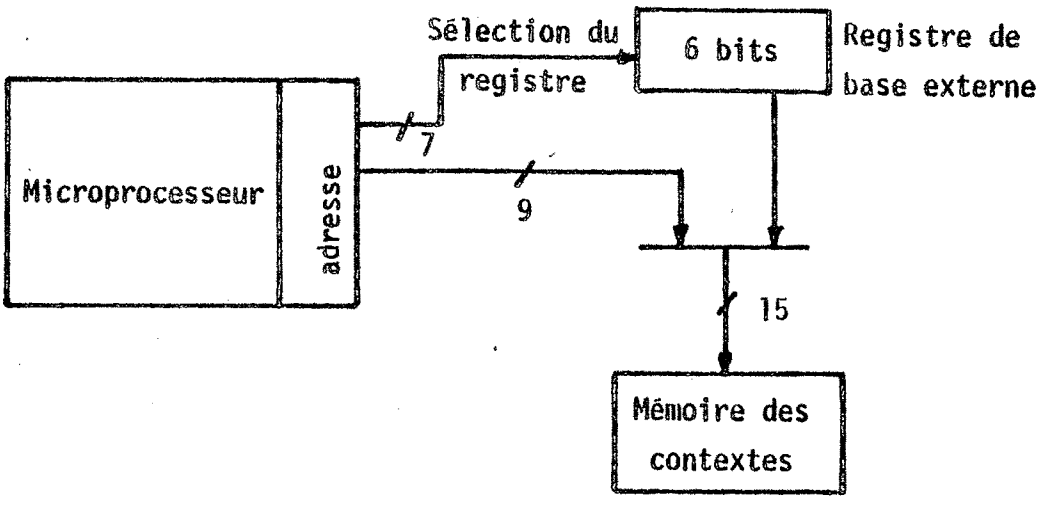


Fig VI.6 - Adressage de la mémoire des contextes

L'utilisation du registre de base externe permet d'augmenter la capacité d'adressage du microprocesseur. En réalité, les 32 K octets de la mémoire des contextes ne vont être représentés que juste par une fenêtre de 512 mots dans le champ d'adressage de 64 K octets du microprocesseur (Fig. VI.7)

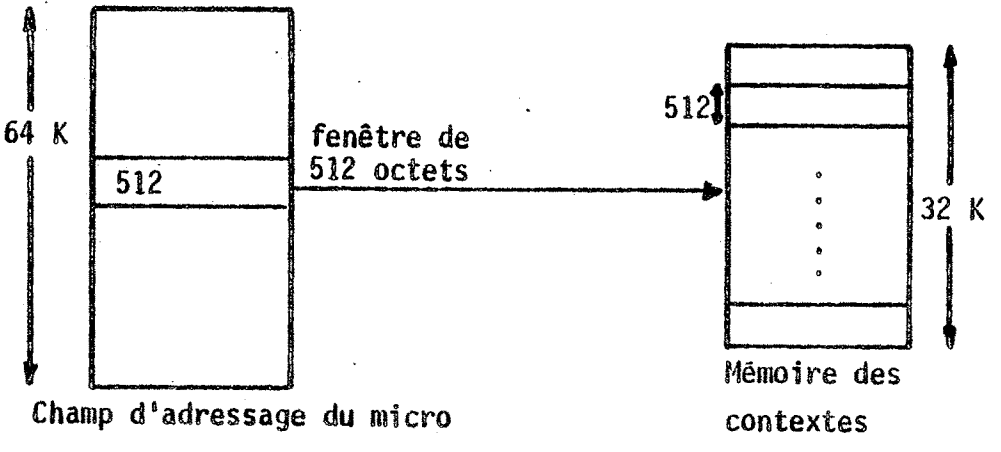


FIG VI.7 - Fenêtre d'adressage des microprocesseurs

A chaque processeur, P1 et P2, on associe un registre de base qui lui indique le contexte avec lequel il travaille. Ces bases sont

chargeables avec la valeur du "pointeur contexte" comme une position de mémoire quelconque. Chaque fois qu'un processeur veut travailler avec un contexte, il commence par charger le registre de base avec le numéro du contexte désiré, puis il effectue le traitement réalisé sur les contextes (d'après les programmes de MAGE), les processeurs ne passent pas d'un contexte à l'autre, mais au contraire traitent un contexte avant de passer à l'autre, ce qui ne nécessite que la présence d'un seul registre de base par processeur [NAV.77] . Toutefois, il est envisageable d'associer une deuxième base à un des processeurs, dans le cas où il aurait à travailler simultanément sur deux contextes. Avec une telle structure, le processeur aurait deux fenêtres de 512 octets pour adresser simultanément deux contextextes et ce n'est que par ces deux champs d'adresse que les contextes pourraient être accédés.

VI.2.6 Processeurs et mémoires

Dans le PBD l'utilisation des processeurs et mémoires a été standard (voir chap. VI.3) et donc il n'y a pas grand intérêt de les présenter ici, cependant il est intéressant de formuler quelques remarques.

Les processeurs emploient comme microprocesseur le M 6800 ou SFS 96800, les mémoires qui leur sont rattachées sont de deux types, des mémoires mortes (ROM) pour les programmes et des mémoires vives (RAM) pour les variables et les liens. Pour le prototype, seules les mémoires vives sont utilisées. Chaque processeur a 16 K octets de mémoires, ce volume étant déterminé d'après les estimations.

Pour la mémoire des contextes, sa dimension de 32 K octets, rend intéressant d'employer des mémoires dynamiques pour minimiser les coûts ; cela oblige cependant à avoir un mécanisme de rafraîchissement. Etant donné qu'il existe déjà un conflit d'accès des processeurs P1 et P2 à la mémoire, il est nécessaire d'avoir un algorithme câblé, entre trois utilisateurs, pour résoudre le conflit entre le rafraîchissement et les deux processeurs. La figure VI.8 donne l'organigramme de cet algorithme. Cet algorithme sera réalisé par le bloc de contrôle (Fig VI.3)

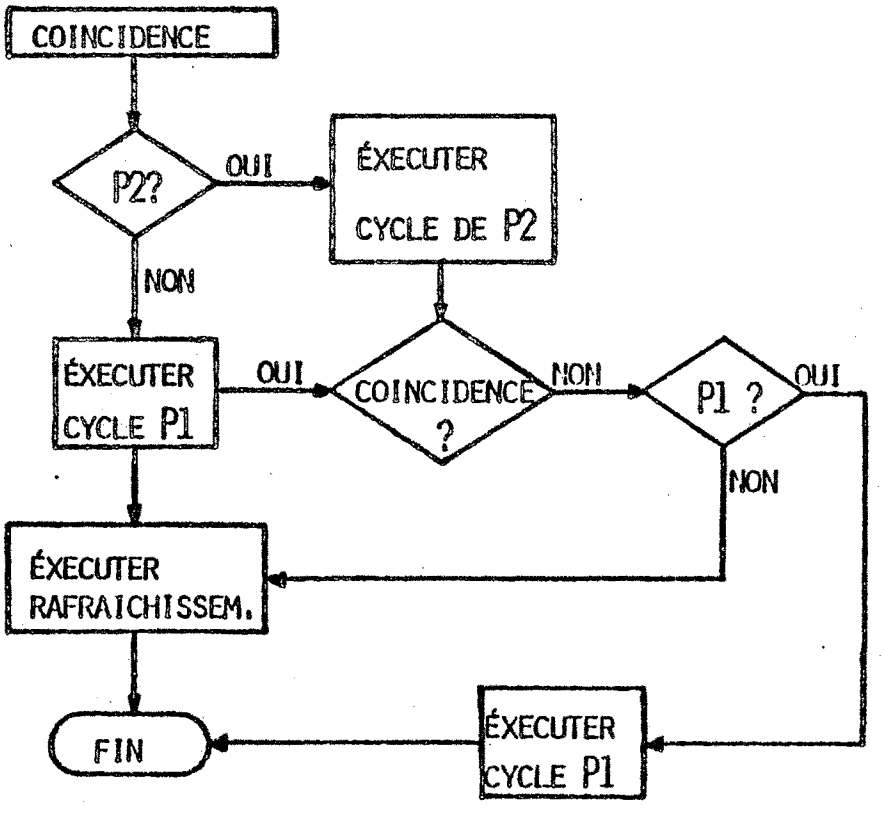


Fig VI.8 - Algorithme temporel d'allocation de la mémoire des contextes

Cet algorithme temporel d'allocation de la mémoire des contextes utilise un mécanisme de priorité pour départager les conflits. La priorité la plus haute est donnée au processeur P2, compte tenu de sa grande charge de traitement, ensuite vient le processeur P1, puis le rafraîchissement. Ce mécanisme oblige le rafraîchissement de gérer le contrôle de deux signaux, l'un de demande de rafraîchissement (refresh request) et l'autre autorisant le rafraîchissement (refresh grant) [NAV.77]

VI.2.7 Communication entre le contrôleur disque et le processeur P2 : la mémoire tampon.

Le partage de la mémoire buffer (tampon) entre le processeur P2 et le contrôleur disque, présente des problèmes différents de ceux retrouvés dans le partage de la mémoire des contextes entre les deux processeurs. Pratiquement, le flux des données provenant du contrôleur disque est élevé, de l'ordre de 1,2 M octets/secondes, et continu pendant le transfert. Cela oblige à avoir un mécanisme de communication rapide pour transférer un octet par microseconde. A cette fréquence, la mémoire buffer ne peut plus être accédée par le microprocesseur P2 pendant le transfert.

Pour résoudre ce problème, il est nécessaire de connaître le temps de transfert moyen d'un secteur de données vers la mémoire buffer. Avec une unité de disque d'un temps de rotation de 16 ms et 64 secteurs par piste (par exemple), le temps de transfert d'un secteur sera de l'ordre de 0,25 ms. Ce temps par rapport aux 29 ms (chap. V 5.3) de temps total moyen pour un accès disque (seek time et latency) donne un pourcentage de moins de 0,9 %. Le temps de transfert d'un secteur représente donc 0,9 % du temps moyen d'un accès disque pour des unités 9760 [T.CDC.77]. Cela est faible si on

considère que les évaluations (chap V.5.3.) indiquent que les microprocesseurs auront un temps d'inactivité de plus de 50 %. Les 2 % ne vont donc pas changer la performance du processeur P2. Par conséquent pour résoudre ce problème de partage de la mémoire buffer, il est indiqué de bloquer le microprocesseur lors du transfert, utilisant le signal de "Halt".

L'arrêt, du processeur P2 lors d'un transfert disque est la méthode la plus économique pour réaliser le partage de la mémoire buffer entre le contrôleur disque et P2. Cette approche n'introduit aucun inconvénient, vu que l'origine de l'interruption de P2 est le disque lui-même. Si P2 est arrêté par un transfert disque il ne pourra pas être interrompu par un autre événement que celui de la fin du transfert lui-même. Un autre avantage de cette technique est que l'arrêt , comme la reprise, du microprocesseur est fait immédiatement sans des retards dus aux sauvegardes de registres. Ce mécanisme de communication est une sorte de DMA par bloc (Fig.VI.9)

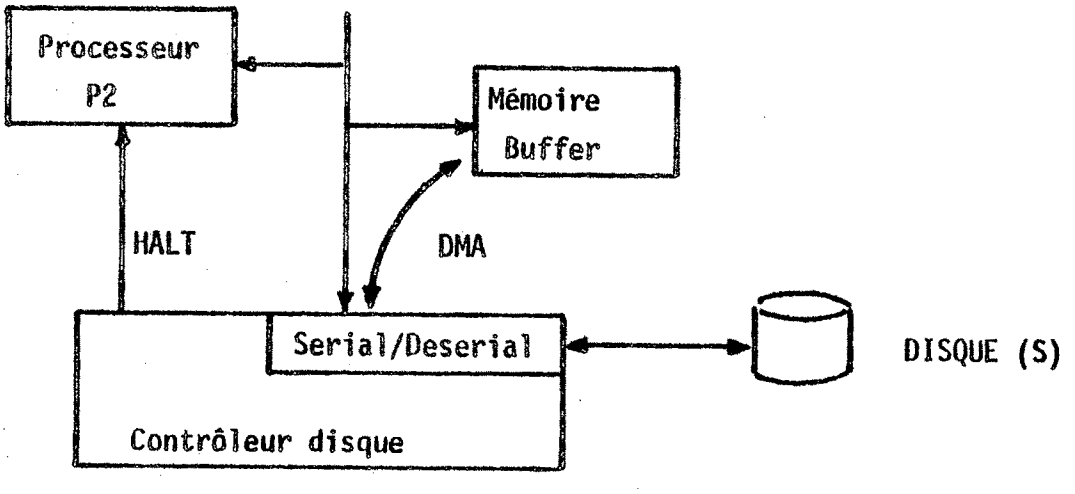


Fig VI.9 - Mécanisme de communication avec la mémoire buffer.

VI.2.8 Processeur Disque

Le contrôleur disque est l'unité qui permet le transfert des données entre le processeur P2 et le disque, via la mémoire buffer. Son but est de recevoir les ordres du processeur P2 et de les transformer en ordres pour le disque, de manière à exécuter les lectures et les écritures de données.

La fonction processeur disque s'étend jusqu'au processeur P2. Elle comprend en plus du contrôle du disque, lui-même l'optimisation du positionnement des têtes, la conversion des adresses, la gestion des requêtes, l'initialisation, le transfert des données, etc... Les disques utilisés CDC 9760 ou AMPEX 900 ont un taux de transfert des bits de 10 MHz, il n'est donc pas possible d'utiliser directement un microprocesseur pour contrôler les disques. Le transfert des données est effectué avec des circuits TTL rapides, et des tranches de processeurs (bit-slice) qui gèrent les mécanismes de transfert. Le microprocesseur 6800 (P2) est utilisé pour des fonctions comme le calcul de l'optimisation du déplacement du bras, la conversion d'adresse, la gestion des requêtes et d'autres tâches nécessitant du traitement. De la fonction processeur disque l'unique partie qui ne fait pas partie de l'unité contrôleur disque c'est le processeur P2 avec sa mémoire, bien que le rôle principal de P2 ne soit pas le contrôleur disque.

Dans le contrôleur disque, il existe deux chemins distincts : un pour les données et l'autre pour les signaux de contrôle. Les principales fonctions exécutées sont la lecture, l'écriture, le formatage du disque, la comparaison des données et la recherche de mots-clés pour accélerer les accès aux fichiers.

Cette unité sera l'objet des deux prochains chapitres.

VI. 3 REALISATION DU PROTOTYPE

A l'occasion de tout développement de projet qui abouti à une réalisation matérielle, le moment de prendre la décision de passer du projet papier au prototype arrive toujours. La construction de celui-ci pose le problème de décider sur quel type d'environnement doit être connecté le projet ? quel standard ? quel modèle de cartes ? de connecteur ? etc...

L'évolution de la technologie aujourd'hui, fait que chaque jour apparaissent de nouveaux boitiers, intégrant en un seul composant ce qui état réalisé par plusieurs boitiers il y a quelques années ou même quelques mois. Cela oblige les projets à aller vite pour que le produit puisse arriver sur le marché à temps pour avoir une durée de vie suffisante.

La tâche d'un architecte de machines électroniques évoluées n'est plus de faire le projet au composant près, mais d'assembler des blocs avec le but d'obtenir une nouvelle architecture. Cela n'empêche pas, que lorsque les contraintes sont spécifiques, de réaliser le projet du bloc particulier au niveau du composant.

Par exemple si la notion de carte de circuits est associée à celle de bloc, le but de l'architecture sera dans ce cas, de réaliser le projet au niveau des blocs et ensuite d'utiliser les cartes pour la construction du prototype par substitution des blocs. Il n'y a pas d'intérêt, par exemple, à développer une plaque d'UC ou une mémoire, car elles existent déjà sur le marché prêtes et testées. Cette technique de développement de projets peut être même poussé, dans certains cas, au niveau de l'industrialisation du produit, quand il s'avère plus économique d'utiliser des cartes existantes plutôt que de les construire.

Un autre point important qui amène l'utilisation de plaques standard, sont les outils de mise au point associés (débug). Ces outils logiciels de test et de mise au point sont très importants dans le développement d'un prototype, cars ils sont nécessaires pour la mise au point et les test d'une carte. L'utilisation de ceux-ci permet de minimiser le temps, ne serait-ce qu'en fournissant des programmes de gestion d'entrée/sortie des données, codage ASC II pour téléimprimeur, conversion hexadécimal, etc.. En plus de cela, il y a les outils de mise au point, comme la lecture/modification d'une position de la mémoire, l'utilisation de points d'arrêts, l'exécution de programmes pas à pas etc...

Le fait d'avoir des outils de test et des cartes au point, permet à l'architecte de se dégager de la préoccupation du fonctionnement du matériel de base et donc de porter son effort sur l' architecture au niveau des blocs fonctionnels.

Comme conclusion, on peut dire que la construction d'un prototype basée sur des outils de développements standards est d'un très grand intérêt. Même si son coût est un peu plus élevé, les facilités des logiciels de test, la modularité et par conséquence la possibilité d'évoluer, rendent économiques ces outils à moyenne échéance.

VI.3.1 Le prototype du PBD-MAGE

Comme suggestion des ingénieurs de la SAGEM, cette technique d' utiliser des outils de développements prêts, a été employés pour le PBD-MAGE, avec le but de minimiser les temps de développement et de réalisation du prototype. L'utilisation de plaques standard (Motorola Micromodule) [T. MOT.77] a été adoptée pour remplacer la plupart des blocs du schéma de la FigV.4. Les blocs

spécifiques qui ne pouvaient pas être remplacés par des plaques standards, ont été construits sur des plaques vierges (à "wraper"). Cela a permis de maintenir entre toutes les plaques, les mêmes spécifications physqiues (dimensions, connecteurs, etc..)

La figure VI.10 présente le projet de l'utilisation de ces plaques standards pour la construction du prototype du PBD. La plaque du processeur P1 a un téléimprimeur qui lui est connecté directement. Cela permet au téléimprimeur d'être utilisé dans un premier cas à la mise au point du projet et plus tard pour la simulation des ordres de l'utilisateur. Les plaques standard UC possèdent les circuits de communications avec des terminaux.

Le processeur P2 pourra aussi avoir un tléléimprimeur de mise au point. Les deux processeurs, P1 et P2, seront premièrement testés indépendamment avant qu'ils communiquent entre eux.

Des plaques à câbler, la principale est le contrôleur disque, objet du prochain chapitre, les autres plaques sont utilisées pour la communication entre les bus et contiennent très peu de composants. Elles se chargent également de réaliser le partage de la mémoire des contextes entre les deux processeurs.

Le projet du PBD se présente comme une architecture de blocs avec des fonctions standard telles que, processeurs, mémoires, etc.. Cela dit, l'utilisation des plaques standards des fabricants, cadre bien dans les besoins du projet. Il est certain que quelques plaques spécifiques seront câblées.

Les possibilités d'évolution permettent, au niveau du prototupe, d'essayer de nouvelles technologies (nouvelles plaques) sans mettre en cause tout le projet. Par exemple, si le processeur P2

est trop chargé, il est possible de changer la plaque processeur réalisée avec un microprocesseur M 6800, par une plaque processeur réalisée avec un M 6809, sans que cela cause aucune modification matérielle. Quant aux programmes en mémoire vive RAM, au fur et à mesure qu'ils seront mis au point, ils pourront passer à des plaques de mémoire morte du type REPROM. Tous ces facteurs plaident pour l'utilisation des plaques standards.

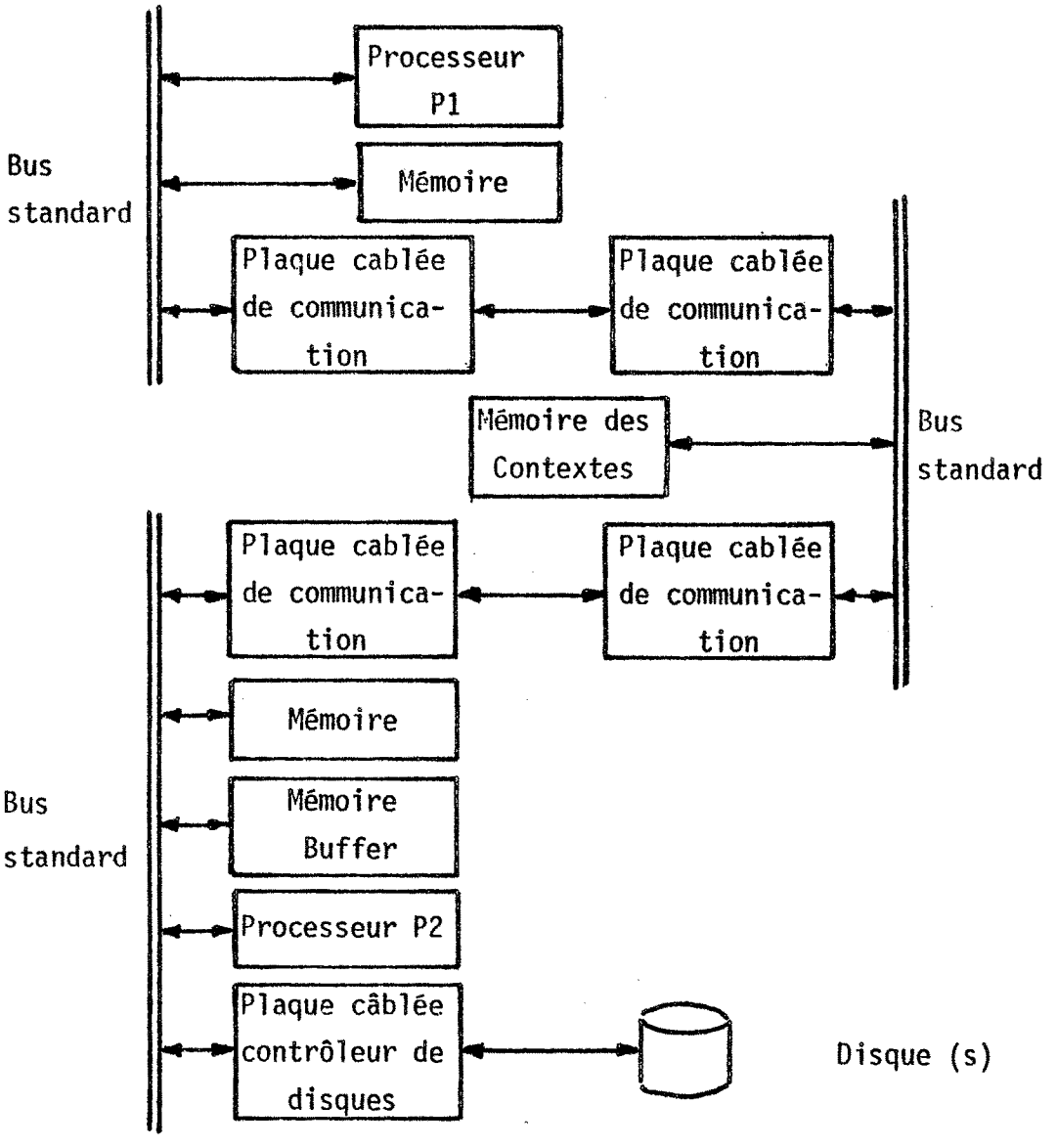


Fig VI.10 La suggestion de construction du PBD avec les plaques standard

VI.4 VIE D'UN PROJET

La vie d'un projet peut être mesurée, entre autres paramètres, par l'évolution de la technologie. Actuellement le nombre de transistors intégrables dans un chip double à peu près chaque année. Si on considère la vie utile d'une machine quelconque, comme la vie de ses cartes, il est possible de calculer cette relation.

D'une manière générale une carte avec ses composants est périmée quand elle peut être remplacée par un seul composant ou plus d' une nouvelle technologie. C'est le cas par exemple des circuits intégrés bascules qui ont substitué les cartes à transistors qui réalisaient la même fonction (flip-flop).

Prenons comme un postulat, qu'un ensemble de circuits composés par "n" portes logiques deviendra périmé lorsque la technologie pourra le réaliser en un seul circuit. Comme actuellement la technologie progresse à une facteur annuel d'intégration de deux, il est possible d'exprimer celle evolution technologique par 2t où t représente le nombre d'années et 2 le facteur d'évolution technologique. A partir de cela on obtient la formule de :

$$N = 2^t \qquad t = \log_2 n$$

pour le temps de vie d'un ensemble de circuits. Si l'ensemble de circuits représente une carte, c'est la carte qui sera périmée, s'il représente un système, c'est tout le système qui est dépassé technologiquement.

Une généralisation de cette formule donnerait :

$$\frac{s}{r} = \phi^t \qquad t = \log_\phi \frac{s}{r}$$

où : s - représente le nombre de portes logiques de l'ensemble de circuits ;
r - le nombre moyen de portes logiques que la technologie peut compacter dans un circuit
ϕ - le facteur d'évolution technologique par année
t - nombre d'années.

Le facteur s/r permet de déterminer une moyenne du nombre de portes logiques que peut contenir un circuit dans la technologie de l'époque.

L'objectif de cette formule empirique et simpliste, n'est pas d'être une mesure très exacte de la vie d'une carte ou système, car beaucoup plus de paramètres entrent en jeu. L'idée est plutôt de montrer à travers un des paramètres, l'évolution technologique, les problèmes qui se posent lors d'un projet et que les architectes de systèmes matériels doivent affronter.

En utilisant comme exemple une carte de mémoire de 2 K octets réalisée avec des boitiers 2102 qui contiennent chacun 1 K de bits. La vie d'une telle carte sera :

$$\frac{2\ K.8}{1\ K} = 2\ t \qquad 16 = 2\ t \qquad t = 4$$

Donc la vie de cette carte est d'environ quatre années. On a pris cet exemple d'une carte avec des circuits déjà anciens, pour montrer que la formule a un certain degré d'exactitude. Ces plaques de 2 K octets ont été lancées aux environs de 74 et aujourd'hui on les trouve difficilement dans les catalogues des fabricants.

La conclusion que l'on dégage de tout cela est que l'évolution de la technologie est telle, qu'actuellement, un projet doit être mené d'une manière astucieuse pour qu'il puisse prendre en compte les nouveaux composants, mais d'autre part il ne peut pas trop

s'étendre dans le temps pour que son industrialisation soit rentable. il ne faut pas oublier qu'après la phase de projet, il y a la phase de préparation pour l'industrialisation, avant que le produit puisse être placé sur le marché.

CHAPITRE VII

LE PROCESSEUR DISQUE

1. INTRODUCTION
2. PRESENTATION DU PROCESSEUR DISQUE DU PBD-MAGE
3. CARACTERISTIQUES GENERALES DU CONTROLEUR DISQUE
   - 1- ADRESSAGE DU DISQUE
   - 2- VERIFICATION D'ADRESSE SECTEUR
   - 3- TRANSFERT DES DONNES
   - 4- VERIFICATION D'ERREUR DE DONNEE
   - 5- PROTECTION SECTEUR
   - 6- VERIFICATION DES DONNEES
   - 7- IDENTIFICATION DES SECTEURS DEFECTUEUX
   - 8- LES REGISTRES DU CONTROLEUR
   - 9- RECHERCHE DE MOTS CLES A LA VOLEE
   - 10- CONTROLES DU SERVO DE PISTE ET DE LA VALIDATION DES DONNES
   - 11- ENTRELACEMENT DES SECTEURS
4. DESCRIPTION FONCTIONNELLE DU CONTROLEUR DISQUE
   - 1- INTERFACE AVEC L'EXTERIEUR
   - 2- DECODEUR D'ADRESSE
   - 3- REGISTRE UNITE DISQUE
   - 4- REGISTRE GENERAL
   - 5- REGISTRE DE VALIDATIONS

VII PROCESSEUR DISQUE

Le chapitre II a montré l'évolution des Processeurs Base de Données. Cette évolution était dirigée vers la construction de machines non-numériques qui nécessitaient la conception d'unités de contrôle de disques et même d'unités de disque d'architecture complètement différentes. Cependant, cette évolution n'étant pas encore viable économiquement, la construction des unités de contrôle de disques reste la même dans ses lignes générales avec seulement l'addition de quelques fonctions.

Ce chapitre présente la définition, les caractéristiques et la conception d'une unité contoleur de disques adaptée aux besoins du PBD-MAGE. Cette unité est présentée plus en détails, par rapport aux autres blocs du système (voir Fig. VI.3), puisque sa structure n'est pas traditionnelle. Le prochain chapitre présentera son implémentation.

VII. 1 INTRODUCTION

Avec l'apparition des microprocesseurs MOS, est venue l'idée de les utiliser dans les contrôleurs disques. L'emploi de leurs programmes permet d'avoir une très grande souplesse dans la modification ou l'addition des fonctions. Pour changer une fonction il suffit de reprogrammer et éventuellement d'ajouter plus de mémoire (ROM)

Anciennement, les contrôleurs construits avec des circuits standard, souffraient d'un sérieux handicap dû à la presque impossibilité de changer leurs fonctions. Le changement des circuits d'une fonction affectait les autres fonctions. C'est une des principales

raisons qui rend l'utilisation d'une technique programmée attirante. Cependant, jusqu'à présent les microprocesseurs souffrent d'un sérieux handicap : il sont trop lents pour ce type d'application.L'utilisation de processeurs en tranches (bit slice) [T.AMD.78] [T.INT.75] permet de résoudre œ problème. Ces circuits, à mi-chemin entre les circuits standars LSI et les microprocesseurs monolithiques, peuvent être microprogrammés et ont des vitesses de l'ordre de 5 à 10 fois celles de microprocesseurs MOS. Le fait qu'ils soient microprogrammés permet de changer facilemement leurs programmes stockés dans les mémoires ROM ou PROM et donc de modifier les fonctions qu'ils exécutent.

Avec des microinstructions de l'ordre de 40 bits, les processeurs en tranches assurent un grand degré de parallèlisme dans l'exécution des ordres. Comme chaque champ de la microinstruction est concerné par le contrôle d'une seule fonction, il existe donc une indépendance entre les divers blocs fonctionnels. Cela permet le développement séparé de chaque fonction et diminue les problèmes de modifications. Deux types de modifications principales peuvent être faites avec un système basé sur des circuits en tranches :
a) changement des programmes en mémoire morte (ROM), pour obtenir d'autres fonctions Dans ce cas il faut carrément remplacer le circuit de mémoire ROM par un autre contenant le nouveau programme.
b) addition de circuits : s'il est seulement nécessaire d'ajouter une fonction avec des circuits pour augmenter les performances, il suffit seulement d'ajouter quelques bits au mot de microinstruction pour contrôler les nouveaux circuits. Cela ne pose pas de grands problèmes et la structure du contrôleur ne sera pas remise en cause par ces modifications.

VII.2 PRESENTATION DU PROCESSEUR DISQUE DU PBD-MAGE

Le processeur disque (PGDISQ) exécute plusieurs fonctions. Hors le contrôle du disque lui même, le processeur réalise l'optimisation du déplacement de bras, la conversion d'adresse, la gestion de requêtes , l'initialisation, etc... Les unités de disque utiliséesde 10 MHz [T.CDC.77] obligent à avoir un contrôleur adapté à ces vitesses. Comme il a déjà été dit, un microprocesseur ne peut pas prendre en compte ces vitesses, le processeur disque nécessite donc plusieurs niveaux de traitement intermédiaires pour amener les données à une vitesse compatible avec celle des microprocesseurs.

Pour résoudre ce problème des contraintes de vitesse, le processeur disque a dû utiliser une architecture basée sur une structure matérielle hiérarchique. Le niveau le plus haut est mené par le microprocesseur (P2), le niveau intermédiaire par des processeurs en tranches et le niveau inférieur par des circuits TTL [T. TEX.77] . Les caractéristiques de ces trois niveaux sont (Fig VII.1 :

- le niveau haut : vitesse lente, language puissant.
- le niveau intermédiaire : vitesse moyenne, language de microprogrammation
- le niveau bas : grande vitesse, ordres directs.

Les processeurs actuels en tranches peuvent supporter des fréquences de 10 MHz comme limite supérieure. Cela ne leur permet donc que de réaliser une partie de la fonction de contrôle. Il n'est donc pas possible d'exécuter toutes les fonctions du contrôleur, car les tranches tournent àla même vitesse que les unités de disque et il est nécessaire d'exécuter plus d'une microinstruction entre deux événements. La solution est donc d'avoir des circuits TTL Schottky

pour réaliser les fonctions les plus rapides.

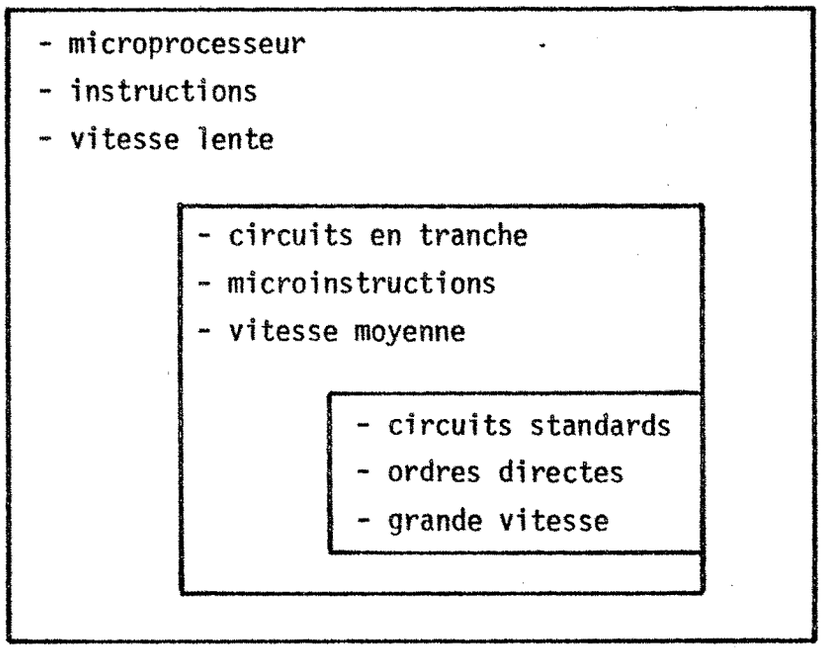


Fig. VII.1 - Structure hiérarchique du processeur disque

Les circuits standards exécuteront la rérialisation/désérialisation sur 8 bits des données provenant du disque et le calcul du CRC (Cyclic Redundancy Check). Cela abaisse la fréquence des événements à 1,2 MHz. A cette fréquence, les tranches peuvent traiter plus facilement des données.

Les processeurs en tranches auront comme principales fonctions le contrôle :

- des circuits de sérialisation/désérialisation des données
- du transferts des données de la mémoire vers le sérialisateur/desirial et vice-versa
- de l'incrémention du registre d'adresse des données du test du CRC
- de la détection de la synchronisation
- de la vérification des données par comparaison, etc ...

Avec toutes ces commades et procédures, les unités en tranches, réalisent quatre fonctions primitives : la lecture des données, l'écriture des données, la vérification des données et la recherche de profils (mots clés). Les tranches sont initialisées par le microprocesseur MOS grâce au transfert de paramètres et à des signaux d'activation.

Le microprogramme des tranches est le centre de contrôle du transfert des données. Les bits des microinstructions, activent directement une grande partie des circuits logiques. Les fonctions les plus complexes comme des comparaisons de données et comptage avec tests sont exécutées par les unités arithméthiques des processeurs en tranches.

Tous ces mécanismes sont contrôlés à un niveau supérieur par le microprocesseur MOS. C'est lui qui active les processeurs en tranches et aussi qui exécute les fonctions nécessitant du traitement telles que la conversion d'adress, le positionnement des t^etes, la gestion des requêtes , l'optimisation du déplacement du bras et d'autres. C'est le microprocesseur associé à des circuits standards qui exécutera la cinquième fonction primitive : la recherche de cylindre (seek) (Fig VII.2) :

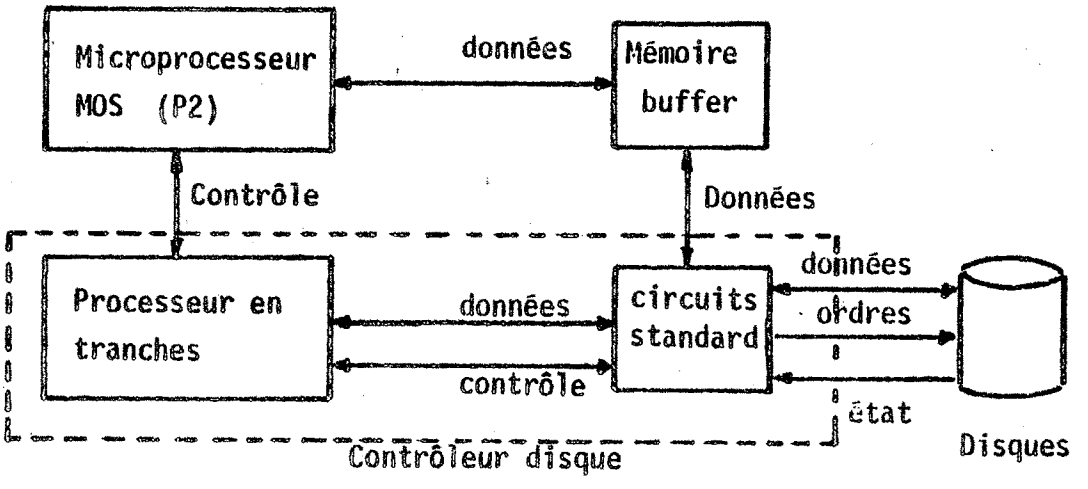


Fig VII.2 - Architecture du processeur disque

VII.3 CARACTERISTIQUES GENERALES DU CONTROLEUR DISQUE

La fonction globale "processeur disque" est réalisée par le processeur P2 (le microprocesseur) et l'unité contrôleur disque (processeurs en tranches et circuits standards). Il est difficile de les désassocier complètement pour présenter juste le contrôleur disque. C'est pour cela que les caractéristiques présentées ici concernent plutôt l'unité de contrôle, cependant quelques fonctions ne pouvant pas être séparées sont présentées ensemble.

Les principales caractéristiques du contrôleur disque sont [T.NAV1.78] :

- Le compteur de mots est tel qu'il peut transférer jusqu'à 64 K octets.
- Il est possible de transférer plusieurs secteurs en une seule opération.
- La séquence de vérification du descripteur secteur garantit que l'adressage du secteur est correcte.
- La détection des erreurs de lecture ou d'écriture est réalisée à l'aide d'un caractère de CRC.
- La mémoire tampon a une taille d'au minimum deux secteurs.
- La séquence de vérification des données est réalisée à travers la comparaison du contenu de la mémoire tampon avec les données du disque.
- Les secteurs du disque peuvent être protégés individuellement.
- Les mauvais secteurs peuvent être marqués même après le formatage du disque.
- Le contrôle du bon déroulement d'un transfert se fait en lisant les registres de l'état du disque et de l'état du contrôleur.
- Le signal ajustement du déplacement du bras, ainsi qu'ajustement de lecture peuvent être utilisés par programme pour effectuer des procédures de récupération des données.

- La reconnaissance de mots-clés peut être réalisée directement sur le disque.
- La possibilité de modifier un secteur par la lecture et réécriture au tour suivant.

VII.3.1 Adressage du disque

Chaque secteur est identifié par une adresse constituée par une adresse cylindre, une adresse tête et une adresse secteur. L'adresse cylindre de 10 bits de longueur spécifie le cylindre où réside le secteur recherché. Pour des unités disque de 300 M octets et de 80 M octets le nombre de cylindres est de 823, pour des unités de 150 M ou 40 M le nombre de cylindres est de 411.

L'adresse tête est de 5 bits et spécifie une des surfaces. L'adresse cylindre et l'adresse tête déterminent une piste du disque. Les unités de 40 M et 80 M ont 5 têtes donc cinq surfaces de données, les unités de 150 M et de 300 M ont 19 têtes.

Finallement l'adresse secteur de 8 bits indique le secteur choisi. Les possibilités d'adressage de ces 8 bits ne sont pas toujours tous utilisés, en fonction de la sectorisation choisie de l'unité disque.

VII.3.2 Vérification d'adresse secteur

Le contrôleur vérifie automatiquement, avant toute opération de lecture ou d'écriture, si le secteur sous les têtes est le secteur choisi. Cette vérification est réalisée par la comparaison de l'adresse secteur désirée avec l'adresse écrite sur le descripteur du secteur. Cette adresse secteur est écrite sur le disque lors du formatage du secteur.

VII.3.3 Transfert des données

Les données du disque sont transférées directement entre le disque et la mémoire buffer du microprocesseur (P2) et vice-versa sans étape intermédiaire. Si les données sont écites seulement dans une partie du secteur, le contrôleur se charge d'écrire des uns (1) dans le reste du secteur.

VII.3.4 Vérification d'erreur de donnée

Les données lues à partir d'un secteur sont vérifiées à l'aide d'un caractère de CRC (Cyclic Redundancy Check), qui est toujours enregistré après les données de chaque secteur. Ce caractère est automatiquement généré par le contrôleur lors d'une opération d'écriture et est enregistré sur le disque. Pendant l'opération de lecture deux octets de vérification sont générés dans le contrôleur pour être comparés avec le caractère de CRC écrit à la fin du secteur des données. S'il y a différence, les données de ce secteur ont été par exemple mal lues et le bit d'erreur correspondant est donc positionné.

VII.3.5 Protection secteur

Tout secteur peut être protégé individuellement contre l'écriture lors du formatage. Quand le bit de protection du secteur est activé, si une tentative d'écriture sur un secteur protégé a lieu, l'opération est stoppée et un bit d'erreur de protection est positionné par le contrôleur. Pour changer l'état de la protection d'un secteur, il faut reformater ce secteur. Cette action détruit normalement les données du secteur.

VII.3.6 Vérification de données

Une séquence de vérification des données peut être commencée pendant une opération de lecture ou d'écriture. Si cette opération est choisie lors d'une opération d'écriture/vérification, les données qui seront écrites sur le disque, seront ensuite lues et comparées mot par mot avec les données en mémoire. Si une opération de vérification simple a été sélectionnée, les données en mémoire sont comparées mot par mot avec celles du disque. Dans les deux cas, si la comparaison n'est pas correcte un bit d'erreur de vérification est activé.

VII.3.7 Identification des secteurs défectueux

Les secteurs défectueux d'un disque peuvent être identifiés et signalés lors du formatage de ce disque. Le bit de secteur défectueux est localisé dans le mot de "Protection d'écriture/ mauvais secteur" au début de chaque descripteur secteur. Si une opération de lecture ou d'écriture est réalisée sur un secteur défectueux, l'opération est stoppée et une erreur de mauvais secteur signalée.

VII.3.8 Les registres du contrôleur

Les registres du contrôleur peuvent être lus pour vérifier leurs états sans restriction. Il n'y a pas de contraintes temporelles sur leur accès. Les registres suivants peuvent être lus : état du disque, état du contrôleur, adresse secteur et, évidemment, la mémoire tampon (buffer)

VII.3.9 Recherche de mots-clés à la volée

Le contrôleur peut effectuer automatiquement la recherche de mots-clés (profils), d'une longueur maximum de 8 octets, sur plusieurs secteurs d'une même piste, et même sur plusieurs pistes, sous le contrôle du microprocesseur. Cette technique est employée spécialement pour accélérer la recherche du début de certains fichiers. Dans le cas du PBD-MAGE cette technique permet : d'accélérer et améliorer l'accès au dictionnaire ; accélérer l'accès aux données, une réalisation d'entité étant identifiée dans un secteur (ou dans un groupe de secteurs) par son "nom interne" ; accélérer et améliorer les accès associatifs aux données qui se font sans cela par tables d'index et "hash-coding". Au niveau du prototype la longueur des mots-clés est fixée à 4 bits. Cette recherche peut être continue à l'intérieur d'un secteur ou alors suivant un déplacement prédéterminé à partir du premier mot des données.

VII.3.10 Contrôle du Servomécanisme de piste et de la validation des données

Les contrôles de l'ajustement du servo de piste et de l'ajustement de la "validation des données" peuvent être commandés par programme depuis l'extérieur. En activant les bits convenables, le programmeur peut déplacer les têtes d'une "valeur" fixe vers l'intérieur ou vers l'extérieur du centre de la piste. Cela permet de corriger des petites erreurs de positionnement de bras. De la même manière, le programmeur peut activer la validation des données plus tôt ou plus tard. Ces deux contrôles sont très utiles à l'occasion d'une récupération de données.

VII.3.11 Entrelacement des secteurs

Cette technique est intéressante lors d'une application où plusieurs secteurs doivent être transférés successivement. Pour les applications où seule un secteur est transféré, l'ordre des secteurs sur la piste peut être séquentiel comme 0, 1, 2 ...

Dans les cas de transfert multisecteurs, l'ordinateur peut être trop lent pour réussir à prendre en compte les secteurs l'un après l'autre. C'est pour résoudre ce problème qu'une technique d'entrelencement a été utilisée dans laquelle les secteurs se présentent sur une piste dans l'ordre 0, 16, 32, 48, 1, 17, 33, 49, 2, 18, 34, 50. Cette organisation permet à l'ordinateur de prendre du souffle pendant le temps de défilement de trois secteurs.

Pour utiliser la technique de la recherche à la volée dans le PBD-MAGE, a été employé un entrelacement d'un secteur, pour permettre au processeur P2 de vérifier l'état de la recherche (Fig. VII.3)

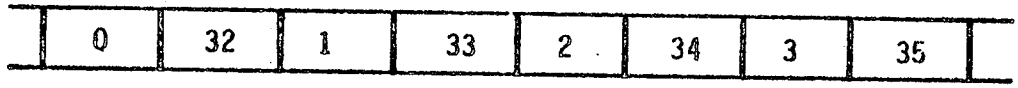


Fig VII.3 - Entrelacement de secteur de longueur 1 pour une sectorisation de 64 (utilisé dans le PBD-MAGE)

VII.4 DESCRIPTION FONCTIONNELLE DU CONTROLEUR DISQUE

Le processeur disque comme il a déjà été dit, est composé par l'ensemble de l'unité contrôleur disque, du processeur P2 et de la mémoire buffer (Fig VII.4). La partie du microprocesseur, utilisant des plaques standards, ne nous intéresse pas ici, ni ses fonctions qui sont implantées par programme (chp VII.5)

L'unité contrôleur disque est constitué par une grande plaque où sont les processeurs en tranche et les circuits TTL. Cette section abordera le contrôleur en décrivant ses principaux blocs fonctionnels.

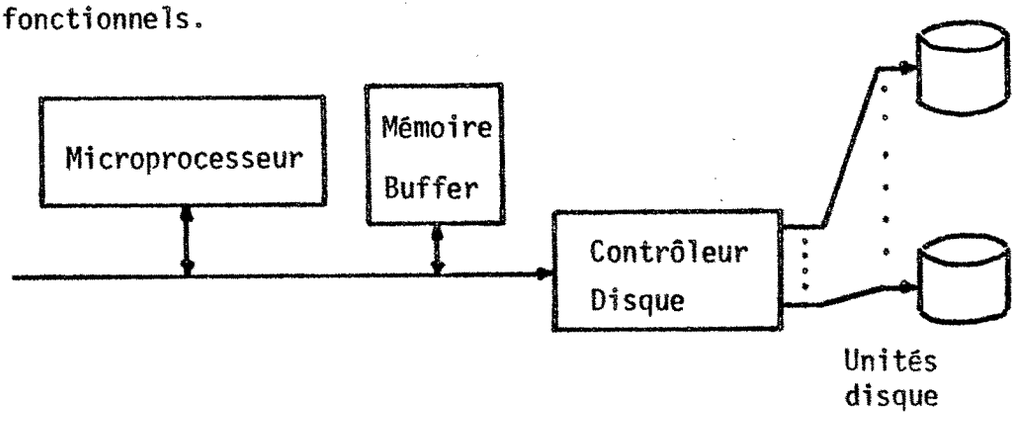


Fig VII.4 - L'ensemble du processeur disque

La plaque contrôleur disque est reliée d'un côté au bus du microprocesseur qui est son moyen de communication avec l'extérieur, et de l'autre côté aux unités disque à travers un ensemble de deux câbles définis par les normes SMD (chap V.10 et annexe 1). Un des câbles, le câble A, véhicule les commandes aux unités disque pendant que l'autre, le câble 3, véhicule les données.
La figure VII.5 montre une vision globale des principaux blocs, composants l'architecture matérielle du contrôleur.

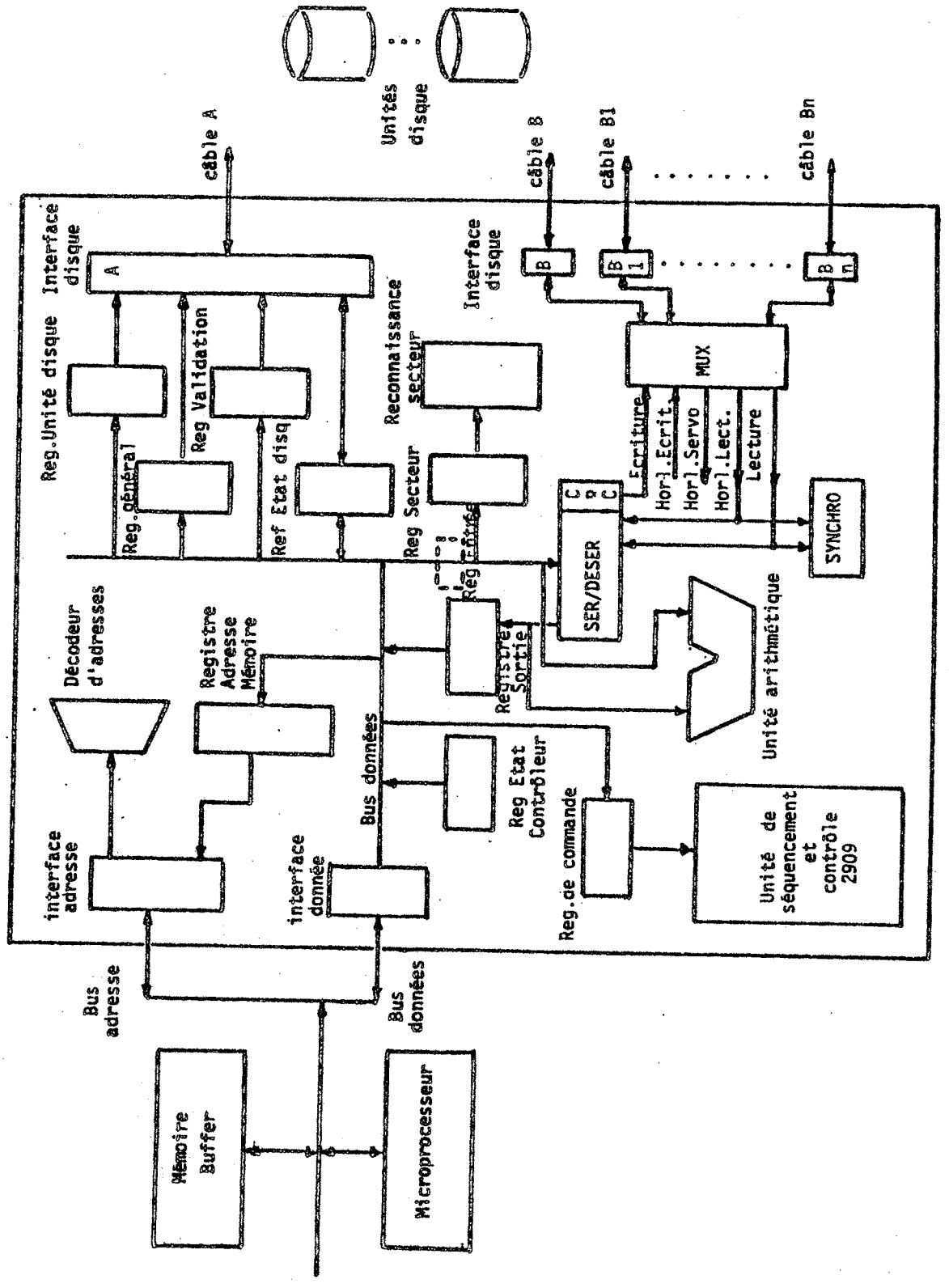


Fig.VII.5 - Principales fonctions matérielles du contrôleur

VII.4.1 Interface avec l'extérieur

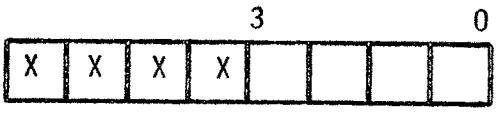
Ces circuits d'interface permettent la communication entre la plaque contrôleur disque et le processeur P2 et la mémoire buffer. Le bus de données, le bus adresse, ainsi que les signaux de contrôle tels que l'horloge 2, le signal d'interruption, le signal de reset, le signal de "Halt", le signal d'écriture/lecture, etc.. passe par des circuits émetteurs/récepteurs.

VII.4.2 Décodeur d'adresse

Le décodeur d'adresse permet d'obtenir, de décoder, les 16 adresses qui concernent le contrôleur. Décodées elles sont de deux types : a) adresses de registres ; b) adresses qui auront un caractère de signal d'activation d'une fonction. Ces adresses font partie du champ d'adressage du microprocesseur et peuvent ^être disloquées gr^ace à des interrupteurs sur la plaque, permettant de positionner le contrôleur à la place la plus convenable pour le microprocesseur. Le décodage est fait sur les 4 bits les moins significatifs du bus adresse du processeur P2.

VII.4.3 Registre Unité disque

Ce registre de 4 bits permet de choisir une des 16 unités disque avec laquelle le contrôleur dialogue. L'information de ce registre est validée pour une unité disque à travers le signal de "validation d'unité disque". Ce registre ainsi que les autres du contrôleur, est chargé comme une position de la mémoire du microprocesseur. Celui-ci utilise les quatre bits les moins significatifs du bus de données du microprocesseur pour son chargement.



VII.4.4 Registre Général

Ce registre général de 10 bits peut avoir trois fonctions différentes suivant le type de signal de validation utilisé. Le chargement de ce registre se fait en deux étapes par le microprocesseur, une de 8 bits et l'autre des 2 bits restants. La table VII.1 ci-dessous donne la liste des signaux par fonction.

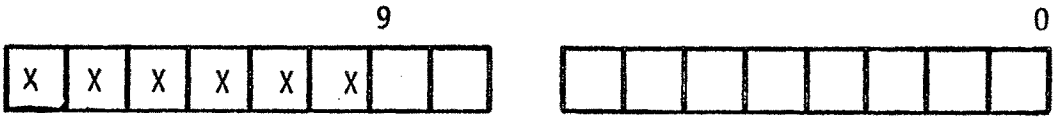
fonction / bit	adresse cylindre	adresse tête	fonction disque
	2^0	2^0	porte d'écriture
1	2^1	2^1	porte de lecture
2	2^2	2^2	déplacement du servo en avant
3	2^3	2^3	déplacement du servo en arrière
4	2^4	2^4	remise à zéro d'une faute
5	2^5	X	recherche de marque d'adresse
6	2^6	X	retour à zéro
7	2^7	X	validation des données anticipées
8	2^8	X	validation des données postérieures
9	2^9	X	X

Table VII.1 - Fonctions des bits du registre général

Si le signal de validation pour le disque est du type "cylindre" la valeur du registre général représente une adresse cylindre.

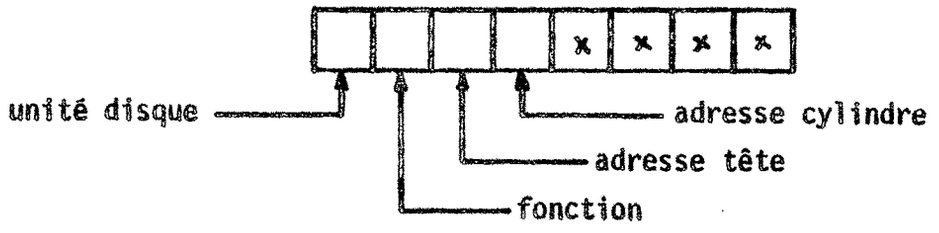
Si la validation est "tête" les bits du registre représentent l'adresse tête choisie. Et si la validation est du type "fonction", le registre fournira des ordres vers le disque. D'une manière succinte les ordres, fonctions, peuvent être décrites comme :

- Porte d'écriture - autorise une écriture sur le disque.
- Porte de lecture - autorise une lecture sur le disque.
- Ajustement du servo en arrière ou avant - oblige les têtes à se déplacer d'un écart de part et d'autre de leur position normale sur un cylindre. Ces signaux sont utilisés pour la récupération des données.
- Remise à zéro d'une faute - ce signal permet de faire la restauration d'une unité disque quand il y a une faute.
- Recherche de marque d'adresse - ce bit autorise la recherche de la marque d'adresse qui indique le début d'un secteur. Ce signal est utilisé pour des pistes à secteurs de longueur variable.
- Retour à zéro - ce bit force le servo à remettre les têtes sur le cylindre zéro.
- Validation des des données anticipées ou postérieures. Ces deux signaux permettent ainsi que l'ajustement du servo, la récupération des données. Ils forcent la lecture des bits de données par le disque, d'un temps avant ou après leur position normale.



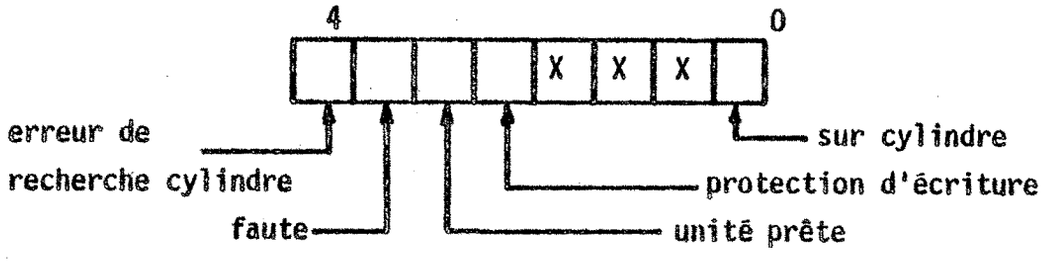
VII.4.5 Registre de Validations

Ce registre valide le registre d'unité disque et valide aussi le type d'information stockée dans le registre général. Il a 4 bits et est chargé à partir des 4 bits les plus significatifs du bus données.



VII.4.6 Registre du disque

Ce registre contrairement aux précédents est lu par le microprocesseur. Il l'informe sur les erreurs qui surviennent au niveau de l'unité disque, ainsi que de son état. Il a cinq bits et ces bits proviennent du bus de données comme ci-dessous.



- Erreur de recherche cylindre- C'est une erreur due à une adresse cylindre trop grande envoyée à l'unité disque, ou au fait que l'unité n'a pas réussi à trouver le cylindre dans le délai réglementaire.
- Faute - Diverses opérations au niveau du disque peuvent déclencher une faute, par exemple : manque d'une des alimentations ; activation d'une écriture simultanément à une lecture.
- Unité prête - Ce signal indique si l'unité est branchée

- Protection d'écriture - Indique que cette unité a l'interrupteur de protection écriture activée, donc que toute écriture est interdite.

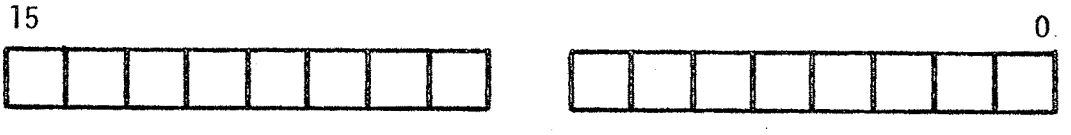
- Sur cylindre - Informe que les têtes du disque sont positionnées sur un cylindre et qu'une recherche de cylindre vient d'être finie correctement. Indique aussi que l'unité est prête à recevoir un nouvel ordre de recherche de cylindre.

VII.4.7 Registre adresse secteur

L'adresse secteur chargée dans ce registre, par les microprocesseurs MOS permet au contrôleur de trouver, à travers les circuits de l'unité de reconnaissance secteur, le secteur voulu pour une lecture ou une écriture sur le disque. Ce registre a 8 bits, mais ces bits seront utilisés en fonction de la sectorisation choisie pour le disque.

VII.4.8 Registre adresse mémoire

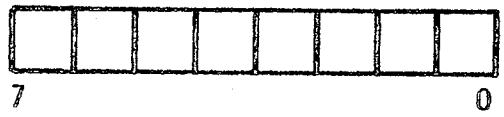
Ce registre permet au microprocesseur d'indiquer au contrôleur l'adresse du premier mot de la mémoire buffer. Le microprocesseur charge ce registre en deux fois 8 bits, autorisant le contrôleur à lire (écire) les données de (dans) la mémoire buffer. Le contrôleur incrémente ce registre automatiquement au fur et à mesure du transfert des données de la mémoire vers le disque ou vice-versa.



Le fait que ce registre ait 16 bits, permet théoriquement d'adresser un buffer jusqu'à 64 K octets, bien qu'il soit pratiquement plus petit.

VII.4.9 Registre d'entrée

Ce registre permet le transfert des données de la mémoire buffer (tampon) vers le disque. Ce registre est virtuel car il n'existe pas en réalité et représente le chemin de données d'entrée dans le contrôleur. Il est utilisé aussi lors de la vérification des données, pour amener les données de la mémoire et les comparer à celles enregistrées sur le disque.

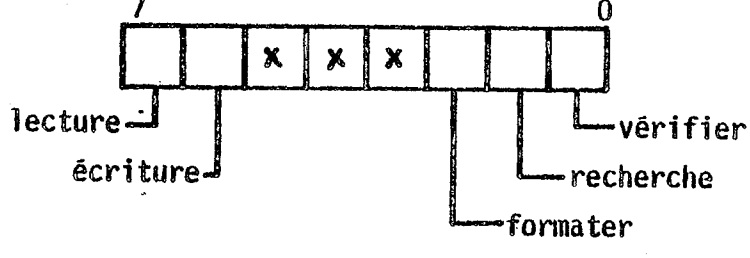


VII.4.10 Registre de sortie

C'est à travers ce registre que les données enregistrées sur les disques peuvent être transférés vers la mémoire buffer. Il a une longueur d'un octet. Le mécanisme utilisé pour la lecture, comme pour l'écriture des données est l'accès direct à la mémoire (DMA).

VII.4.11 Registre de commande du contrôleur

Ce registre, d'une largeur de 5 bits, indique au contrôleur le type d'opération qu'il doit exécuter et quel est le microprogramme de l'unité de séquencement qui doit être déclenché.



Vérifier - Indique une opération de vérification des données enregistrées sur le disque. Si cet ordre est associé avec une écriture, l'opération sera une écriture suivie d'une vérification des données écrites. Si l'ordre est associé à une lecture, l'opération déclenchée sera une simple vérification des données enregistrées. Une vérification est réalisée par la comparaison du contenu d'un secteur du disque avec celui de la mémoire buffer.

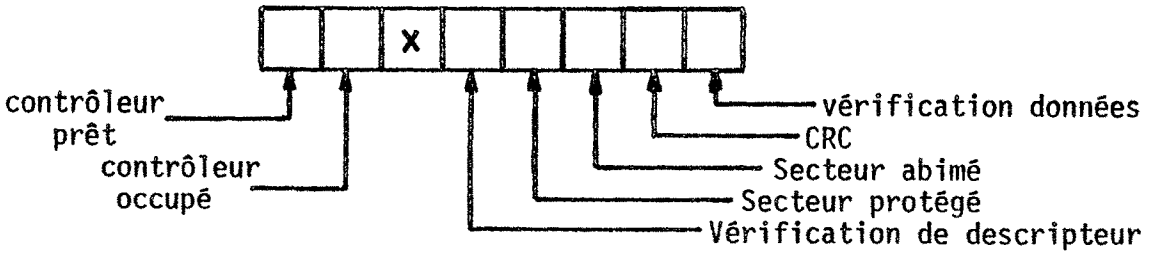
Recherche - Cette opération déclenche la recherche d'un profil (mot clé) sur une piste. Le profil a été fourni au contrôleur. Lors de la rencontre du profil recherché, les données qui lui succèdent peuvent être lues. Cette recherche est faite directement sur les données du disque et est connue sous le nom de "recherche à la volée".

Formater - Ce bit indique au contrôleur qu'il doit réaliser une opération de formatage d'un secteur.

Lire ou écrire - Le nom même indique l'opération déclenchée par ces bits.

VII.4.12 Registre d'état du contrôleur

Ce registre de 7 bits, indique au microprocesseur quel est l'état du contrôleur et les erreurs s'il y a lieu.



Contrôleur prêt - Indique que le contrôleur marche, qu'il est alimenté par exemple.

Contrôleur occupé - Indique sue le contrôleur est entrain de réaliser une opération.

Vérification de descripteur - Indique que le contrôleur a trouvé le bon secteur. Deux cas peuvent arriver : si le CRC est mauvais aussi, le contrôleur a rencontré des problèmes à la lecture du descripteur du secteur. Si le CRC n'est pas activé, le contrôleur simplement n'a pas trouvé le bon secteur.

Secteur protégé - Le secteur que le contrôleur a essayé d'écrire était protégé. L'opération est arrêtée automatiquement.

Secteur abîmé - Le secteur est abîmé et il est interdi d'écrire ou de lire. Une opération sur ce secteur est annulée et le bit d'erreur est activé.

CRC - Cyclic redundancy check. Ce bit indique que le test d' erreur en lecture a été positif et que donc les données lues ne sont pas correctes.

Vérification de données - indique que l'opération de "vérification simple" ou "d'écriture avec vérification" a été juste. En cas de différence entre les mots de la mémoire et ceux enregistrés sur piste, le bit sera activé.

VII.4.13 Interface avec les unités disque

Les interface sont réglés par les normes SMD (chap IV.10 et annexe 1) de communication avec les unités disque, ils utilisent pour cela des circuits émetteurs et récepteurs différentiels de ligne. Il y a deux types d'interfaces : une destinée aux signaux de contrôle : câble A, et l'autre aux données et aux horlogeries associées : câble B.

Le contrôleur utilise la technique "daisy chain" (chap IV.11) pour se connecter à plusieurs unités disque. Par conséquent il y a un interface pour le câble A qui parcours en série toutes les unités disque, et il y a plusieurs interfaces, une pour chaque câble B. Chaque câble B n'étant connecté qu'à une seule unité disque.

VII.4.14 Multiplexeur des interfaces B

Une unité de Multiplexage associée aux interfaces B, signaux de données et d'horloge, permet de choisir vers quelle unité disque les données doivent être écrites ou lues (envoyées ou reçues). Ce circuit de multiplexage aiguille les signaux provenant de l'unité de sérialisation/désérialisation vers l'interface B choisie par le contrôleur.

VII.4.15 Circuits à décalage et CRC

Le circuit à décalage permet de sérialiser ou de désérialiser les données de la mémoire buffer, octet par octet, vers ou provenant des disques, selon qu'ils sont écrits ou lus respectivement bit par bit du secteur. Cette sérialisation/désérialisation sur 8 bits amène la fréquence de transfert de 10 MHz (transfert d'un bit) à 1,2 MHz (transfert d'un octet), permettant l'action des processeurs en tranche.

Le CRC (Cyclic Redundancy Check), circuit de détection d'erreur est associé au circuit de décalage. Ce circuit permet d'introduire, à la fin de l'écriture d'un secteur sur le disque, un code de deux octets. Ce code sera ensuite utilisé à la lecture pour vérifier si

elle se fait correctement. Cela est réalisé grâce à l'activation de ce circuit (CRC) à la lecture d'un secteur. Après la lecture des données et du CRC enregistré le contenu du circuit doit être zéro s'il n'y a pas eu d'erreur.

VII.4.16 Unité arithmétique

Cette unité composée de circuits AMD 2901 T.AMD.78 permet au contrôleur de réaliser toutes les opérations qui concernent une unité d'arithmétique et logique (UAL). C'est le cas par exemple, du comptage et test du nombre d'octets transférés. Une autre opération réalisée est la reconnaissance de profils. Cette unité exécute la comparaison des données provenant de la mémoire buffer avec ceux enregistrés sur le disque.

VII.4.17 Unité de Séquencement et de Contrôle

Cette unité est composée de circuits séquenceurs 2909 T.AMD.78 avec la mémoire de microprogramme associée, qui est l'âme du contrôleur. C'est de cette unité que partent tous les ordres pour la commande du contrôleur de disque. Selon les paramètres reçus, cette unité déclenche la séquence de microprogramme appropriée pour exécuter une fonction telle que la lecture d'un secteur disque, par exemple. L'activation de l'unité de contrôle et de séquencement est faite par le microprocesseur.

VII.4.18 Unité de Synchronisation

Cette unité permet au contrôleur disque de retrouver le début des données valides dans un secteur. Elle est également utilisée aussi

pour retrouver le début du descripteur secteur. C'est à travers la lecture et reconnaissance d'un caractère spécial que l'unité synchronise le CD avec les données du disque.

VII.4.19 Unité de Reconnaissance Secteur

Cette unité permet au CD de déterminer sur quel secteur les têtes de l'unité disque sont positionnées. Un Registre Compteur Secteur reçoit des signaux du disque, ce qui lui permet d'avoir l'adresse secteur courante. Ce registre est comparé avec le mot stocké par le microprocesseur dans le Registre Adresse Secteur pour déterminer quand les têtes sont sur le secteur choisi par le contrôleur pour dialoguer.

VII .4.20 Mémoire des Buffers

La Mémoire des Buffers a deux fonctions. Hors le transfert de données entre le disque et le microprocesseur, elle est aussi le moyen de communication pour l'envoi d'une partie des paramètres vers le CD, spécialement vers l'UAL. La Mémoire des Buffers peut avoir plusieurs buffers, le minimum étant 2 buffers et le maximum fonction des performances d'accès disque nécessaire. La distribution des données et paramètres dans une mémoire buffer est présentée ci-dessous.

Adresse 0	- Etat Secteur	- 1 octet
" +1	- Adresse Cylindre Supérieur	- 1 "
" +2	- Adresse Cylindre Inférieur	- 1 "
" +3	- Adresse tête	- 1 "
" +4	- Adresse secteur	- 1 "

Adresse +5 - Longueur des données à transférer	- 1	octet
" +6 - Déplacement par rapport au début du secteur	- 1	"
" +7 - Pas utilisé	- 1	"
" +8 - Profil à rechercher	- 4	"
" +12 - Unité - Numéro d'unité disque	- 1	"
" +13 - Nombre de Secteurs pour l'enchainement	- 1	"
" +14 - Pas utilisé	- 1	"
" +15 - Pas utilisé	- 1	"
" +16 - DONNEES	- 256	"

Le mot d'Etat Secteur est utilisé par le microprocesseur lors d'une opération de "formatage" pour indiquer si le secteur est protégé contre l'écriture ou abimé. L'ensemble de l'Adresse Cylindre, Tête et Secteur permet au CD de vérifier, par comparaison avec le descripteur secteur, si les têtes sont sur le bon secteur. Les paramètres Déplacement et Longueur concernent "la recherche de profil", ils indiquent le déplacement où se trouve le profil à partir du début secteur et la longueur de l'enregistrement. Le Profil est le mot clé qui sera recherché par le contrôleur pour déterminer le début d'un enregistrement. Sa longueur maximum est de 8 octets, mais pour le prototype elle est fixe et de 4 octets.

Après les 16 octets de paramètres commence la zone de 256 octets de la Mémoire Buffer destinée aux données.

VII.5 DESCRIPTION OPERATIONNELLE DU CONTROLEUR

Le processeur disque, dans son ensemble, réalise plusieurs fonctions, le but de cette section est de décrire seulement les fonctions primitives qui concernent l'unité de contrôle. Chaque

fonction primitive sera décrite indépendamment des autres. Elles engagent l'action de l'ensemble de la plaque contrôleur, du microprocesseur et de la mémoire tampon (buffer). Dans les diagrammes en blocs, la répartition entre les actions réalisées par le microprocesseur et celles réalisées par la plaque de contrôle sera indiquée.

VII.5.1 OPERATION DE RECHERCHE CYLINDRE

Une opération de recherche cylindre est commencée dès qu'arrive une requête au microprocesseur qui sollicité l'accès d'un nouveau cylindre. La première opération du microprocesseur est alors d'envoyer l'adresse de l'unité, de la valider et de vérifier si cette unité est prête . Si l'unité n'est pas prête , le bit d'erreur correspondant le signalera au microprocesseur et l'opération est arrêtée . Le deuxième test est de vérifier si l'unité n'a pas une "erreur de recherche", si le bit est activé l'opération est également annulée. On réalise de la même manière le test sur le signal de "faute". Ces tests constituent une sécurité contre des anomalies du fonctionnement du disque et du contrôleur. Si tout est en ordre, le micro envoie l'adresse cylindre sur le registre général et valide cette adresse, initialisant ainsi l'opération de recherche d'un nouveau cylindre.

Le microprocesseur, dans une opération simplifiée sans du parallèlisme de traitement, restera à surveiller le résultat de son ordre de recherche cylindre. Pour cela, il bouclera sur les signaux "d'erreur de recherche" et de "faute" par mesure de sécurité et "sur cylindre". L'opération sera considérée comme achevée lorsque le signal "sur cylindre" sera activé. Tous les autres signaux arrêteront l'opération et les bits d'erreurs respectifs le signaleront au micro, qui prendra les mesures nécessaires.

Cette action est menée entièrement sous le contrôle du microprocesseur MOS sans que les processeurs en tranche de la plaque de contrôle n'interviennent (Fig. VII.6)

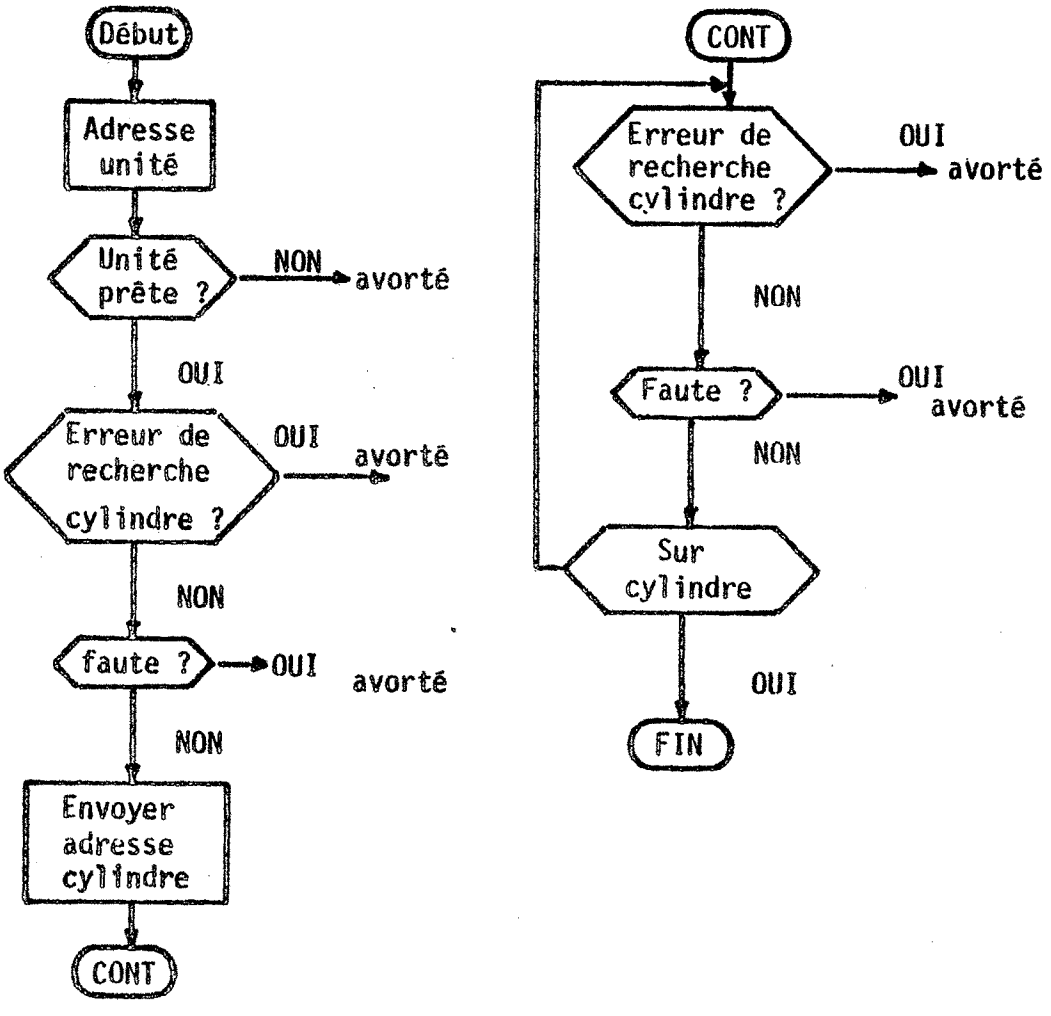


Fig VII.6 - Opération de recherche cylindre

Le microprocesseur MOS ne reste pas en attente active sur les signaux dans une opération avec optimisation du temps de celui-ci. Le microprocesseur lance l'opération et va réaliser d'autres traitements tels

que la conversion d'adresses, la préparation d'une nouvelle requête d'accès au disque, le traitement BD etc.. De temps à autre il vient vérifier l'état des signaux "sur cylindre", "faute" et "erreur de Recherche" pour déterminer la fin de l'opération de "Recherche Cylindre".

VII.5.2 OPERATION DE LECTURE

L'opération de lecture commence lorsque le microprocesseur reçoit une requête qui implique une lecture. S'il n'y a pas de file d'attente de requêtes, l'opération sera déclenchée immédiatement. On suppose que le micro a déjà positionné les têtes sur le bon cylindre et que le transfert des paramètres vers le contrôleur a déjà été réalisé.

Le micro vérifie si le bit de faute n'est pas positionné et si les têtes sont "sur cylindre" par mesure de sécurité. Si tout est correct l'opération de lecture est activée. Le micro commence par envoyer l'adresse tête et l'adresse secteur et ensuite vérifie le bit de faute pour déterminer s'il n'y a pas eu d'erreur au niveau de l'activation de la tête. La tête activée, le micro transfére le contrôle de l'opération à la plaque de contrôle.

Le contrôleur exécutera les opérations d'initialisation et se mettra en attente du signal indiquant que les têtes sont sur le bon secteur. La détermination de l'adresse secteur est réalisée à travers le comptage des signaux "secteur" provenant du disque. Un comparateur permet de détecter l'occurence du bon secteur. A cette occasion un signal de coïncidence secteur est généré [T.NAV.78]

Arrivé au bon secteur, le contrôleur lira premièrement les bits de synchronisation constituant le préambule. Cela permet au contrôleur de se synchroniser pour lire le descripteur secteur. Le mot d'état signalera si le secteur est abimé et un bit d'erreur correspondant sera éventuellement positionné. L'adresse du secteur obtenue est ensuite comparée avec celle demandée, s'il y a erreur, le bit d'erreur de vérification d'adresse est activé. L'erreur peut être due aussi à une erreur de lecture et dans ce cas le bit de CRC l'indiquera. Dans tous ces cas d'erreur, l'opération sera stoppée.

S'il n'y a pas eu d'erreur jusqu'à ce point, le contrôleur est alors autorisé à lire les données du secteur, pour cela il devra se resynchroniser. Le transfert des données se fera du secteur vers la mémoire buffer. A la fin de la lecture des données, le contrôleur ira vérifier le CRC, s'il y a erreur le bit d'erreur de CRC sera signalé au micro et indiquera que les données transférées ne sont pas correctes. A la fin de chaque transfert secteur le micro vérifie s'il ne doit pas transférer le prochain secteur, si oui, il recommence l'opération (Fig. VII.7). Le microprocesseur est arrêté, en "Halt", durant le temps que le contrôleur disque est actif.

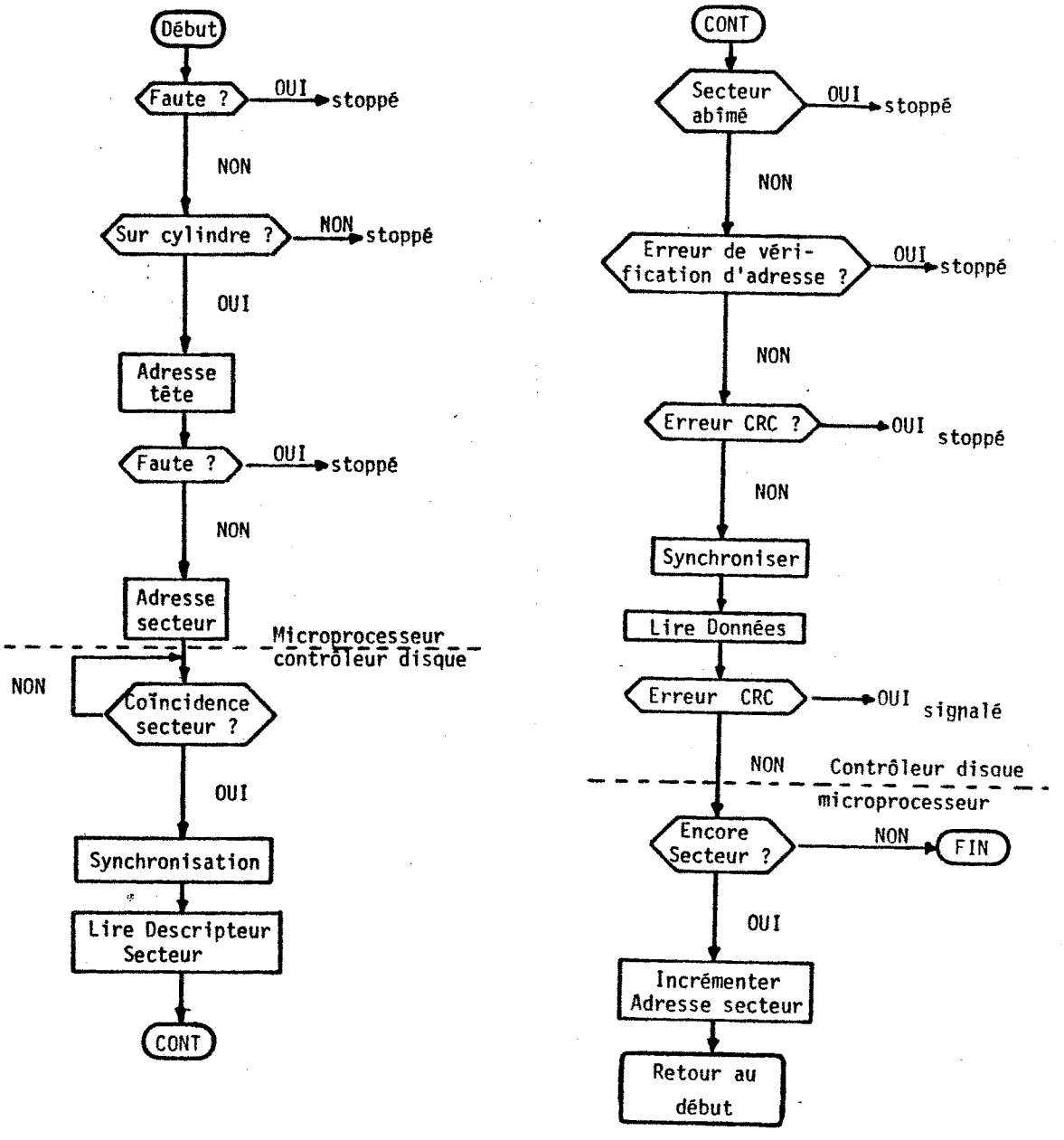


Fig.VII.7 - Opération de lecture

VII.5.3 Opération d'écriture

Cette opération est assez semblable à celle de lecture. Pour éviter les redites, on reprend le déroulement de l'opération juste après que le contrôleur a lu le descripteur secteur et fait les vérifications correspondantes. Avant cela la tête a été activée, l'adresse secteur lue et vérifiée, le CRC testé, l'état du secteur testé pour déterminer si le secteur est abîmé et, ce qui est très important, si le secteur n'est pas protégé contre l'écriture. Un autre test a été exécuté à priori, avant de lancer le contrôleur, lors des tests d'erreur le micro a vérifié si l'unité disque n'était pas protégée dans son ensemble contre l'écriture par l'interrupteur du panneau avant.

Donc, en reprenant l'opération à ce point, s'il n'y a pas eu d'erreurs, le contrôleur est alors autorisé à écrire les données. Pour cela, juste après le test du CRC du descripteur secteur, le contrôleur a donné l'ordre à la tête de passer du mode lecture au mode d'écriture, puis les bits de préambule, de synchronisation, ont été écrits. La prochaine étape est le transfert des octets de données vers le secteur et à la fin l'écriture du CRC. Les données sont transférées directement de la mémoire buffer pour être enregistrés est plus petit que la capacité du secteur, le contrôleur s'occupera d'écrire des "uns" pour compléter la zone des données (Fig. VII.8).

L'opération de transfert des données étant dinie, le contrôleur déterminera s'il doit transférer une nouveau secteur, si oui, il recommencera l'opération depuis le début mais un autre teste doit être fait avant de rendre le contrôle, pour déterminer si le bit de vérification du registre de commande n'est pas activé, si cela est le cas, le contrôleur attendra d'^etre de nouveau sur le secteur

pour lire les données écrites et les comparer avec celles de la mémoire buffer. C'est une opération conjointe connue sous le nom d'écriture/vérification. L'opération de vérification sera décrite plus loin dans cette section.

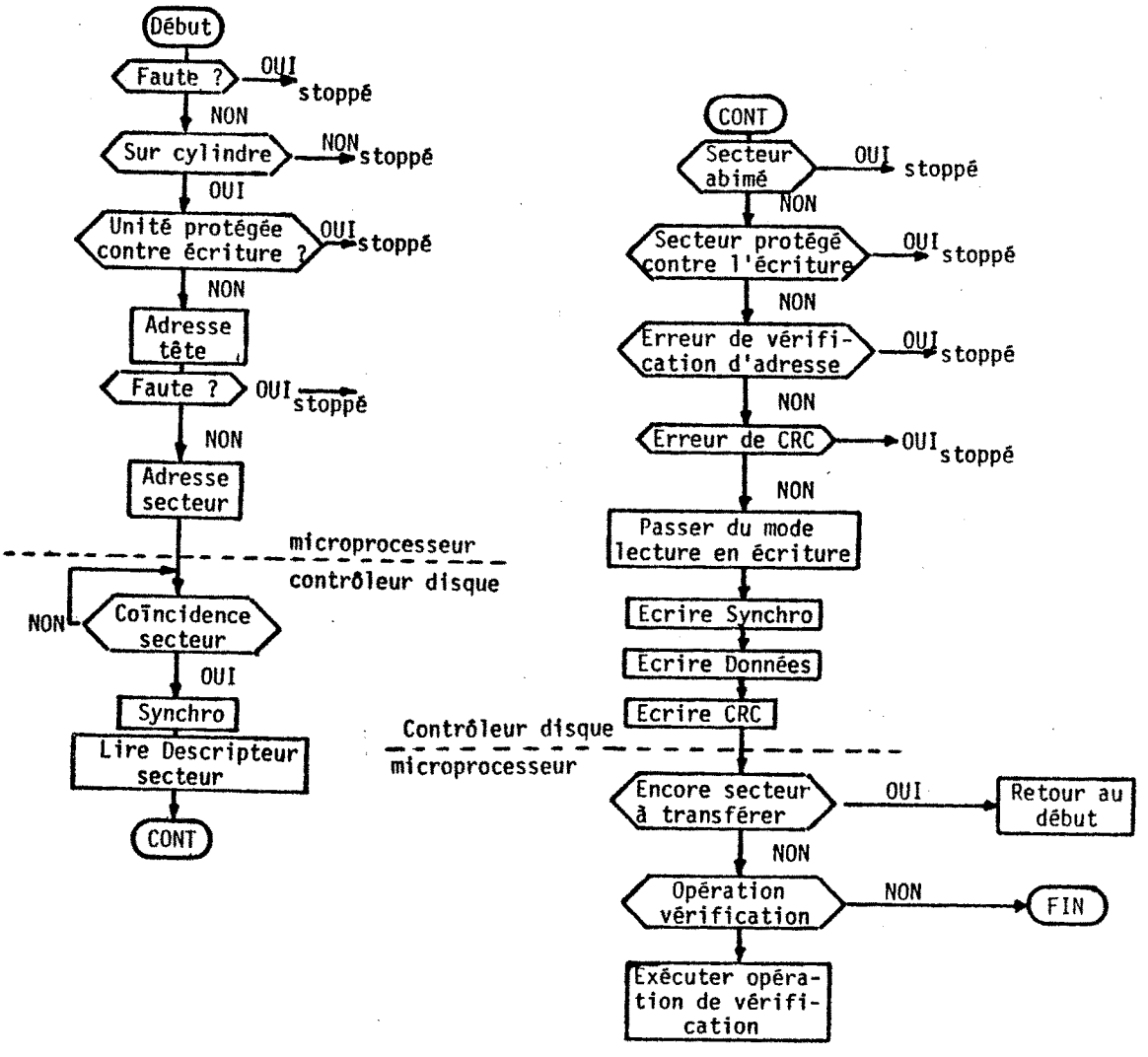


Fig.VII.8 - Opération d'écriture

VII.5.4 Opération de vérification

Cette opération est déclenchée par une opération de vérification simple ou par une opération d'écriture/vérification. La première a pour but de comparer le contenu d'un secteur avec celui de la mémoire. La deuxième consiste à comparer le contenue du buffer mémoire avec celui du secteur, elle est différente de la précédente par le fait que l'opération de vérification est déclenchée juste après une écriture et a pour but de déterminer si le contenu de la mémoire a été correctement enrgistré sur le secteur. L'opération de vérification est la même dans les deux cas, mais son but est différent.

L'opération de vérification est presque la même que l'opération de lecture. Le contrôleur teste les erreurs, positionne la tête, vérifie l'adresse secteur, le CRC du descripteur secteur, vérifie si le secteur n'est pas abîmé et se synchronise pour lire les données. Mais à ce moment les données lues sur le disque n'iront pas vers la mémoire buffer mais resteront au niveau du contrôleur et seront seulement comparées avec celles de la mémoire buffer qui ont été lues en parallèle. Les octets sont donc comparés, l'un après l'autre et s'il y a différence, l'opération est avortée et le bit d'erreur de vérification est positionné pour le signaler au microprocesseur. Cette opération est exécutée sur tous les octets du secteur. A la fin, le CRC est vérifié pour déterminer s'il n'y a pas eu d'erreur de lecture.

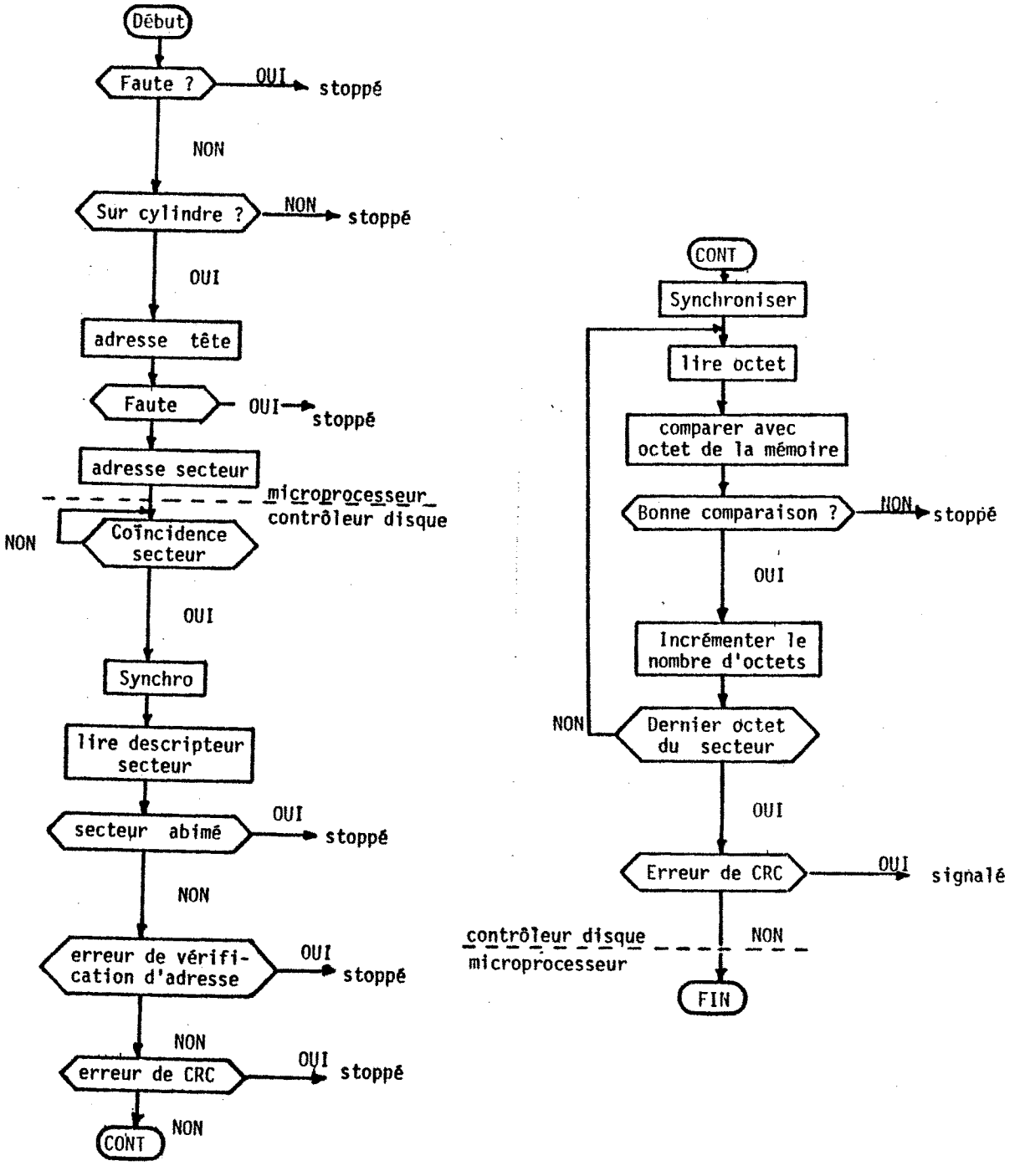


Fig.VII.9 - Opération de verification

VII.5.5 Opération de recherche de profil

Cette opération a quelques ressemblances avec celle de vérification. Son but est de trouver la localisation de données grâce à la comparaison d'un profil : un mot-clé, qui indique le début d'une zone de données recherchée. Cette action de comparaison est réalisée directement sur les données du disque sans passer par une mémoire intermédiaire. C'est pour cela que cette opération est aussi connue sous le nom de recherche à la volée. Cette recherche peut être menée sur plusieurs ou tous les secteurs d'une même piste.

L'opération commence par l'envoi de la part du microprocesseur au contrôleur des paramètres nécessaires, la longueur des données à transférer (nombre d'octets), le profil lui-même, le premier secteur à partir duquel doit être réalisée la comparaison, le nombre de secteurs sur lesquels doit être faite la recherche etc... Ces paramètres sont transférés à travers la mémoire tampon (buffer) vers le contrôleur. Il est aussi possible d'indiquer le "déplacement" à partir du début du secteur où doit être localisé le premier profil. Avec ce type d'ordre, le contrôleur fait les comparaisons uniquement sur les emplacement situés à d + n.l (d-déplacement, l-longueur à transférer) du début du premier secteur jusqu'à la fin du dernier secteur. Ce type de comparaison est par exemple utile pour retrouver les entrées dictionnaire et les réalisations d'entité du PBD-MAGE, entre autres.

Après cette phase d'initialisation, l'opération se déroule comme une lecture. Le contrôleur lit le ou les secteurs en comparant les octets aec le profil. A l'occasion du premier profil correct trouvé, la recherche est arrêtée, le microprocesseur prévenu que la recherche a été réussie et les données qui suivent le profil sont alors transférées vers la mémoire buffer. Uniquement l'enregistrement est transféré.

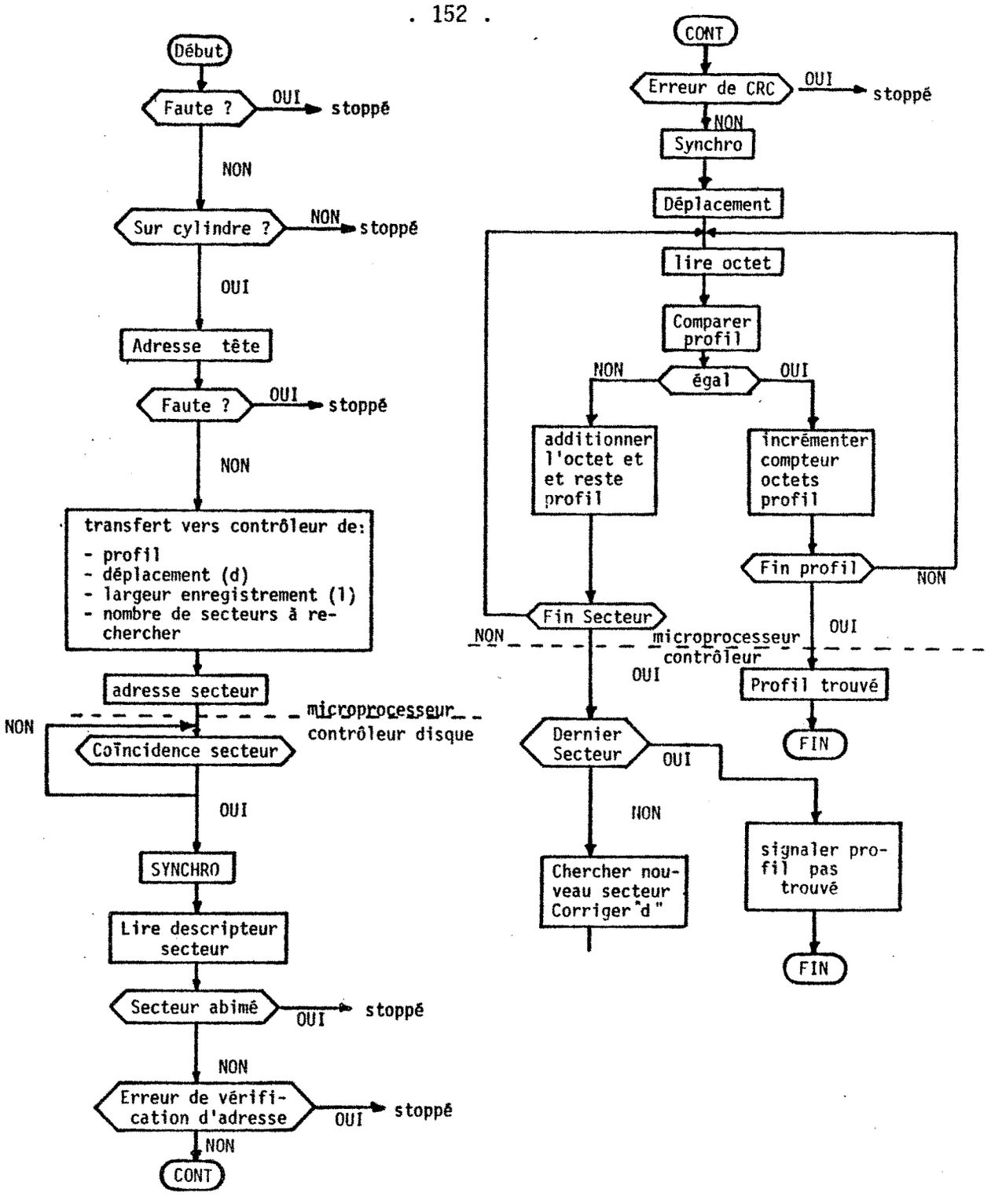


Fig.VII.10 - Opération de recherche de profil

Une opération d'écriture après la recherche d'un profil n'est pas possible compte tenu des problèmes de tolérance des unités disque. Il suffit qu'une tête soit un peu déplacée pour que les données perdent leur parfaite continuité, avec comme conséquence la perte de toute l'information du secteur. Pour écrire quelques octets sur un secteur, il est nécessaire de le récrire tout entier, ne serait ce qu'à cause du CRC.

L'opération de recherche de profil est considéré comme finie après la première comparaison avec succès ou si à la fin des secteurs d' une piste, aucune vérification n'a été réussie. Si un profil doit être recherché sur plusieurs secteurs, le microprocesseur doit relancer l'opération successivement sur plusieurs secteurs. C'est le bit de vérification du mot d'état du contrôleur qui indique le succès d'une opération de recherche.

Un profil, un mot clé ne peut être plus long que huit octets, pour le contrôleur disque. Le prototype du contrôleur MAGE utilise un profil fixe de 4 octets, ce qui lui permet d'avoir une longueur standard de certains opérations ainsi comme une simplification au niveau des paramètres et microprogrammes des processeurs en tranche.

VII. 6 FORMAT D'ENREGISTREMENT

Tout disque avant d'être utilisé, donc de recevoir des données, doit être formaté. Ce formatage consisté à écrire certaines configurations de bits et des informations d'adresse et de contrôle au début de chaque secteur. Cela permet au contrôleur disque d'avoir des repères pour connaître le point physique où commencent les données, ainsi que de pouvoir vérifier l'état du secteur et si son adresse est bonne. Cette information au début de chaque secteur est appelée "descripteur du secteur".

Un disque est formaté par un programme spécial qui est exécuté une fois à l'initialisation du disque. Ce programme ne sera plus utilisé sur le même disque, à moins qu'une mauvaise opération n' efface les descripteurs. Le formatage d'un disque est réalisé par le contrôleur.

Le format que doit adopter un secteur, et donc le disque, est variable d'un constructeur à l'autre et plutôt d'un contrôleur à l' autre. Le format utilisé pour les disques du PBD-MAGE, est basé sur les suggestions des fabricants [T.CDC.77] [T.AMP.77] et sera celui décrit dans cette section. Il n'existe pas une régle exacte pour déterminer quel est le meilleur format à adopter pour tel ou tel disque, mais en contre-partie, il existe des conditions de contour qui fournissent des limites. Par exemple, plus grands sont les secteurs, plus de données il peuvent contenir, mais plus une erreur peut survenir et rendre inutilisable toute l'information qu' il contient. Il existe une fourchette de longueur conseillée où la probabilité d'erreur est faible par rapport au nombre de données du secteur.

VII.6.1 Le Format

Le début physique d'un secteur est déterminé par des moyens mécaniques, magnétiques ou électriques, dépendant du type d'unité disque. L'unité disque fournit au contrôleur un signal indiquant le début de chaque secteur et un autre signal indiquant quel est le premier secteur d'une piste, l'index. Le signal de début secteur est utilisé par le contrôleur pour travailler avec des secteurs de longueur fixe, c'est le cas du PBD-MAGE. Si le contrôleur est conçu pour des secteurs de longueur variable, le formatage est assez différent. Cette section abordera juste le format des secteurs de longueur fixe.

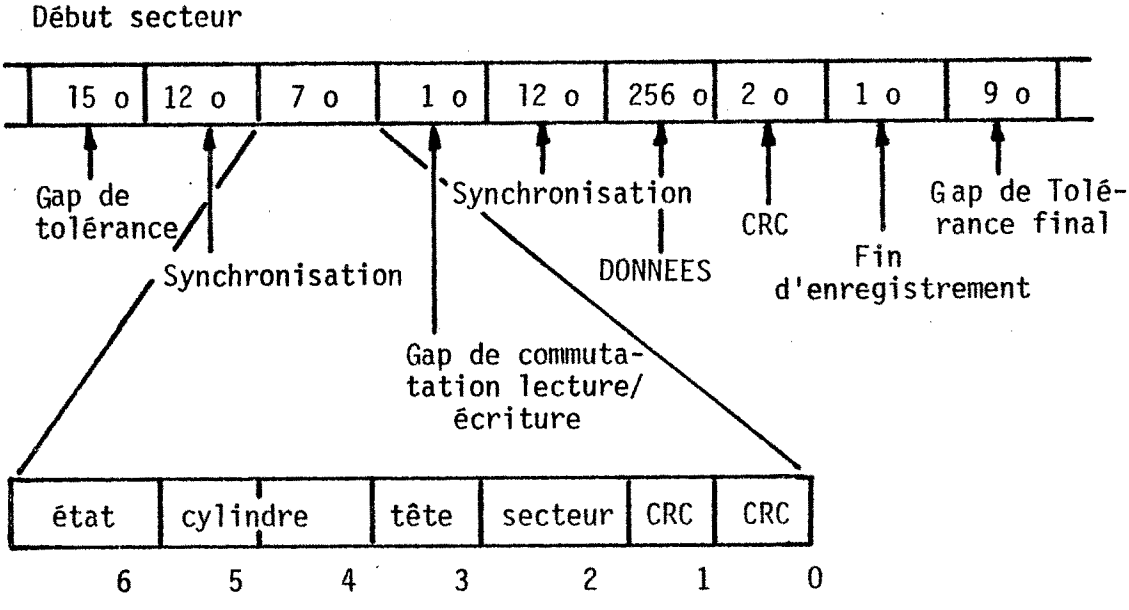
Le format d'un secteur est composé de plusieurs zones. Il est le suivant par ordre de formatage :

a) Gap de tolérance - cet espace est composé de 15 octets à zéro qui correspondent au temps de réaction du contrôleur. Cela permet de changer le disque "pack" d'une unité à l'autre sans que le temps de réponse des contrôleurs puisse poser de problèmes.

b) Préambule ou Profil de Synchronisation d'Adresse. Le profil est composé de 11 octets à zéro (88 bits) et un octet de un. Cela permet au contrôleur de savoir où commence les informations utiles. Les données commencent juste après le dernier bit à un du préambule. Ce profil est choisi de manière à ce qui'il soit presque impossible de trouver des données ayant un aspect semblable. Cela poserait des problèmes au contrôleur pour déterminer le début de l'information utile. Certains contrôleurs adoptent un profil de synchronisation constitué de zéros suivi par un seul bit à un. Ce choix varie d'un constructeur à l'autre et est plutôt une conséquence du matériel, des circuits, du contrôleur.

c) Descripteur du secteur - Ces 7 octets permettent d'identifier le secteur. Le premier octet fournit l'état du secteur, s'il est abîmé et s'il est protégé contre l'écriture. Les deux prochains

octets indiquent l'adresse cylindre, le prochain l'adresse tête et l'autre octet donne l'adresse secteur. En dernier lieu viennent les deux octets de CRC qui permettent au contrôleur de vérifier si la lecture du descripteur du secteur est correcte.



d) Gap de commutation - cet octet donne au contrôleur le temps pour commuter la tête de lecture en écriture, si c'est le cas.

e) Perfil de synchronisation de donnée - les 12 octets, 11 octets de zéros et 1 octet de uns permet au contrôleur de se synchroniser sur le début des données.

f) Données - ces 256 octets de données sont l'espace utile du secteur. C'est ici que sont stockées les informations de l'utilisateur de l'unité disque.

g) CRC - Cyclic Redundancy Check - les deux octets du code permettent de vérifier si les données ont été lues correctement.

h) Fin d'enregistrement - c'est un octet de zéros qui permet de laisser toujours propre la fin d'un nouvel enregistrement. Cela évite qu' il restequelques bits d'un enregistrement antérieur, dus à des différences de tolérance de l'enregistrement du contrôleur.

i) Gap de tolérance final - ce dernier gap de 9 octets est une sécurité pour éviter que le données ne soient écrites sur le prochain secteur et vice versa. Il évite la perte d'informations à la fin d'un secteur due à des différences de positionnement des têtes d'une unité à l'autre.

Ce format de secteur donne l'efficacité suivante d'occupation :

$$\text{efficacité} = \frac{\text{nombre d'octets de données/secteur}}{\text{nombre total d'octets/secteur}} = \frac{256}{315} = 81\ \%$$

Avec ce format le nombre de secteurs que peut avoir une piste appartenant à des unités disque 9766 T.CDC.77 de 300 M octets est de :

$$\text{nombre de secteurs} = \frac{\text{nombre total d'octets/piste}}{\text{nombre total d'octets/secteur}} = \frac{20160}{315} = 64 \text{ secteurs}$$

VII.6.2 Enregistrement du format

Tout disque avant son utilisation doit être formaté. Cette opération consiste à enregistrer le format sous l'action du contrôleur :

-Premièrement sélectionner l'unité disque, le cylindre, la tête et le secteur. Attendre le signal de début secteur.
- Activer la tête d'écriture et écrire 26 octets de zéros suivis d'un octet de uns.
- Ecrire les octets d'état du secteur, l'adresse cylindre, tête et secteur, plus les deux octets de CRC.
- Ecrire 12 octets à zéro qui correspondent au gap de commutation et au profil de synchronisation, écrire l'octet de uns.

- Ecrire le champ des 256 octets de données avec des uns. Après écrire les deux octets de CRC correspondants.
- Ecrire l'octet de zéros de fin d'enregistrement.
- Rien n'est écrit dans le gap de tolérance finale et l'opération de formatage du secteur s'arrête à l'octet de fin d'enregistrement.

CHAPITRE VIII

IMPLEMENTATION DU PROCESSEUR DISQUE :

LE PROTOTYPE

1. METHODOLOGIE D'IMPLEMENTATION DU PROTOTYPE
2. OUTIL DE DEVELOPPEMENT : L'EXORCISER
3. LIAISON "DAISY CHAIN" DU CD
4. CONNEXION MATERIELLE DU CONTROLEUR COTE MICROPROCESSEUR
5. CONNEXION MATERIELLE DU CONTROLEUR COTE DISQUE
6. REGISTRES DES SIGNAUX DE CONTROLE DU DISQUE
7. SIGNAUX DE VALIDATION DES ORDRES DU DISQUE
8. SIGNAUX DE CONTROLE PROVENANT DU DISQUE
9. UNITE DE RECONNAISSANCE DE L'ADRESSE SECTEUR
10. DECODEUR D'ADRESSES DU CONTROLEUR
11. LECTURE DES DONNEES DU DISQUE
12. ECRITURE DES DONNEES DU DISQUE
13. MULTIPLEXAGE DES INTERFACES POUR LE CABLE B
14. TRANSFERT DES DONNES ENTRE CONTROLEUR ET MEMOIRE BUFFER
15. SIGANL DE HALT
16. UNITE DE SYNCHRONISATION
17. ARCHITECTURE DE PROCESSEURS

VIII - IMPLEMENTATION DU PROCESSEUR DISQUE : LE PROTOTYPE

Le chapitre précédent a donné les caractéristiques du processeur disque, ce chapitre aura pour but de présenter les principaux aspects de sa réalisation matérielle.

Les principales caractéristiques de la plaque contrôleur du processeur disque sont : qu'elle est microprogrammée et qu'elle se synchronise sur l'horloge des unités disque. L'utilisation des microprogrammes permet une meilleure action de contrôle sur tous les signaux du contrôleur. D'autre part, le fait qu'il se synchronise sur l'horloge du disque lui permet d'avoir une plus grande souplesse de communication avec les unités de disque.

Ces deux caractéristiques représentent les principales originalités du contrôleur disque. Le fait de synchroniser le contrôleur à imposé l'utilisation d'une architecture à haute vitesse pour le processeur en tranches. Avec toutes les possibilités de la carte contrôleur, plus celles du microprocesseur, le processeur disque dispose d'une grande versatilité et d'un pouvoir d'évolution. Cela lui permet d'accompagner aisément la progression de la technologie de disque, ainsi que sa propre évolution.

VIII.1 METHODOLOGIE D'IMPLEMENTATION DU PROTOTYPE

Une des décisions importantes, déjà mentionnée au ch. VI.3, qui doit être prise lors de la construction d'un prototype, concerne la démarche à suivre pour sa construction. La technique utilisée pour le processeur disque a été la même que celle proposée pour le reste de l'architecture du PBD-MAGE. Elle consiste à employer

dans la mesure du possible, des outils de développement déjà existants pour faciliter la construction du prototype.

Pour le processeur disque, deux outils ont été employés, l'Exorciser [T.MOT.75] pour la partie microprocesseur et MADAM [T.SCH.78] pour la partie processeurs en tranche. La plaque contrôleur disque a été construite en respectant les standards Exerciser de manière à ce qu'elle puisse être connectée directement sur le bus EXORCISER. Dans un premier temps, l'Exorciser est utilisé pour tester chaque fonction de la carte, ensuite, quand la carte sera finie, il assumera le rôle du microprocesseur disque en envoyant les requêtes au contrôleur disque (CD).

L'emploi de l'outil MADAM est différent de celui de l'Exorciser. MADAM est utilisé seulement pour charger et modifier la mémoire de microprogramme des processeurs en tranche. L'outil MADAM sera présenté ultérieurement dans les annexes.

La technique employée pour la construction du PBD a été de partir de la plaque CD (contrôleur disque) et d'implémenter successivement les autres plaques. Commencer du côté de l'unité disque pour finir par l'utilisateur. Par cette technique l'Exorciser assume toujours les fonction du niveau supérieur. Par exemple, vis-à-vis du CD, l'Exorciser représentera le processeur P2 et le reste du PBD. Au dernier niveau, lors de la construction du processeur P1, l'Exorciser simulera la fonction de l'utilisateur.

VIII.2 OUTIL DE DEVELOPPEMENT : l'EXORCISER

Le contrôleur communique d'un côté avec les unités disque et de l'autre il est relié à un Exorciser [T.MOT.75] comme moyen d'interface avec le monde extérieur, pour le prototype. L'Exorciser est

un microordinateur construit autour d'un microprocesseur 6800 [T.MOT.76] , conçu pour être utilisé ou comme un petit système d'usage général, ou comme un outil de développement. Il possède des facilités de logiciel de test et d'interface qui permettent une mise au point rapide des cartes construites. Ces cartes peuvent être externes à l'Exorciser et reliées à lui à travers un câble, ou elles peuvent être construites sur le standard de cartes de l'Exorciser et à ce moment, elles seront directement connectées sur son bus ; c'est le cas du prototype du CD.

Le prototype a été conçu de manière à ce que ses cartes puissent être connectées à l'Exorciser. Le CD pour des raisons d'implantation est réalisé en deux cartes. L'Exorciser pour les plaques CD représente le microprocesseur, le processeur P2 et la mémoire buffer. Donc les fonctions lentes du microprocesseur, décrites dans le chapitre précédent, seront exécutées par l'Exorciser. Vis-à-vis de la plaque CD, il simulera, hors les fonctions de microprocesseur, les requêtes du monde extérieur et ainsi il englobe les aspects d'outil de développement et de partie intégrante de l'architecture du PBD (microprocesseur disque) et aspect de simulation des requêtes externes.

La figure VII.1 donne une idée de l'aspect matériel du prototype. Le CD d'un côté est relié à l'Exorciser et de l'autre côté il communique avec les unités disque à travers les interfaces SMD (Norme de communication voir chap IV.10°). Le prototype peut se raccorder à deux unités disque.

Le prototype permettra de déterminer le comportement du contrôleur disque, déterminant les performances et exécutant des tests.

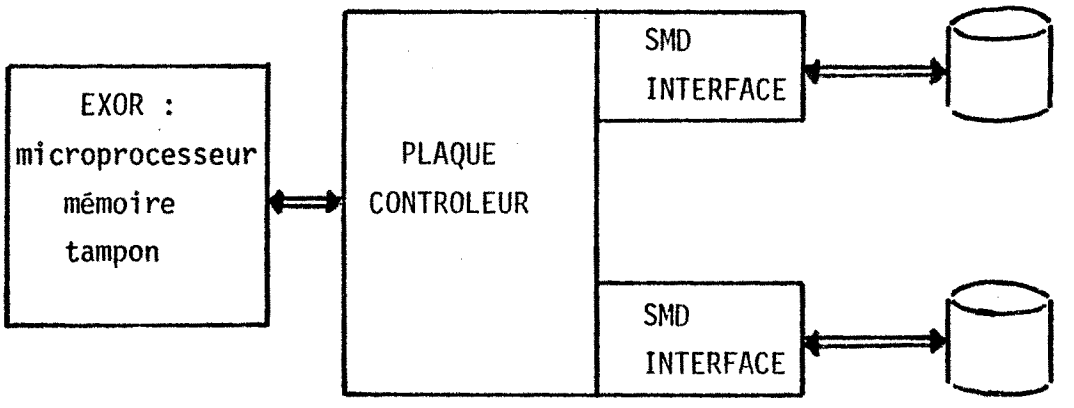


Fig VIII.1 - Aspect général du prototype du CD

VIII.3 LIAISON "DAISY CHAIN" du CD

Pour le CD du PBD-MAGE le choix du type de liaison entre contrôleur disque et unité disque, est retombé sur la connexion "daisy chain"(voir chap IV.10). Les performances nécessaires au contrôleur, ainsi que le peu d'intérêt de connecter plus de 4 unités disque, n'incitaient pas à sophistiquer et rendre coûteux le CD. Dans l'autre type de liaison,la liaison étoile (chap. IV.10), il est nécessaire pour chaque câble A d'avoir tous les registres d'adressage et circuits d'interface, ce qui accroît le prix du contrôleur très rapidement. Donc, pour le PBD-MAGE, le contrôleur possède juste un câble A, avec les circuits associés, les câbles B étant multiplexés sur la logique d'écriture/lecture (Fig VIII.2)

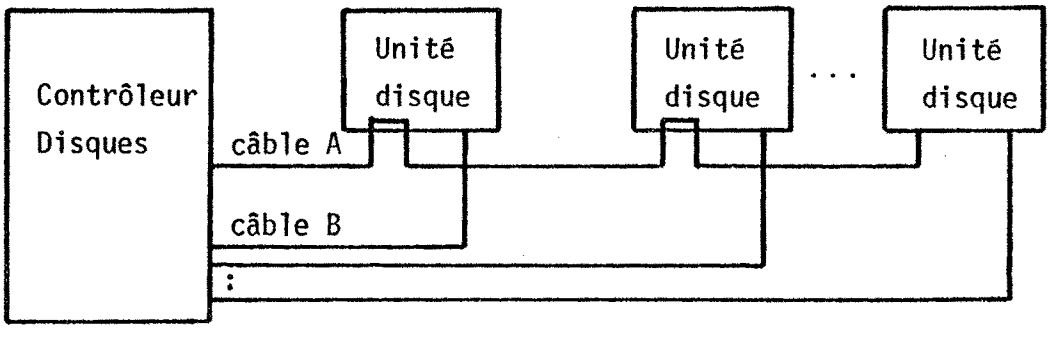


Fig. VIII.2 - Liaison "daisy chain" du CD

VIII.4 CONNEXION MATERIELLE DU CONTROLEUR COTE MICROPROCESSEUR

Les principaux signaux qui co posent la connexion du contrôleur vers le microprocesseur sont le bus adresse, le bus données et quelques signaux de contrôle. Ces signaux utilisent la logique trois états et pour cela des circuits amplificateurs de ligne TTL bidirectionnels [74LS245] et unidirectionnels [74LS240] [T.TEX.77] sont employés. Tous les signaux doivent passer par un de ces circuits avant d'arriver au bus de l'Exorciser. Cela permet la mise au niveau des signaux, une meilleure immunité aux bruits extérieurs, ainsi qu'une protection contre les courts circuits. L'utilisation de ces signaux sera décrite, au fur et à mesure des besoins, tout au long des prochains chapitres.

VIII.5 CONNEXION MATEREILLE DU CONTROLEUR COTE DISQUE

Ces signaux doivent suivre les normes SMD (chap IV.10) assez strictes pour la communication avec les unités disque. Ces normes sont différentes pour le connecteur du câble A ou celui du câble B, cela est une conséquence, en partie, des vitesses différentes des signaux qui circulent dans un câble et l'autre. Des circuits amplificateurs différentiels de ligne (75110 et 75108 ou 75 107) sont utilisés comme transmetteurs et récepteurs de lignes. Toutes ces lignes possèdent des terminaisons passives d'un ou des deux côtés, dépendant du câble. Ces signaux seront décrits plus loin dans ce chapitre (Fig VIII.3)

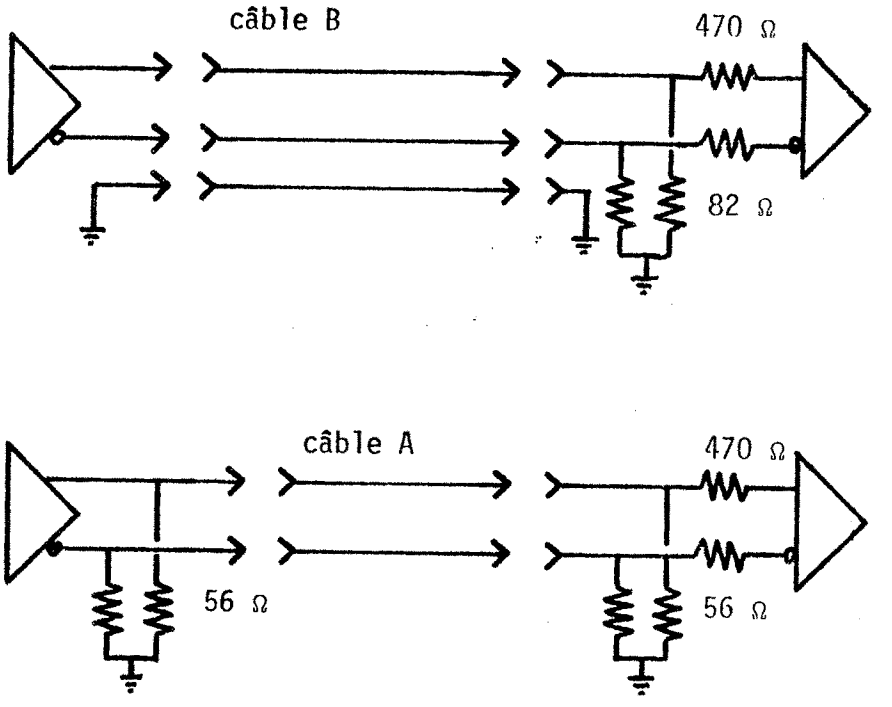
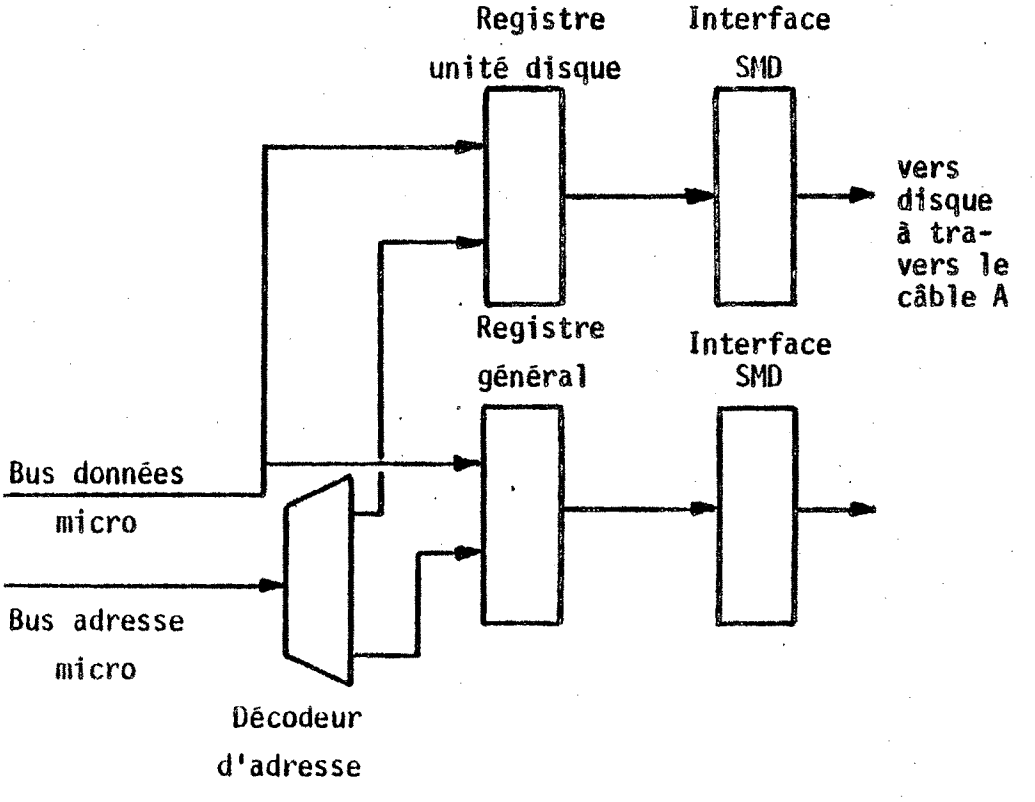


Fig VIII.3 - Terminaisons des connexions CD/unités disque

VIII.6 REGISTRES DE SIGNAUX DE CONTROLE DU DISQUE

D'une manière générale, l'envoi d'ordres vers le disque, est faite par leur stockage, premièrement dans des registres et ensuite par l'activation du signal de validation respectif, autorisant le disque à lire les registres. Ces signaux étant plutôt lents (1 MHz), la logique associée est simple. Ce chargement des registres par le micro (l'Exorciser) est réalisée par une instruction du type chargement de mémoire (store). Les données, ordres, viennent par le bus données micro, et leur rangement dans les registres respectifs est validé par le décodage du bus adresse micro.

Les signaux concernés sont la sélection d'unité disque, l'adresse cylindre, l'adresse tête et les fonctions. Ces trois derniers type de signaux utilisent le même registre, registre général, pour envoyer les ordres aux disques (chp VII.4.4)



LDAA # X chargement du registre A du micro avec la valeur X

STAA ADR, envoi de la valeur X pour être stockée dans le registre CD d'adresse ADR

Fig VIII.4 - Envoi d'ordres vers les unités disque.

VIII.7 SIGNAUX DE VALIDATION DES ORDRES POUR DISQUES

Ces signaux autorisent le transfert des ordres vers le disque ; en d'autres termes, ils informent les unités disque qu'il y a une information rangée dans un des registres du contrôleur et que donc les disques peuvent lire cet ordre. Il y a quatre lignes de validation, une pour chaque ordre : unité disque, adresse cylindre, adresse tête et fonction.

Deux de ces signaux étant impulsionnels,il a été nécessaire d'utiliser une technique non standard pour économiser le matériel et aussi pour permettre d'avoir les mêmes circuits pour les quatre validations. La technique utilisée a été de rendre ces signaux de validation identiques aux autres et donc stockés dans des registres. C'est le microprocesseur qui les active et désactive directement, à travers le chargement successif des valeurs "un" et "zéro". Donc, pour avoir un signal pulsé de plus de 1 μs, il suffit d'avoir deux instructions successives de chargement, une pour "un", l'autre de "zéro" (Fig VIII.5)

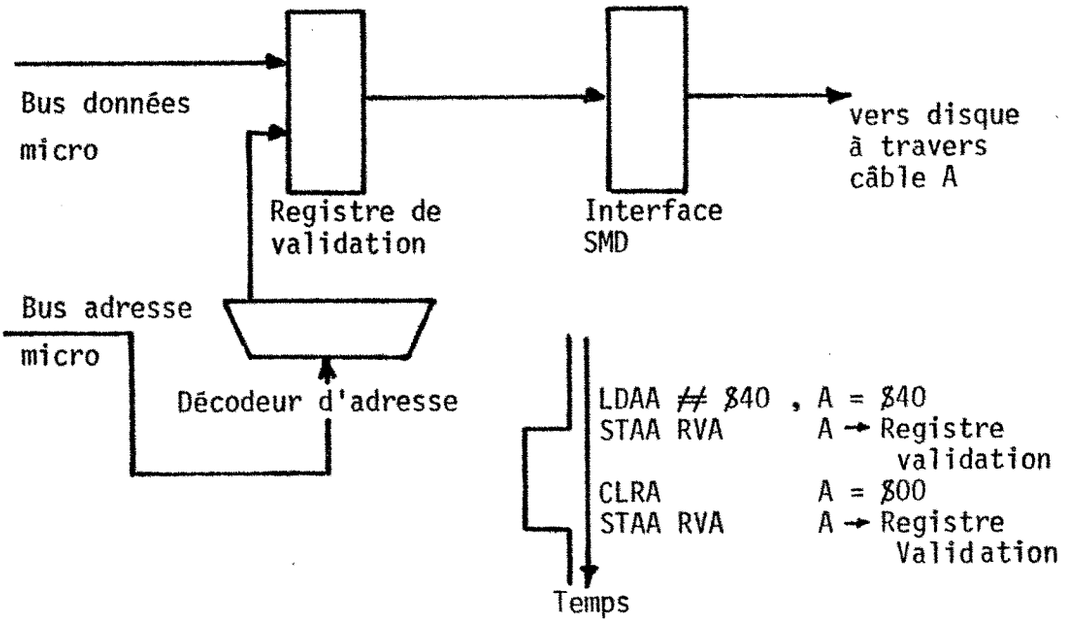


Fig VIII.5 - Signaux de validation d'ordres disque

VIII.8 SIGNAUX DE CONTROLE PROVENANT DU DISQUE

"Faute, "Erreur de Recherche" et "Unité prête", sont trois lignes provenant du disque , qui indiquent une anomalie par rapport au fonctionnement normal du disque. Ces informations sont, à travers des circuits, aiguillées vers le bus données de l'Exorciser pour que celui-ci puisse lire directement l' information. Cette lecture est faite d'après le programme d'utilisation du disque, les primitives (chap VII.5). Par exemple, la lecture peut être faite avant tout lancement d'opération de manière à savoir s'il y a anomalie et après l'opération pour savoir si l'opération lancée s'est bien déroulée (Fig VII.6).

Deux autres signaux, provenant du disque, sont aussi aiguillés vers l'Exorciser, ce sont "Unité Protégée" et "Sur cylindre". Le premier indique que l'unité disque est protégée contre l' écriture et l'autre indique à l'Exorciser que les têtes du disque sont positionnées sur un cylindre et que donc une opération de recherche cylindre a été réalisée avec succès. Tous ces cinq signaux sont dirigés vers le microprocesser(Exorciser) et c' est lui qui prendra la décision de l'action à mener. Ces signaux représentent pour le micro le Registre d'Etat du Disque. Ces lignes peuvent déclencher une interruption pour informer le microprocesseur de leur modification.

Les trois signaux de contrôle restants provenant du disque : "Index", "Marque de Secteur" et "Marque d'Adresse" concernent la position angulaire du disque. Les deux premiers permettent d'établir le début du secteur et son adresse, pour des disques à sectorisation fixe. Le signal de "Marque d'adresse", par contre , est employé lors de l'utilisation de secteurs de longueur variable où il indique aussi le début d'un secteur.

Pour le contrôleur du PBD-MAGE, une sectorisation fixe a été choisie et donc ce sont les deux premiers signaux qui sont employés au niveau de la plaque contrôleur pour déterminer l'adresse d'un secteur. La prochaine section présente la détermination de l'adresse secteur.

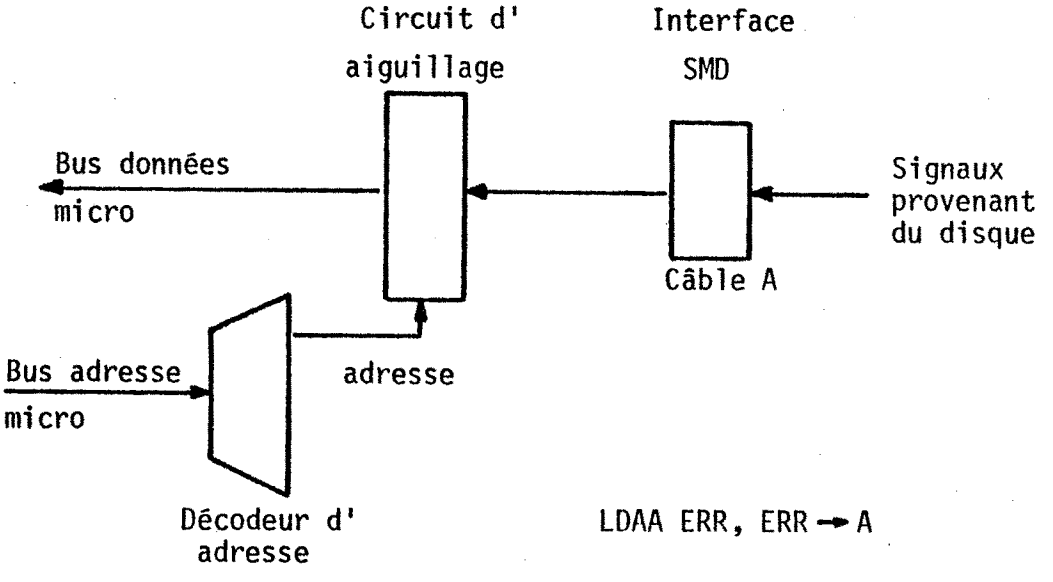


Fig VIII.6 - Signaux de contrôle provenant du disque

VIII.9 UNITE DE RECONNAISSANCE DE L'ADRESSE SECTEUR

L'Index et la Marque de secteur permettent au contrôleur de savoir à tout instant le secteur sur lequel passent les têtes de lecture/écriture du disque concerné. Dans le CD, ces deux signaux vont vers un Registre Compteur de Secteur qui utilise la "Marque de Secteur" pour s'incrémenter et le signal "Index" pour se remettre à zéro. De cette manière, le "Registre Compteur Secteur" contient, à tout instant, la connaissance de l'adresse du secteur sur lequel passe la tête.

Pour que le microprocesseur adresse un secteur, il faut qu'il charge d'abord le "Registre Adresse Secteur" avec l'adresse choisie et qu'ensuite ce registre soit comparé avec le registre-compteur jusqu'à la coïncidence de l'adresse. A cet instant, le contrôleur peut démarrer une opération de lecture ou d'écriture sur le secteur.

La comparaison sera réalisée par des circuits comparateurs (SN 7485 [T.TEX.77]) et elle est autorisée par le signal "Sur Cylindre". Cela permet d'effectuer la comparaison dès que la tête est stabilisée sur une piste. Du circuit comparateur sort un signal qui active l'Unité de Séquencement et de Contrôle permettant de démarrer une opération sur ce secteur (Fig VII.7)

Comme option (non implanté sur le prototype) le Registre Compteur Secteur peut être relié au bus données du microprocesseur et permettre ainsi sa lecture. Le micro peut ainsi connaître l'adresse courante secteur. Cette information peut être utilisée pour des opérations de test d'erreurs lors de la maintenance, pour des optimisations d'accès et pour certains types d'opérations de recherche à la volée.

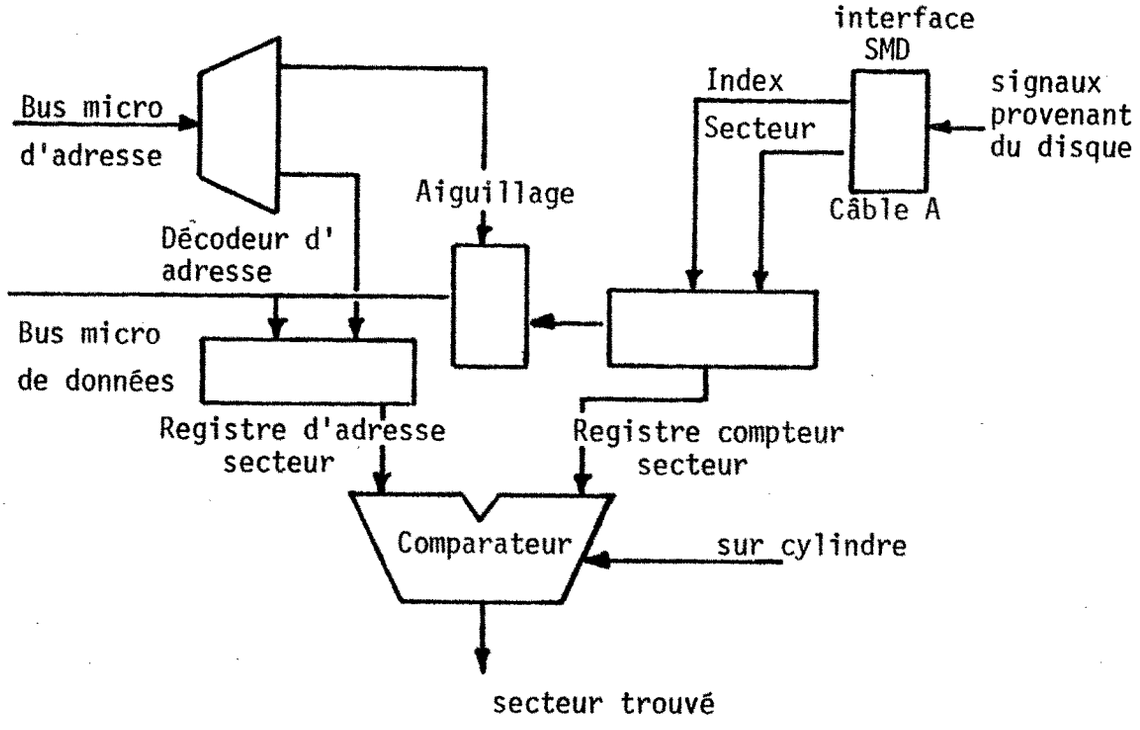


Fig VIII.7 - Circuit de Recherche Secteur

VIII.10 DECODEUR D'ADRESSES DU CONTROLEUR

La section VII.4.2 a donné les caractéristiques du décodage des adresses provenant du microprocesseur. Le contrôleur décode 16 adresses qui vont correspondre à des registres ou à des signaux d'activation. Ces registres sont un des moyens de communication entre le contrôleur et l'extérieur, l'autre étant le DMA des données à partir de la mémoire buffer. Ces adresses décodées peuvent être employées aussi bien pour la lecture ou l'écriture d'un registre.

Le décodage de l'adresse est réalisé par un circuit décodeur (SN 74 154 T.TEX.77) sur les 4 bits de poids faible de l' adresse micro. Les autres bits sont décodés dans un point fixe de la mémoire établissant la position des 16 bits dans l'espace d'adressage du micro.

Cependant, pour permettre de déplacer la position des registres (adresses) du contrôleur dans l'espace mémoire-micro, les 4 bits de poids fort passent à travers un circuit comparateur pour choisir cette implantation. Ce comparateur fait la comparaison entre les 4 bits de poids fort du bus adresse-micro avec la valeur de 4 clefs (interrupteurs), ce qui permet de changer les adresses du contrôleur en modifiant ces clefs (Fig VIII.8).

Les registres et signaux d'activation concernés sont :

- Registre Unité Disque, en écriture
- Registre Général, en écriture, 2 octets
- Registre de validation, en écriture
- Registre Spécial, en écriture, (tests prototype)
- Registre Etat Disque, en lecture
- Registre Adresse Secteur, en écriture (en lecture sur option)
- Registre Commande, en écriture
- Registre Adresse Mémoire, en écriture, 2 octets
- Activation de l'Unité de Séquencement et Contrôle

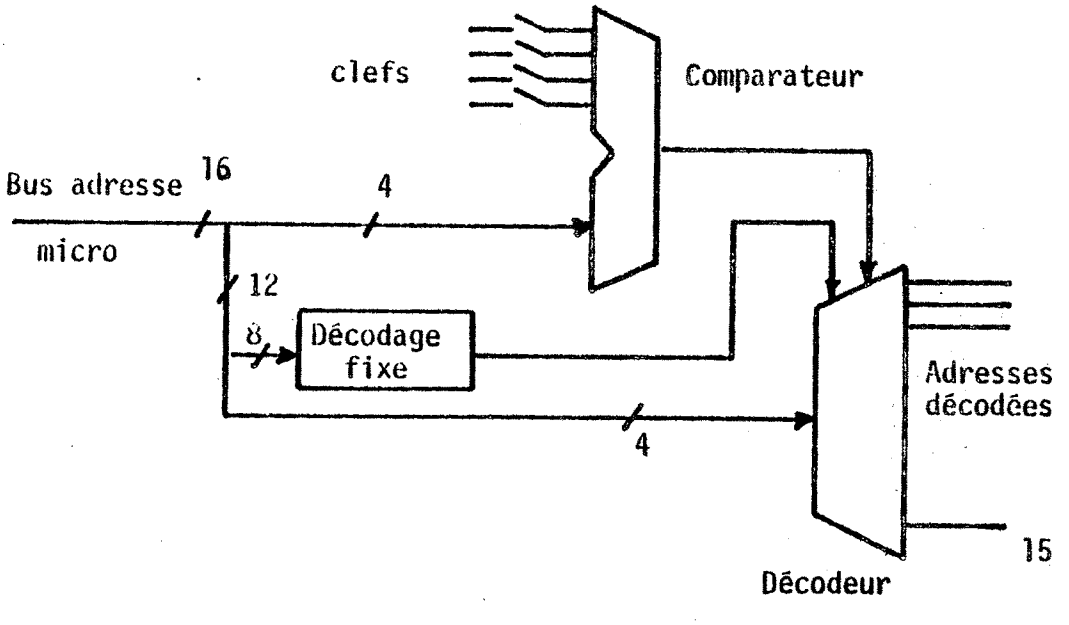


Fig.VIII.8 - Decodeur d'adresses

VIII.11 LECTURE DES DONNEES DU DISQUE

Les signaux pour la lecture et l'écriture des données du disque proviennent du câble B. Ils sont émis ou reçus en série. Les lignes provenant du disque concernées par l'opération de lecture sont "Lecture de Données" et "Horloge de Lecture".

Dans l'opération de lecture d'un secteur disque, les bits, codés en NRZ, arrivent en série à la carte contrôleur par la ligne "Lecture des Données". Ces bits sont transmis à l'"Unité de Sérialisation/Désérialisation où ils sont transformés en octets pour être transférés vers le "Registre Sortie de Données". Le signal qui commande le décalage du désérialisateur provient de la ligne "Horloge de Lecture". Ce signal est généré par l'unité dique et l'occurence des bits sur la ligne "Lecture des Données". Toute l'opération reste sous le contrôle de l'Unité de Contrôle et de Séquencement, UCS (les processeurs en tranche). Les prochaines sections aborderont le mécanisme de transfert de ces données ainsi que les séquences de microprogrammes utilisées.

Les données provenant du disque, sont aussi envoyés à un circuit de CRC (Cyclic Redundancy Check) qui permet de déterminer si les données ont été lues correctement à la fin d'un transfert secteur.

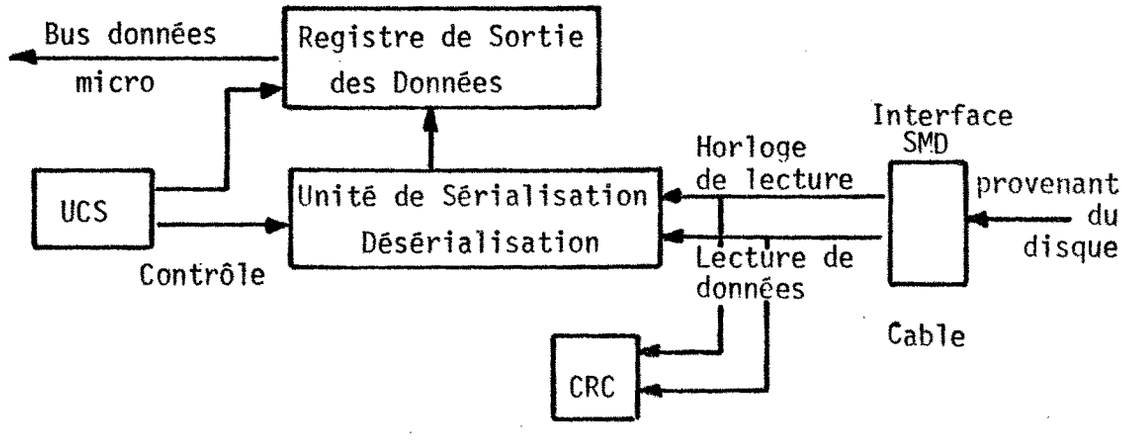


Fig. VIII.9 - Lecture des Données

VIII.12 ECRITURE DES DONNEES DU DISQUE

Pour une opération d'écriture sur disque, la démarche est semblable à celle d'une lecture, à l'exception des données, qui viendront directement de l'Unité de Sérialisation/Désérialisation à partir de la mémoire tampon (buffer) du microprocesseur. Le transfert est exécuté sous le contrôle de l'USC. Par décalages successifs le sérialisateur envoie l'octet sous forme série vers le disque dans un code NRZ. Le transfert des données vers le disque est fait à travers la ligne d'Ecriture des Données. Chaque bit envoyé est accompagné d'un top d'horloge sur la ligne "Horloge d'Ecriture". Cette horloge signale au disque que le bit de données est valide. L'Horloge d'Ecriture est générée à partir de l'Horloge Servo qui provient du disque. Cette dernière est générée par la lecture, sur une piste spéciale du disque, de signaux d'horloge préenregistrés. Cette technique est nécessaire compte tenu de la grande densité d'enregistrement.

Les données à écrire passent aussi par un circuit de CRC, qui génère un code détecteur d'erreur qui est enregistré à la fin du transfert des données d'un secteur. Toute l'opération d'écriture s'effectue sous le contrôle de l'USC (voir prochaines sections pour plus de détails).

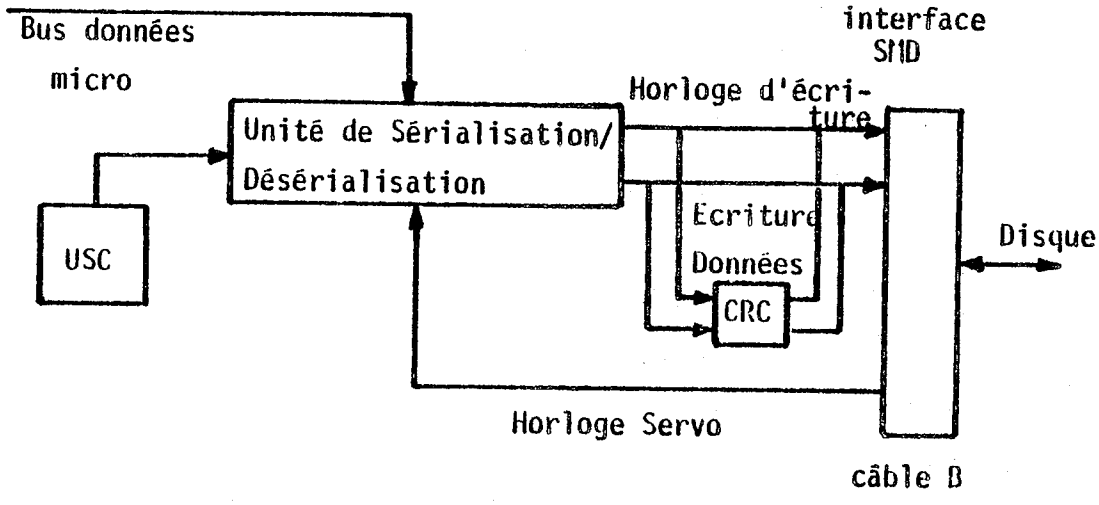


Fig VIII.10 - Ecriture des données

VIII.13 MULTIPLEXAGE DES INTERFACES POUR LES CABLES B

Il existe dans une liaison du type de "daisy chain" entre CD et les unités disque, plusieurs interfaces B, une pour chaque unité disque jusqu'à un maximum de 16. Les signaux provenant de l'Unité de Sérialisation/Désérialisation ainsi que ceux des horloges doivent être multiplexés entre les divers interfaces B de manière à ce qu'ils puissent etre connectés à l'interface choisi.
Le choix de l'unité avec laquelle le CD ira dialoguer est déterminé par l'adresse stockée dans le Registre d' Unité Disque. L'adresse de ce registre de 4 bits est décodée et les sorties du décodeur activent les interfaces SMD pour les câbles B. (Fig VIII.11

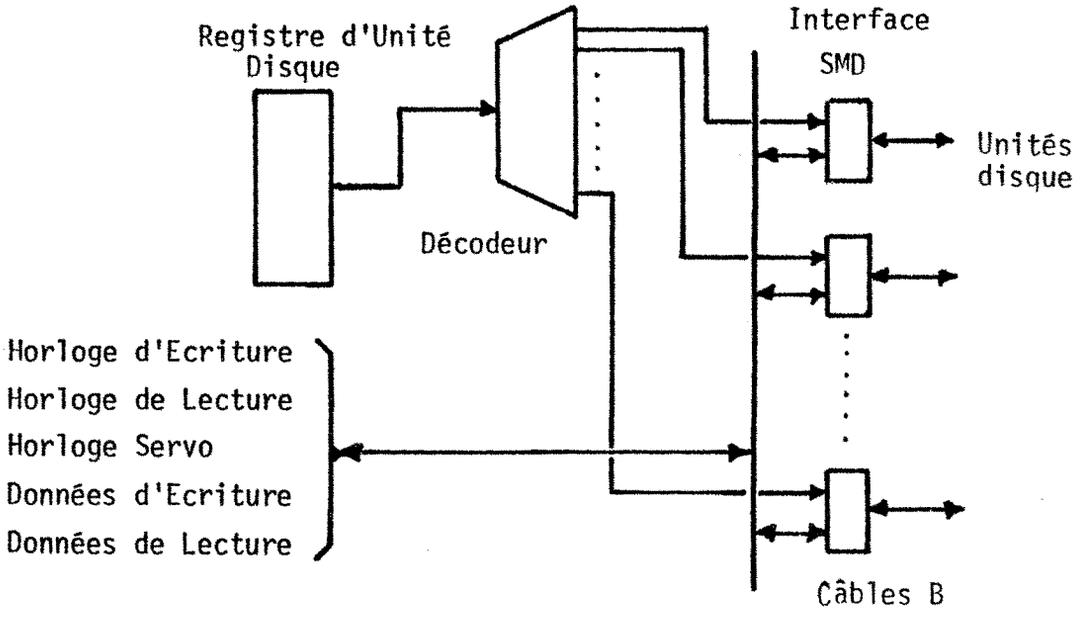


Fig VIII.11 - Multiplexage des interfaces B

VIII.14 TRANSFERT DES DONNES ENTRE CONTROLEUR ET MEMOIRE BUFFER

La technique utilisée pour le transfert des données entre le contrôleur et la mémoire buffer est du type accès direct à la mémoire, DMA (Direct Mémory Access) Avant le transfert, le microprocesseur charge le Registre Adresse Mémoire du contrôleur avec l'adresse du premier mot de la zone de Mémoire Buffer choisie. Ce Registre Adresse estde 16 bits, ce qui lui permet d'adresser la Mémoire Buffer quelle que soit sa place dans le champ d'adressage du microprocesseur.
Le travail de l'Unité de Séquencement et de Contrôle (USC) consiste à valider le mot et à incrémenter successivement le Registre d' Adresse pour chaque mot (octet) transféré, permettant ainsi le balayage de toute la zone buffer. Cette action sera la même tant en lecture lecture qu'en ecriture, la différence étant que dans un cas les données sont lues de la mémoires et sont chargés directement dans l'Unité de Sérialisation/ Désérialisation (USD) du contrôleur, tandis que dans l'autre cas, les données viennent du Registre de Sortie Données pour être écrit dans la mémoire buffer (Fig VIII.12)

Un transfert en lecture ou en écriture disque est lancé par l'USC quand elle arrive sur le secteur choisi. Cela est fait automatiquement d'après la séquence de microprogramme activée. Une description plus détaillée des microprogrammes de transfert est présentée dans une autre section de ce chapitre.

Etant donné le tauxde transfert de l'ordre de 1 MHz, l'unique technique viable est le DMA. Quand le disque débite en lecture ou en écriture ses données, il ne peut pas être arrêté et doit réaliser le transfert des données jusqu'à la fin du secteur. Cela empêche

le microprocesseur d'accéder cette mémoire pendant le transfert. La section VI.2.7 a déjà mentionné la solution adoptée pour résoudre ce problème en utilisant la technique du "Halt" (arrêt) du microprocesseur lors du transfert. Quand le contrôleur détecte le secteur choisi, l'USC arrête le microprocesseur grâce au signal de "Halt" et le transfert des données commence dans un sens ou dans l'autre avec l'aide du Registre d'Adresse Mémoire. A la fin du transfert du secteur, l'USC libère le microprocesseur par la désactivation du signal de "Halt" et celui-ci continuera à exécuter son programme normalement à partir de l'instruction où il a été arrêté.

L'activation du Halt n'est pas immédiate, le signal n'étant pris en compte qu'à la fin de l'instruction, cela peut prendre quelques cycles du microprocesseur, mais ne pose aucun problème, car après l'activation du Halt par l'USC, lors de l'arrivée des têtes sur le bon secteur, il existe un temps de l'ordre de 47 μs avant le transfert. Ce temps est utilisé par le contrôleur pour se synchroniser et pour vérifier l'adresse du secteur.

Ce mécanisme utilisé pour le transfert des données-secteur, est aussi utilisé pour le transfert des paramètres du microprocesseur vers l'Unité Arithmétique et Logique (UAL) du contrôleur. C'est le cas de l'opération de "Recherche à la Volée" où plusieurs paramètres doivent être envoyés à l'UAL du contrôleur.

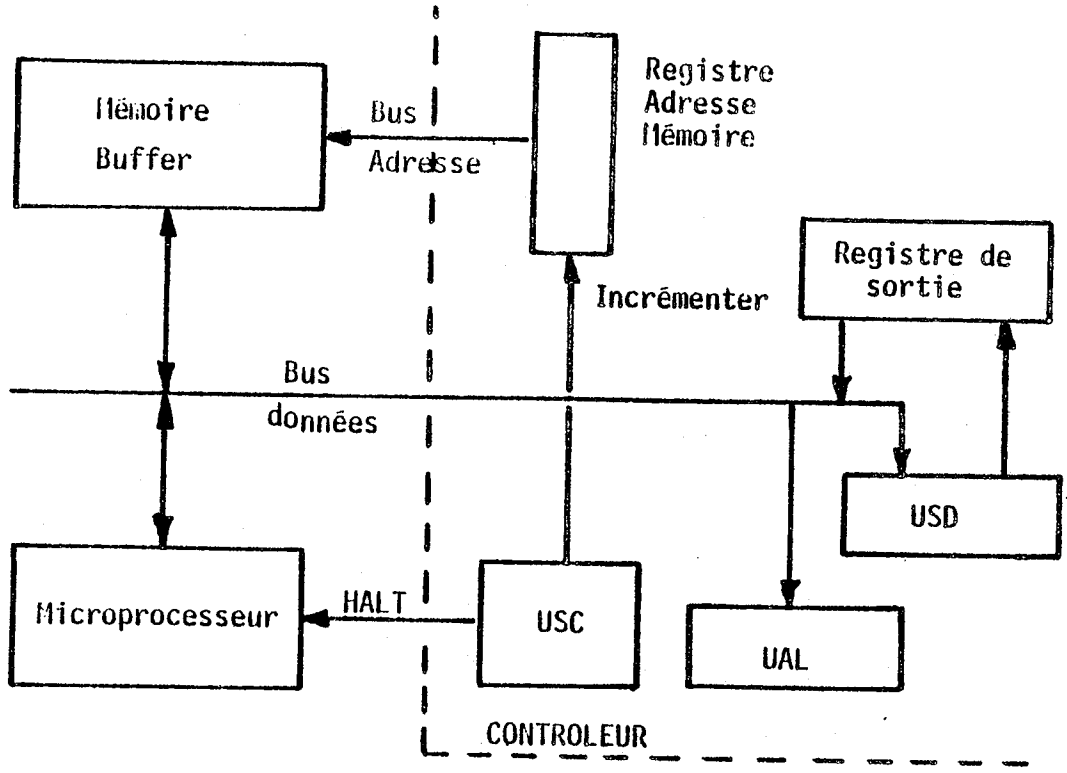


Fig VIII.12 - Transfert des données dans le processeur disque

VIII.15 LE SIGNAL DE HALT

Cette section a pour but de présenter et de discuter les moyens qui existent entre le contrôleur et le micro pour qu'ils s'informent de l'occurence d'un événement. Le microprocesseur est dédié au contrôleur dans le sens qu'il ne reçoit qu'un type d'interruptions (dans le sens général) externes provenant du contrôleur disque. Le micro doit donc savoir à tout instant s'il a lancé un ordre au disque. Un exemple; si le micro lance un ordre de lecture, il va attendre la fin de cet ordre pour réaliser une autre tâche. Pendant la durée de cet ordre de lecture rien ne pourra venir interrompre ce dialogue entre le micro et le contrôleur.

Il n'est pas nécessaire d'envoyer un signal au micro lors d'une erreur de la part du disque ou du contrôleur. Il n'est pas nécessaire, par exemple, d'utiliser un signal d'interruption pour alerter le micro d'une erreur. Pour détecter une erreur, il suffit en effet, au microprocesseur de lire l'état du contrôleur avant, après, ou éventuellement pendant une opération. Cela permet d'éviter d'utiliser tout un matériel supplémentaire sans améliorer les performances.

Donc, s'il n'est pas nécessaire de signaler directement les changements d'état du contrôleur, il ne reste juste qu'à synchroniser le microprocesseur avec le CD, de manière à éviter leur accès simultané à la mémoire buffer. Ce moyen de synchronisation est réalisé par le signal "Halt". Ce signal est très pratique et économique car, il arrête le micro juste après la fin de l'instruction courante et n'a pas besoin de perdre du temps en sauvegarde des registres.

En prenant l'exemple d'une opération de lecture ; le microprocesseur enverra, premièrement les paramètres du contrôleur et ensuite activera l'USC démarrant ainsi l'opération de lecture. Après son activation l'USC indiquera au micro qu'elle est occupée par le bit d'état du contrôleur. L'USC arrêtera le micro (Halt) seulement quand le contrôleur sera sur le bon secteur. Avant cela le micro peut continuer ses travaux en préparant, par exemple d'autres requêtes mais il ne pourra pas les lancer tant que le bit d'état du contrôleur l'indique occupé. Quand le contrôleur arrive sur le bon secteur, l'USC arrête le micro et effectue l'opération de lecture. A la fin, elle libère le micro, en même temps qu'elle change l'état du bit occupé. Le micro saura lorsqu'il viendra lire le bit d'état , que l'opération est finie. A ce moment il pourra aller chercher une prochaine opération dans sa file d'attente (Fig VIII.13)

L'USC, après avoir réalisé une opération, se met en attente active jusqu'à son prochain réveil par le microprocesseur.

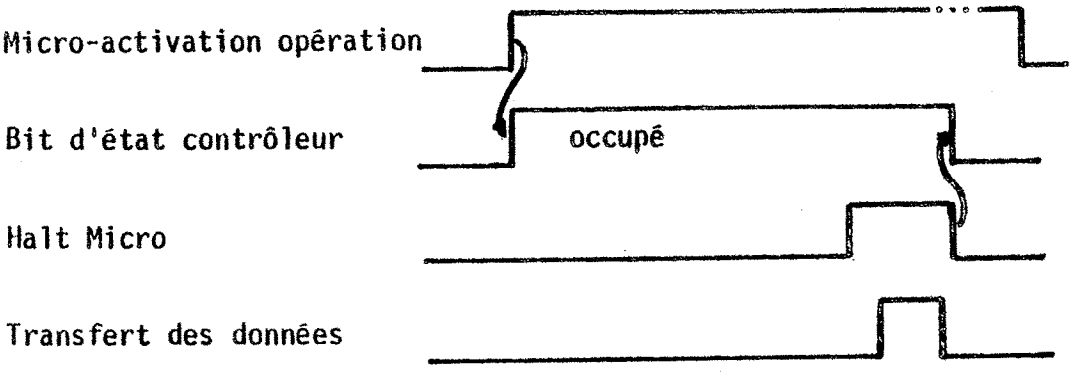


Fig VIII.13 - Utilisation du signal Halt

Ce type d'emploi du signal de Halt est celui qui économise le plus de temps d'exécution du micro et qui donne les meilleures performances. S'il n'y a pas de contraintes de temps pour le micro, il peut être arrêté à l'activation de l'opération et à ce moment il ne continuera pas à traiter d'autre requêtes. Il sera libéré juste à la fin du transfert lors de la libération du signal de Halt. Dans ce type de technique le micro n'est dédié qu'à une opération à la fois et à la fin de celle-ci il reprend son activité en relançant par exemple une autre opération. Dans ce cas, il n'y a pas besoin de vérifier le bit d'état du contrôleur (Fig VIII.14).

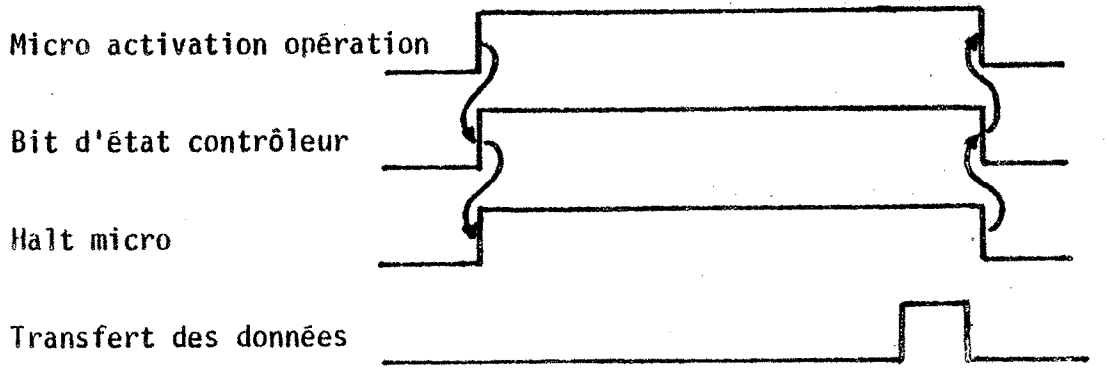


Fig VIII.14 - Utilisation du Halt par une technique plus simple

VIII.16 UNITE DE SYNCHRONISATION

La synchronisation est l'action du contrôleur disque de recherche d'un profil (configuration binaire) déterminé et fixe qui indique le but des données valides. C'est le moyen utilisé par le contrôleur pour se synchroniser sur les données, lui permettant de connaître automatiquement la position des premiers octets valides. Cette action étant rapide (10 MHz), elle ne peut pas être accomplie par le processeur en tranche. Il est nécessaire d'avoir une logique auxiliaire : un automate, réalisé avec des circuits TTL Schottky [T. TEX.77].

Dans la section VII.6 a été présente le format du profil de synchronisation, il est composé de 11 octets à Zéro suivis d'un octet avec tous ses bits à uns. La logique de détection est assez simple et consiste à compter le nombre de bits de données à zéro jusqu'au moins 88 à partir de cet instant la détection de l'octet à un est autorisée. Ce compteur de Synchronisation est remis à zéro (reset) par toute arrivée d'un "un" incorrectement placé, et provoque la réinitialisation du comptage des zéros. Les signaux du disque (les bits) passent à travers l'USD avant d'arriver au compteur, ce qui fait que lorsque le profil binaire est trouvé, l'octet de "uns" est déjà dans le circuit de décalage de l'USD; Il suffit donc, à partir de l'autorisation du compteur d'attendre la détection de l'octet à "un" dans le circuit de décalage et de signaler à l'USC que le profil est trouvé. (Fig. VIII.15)

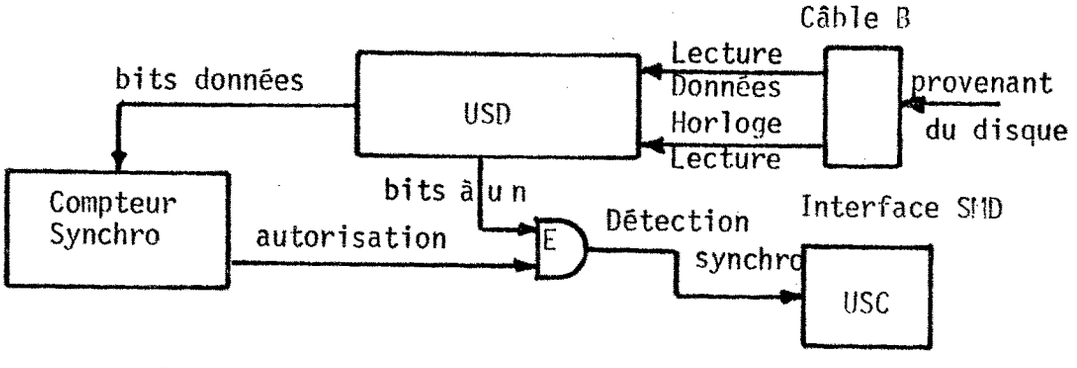


Fig VIII.15 - Unité de synchronisation

VIII.17 ARCHITECTURE DE PROCESSEURS

VIII.17.1 Introduction

Cette section présente l'architecture matérielle employée pour le processeur en tranche. Le but était de faire fonctionner les tranches le plus vite possible, de manière à ce qu'elles puissent se synchroniser sur les horloges des unités disques de 10 MHz. Le problème était lié au fait que les circuits en tranches (bit slice) ont des vitesse maximales de l'ordre du 10 MHz, ce qui additionné au temps d'accès des mémoires de microprogramme et au temps de traversée des registres, donnait des valeurs de temps de cycle de l'ordre de 200 ns. Il y avait intérêt donc à employer des architectures à haute vitesse, de manière à ce que le séquenceur puisse trouver à 100 ns et émettre ses ordres synchronisés avec les unités disque.
Cette section présentera d'abord les architectures courantes pour ensuite aborder les plus rapides.

VIII.17.2 Organisation de la partie contrôle des machines

Normalement toute machine (CPU) exécute une instruction en deux temps principaux "Fetch" et "Exécuté". Dans le cycle d'une instruction, le CPU lit l'instruction de la mémoire, la décode (Fetch) et après l'exécute (Exécute). Cette exécution prend un double sens car une partie du code instruction est utilisé pour la réalisation de l'opération dans l'unité arithmétique, pendant que l'autre partie du code permet de calculer l'adresse de la nouvelle instruction à exécuter. Comme les programmes sont écrits séquentiellement et qu'ils sont stockés en ordre dans la mémoire,

l'adresse de la prochaine instruction est normalement la suivante à moins qu'il n'y ait un branchement.

En juxtaposant les temps "Fetch" et "Exécute" l'un après l'autre on obtient des architectures à deux barrières temporelles. La Fig VIII.16 présente une de ces architectures avec comme barrière un registre d'instruction

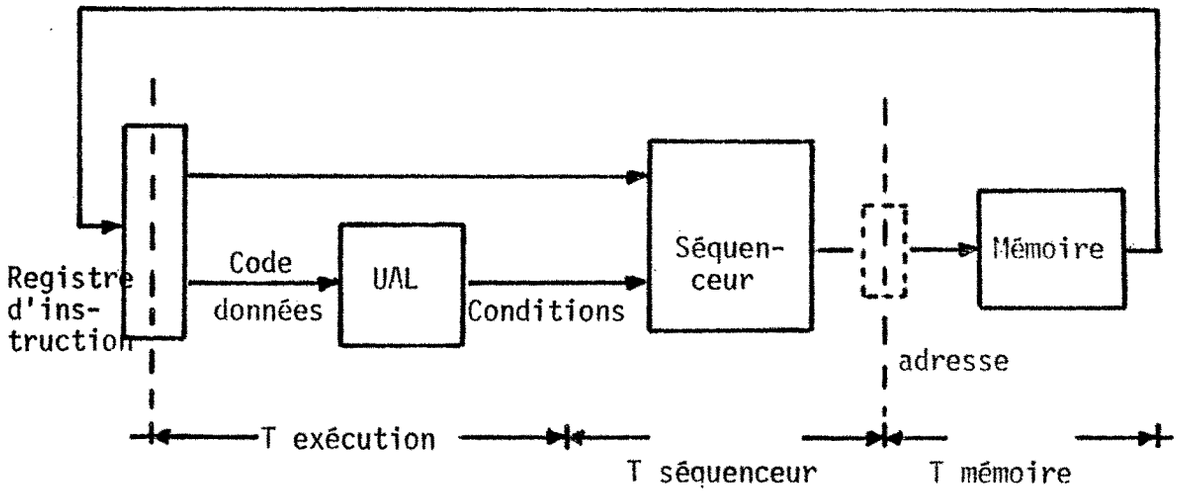


Fig VIII.16 - Architecture simple avec Registre d'Instruction

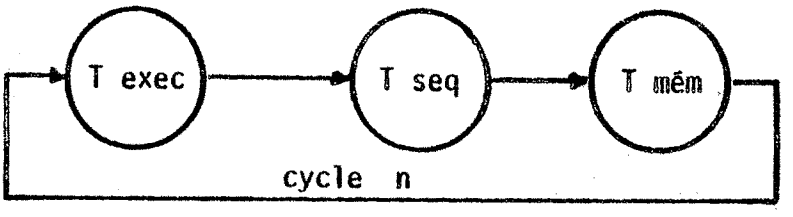
Cette architecture ne possède aucun parallèlisme et le temps d'un cycle d'instruction est l'addition des trois temps :

T cycle = T exec. + T seq. + T Mem

Cela donne des instructions du type

- Comparer A avec B et saut si égaux

Son avantage est de pouvoir réaliser en un cycle un branchement, fonction des résultats de l'exécution de l'opération dans l'UAL. Son désavantage est la longueur du cycle et donc de donner une machine lente. En utilisant un diagramme de Pert on montre l'enchaînement des opérations [ANC.76] :



Deux autres architectures simples sont possibles en changeant de place la barrière temporelle, le registre : l'architecture avec registre d'adresse (Fig VIII.77) et celle avec registre de données et conditions (Fig VIII.18)

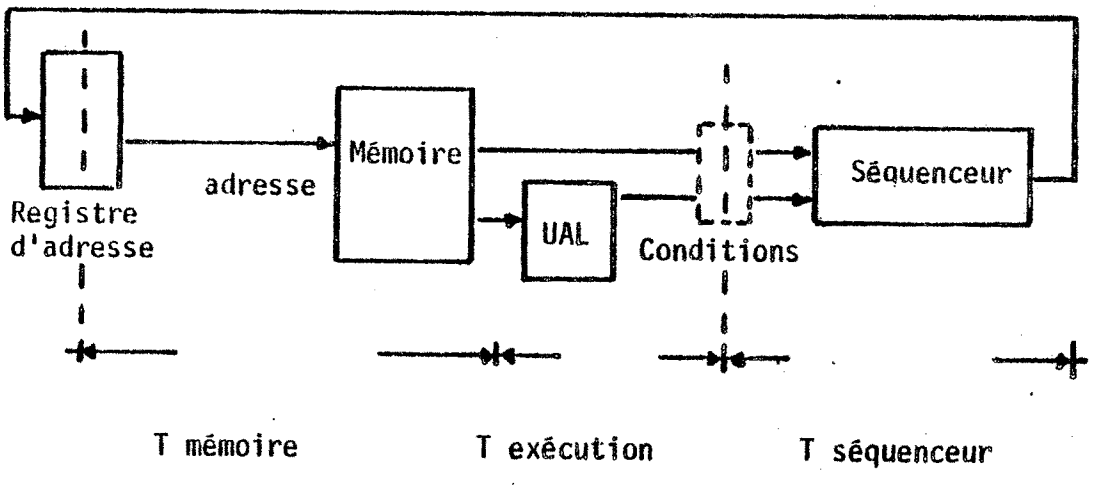


Fig VIII.17 - Architecture simple avec Registre d'Adresse

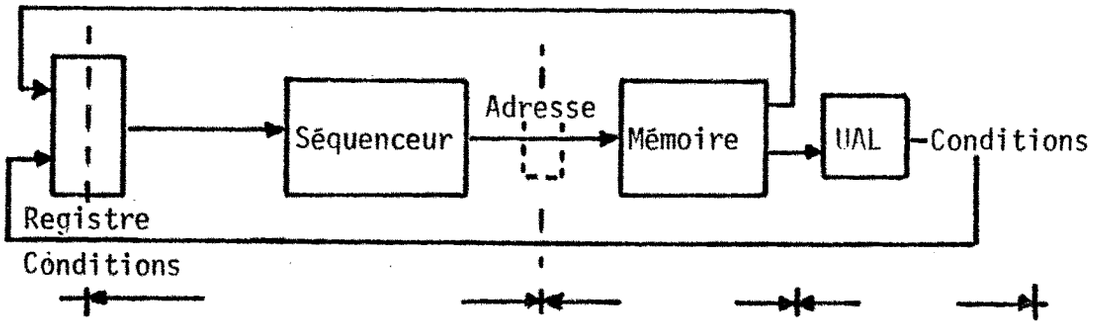


Fig VIII.18 - Architecture simple avec Registre de Conditions

VIII.17.3 Machine à un niveau de parallelisme

Le fait que les programmes étaient structurés de manière séquentielle dans la mémoire et qu'ils étaient donc exécutés en ordre sauf en cas de branchement, a amené il y a 15 ans les chercheurs à adopter une autre architecture. Il n'y avait pas intérêt à attendre par chaque instruction le résultat des conditions de l'UAL, car le taux d'instructions de branchement n'était pas grand. Cela a amené à l'architecture de la Fig. VIII.19 où la partie opérative est commandée en parallèle avec la lecture de la mémoire d'instructions et le séquencement. C'est l'architecture utilisée actuellement dans plusieurs ordinateurs.

Ce type d'architecture peut avoir un Registre Adresse au milieu du cycle d'instruction, de manière à séparer la partie séquencement de l'accès à la mémoire.

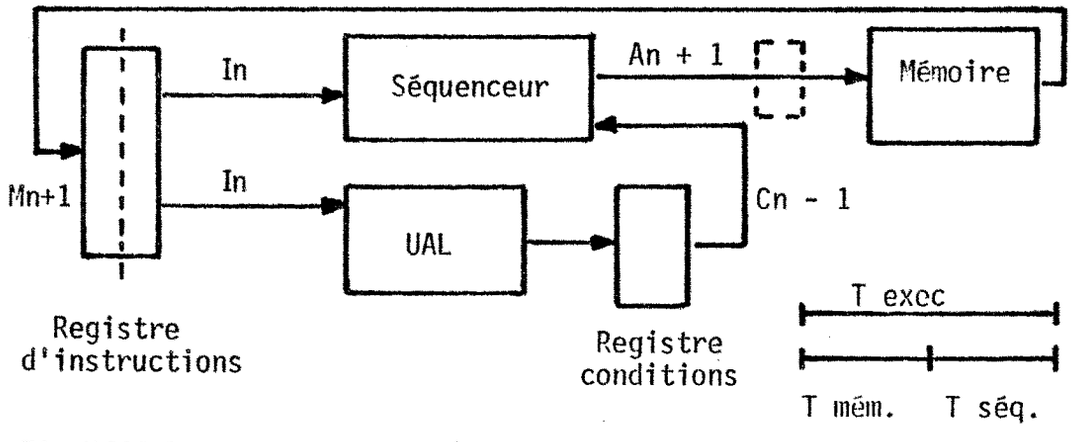
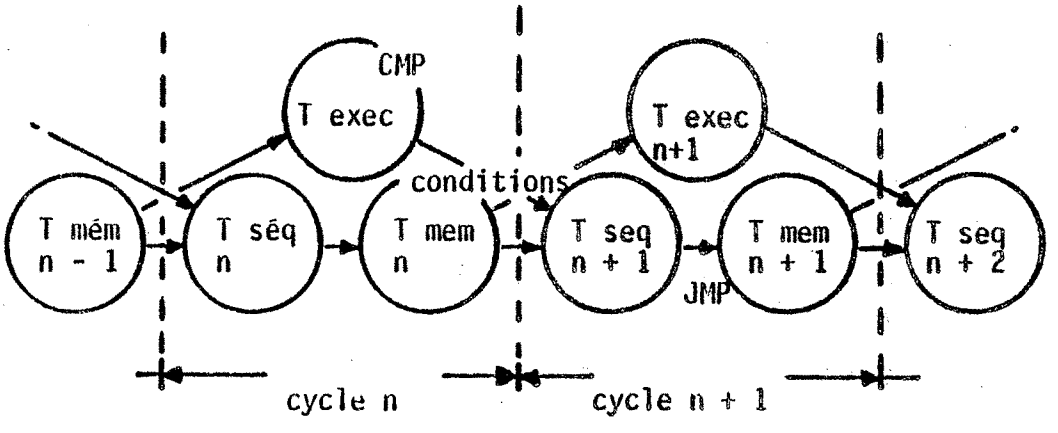


Fig VIII.19 - Architecture à un niveau de parallèlisme

De la mémoire sortent les instructions qu'iront activer en parallèle le séquenceur et l'unité arithmétique. Les résultats de l'opération de l'UAL, les conditions, seront prises en compte par la prochaine instruction pour calculer la nouvelle adresse. Cela donne des instructions du type :

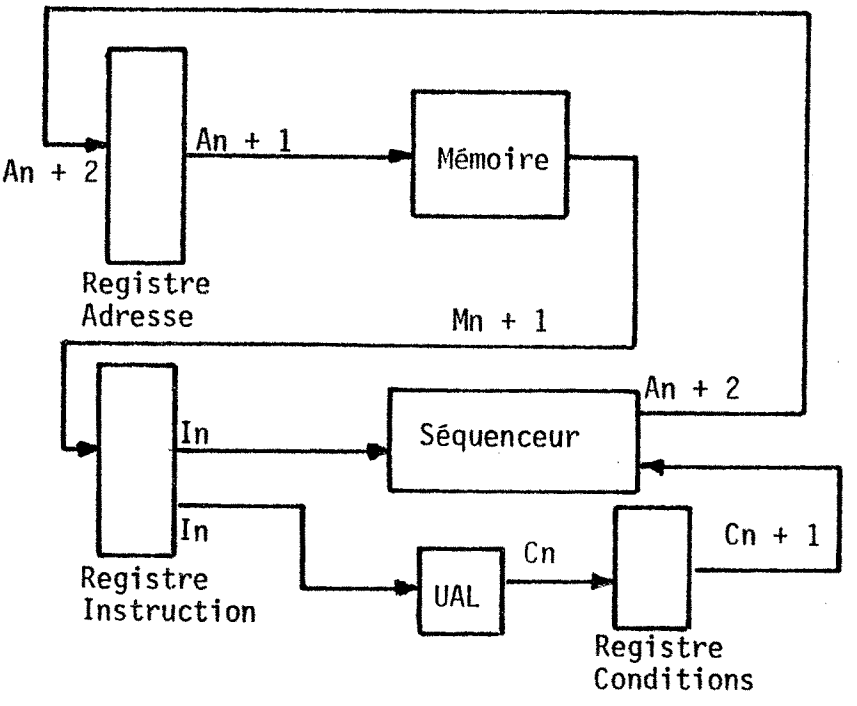
- comparer A avec B
- Branchement si égaux

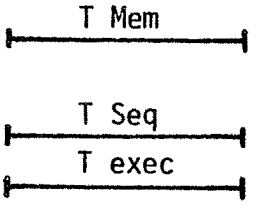
L'avantage de cette architecture est le gain de vitesse dû au parallèlisme entre le "fetch" et l'"execute". Le diagramme de Pert devient



VIII.17.4 Architecture à double parallèlisme

Comme il a été dit dans l'introduction de cette section, l'objectif pour le contrôleur disque est d'obtenir une architecture pour les processeurs en tranches qui puisse tourner avec un cycle de base de 100 ns. N'ayant pas obtenu de telles performances avec les autres architectures, il a été nécessaire d'en trouver une autre plus performante. La solution adoptée a été de mettre en parallèle les temps d'accès à la mémoire, de l'exécution des opérations (UAL) et du séquencement. L'architecture se présente comme dans la Figure VIII.20, les trois registres se changeant en même temps.

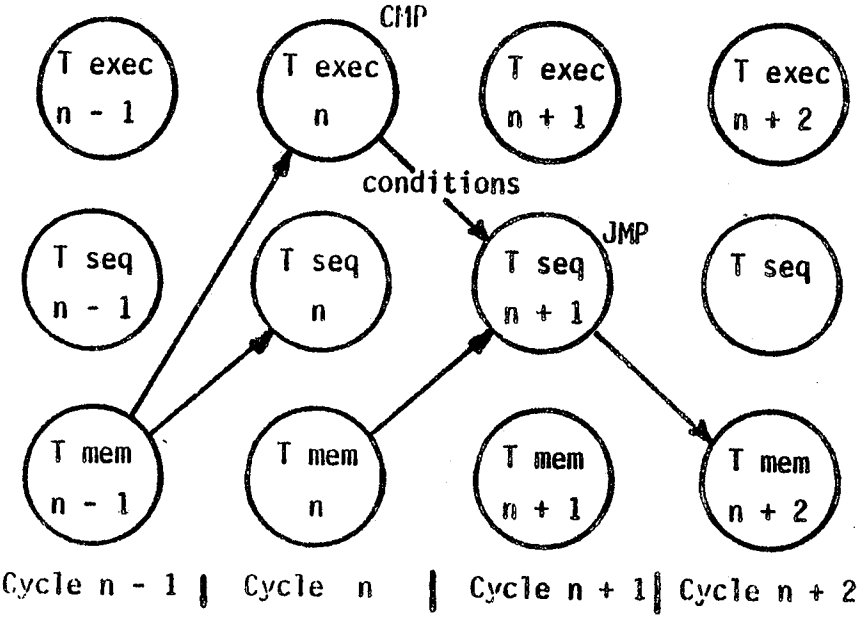




Cycle d'instruction

Fig VIII.20 - Architecture à double parallèlisme

Le resultat de ce triple parallèlisme donne le diagramme de Pert suivant:



Il y a maintenant un décalage de deux instructions entre le calcul et l'action du au test. Cela implique que le branchement sera réalisé deux instructions après le test, l'exemple précédent deviendrait

```
                                  Comparer A avec B
                                  Branchement si égaux
                                  NOP
Execution du Branchement  ------>
```

Un programme qui se déroule avec cette architecture se comporte d'une manière tout à fait semblable à ce qu'il ferait sur l'architecture précédente tant que ses instructions se déroulent séquentiellement, la différence apparaît lors des branchements où il est nécessaire d'attendre une instruction de plus pour les réaliser. Tout branchement conditionnel doit donc être suivi d'une instruction vide (NOP) ou d'une instruction qui n'ait pas de rapport avec le test.

Programmer avec ce type d'architecture n'est pas un problème mais plutôt une question d'habitude. Vu son parallèlisme cette architecture est intéressante pour des applications ou il y a besoin de grandes vitesses.

Quelques exemples de séquences d'instructions :

1) Pour un branchement inconditionnel qui ne dépend pas des conditions de l'UAL, l'instruction de branchement (JMP) doit être écrite une instruction avant pour minimiser le temps en prévoyant déjà son décalage.

```
JMP  LOIN
ADAB
```

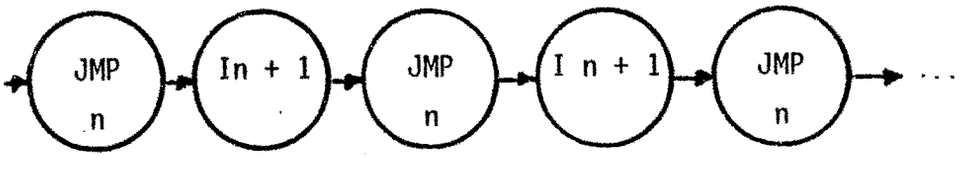
Le saut s'exécutera après l'exécution de l'instruction d'addition (ADAB).

2) Une boucle d'attente sur une instruction avec test du type :
JMP * if bit non zéro devra être écrit

```
      JMP ICI
ICI   JPM ICI if bit not clear
```

Cela permet à l'instruction de boucler sur le test.

Sans le premier JMP, le déroulement du programme se ferait de la manière suivante :



VIII.18 ARCHITECTURE DES PROCESSEURS EN TRANCHE DU CD

C'est cette dernière architecture qui est présentée ici (à double parallèlisme) qui a été choisie pour les processeurs en tranche du contrôleur disque. Ce choix a permis de faire tourner le sequenceur à la même vitesse que les unités disque (10 MHz) autorisant ainsi un contrôle plus direct des actions (Fig VIII.21)

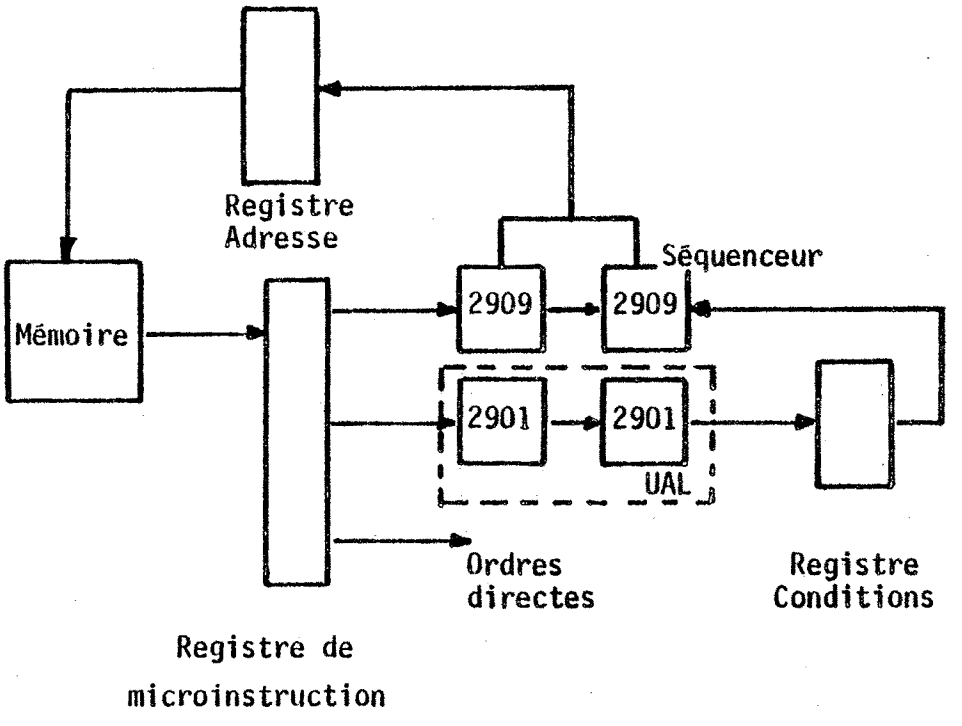


Fig VIII.21 - Architecture des processeurs en tranches du CD

Cette architecture est composée par une mémoire de 256 mots de 44 bits. Le registre de microinstruction a par conséquent 44 bits qui sont répartis en trois champs, le champ des ordres au séquenceur (2909), le champ des ordres à l'unité arithmétique (2901) et celui des ordres directs. Ces ordres iront contrôler directement des circuits ou seront utilisés pour informer d'autres éléments de l'état du contrôleur. Deux circuits 2901 composent

l'unité arithmétique (l'UAL) qui travaillesur un octet. Les deux circuits 2909 font partie de l'USC et calculent les adresses pour la mémoire de microprogramme. Chacun travaille sur 4 bits ce qui totalise 8 bits, cela permet d'adresser une mémoire de 256 mots. Le registre d'adresse aura pas conséquent 8 bits. Le registre de conditions, garde certains bits du résultat des calculs de l'UAL et d'autres unités du contrôleur, pour qu'elles puissent être utilisées par le séquenceur pour déterminer la prochaine adresse.

VIII.19 UNITE DE SEQUENCEMENT ET CONTROLE-USC

L'Unité de Séquencement et Contrôle, USC, est l'organe le plus important du contrôleur. C'est lui qui, à travers ses micro-programmes, commande toutes les actions qui concernent les données : opération de lecture, d'écriture, de formatage, de vérification et d'opération de recherche de profil à la volée . L'USC est composée de deux circuits séquenceurs en tranches 2909 [T.AMD.78] de la mémoire de microprogramme et les registres et des circuits auxiliaires (Fig VIII.22).

Les séquenceur 2909 calculent l'adresse de la prochaine micro instruction pour le déroulement de la séquence des microprogrammes. Ils sont câblés de manière à travailler sur deux types d' instructions : l'incrémentation d'adresse et le saut (JMP). Si le code de l'instruction est incrémenter l'adresse, le séquenceur le fera automatiquement et le programme se déroulera en séquence. Si l'instruction est le "saut" il pourra être conditionnel ou non, les conditions étant représentées par des signaux provenant d'événements spécifiques, tels que, le signal de "synchronisme réalisé".

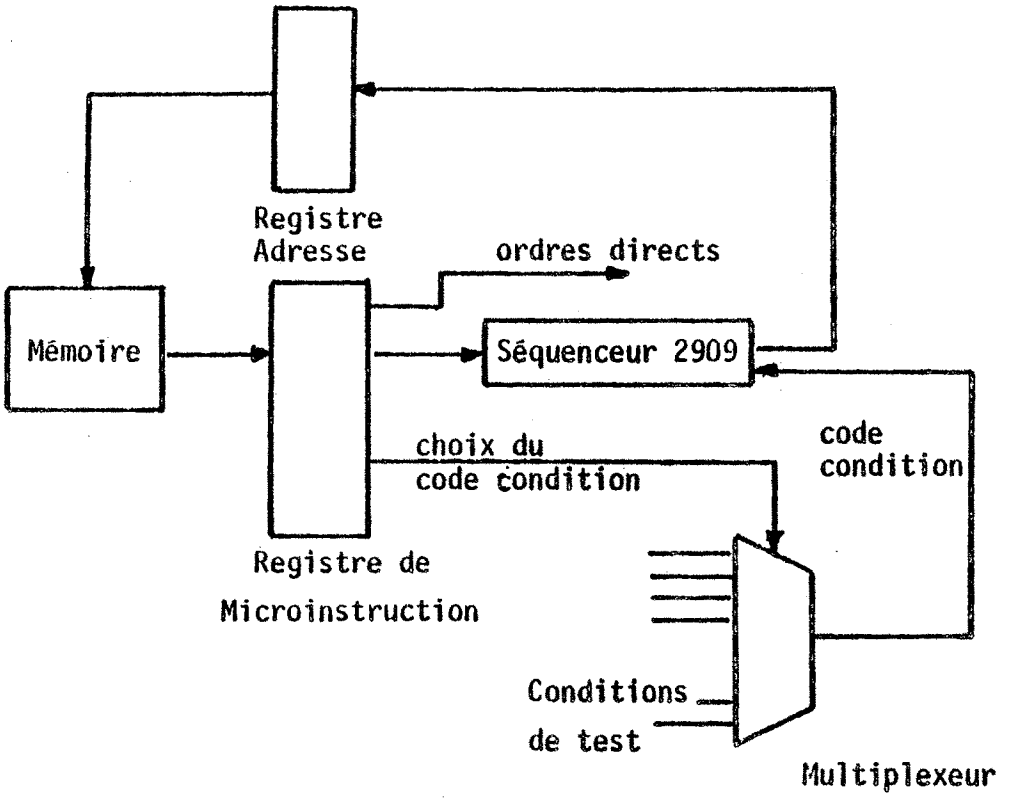


Fig VIII.22 - L'unité de Contrôle de Séquencement.

Le choix est retombé sur ces deux instructions car pour le contrôleur disque avec ses microprogrammes fixes, il ne s' avérait pas nécessaire d'utiliser toutes les possibilités du 2909, comme l'instruction "Saut sous routine" (JSB).

La plupart des ordres qui émanent de l'USC proviennent directement du décodage des bits de la mémoire de microprogramme à travers le Registre de Microinstructions. Le retour de ces ordres est représenté par les signaux d'état qui arrivent au circuit séquenceur du 2909 et qui donnent les conditions de saut.

La mémoire de microprogramme possède tous les programmes nécessaires pour l'exécution des diverses opérations primitives du contrôleur. Ses 256 mots représentent la capacité maximale

que peut adresser les 8 bits des deux circuits séquenceurs (2909). Le démarrage du séquenceur est réalisé grâce à un signal d'activation provenant du microprocesseur. Ce signal forcera le séquenceur à commencer à l'adresse "00".

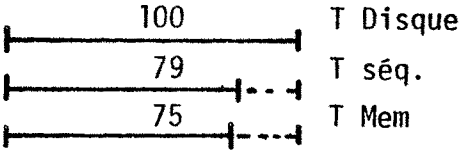
VIII.19.1 Temps de propagation dans l'USC

En considérant chaque bloc de l'architecture de l'USC on obtient pour les temps de propagation :

T MEM = t reg. adr . + t mem.+ t set up reg. instruc. = 10 + 60 + 5 = 75 ns

T SEQ = t reg. instr. + t seq. 2909 + t max + t set up reg adr = 10 + 55 + 9 + 5 = 79 ns

Cela montre qu'avec cette architecture l'USC peut travailler en dessous du 100 ns. Le temps total mémoire est composé du temps de traversée du registre d'adresse plus le temps de traversée de la mémoire et du temps de chargement du registre de microinstruction. Le temps total du séquenceur, du 2909, est composé du temps pour exécuter son opération la plus longue, du temps de traversée du multiplexeur, du temps de chargement du registre d'adresse et du temps de traversée du registre de microinstruction.
Tous ces temps ont été calculés avec les valeurs maximales du fabricant et pour les conditions extrêmes de fonctionnement. Dans des conditions normales, ces temps sont très inférieurs à ces valeurs. Comme la somme est toujours inférieure à 100 ns, il n'ya a pas de problèmes pour faire tourner l'USC à la vitesse des unités disque permettant ainsi le contrôle direct des signaux provenant des disques.



VIII.20 UNITE ARITHMETIQUE ET LOGIQUE

Cette unité construite autour de deux processeurs arithmétiques en tranches 2901 [T.AMD.78]exécute principalement les opérations de comparaison et de comptage. Son fonctionnement est complètement sous le contrôle de l'USC et ses données proviennent de la Mémoire de Microprogramme, du microprocesseur, à travers le bus données, ou alors du disque à travers de l'USD (Fig VIII.23) L'UAL travaille sur des mots de 8 bits.

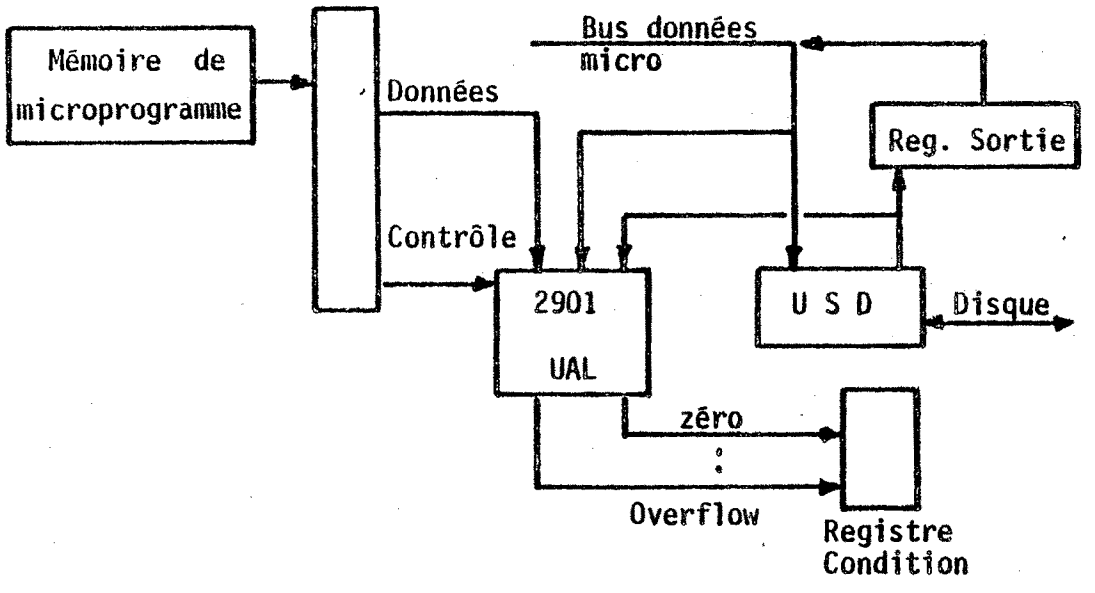


Fig VIII.23 - L'unité Arithmétique et Logique

Les valeurs immédiates proviennent directement de la mémoire de microprogramme. Pour réaliser une comparaison avec un octet provenant du disque, il est nécessaire pour l'UAL de lire directement la sortie de l'USD (désérialisateur). Pour lire des paramètres et des données, tels que le descripteur secteur à vérifier, le profil à trouver et d'autres, l'UAL devra les prendre

directement dans la mémoire buffer, à travers un mécanisme semblable au DMA pour les données du disque. Ces paramètres sont ensuite rangés dans la petite mémoire du 16 mots des 2901.

VIII.20.1 Temps de propagation dans l'UAL

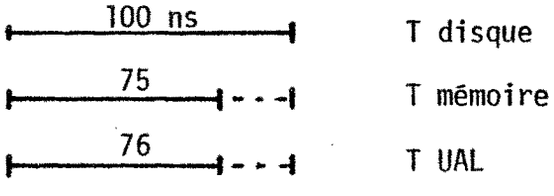
Les temps de propagation des signaux de l'UAL sont calculés avec les données du fabricant sur les circuits 2901 B. Les résultats pour des conditions normales d'utilisation sont :

T MEM = t reg adr + t mem = t set up reg ins = 10 + 60 + 5 = 75 ns

T UAL = t reg ins + t op. + t carry + t set up reg adr = 10 + 39 + 22 + 5 = 76 ns

Ces temps sont tous inférieurs à 100 ns. Le temps total de lecture de la mémoire de microprogramme est le même que dans le cas de l'USC. Le temps total de l'UAL est composé du temps de traversée du registre de microinstruction du temps d'une opération jusqu'à la sortie du report (carry) du temps de traversée du deuxième processeur 2901 B depuis l'entrée report entrant (carry in) jus qu'à la sortie FO de test de bits à zéro et du temps de chargement du registre d'adresse. Ces temps sont toujours pris par rapport au chemin le plus long que peut parcourir un signal.

Ce calcul montre que l'architecture à double parallèlisme permet d'avoir des temps assez courts tant pour l'USC que pour l'UAL



Cependant, pour l'UAL si les temps employés sont les valeurs maximales du fabricant, il apparaît quelques différences

TUAL max = T reg instr + top max + t carry max + t set up reg adr = 10 + 59 + 37 + 5 = 111 ns

Ce temps est supérieur au 100 ns de l'horloge du disque et donc les deux processeurs 2901 B ne pourraient pas tourner. Toutefois comme ces temps sont des valeurs maximales donc en fonctionnement normal l'UAL doit pouvoir travailler. Mais comme le moindre aléa dans l'exécution d'une opération peut amener à avoir des temps se rapprocahnt des valeurs maximales et que la fréquence des unités disque peut avoir un décalage de 1 03, 3 $\pm$ 10 ns, il a été nécessaire d'envisager des solutions pour pallier à ces problèmes. Ces solutions sont présentées dans les deux prochaines sections.

VIII.21 EXECUTION D'OPERATIONS PLUS LONGUES QUE L'HORLOGE

Les problèmes posés par certaines opérations d'une durée supérieure au cycle d'horloge et par l'existence de temps de propagation pour certaines unités inférieures à 100 ns, nous ont amené à développer certaines techniques.

Le temps total d'une opération T_{TOT} est égal à la somme des temps de plusieurs opérations (dans le sens général de propagation)

$$T_{TOT} = Tt1 + Tt2 + Tt3 + \ldots\ldots + Ttn$$

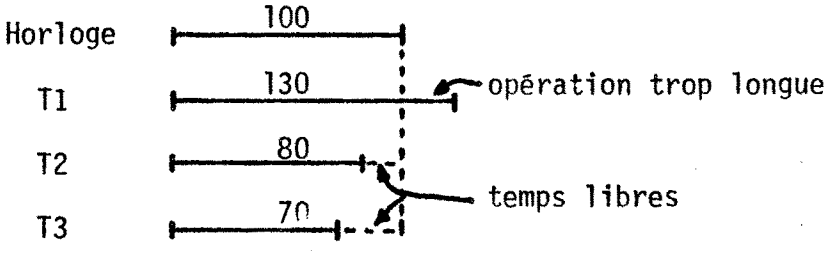
T_{TOT} représente le temps total pour l'exécution d'une instruction et les Ttn, les temps de divers cycles qui la composent. Chaque cycle représente une des diverses petites opérations nécessaires pour réaliser une opération. Normalement dans une machine, les cycles

ont les même durées et sont égales au cycle de l'horloge, d'où :

$$T_{TOT} = n\ Ttn$$

Lorsque les opérations ont des durées inférieures à l'horloge qu'elles utilisent, il en résulte des temps libres (sans opération) Le motif qui amène à utiliser la même horloge pour tous les cycles est d'ordre économique, ainsi que de simplicité. Il est en effet plus simple d'avoir des machines avec une horloge unique et fixe.

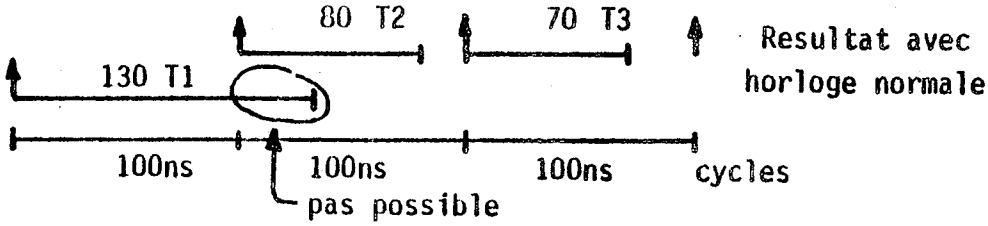
Pour mieux expliquer la technique d'utilisation de cycles de durée différentes, prenons un exemple. Supposonsun cas semblable au contrôleur disque, où une microinstruction nécessite trois opérations séquentielles pour la réaliser. Ces tâches ont des temps de 130 ns, 80 ns et 70 ns et l'horloge de base utilisée est de 100 ns. Apparemment la tâche de 130 ns ne pourrait pas être exécutée avec une horloge de 100 ns.

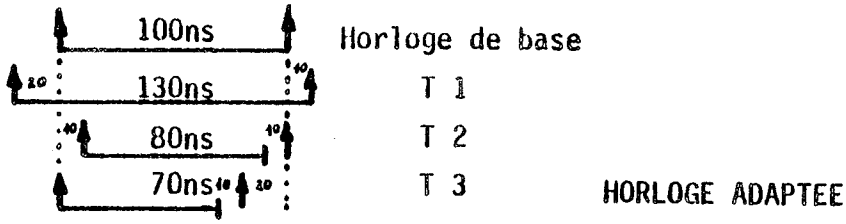


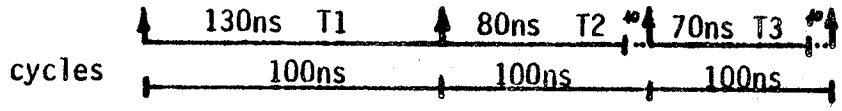
Cependant la somme des trois tâches donne 280 ns ce qui est inférieur à trois cycles d'horloge (300 ns)

$$T_{TOT} = T_1 + T_2 + T_3 = 130 + 80 + 70 = 280\ ns$$

La technique est donc de ne pas utiliser l'horloge telle qu'elle est est, mais de l'adapter légèrement aux besoins des opérations. Avec l'horloge adaptée les cycles seront variables, mais le total de leurs temps sera toujours égal à un multiple de l'horloge de base, dans ce cas 300 ns. C'est le concept d'utiliser une horloge floue.







Cela montre que des opérations de durée légèrement supérieure à l'horloge, peuvent être exécutées sans problème, dans la mesure où d'autres opérations ont des temps inférieurs à l'horloge. Ces horloges adaptées peuvent être obtenues à partir de l'horloge de base, à travers de petits circuits de retard. Cela n'est valable que pour de petites différences de temps.

Cette technique a été utilisée dans le CD du PBD-MAGE pour faire un léger ajustement pour le temps d'opération de l'UAL.

VIII.22 DOUBLE MICROINSTRUCTION

La section précédente, VII.21, a présenté certains artifices qui permettent d'obtenir des meilleures performances pour l'architecture matériellle. Cette section présente des mécanismes logiciels de microprogrammation pour également améliorer ces performances.

Quand la plupart des microinstructions s'exécutent en un (1) cycle d'horloge et que juste quelques unes prennent un peu plus d'un cycle pour s'exécuter, faire tourner toute la machine sur une horloge ralentie seulement pour servir ces instructions entraîne une perte de performances. Cela peut être pallié par le ralentissement de l'horloge à l'occasion de telles instructions, mais cela oblige à avoir un matériel supplémentaire pour les décoder.

La solution qui paraît intéressante est de simplement doubler la durée des instructions longues par leur maintien pendant deux cycles.

Cela est possible à partir du moment où ces instructions sont exécutées d'une manière combinatoire. Par exemple, si la microinstruction CMPA (comparer A) ne peut pas être exécutée en un cycle, elle est simplement maintenue au prochain cycle, ce qui permet au circuit combinatoire d'élaborer son résultat.

```
CMPA
CMPA
JMP  si vrai
```

Toute la machine n'est pas pénalisée s'il n'y que quelques instructions longues. Cette technique est intéressante car la machine peut tourner à sa plus grande vitesse et le matériel n'a pas à être modifié.

VIII.23 SYNCHRONISME DES PROCESSEURS EN TRANCHE AVEC LES DISQUES

A l'occasion du projet du contrôleur disque, un des problèmes qui se posait était de savoir comment se synchroniser avec les unités

disque. Comment le contrôleur pouvait recevoir où émettre des octets de données d'une manière synchrone avec l'horloge du disque. Dans les sections précédentes, il a été montré que les unités disque employées, famille[CDC 9760][T.CDC.77] et la famille Ampex 900 [T. AMP.77] possèdent une horloge maîtresse provenant d'une piste pré-enregistrée en usine (Horloge Servo). Cela oblige le contrôleur à toujours écrire ou lire en synchronisme avec cette horloge.

Plusieurs solutions étaient viables, telles que d'avoir des circuits synchronisés toujours sur l'horloge des unités disque. Ces circuits auraient pour communiquer avec le reste du contrôleur des mécanismes tampons [LAU.79] tels que des mémoires FIFO (first in, first out). Ces mémoires auraient été le moyen utilisé pour absorber les différences de synchro entre les deux zones du contrôleur (Fig VIII.24)

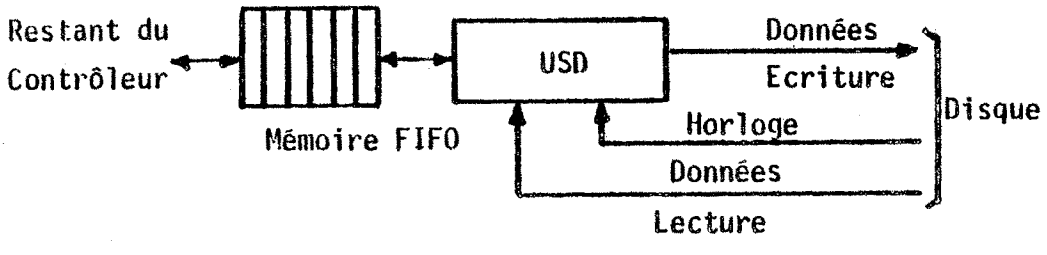


Fig VIII.24 - Synchronisation contrôleur/disque au moyen de FIFOs

Cette solution est assez utilisée par les contrôleurs, mais elle pose le problème que les circuits reliés au disque ont un contrôle lié à l'horloge du disque et donc semi indépendant du reste du CD.

Pour le contrôleur disque du PBD la solution de synchroniser directement, tout le contrôleur avec les horloges des unités disque

a été choisie. Cela permet à l'USC d'avoir un pouvoir sur tous les organes du contrôleur. Le CD dans ce cas est unifié tandis que dans la solution précédente il était divisé en deux, une partie de manipulation des données du côté du disque et une partie qui concerne la communication avec le processeur (Fig VIII.25)

Les processeurs en tranche de l'USC (2909) ainsi que ceux de l'UAL (2901) ont donc comme horloge celle provenant du disque. Cette solution, en plus de l'économie faite sur les circuits FIFO, donne une très grande liberté à l'USC pour activer directement les fonctions du CD comme par exemple l'unité de sérialisation/désérialisation (USD).

Le changement d'une unité disque à l'autre par le contrôleur, ne pose pas de problèmes dus à la perte momentanée de l'horloge des processeurs en tranches car lors d'une opération de recherche d'unité ou de cylindre, le processeur séquenceur, l'USC, est désactivé et dans un état du type de remise à zéro.

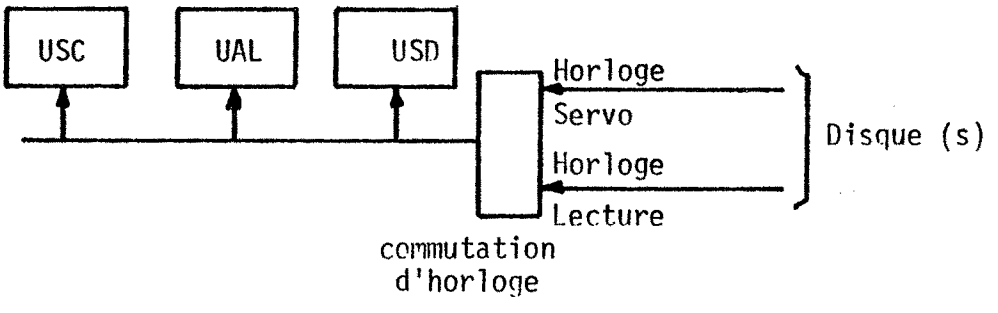


Fig VIII.25 - Synchronisation directe du contrôleur aux disques

Quand le CD passe d'une opération de lecture à une d'écriture, un circuit de commutation d'horloge (passage de l'Horloge Lecture à l'Horloge Servo) empêche qu'il y ait deux impulsions d'horloge trop rapprochées. Ce circuit élimine simplement la première impulsion de l'horloge d'écriture.

VIII.24 REGISTRE DE COMMANDE

Le registre de commande est le moyen qu'utilise le microprocesseur (processeur P2) pour indiquer au CD quelles est la fonction qu'il doit exécuter. D'après la valeur des bits de ce registre, l'USC ira choisir le microprogramme qui sera déroulé. Cinq microprogrammes peuvent être exécutés : la lecture, l'écriture, le formatage, la vérification et la recherche de profil.

Le choix du microprogramme est réalisé par l'USC en testant les bits du registre. A chaque bit correspondra une instruction de branchement (JMP) au microprogramme déterminé. Il est envisageable d'implanter une sixième fonction, donc un sixième microprogramme, qui permettrait par exemple, l'auto test du contrôleur (Fig VIII.27)

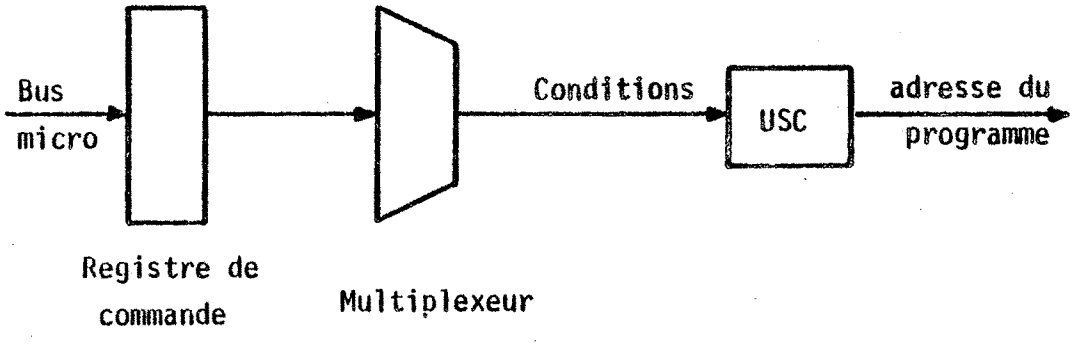


Fig VIII.27 - Registre de commande

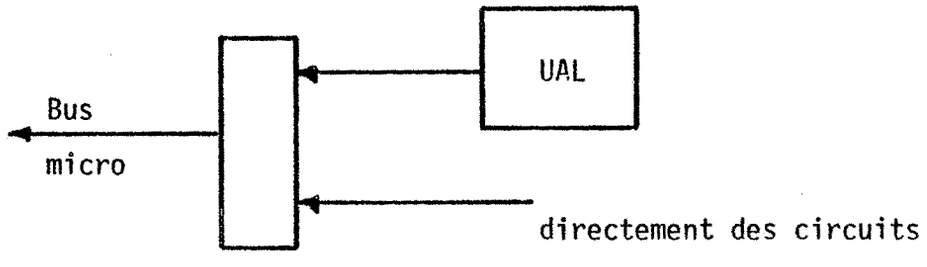
VIII.25 REGISTRE D'ETAT DU CONTROLEUR

De même qu'il existe un registre d'état pour les unités disque, il en existe un pour le contrôleur. Ce registre indique au microprocesseur quel est l'état du CD. Ces informations sont : contrôleur prêt, contrôleur occupé, vérification de descripteur secteur,

vérification de données, test du code correcteur, secteur abimé, secteur protégé.

Ces bits d'information du registre proviennent soit directement des circuits ou de l'UAL par exemple. Les bits "contrôleur prêt" et "contrôleur occupé" proviennent directement des circuits. Les autres bits viennent de l'UAL et sont le résultat d'un calcul ou d'une opération de celle-ci.

Par exemple le bit de vérification de donnée est le résultat de la comparaison des données du disque avec ceux de la mémoire buffer. cette comparaison est réalisée par l'UAL et s'il y a une différence l'USC la détectera et forcera l'UAL à générer une bit concernant l'erreur sur son bus de sortie. Ce bus est relié au Registre d' Etat qui lira l'information et la gardera jusqu'au moment où le microprocesseur la demandera.



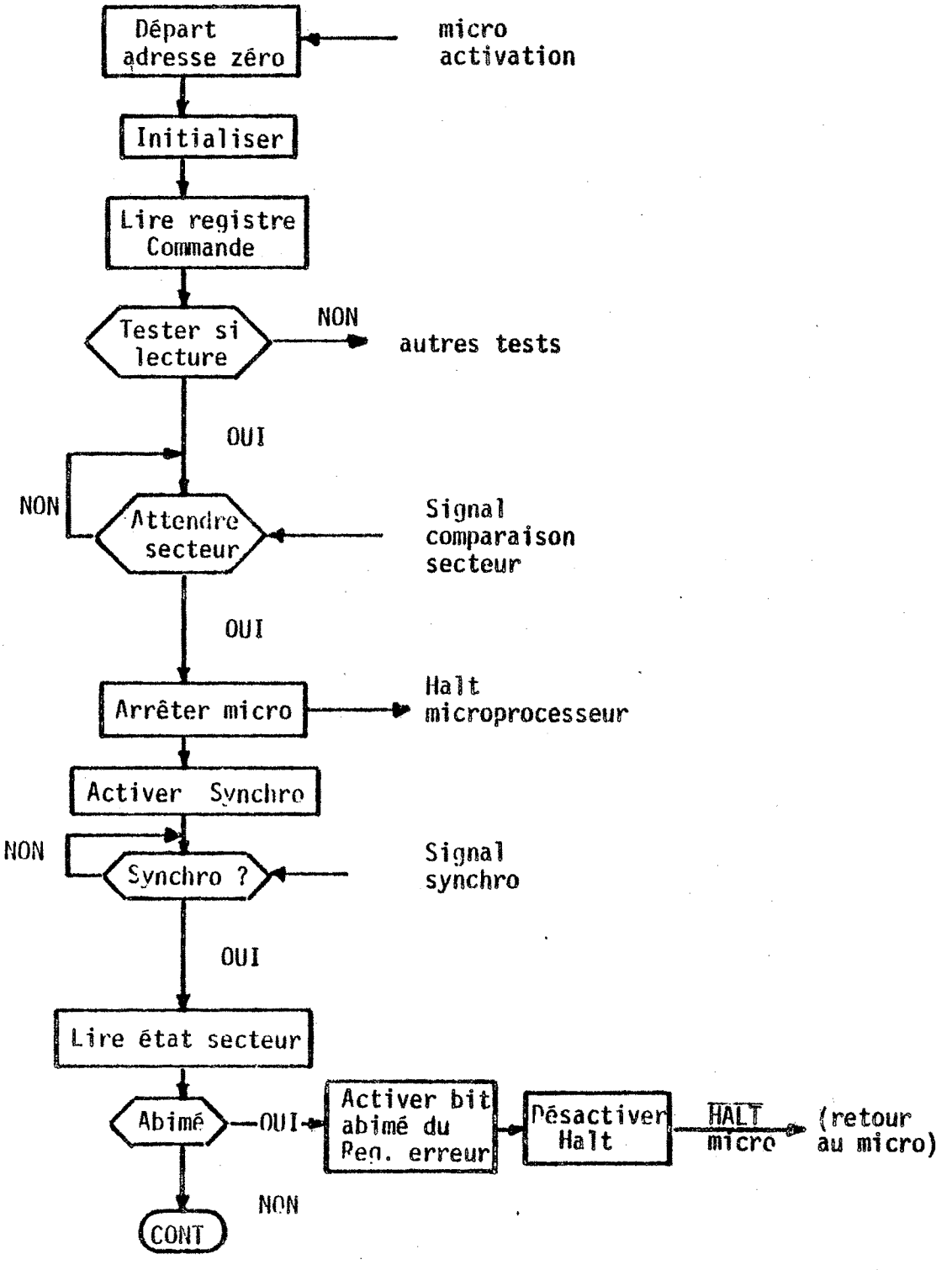
Registre d'état
du contrôleur

Fig VIII.28 - Registre d'état du contrôleur

VIII.26 ORGANIGRAMME D'UNE OPERATION

La lecture a été choisie comme exemple d'une opération primitive exécutée par le CD. Ces diverses tâches concernent juste le contrôleur

et non le micro, c'est en fait le déroulement d'un microprogramme par l'USC (Fig. VIII.29).



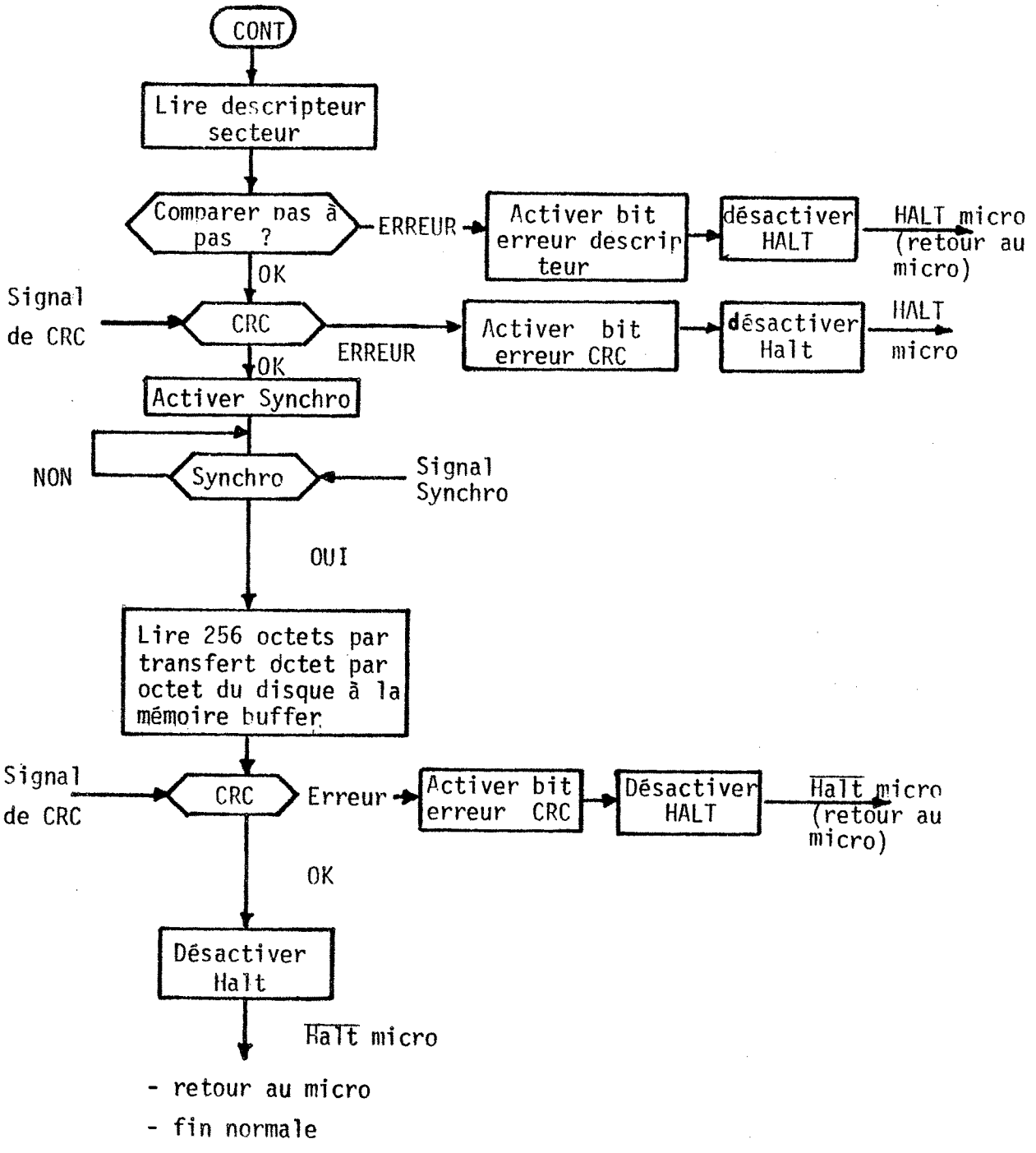


Fig VII.29 Séquence du microprogramme de lecture

Cette séquence montre clairement les principaux signaux qui communiquent avec l'USC. Les signaux de CRC, de Synchro et de Comparaison Secteur qui fournissent les conditions pour les décision de saut (JMP) du séquenceur 2909. Le signal de Halt est une sortie de l'USC et représente un ordre, une réponse de fin de travail, vers le micro. Le diagramme de la Fig. VIII.29 représente les principales opérations et signaux. Il est impossible de présenter tous les détails parce que le diagramme deviendrait incompréhensible.

L'opération de lecture commence par l'activation de l'USC par le processeur P2 (le micro), à partir de ce moment est déclenché le microprogramme. La première tâche de l'USC est de déterminer l'opération à exécuter, cela est fait en lisant le registre de commande. Pour cet exemple ce bit était celui du programme de lecture. Le prochain pas est d'attendre le signal de début du bon secteur, à ce moment l'USC bloque le microprocesseur par le signal de Halt en permettant ainsi qu'une opération de DMA soit exécutée. L'USC attend le signal de synchronisation et à son arrivée commence la lecture et la comparaison du descripteur secteur. L'adresse du secteur provenant du disque est comparée à celle stockée dans la mémoire buffer. La comparaison est réalisée octet par octet dans l'UAL et s'il y a erreur le bit d' Erreur Descripteur est activé et l'opération est abandonnée ce qui entraîne la libération du microprocesseur et sa reprise du contrôle. S'il y a erreur de comparaison ou non, l'USC attend le test du CRC pour déterminer si la comparaison a été fausse (due à une mauvaise lecture ou due à une erreur de positionnement de la tête sur le secteur).

Si la première étape s'est bien passée et que les têtes sont donc sur le bon secteur, l'USC passera à l'étape de lecture de données.

Il devra se synchroniser pour ensuite commencer la lecture des données, octet par octet, transférant du disque vers la mémoire buffer.

Pendant qu'un octet du disque est désérialisé, les tranches incrémentent le registre d'adresse pour préparer le transfert. L'octet prêt dans l'USD (désérialisation) est transféré vers le Registre de Sortie pendant que l'USD reçoit le prochain octet. Le mot (octet) dans le Registre de Sortie sera ensuite transféré vers la mémoire buffer sous le contrôle de l'USC à l'adresse indiquée par le Registre d'Adresse Mémoire. Pendant ce temps, dans l'UAL est réalisé le comptage du nombre d'octets transférés (voir section VIII.28).

A la fin du transfert de tous les octets (256), le CRC est vérifié par l'USC et s'il y a erreur le bit d'erreur CRC est activé. Si tout est correct l'USC libère le microprocesseur ($\overline{\text{Halt}}$) et se met en attente de la prochaine activation du micro.

La prochaine section donnera des détails sur les microprogrammes de transfert des données.

VIII.27 - SEQUENCES DE MICROPROGRAMMES

Cette section a pour but de présenter quelques séquences de microprogrammes écrites sur l'architecture du CD. Ces morceaux de microprammes montrent spécialement l'aspect de la synchronisation des unités disque avec le contrôleur pour le transfert des données.

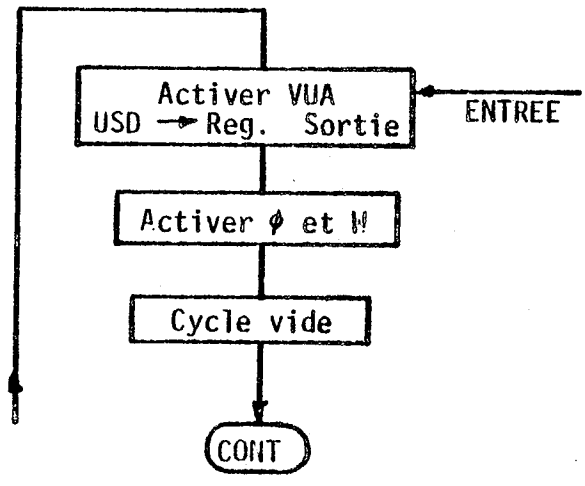
VIII.27.1 Cycle microprogrammé de lecture des données du disque

Quand l'unité de synchronisation annonce au séquenceur (USC) qu'elle a trouvé le caractère de synchro, celui-ci commence l'opération de transfert, par DMA, des données du disque vers la mémoire buffer. La boucle qui réalise ce transfert, exécute l'incrémentation du Registre Adresse Mémoire, pendant qu'un octet du disque est désérialisé. L'octet prêt dans le désérialisateur passera du Registre de Sortie et de là, toujours sous le contrôle du séquenceur, est rangé dans la mémoire buffer à la position donnée par l'adresse du Registre Adresse Mémoire. Le stockage des données dans la mémoire est réalisé grâce à la génération par le séquenceur, d'un signal de validation d'adresse (VUA), d'un siganl d'horloge (ϕ) et d'un signal d'écriture (w). Ces trois signaux sont tous générés par microprogramme (Fig VIII.30)

Cycles - JMP dehors si fin transfert

Sortie
Entrée

1 - Desactiver Validation, Horloge et Ecriture
2 - Incrémenter Registre Adresse Mémoire
3 - Activer Validation. Transfer USD vers Registre Sortie
4 - Activer Horloge et Ecriture
5 - NOP
6 - JMP Boucler
7 - Incrémenter le nombre d'octets transférés



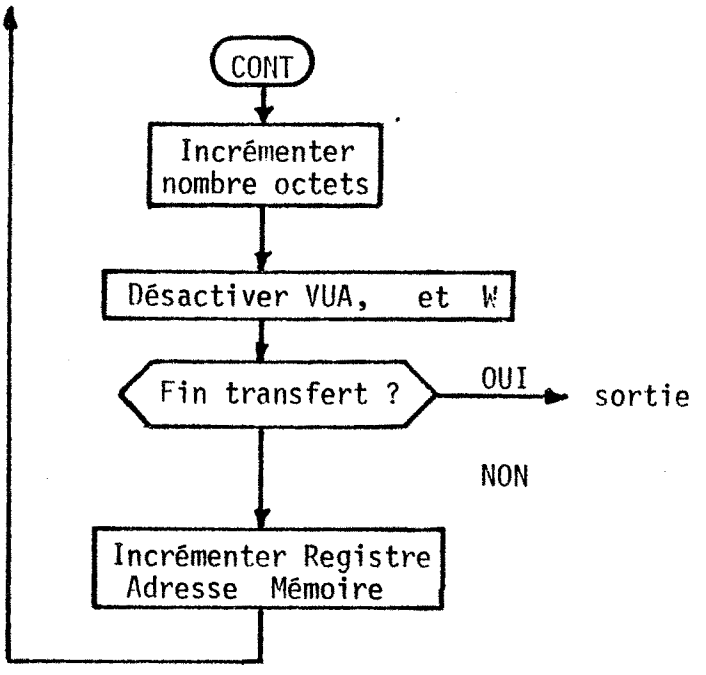


Fig VIII.30 - Microprogramme de lecture données disque

Cette boucle donne une bonne idée du mécanisme de transfert. Elle est composée de 8 microinstructions, soit 8 cycles, qui représentent le temps pour que 8 bits de données provenant du disque soient désérialisées dans l'USD. Le séquenceur n'a pas besoin de compter les bits d'un octet, car la boucle de microprogramme est par elle-même synchrone, elle représente ce temps (grâce à la synchronisation contrôleur/disque). Il restera seulement à compter le nombre total d'octets à transférer ; cela est fait par l'UAL et le résultat de ce comptage est utilisé par le séquenceur pour décider du branchement pour la sortie de la boucle.

Cet exemple montre bien l'emploi de programmes avec l'architecture à double parallèlisme, spécialment les instructions de saut (JMP).

VIII.27.2 Cycle microprogrammé d'écriture des données sur disque

Cette boucle est très semblable à celle de lecture. Pour le transfert

des données vers le disque, le premier octet doit être déjà chargé dans le sérialisateur (USD), pour qu'au signal de synchronisation , il démarre le transfert. La séquence d'écriture des données est réalisée par le transfert des octets de la mémoire buffer vers l'USD. Du sérialisteur les octets sont sérialisés et enregistrés bit à bit sur le disque (Fig VIII.31)

Cycles Sortie	0 - Stocker octet dans USD
	1 - Désactiver Validation, Horloge et Lecture
	2 - Incrémenter Registre Adresse Mémoire
Entrée	3 - Activer VUA
	4 - Activer Horloge et Lecture
	5 - NOP
	6 - JMP Boucle . Incrémenter nombre octets à transférer
	7 - JMP dehors si fin transfert

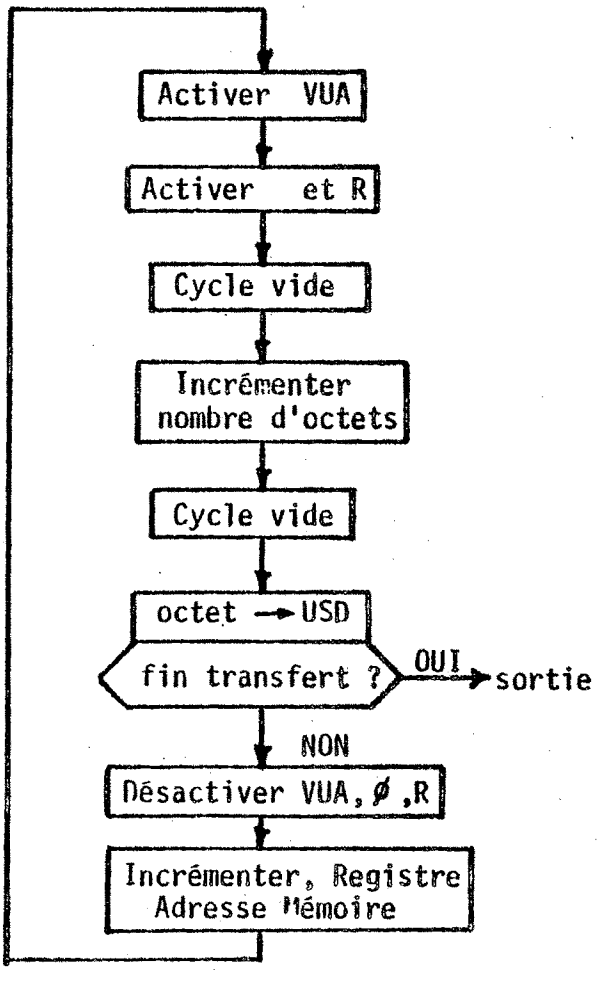


Fig VIII.31 - Microprogramme d'écriture des données sur disque

La boucle de 8 microinstructions permet au contrôleur comme dans le cas de la boucle de lecture, de rester synchronisé au mécanisme de transfert des données. Tout au lont de la boucle, pendant qu'un octet est sérialisé vers le disque, le séquenceur prépare un nouvel octet dans la mémoire buffer, pour transferer dans l'USD au moment donné. Les signaux de validation d'adresse (VUA), de lecture (R) et d'horloge (ϕ) sont utilisés dans la mémoire buffer pour controler le transfer des octets. Le comptage du nombre d'octets à transférer est réalisé par l'UAL et le résultat de ce comptage est utilisé par le séquenceur pour décider de la sortie de la boucle. Il faut observer que l'entrée et la sortie de cette boucle ne se font pas directement, il faut initialiser et finir le cycle. Par exemple, pour entrer dans la boucle le premier octet doit être préalablement socké dans l'USD.

Ce cycle de microprogramme d'écriture est par ailleurs réutilisé pour le formatage.

VIII.7.3 Cycle microprogrammé de vérification de données

Le microprogramme de vérification de données permet au contrôleur de lire les données d'un secteur du disque et de les comparer octet par octet avec les données dans la mémoire buffer. Dans une opération d'écriture/vérification, la vérification permet au contrôleur de s'assurer que les données d'un secteur ont été bien écrites.

L'opération consiste à lire simultanément les données de la mémoire et du disque et de les comparer au niveau de l'UAL du contrôleur. Ce type de comparaison est direct, au moment de la lecture du disque.

Cycles 0 - JMP sortie si fin de transfert
1 - Comparer octet UAL avec octet mémoire buffer
2 - Désactiver Validation, Horloge, Lecture.
JMP si comparaison fausse
3 - Incrémenter Registre Adresse Mémoire
4 - Activer Validation
5 - Activer Horloge et Lecture
6 - JMP Boucler. octet USD stocké en UAL
7 - Incrémenter le nombre d'octets à transférer.

Le contrôleur lit l'octet enregistré sur le disque à travers l'USD, et le garde dans la mémoire de travail de l'UAL.

En parallèle, les signaux de validation d'adresse (VUA), Horloge (ϕ) et Lecture (R) conjugués préparent la lecture de l'octet de la mémoire buffer. A un instant donné les deux octets sont comparés entre eux et s'il n'ya a pas de différence, l'opération de vérification continue cycliquement jusqu'au dernier octet du secteur. S'il y a différence le contrôleur sort de la boucle et le bit d'erreur est activé (Fig VIII.32).

Ce microprogramme est un exemple de la boucle de vérification, elle n'inclut pas les instructions d'initialisation ni celles de terminaison. Cette boucle est semblable à celle de recherche de profil ou la même opération de comparaison est utilisée, mais avec le profil stocké dans la mémoire de l'UAL.

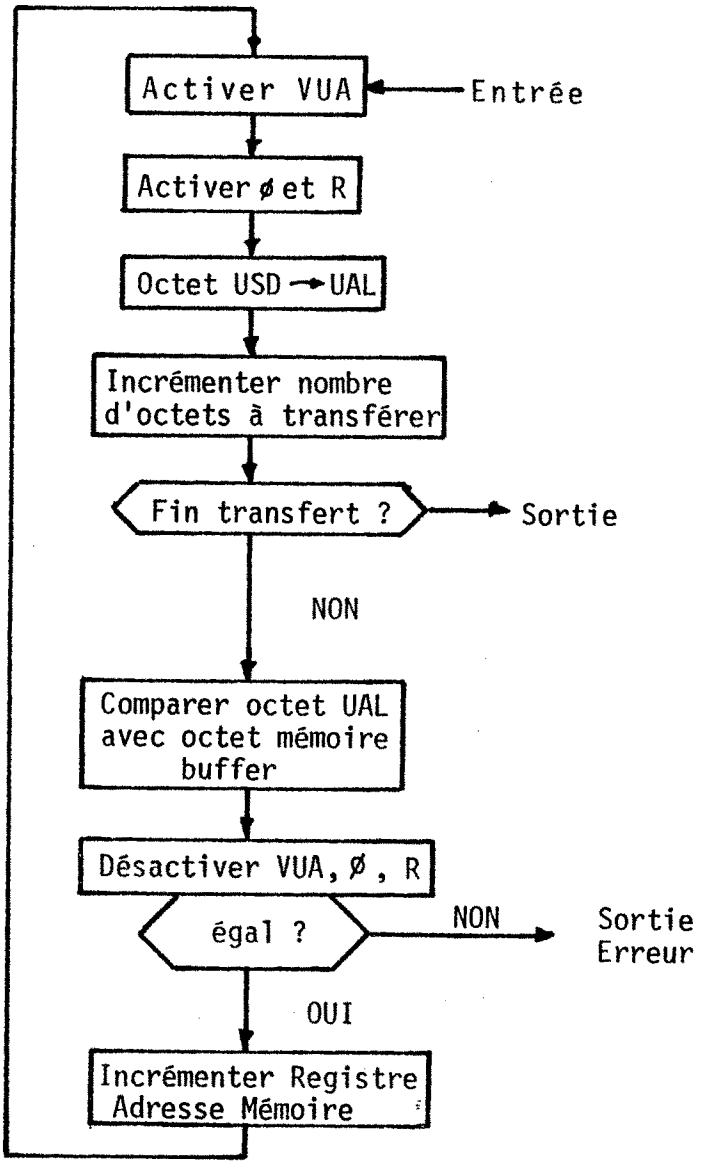


Fig VIII.32 - Microprogramme de vérification des données sur disque

VIII.28 ENTRELACEMENT DE SECTEURS

La section VII.3.11 a montré l'intérêt d'entrelacer les secteurs pour permettre au contrôleur de prendre son souffle entre deux secteurs, dans des opérations multisecteurs. Pour le CD du PDB-MAGE cette technique a été adoptée afin de simplifier les tâches

du contrôleur. L'entrelacement est de niveau 1, ce qui fait que deux secteurs d'adresse "n" et "n + 1" sont à des adresses physiques "i" et "i + 2". Pour des disques sectorisés en 64 secteurs l'ordre est le suivant, 0, 32, 1, 33 ...

La raison de l'utilisation de l'entrelacement pour le contrôleur du PBD est due au choix de simplifier les tâches multisecteurs et donc d'économiser du matériel. Si pour ces tâches la carte contrôleur devait assumer tout le contrôle, cela représenterait d'une part une augmentation de la taille des microprogrammes, donc de la mémoire morte ROM, et d'autre part, d'avoir une mémoire RAM auxiliaire pour garder toutes les variables à manipuler,et finalement plus de circuits. La mémoire interne de l'UAL de 16 mots ne serait pas suffisante. D'autre part, au niveau de la séparation des tâches entre le microprocesseur et la carte de contrôle, il y a intérêt à laisser seulement des opérations primitives au niveau du contrôleur, laissant au microprocesseur la responsabilité de gérer les fonctions plus complexes.

Par exemple, avec cet entrelacement, lors d'une opération de Recherche de Profil sur plusieurs secteurs, après chaque secteur le contrôleur rend le contrôle au microprocesseur, qui vérifie le résultat de la recherche sur ce secteur et réinitialise éventuellement la recherche sur le prochain secteur si le profil n' a pas été trouvé. Dans une configuration de 64 secteurs par piste, le micro-processeur aura donc un temps de l'ordre de 250 μs pour ralancer l'opération, ce qui est largement suffisant pour qu'il puisse réaliser les tests et les opérations nécessaires. Ce type de mécanisme est intéressant car il permet au microprocesseur de vérifier à chaque secteur le déroulement de l'opération en lui donnant le droit de l'arrêter si nécessaire.

D'autre part, sans cet entrelacement, une recherche multi-secteur devrait être menée entièrement sous la responsabilité des processeurs en tranche, ce qui ferait que le microprocesseur ne prendrait connaissance des résultats juste à la fin du processus total cela n'étant pas toujours pratique ni rapide.

VIII.29 MEMOIRE BUFFER

La mémoire Buffer d'après l'architecture du PBD-MAGE est située sur le bus du processeur P2 et est représentée par une plaque de RAM statique. Cependant, au niveau du prototype réalisé avec l'xorcicer, il a été nécessaire de mettre cette mémoire à l'intérieur de la plaque CD. Le système Exorciser n'autorise pas la manipulation de certains signaux nécessaires pour l'accès direct à la mémoire (DMA). D'autre part, l'horloge du microprocesseur est de 1 MHz pendant que le contrôleur et les unités disque tournent à 1,25 MHz (fréquence octet). Cette différence empêche le contrôleur de se synchroniser avec l'horloge du microprocesseur pour le transfert des octets du ou vers le disque.

La solution à ces problèmes a été la construction, pour le prototype, d'une mémoire tampon (buffer) de 1 K octet à l'intérieur de la plaque CD. Cette mémoire peut être accédée aussi bien par le microprocesseur que par le contrôleur.

Pour résoudre le problème du partage de cette mémoire, le signal de "Halt" est utilisé (voir VI.3.7)

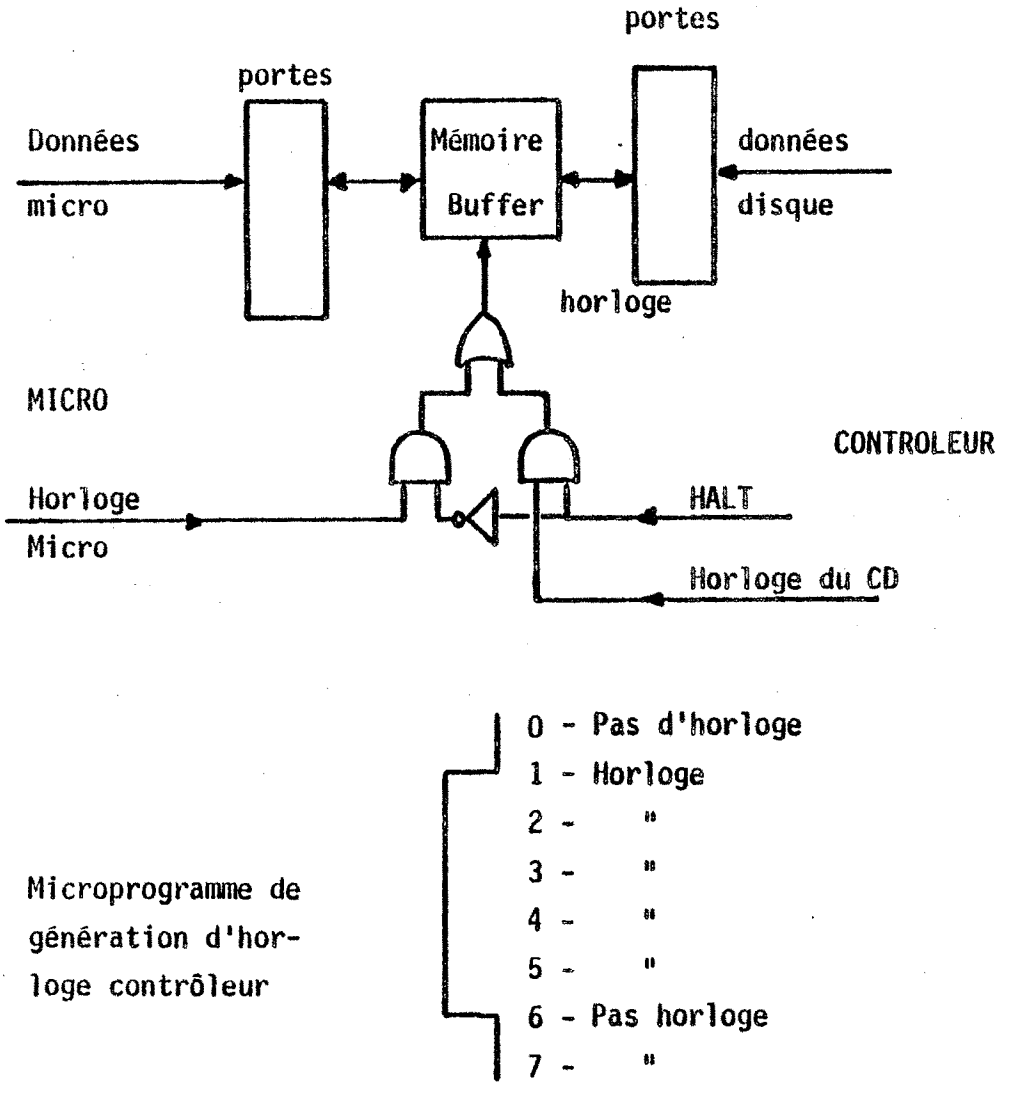


Fig VII.33 - La Mémoire Buffer et son horloge

VIII.30 CONDITIONS D'INITIALISATION

Les conditions d'initialisation sont un aspect toujours très important dans tout projet, ce sont elles qui déterminent la bonne séquence d'états à suivre. Une mauvaise initialisation peut amener la machine dans des états inchohérents sans possibilité d'en sortir si ce n'est que par le redémarrage.

Pour le contrôleur disque, les conditions d'initialisation ont été observées. Comme le contrôleur est sout les ordres du microprocesseur, une bonne partie des conditions sont établies par le micro lui-même. Par exemple, au démarrage, mise sous tension du contrôleur, il y a un registre qui ferme toutes les portes de communication (portes SMD) d'interface avec les unités disque. Le micro peut donc balayer tous les registres, en les initialisant, avant de commander le registre (flip-flop) pour ouvrir les portes. En cas de coupure de tension, ce registre est le premier à tomber en empêchant ainsi tout ordre faux vers les unités disque ou toute initialisation de nouveau transfert avant l'arrêt du microprocesseur ou du contrôleur.

Quant au démarrage de l'USC, il est aussi fait par le microprocesseur. L'USC ne fonctionne que sous les ordres du micro, hors de cela il ne peut pas travailler. Lors de la mise sous tension, l'USC sera automatiquement en attente sous l'adresse zéro. Toute initialisation ou ordre doit venir du micro.

CHAPITRE IX

FUTURS DEVELOPPEMENTS

1. L'EVOLUTION DU PBD-MAGE
2. CIRCUIT CONTROLEUR DISQUE
3. ASPECT SECURITE

IX.1 L'EVOLUTION DU PBD-MAGE

L'évolution de la technologie tout au long du projet du PBD-MAGE a beaucoup influencé l'architecture matérielle. L'arrivée de nouveaux microprocesseurs plus puissants de 16 bits tels que le Z 8000 et le M 68.000 peut inciter à remettre en cause certaines options prises pour la structure matérielle.

Ce qu'il est important de remarquer, c'est qu'un éventuel changement de microprocesseur ne remet pas en cause le projet tout entier. Dans le chapitre VI.3 on avait déjà mentionné la modularité de l'architecture du PBD-MAGE. Cette modularité est basée sur une méthode d'accès aux données réalisée par des grands blocs fonctionnels matériels. A l'intérieur de chacun de ces blocs, le matériel peut être réalisé de plusieurs façons sans mettre en cause la structure globale du PBD-MAGE. Par exemple, un éventuel changement pour des microprocesseurs plus puissants peut amener une amélioration de la performance du PBD.

L'utilisation de microprocesseurs plus puissants peut changer les degrés de responsabilité des blocs du PBD. La plus grande puissance de traitement des processeurs peut permettre de réaliser des fonctions plus sophistiquées. La figure IX.1 suggère une nouvelle structure. (Fig IX.1)

Une structure semblable avait déjà été envisagée [NAV.77] puis repoussée pour des raisons économiques. Cette structure présente un plus grand intérêt, dans le mesure où les processeurs acquièrent une plus grande personnalité. On voit qu'il existe une bonne définition des responsabilités de chaque processeur. Il y aurait un "processeur de communication" réalisé par un microprocesseur 8 bits, qui réaliserait la fonction de gestion du dialogue avec le

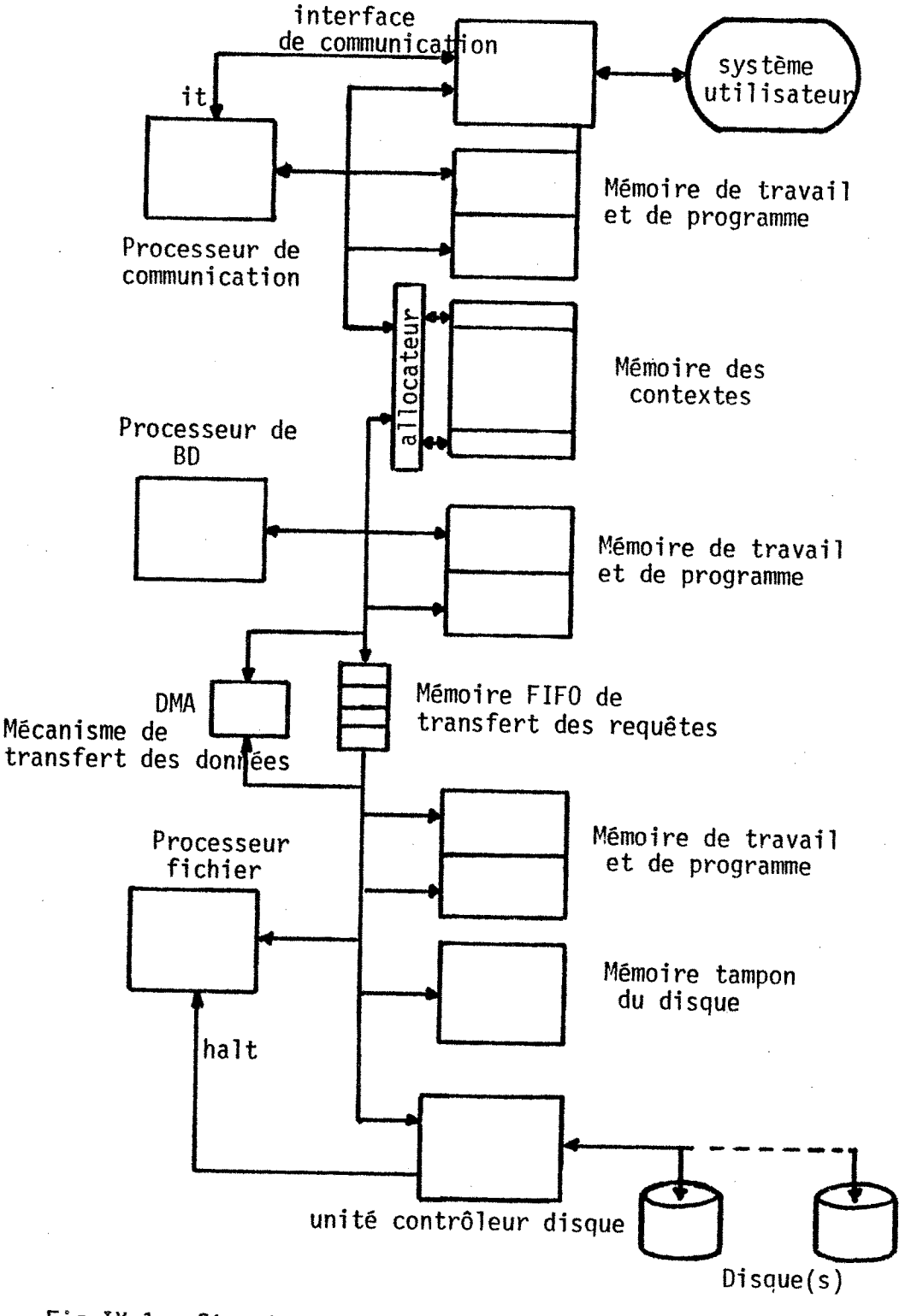


Fig.IX.1 - Structure du PBD-MAGE munis de microprocesseurs plus puissants

monde extérieur. Hors ce dialogue, il préparerait les requêtes utilisateur et les placerait dans la mémoire des contextes. Le deuxième processeur, "le processeur BD", serait responsable de la méthode BD d'accès aux données MAGE. Ce processeur serait réalisé par un microprocesseur puissant de 16 bits. Le dernier processeur serait le "processeur fichier" qui s'occuperait de la gestion des requêtes d'accès aux disques. Semblable à l'actuel "processeur disque" cette unité cependant aurait un plus grand pouvoir, lui permettant de recevoir directement des requêtes de fichiers. Ce processeur serait réalisé par un microprocesseur 8 bits.

Le moyen de communication entre le "processeur de communication" et celui de "BD" continuerait à être la mémoire des contextes. Entre le "processeur BD" et le "processeur fichier" une méthode de communication différente devrait être employée. Une mémoire FIFO (first in first out), type boîte aux lettres, permettrait le transfert des requêtes et des réponses entre les deux processeurs. L'utilisation de cette technique rend assynchrone la communication entre les deux processeurs, ce qui leur donne une totale liberté d'action, l'un vis à vis de l'autre. Pour le transfert des données entre la mémoire tampon (buffer) et la mémoire des contextes un mécanisme d'accès direct à la mémoire (DMA) serait utilisé. Ce mécanisme permet un transfert plus rapide des données et donc une économie de temps.

IX.2 CIRCUIT CONTROLEUR DISQUES

De même qu'il existe aujourd'hui des circuits monolithiques contrôleurs de disquettes, il est envisageable d'avoir dans un proche

futur des circuits contrôleurs d'unités disque. Ce type de possibilités a été fortement ressenti tout au long du projet du PBD-MAGE. Un des principaux facteurs qui permettent cela est la standardisation des connexions disques par la norme SMD, qui est suivie d'une manière générale, par les grands constructeurs d' unités disque.

Le grand nombre de signaux nécessaires pour l'interface SMD fera que, le contrôleur intégré sera probablement construit à partir de deux circuits principaux. Le premier circuit s'occuperait de l'interface avec le câble A (des normes SMD), signaux que nous appelons "lents". Ce circuit pourrait avoir à son intérieur un microprocesseur et éventuellement la mémoire tampon (il est toutefois préférable que la mémoire tampon reste extérieure au circuit par des motifs de modularité) Il aurait aussi tous les registres d'ordres pour les unités disque ainsi que le traitement des signaux d'état en provenance des unités.

Le deuxième circuit s'occuperait de l'interface B (normes SMD) et donc des signaux rapides. Dans un premier temps, dû au niveau de la technologie, ce circuit ne pourrait pas s'occcuper directement du transfert des bits vers le disque ou preovenant du disque, et il nécessiterait un circuit auxiliaire pour sérialiser/désérialiser les bits et amener par exemple la vitesse de transfert à 1,2 Mhz. Ce circuit s'occuperait de comptabiliser le nombre d'octets à transférer, de vérifier le descripteur secteur, d'effectuer la vérification de données écrites sur un secteur, etc...

En résumant, ce circuit serait concerné plus particulièrement par les fonctions de lecture, d'écriture, de formatage, de vérification et de recherche de profil du disque (Fig IX.2)

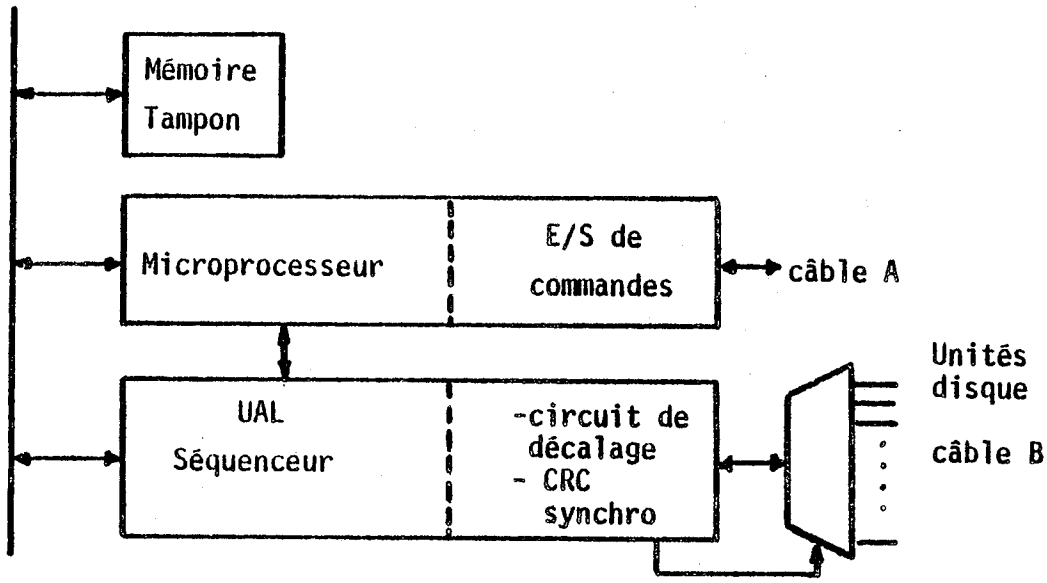


Fig IX.2 - Circuits Contrôleurs Disque

Avec l'évolution du pouvoir d'intégration dans les circuits, il est envisageable d'imaginer que dans un futur proche ces circuits contrôleurs disques deviendront des processeurs fichiers intégrés. Probablement, ils passeront par une première phase où ils deviendront des unités de gestion de mémoire secondaire avant de devenir des processeurs intégrés. Cette évolution nécessiterait qu'une certaine standardisation des accès aux fichiers soit possible.

IX. 3 ASPECT SECURITE

Très peu, pour ne pas dire rien, n'a été dit sur l'aspect sécurité du PBD dans les chapitres précédents. Cet aspect cependant n'a pas été oublié pendant le projet. Mais comme le premier but était de vérifier la possibilité d'implanter un SGBD avec des microprocesseurs, la sécurité a été laissée pour un deuxième temps.

L'aspect sécurité dans un PBD est tellement complexe qu'il mérite en lui-même une étude de l'importance de celle qui a été faite pour définir le PBD-MAGE. Nous ne citerons ici que quelques aspects.

Le fait d'utiliser deux (ou trois) microprocesseurs dans la configuration du PBD-MAGE permet d'avoir un certain degré de surveillance entre les processeurs. Il a été envisagé que lors de la détection d'une anomalie dans le comportement d'un microprocesseur, un programme en mémoire morte ROM, irait effectuer des tests pour localiser l'erreur. Même pendant le fonctionnement normal du PBD il est possible d'effectuer des tests périodiques lorsque les processeurs n'ont pas de tâches à accomplir. Un type de test qui peut être fait systématiquement est celui des mémoires RAM, à travers l'écriture puis la relecture de certains profils.

Les "chiens de garde" [COU.79] constituent un autre outil assez intéressant qui a été envisagé pour les processeurs. Les chiens de garde permettent de détecter la plupart des anomalies des processeurs d'une manière dynamique. Il n'est pas nécessaire que le microprocesseur arrête ses activités pour réaliser la détection. Il suffit que de temps en temps dans son programme, il relance le chien de garde. Cet outil assez simple est spécialement efficace pour des applications comme le PBD-MAGE où les programmes sont fixes.

La sécurité de l'écriture sur disque, est un autre aspect important qui apparaît au niveau du contrôleur disque. Il faut éviter à tout prix qu'à la suite d'une anomalie, le contrôleur écrive intempestivement sur le disque. Pour diminuer la probabilité de ceci nous penson intéressante l'utilisation d'un registre d'un bit qui donnerait l'autorisation à toute écriture de s'effectuer. Pour écrire,

le microprocesseur devrait premièrement positionner ce bit avant de lancer la routine d'écriture. Ce registre permettrait de minimiser les erreurs dues à une anomalie du contrôleur disque, mais n'éviterait pratiquement pas les anomalie provenant du microprocesseur lui-même. Une amélioration au niveau de ce dernier, peut être obtenue en séparant en deux les routines d'activation du registre et de lancement de l'écriture. Un temporisateur qui autoriserait l'écriture pendant un certain temps peut être associé à l'activation de ce registre ; après ce délai, toute tentative d'écriture serait stoppée et signalée comme une erreur.

Un autre aspect lié au contrôleur disque est le traitemene des erreurs provenant du disque. Quel est le meilleur algorithme de décision , par exemple, lorsque le contrôleur n'arrive pas à écrire sur un secteur parce qu'il est abimé ? Combien de tentatives doivent être faites avant de déclarer que ce secteur est abimé ? Nous avons constaté qu'il n'existe pratiquement pas de littérature sur ce sujet et que les algorithmes varient d'un constructeur à l'autre. La plupart du temps les algorithmes ont tendance à utiliser la règle de l'excès plutôt que celle de l'optimisation. Cela est dû à ce que les anomalies sont rares et donc il est préférable de répéter l'action plusieurs fois pour essayer d'obtenir une presque certitude sur la présence de l'anomalie. Par exemple, si le contrôleur découvre qu'il existe un problème sur un secteur, il essayera 10 fois le même action d'accès avant de la signaler.

Pour conclure ce tour d'horizon de quelques aspects de sécurité, nous pensons qu'il est intéressant, pour la maintenance, d'avoir la plupart des programmes de test en ROM déclenchés externement par l'opérateur ou le responsable de la maintenance. Actuellement

il est encore trop onéreux que les systèmes s'autotestent en permanence pendant leur fonctionnement. La technique qui parait la plus conseillée est d'avoir un circuit qui détecte les pannes et arrête le système. Il ferait éventuellement quelques tests simple pour éliminer les fausses pannes (dues à des transitoirs par exemple). C'est le responsable de la maintenance qui localiserait la panne par le lancement de plusieurs programmes de test l'un après l'autre.

X. CONCLUSION

X CONCLUSION

Le projet du PBD-MAGE est conçu pour des petits systèmes de gestion clefs en mains. Nous pensons cependant que les solutions apportées pour ce projet sont d'ordre général et peuvent très bien être prises en compte dans les projets d'autres systèmes.

Deux principaux buts ont été poursuivis et atteints dans ce travail. Le premier était d'utiliser les microprocesseurs pour réaliser une fonction complexe comme une BD. A l'époque du début de ce projet, il y avait des doutes quant aux possibilités de la puissance de traitement des microprocesseurs. Aujourd'hui leur utilisation devient monnaie courante. Avec les nouveaux progrès technologiques qui ont abouti à des microprocesseurs très puissants et performants tels que le Z 8000 et le M 68.000 cette utilisation va se généraliser.

Le deuxième but poursuivi était de réaliser une machine qui puisse aussi bien être utilisée dans les systèmes actuels centralisés comme pour les futurs systèmes distribués. Le PBD-MAGE peut aussi bien être utilisé aujourd'hui pour soulager les ordinateurs centralisés créant ainsi des systèmes décentralisés, comme il pourra être utilisé dans les futurs systèmes distribués, son adaptation se résumant à une modification de l'interface utilisateur et des programmes de communication. Le PBD-MAGE permet de faire aisément ce passage d'un type de système à l'autre. Par cet aspect il peut être considéré comme une machine de transition de technologie des systèmes.

Des systèmes complexes peuvent être décentralisés et des fonctions spécifiques réalisées avec des microprocesseurs. Il n'y a pas de

doute qu'un des facteurs prépondérants qui a permis cette révolution à laquelle nous assistons est le microprocesseur. Son bas coût par rapport à sa puissance de calcul fait qu'on peut avoir de "l'intelligence" un peu partout là où elle se fait nécessaire. Aujourd'hui il existe même des claviers gérés grâce à des microprocesseurs. De plus en plus l'intelligence est placée directement où elle est nécessaire, créant ainsi des processeurs pour des applications spécifiques. L'interconnexion de ces processeurs donne les systèmes distribués.

Anciennement la ressource chère était l'ordinateur, l'UCP, et donc elle était partagée au maximum pour améliorer le rapport performance/prix. Cela n'est plus le cas aujourd'hui, il n'est plus question de partager la ressource "puissance de traitement", l'intelligence. Actuellement la ressource partagée est l'information. Dans le chap III.3 nous avons vu que plus grande était la puissance de traitement, plus petit était le flux d'information nécessaire pour communiquer. Le fait de mettre l'intelligence directement là où elle est nécessaire, permet un meilleur traitement de la fonction spécifique, en même temps qu'elle diminue le flux d'information nécessaire pour dialoguer avec cette fonction.

Quant au prototype du PBD-MAGE, la modularité de ce projet, ainsi que sa construction, lui permet d'évoluer avec les nouvelles technologies. Sa modularité permet un assez haut degré de modifications sans que le projet soit tout entier remis en cause. Cela lui donne une plus grande durée de vie. Un changement dans la technologie des mémoires ou des processeur n'implique pas nécessairement le reformulation du projet. Du moment que le lien de communication matériel et logiciel reste le même, il n'est pas nécessaire de modifier les autres cartes de la machine.

Deux techniques intéressantes ont été employées dans le projet d'architecture du processeur disque du PBD-MAGE. La première utilise des processeurs en tranches dans une architecture à double parallélisme pour permettre au contrôleur de tourner à la fréquence de 10 Mhz. La deuxième consiste à synchroniser le contrôleur disque directement aux unités de disque. Cette dernière technique différente de celle employée par les contrôleurs courants est intéressante par ses possibilitées d'interactions directes donc rapides avec les unités de disque.

Un aspect important dans le futur des BDs est l'arrivée des circuits à mémoirs CCD, mémoires à bulles et mémoires EBAM. Les mémoires à bulles, spécialement changeront l'aspect des PBD dans un futur proche, vue leurnon volatilité. Cependant ces nouvelles technologies n'iront pas prendre la place des unités disque à bras mobile avant les années 90. Cela est un point important, car il montre que pour10 années encore, les projets de PBDs devront prendre en compte les unités disque pour leur mémoire secondaire.

L'évolution actuelle de la compétitivité des mémoires à bulles par rapport aux unités de disque peut être très bien comparée à celle qu'ont eu, à partir de 1970, les mémoires monolithiques par rapport aux mémoires à tores. Aujourd'hui, dix années, après l'avènement des mémoires monolithiques, leur utilisation est courante et n'est plus contestée.

Une extrapolation des courbes du chapitre IV donne une idée del' époque où se croiseront la courbe du prix du bit mémoire à bulles et celle des unités disque à bras mobile (Fig X.1). Même avec un ralentissement dans les progrès technologiques, nous pensons que ces courbes extrapolées donnent une bonne idée de l'évolution pour

les prochaines années, des deux technologies et de leur point de croisement. On voit qu'il se situera au plus tôt vers l'année 90. Ces extrapolations ont été faites en prenant comme base, le prix du bit pour des grandes capacités de mémoire. Le point de croisement probable correspond à celui où les mémoires à bulles deviendraient compétitives en prix pour des grandes capacités.

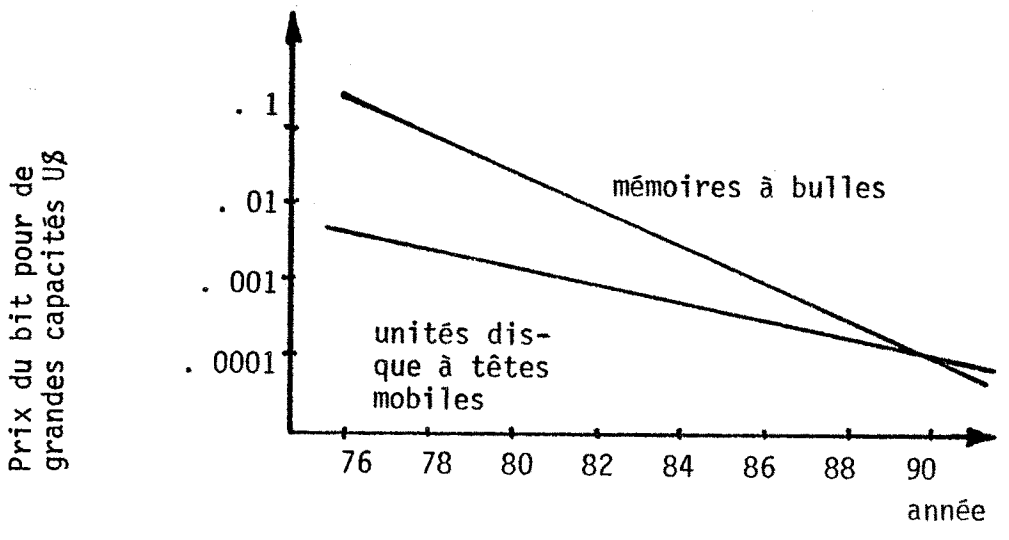


Fig X.1 - Courbe extrapolée de l'évolution du prix du bit.

Si tout au long de ces dix prochaines années il n'y a pas de révolution technologique, ces courbes ont une bonne probabilité de représenter la réalité. Elles considèrent une diminution du prix du bit des unités disque de trois fois toutes les quatre années et de trois fois toutes les deux années pour les mémoires à bulles.

Cependant ces courbes ne veulent pas dire que les mémoires à bulles deviendront compétitives juste en 90. Elle progresseront tout doucement en prenant premièrement le marché des mémoires secondaires de petite capacité, après celui de moyenne capacité et ainsi successivement. Le chapitre IV cite qu'en 78, ces mémoires n'étaient compétitives que pour des capacités mémoires inférieures à 1 M bit et qu'en 80 elles le seront jusqu'à 4 M bit. Une extrapolation de ces valaeur, basée sur les courbes précédentes, permet de montrer l'évolution de la limite de compétitivité des mémoires à bulles vis-à-vis des unités disques tout au long des prochaines années (Fig X.2)

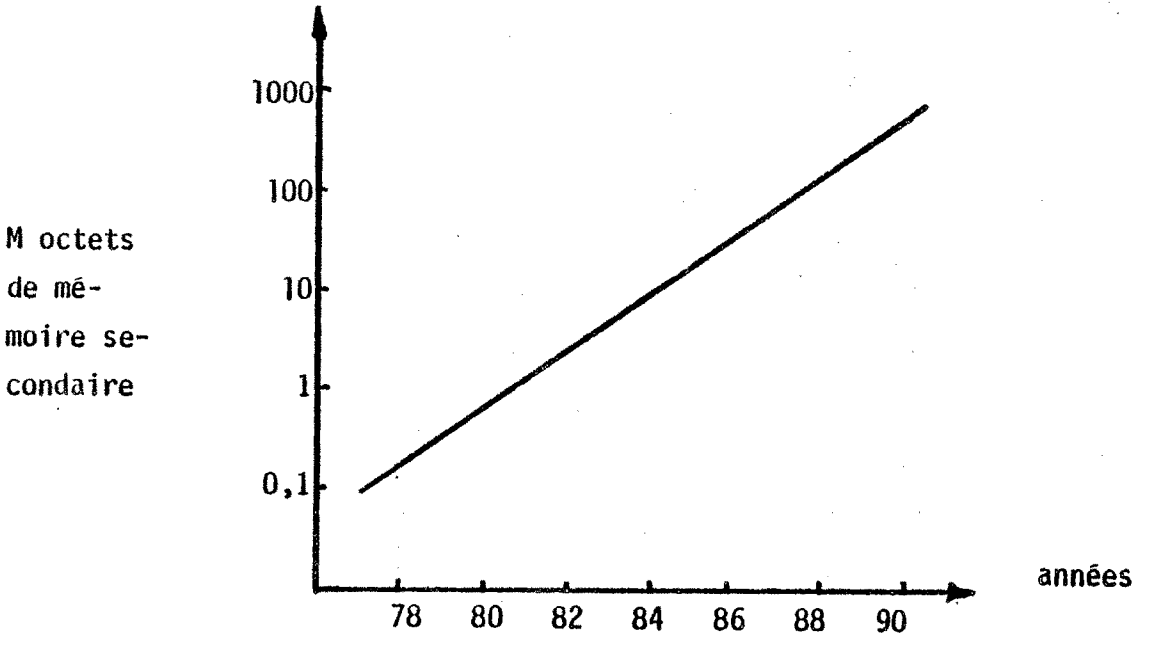


Fig X.2 - Evolution de la compétitivité des mémoires à bulles par rapport aux unités disque à bras mobile.

Ces courbes montrent que c'est juste après les années 84 que les mémoires à bulles auront, avec 8 M octets, des capacités de stockage qui commenceront à être intéressantes et compétitives.

On peut en conclure que ces mémoires à bulles, tout au long des prochaines années, seront utilisées plutôt comme des mémoires tampons et auxiliaires. Dans le cas du PBD-MAGE, dans un premier temps, l'information sur la structure pourrait être mise en mémoire auxiliaire à côté du processeur P1 (par exemple) permettant ainsi de minimiser les accès au disque. Dans une deuxième étape plus lointaine, l'information dictionnaire pourrait elle aussi rester dans une mémoire liée directement à un processeur. Cette évolution amènerait, pour le PBD-MAGE, à avoir toute l'information sur la méthode d'accès en mémoire auxiliaire et ne resteraient dans les unités disque, que les données.

C'est la modularité des accès aux différents fichiers de MAGE qui rend possible une telle évolution, sans une remise en cause de l'architecture. Cet aspect est certainement un des points forts du PBD-MAGE :

- une architecture réalisable avec les technologies actuelles.
- possibilité d'évolution et de progrès avec l'évolution des technologies.

Cela n'est envisageable que si la technologie des microprocesseurs suit en permettant de disposer toujours d'une puissance de calcul suffisante pour "suivre" des mémoires secondaires plus rapides.

Ces deux axes, l'augmentation de l'intelligence des PBDs et l'utilisation de mémoires auxiliaires pour garder l'information de la structure d'accès aux données, sont certainement les principaux créneaux de recherche en PBD pour les prochaines années. Comme il est certain que ce n'est pas pour demain que les unités disque disparaîtronts du marché, il est nécessaire que les recherches en PBD prennent en compte ce fait dans leurs projets :

l'augmentation de l'intelligence des PBD, introduisant de la recherche associative directe sur le disque à travers de cellules, et aussi permettant au PBD d'avoir avec l'extérieur un dialogue à haut niveau, diminuant le flux, ces deux facteurs permettront de diminuer les temps d'accès à l'information. De la même manière l'utilisation de mémoires auxiliaires, pour la structure d'accès, diminue ce temps.

Les principales motivations pour les recherches en BD, et donc en PBD, est l'"efficacité" et cela est représenté par le temps de réponse d'une requête utilisateur. Les deux autres principales motivations sont l'"utilisabilité", un système de BD doit être facile à manipuler par l'utilisateur, et l'autre l'"intégrité", il est nécessaire d'augmenter, spécialement par des moyens matériels, la sécurité et la confidentialité des données dans les BDs.

Il n' y a pas de doute que la PBDs commencent à apparaître de plus en plus, au moins les acticles sur les PBD [HSI.75], et qu'ils seront dans le futur une des parties des plus importantes dans les systèmes. Le PBD-MAGE peut être considéré comme une machine intermédiaire entre la génération des SGBD réalisées par les ordinateurs et les futurs PBD qui utiliseront intensément la recherche associative directe sur les disques. Le PBD-MAGE réalise la méthode d'accès aux données d'une manière classique, mais en utilisant des microprocesseurs et dans le cadre d'un processeur fonctionnel. Donc, d'une part la méthode d'accès est classique, mais d'autre part, elle est partie intégrante d'un processeur spécialisé. C'est pour cela que le PBD-MAGE peut être considéré comme un lien, une machine de transition, entre les modèles de BD classiques et les futurs systèmes.

Comme dernier point, nous aimerions aborder le sujet des architectures des machines. Nous pensons que l'aspect recherche en architecture de machines, le matériel, a été laissé trop de côté tout au long des vingt dernières années, au bénéfice du logiciel, de l'aspect système, etc... Tout au commencement, lors des premiers pas de l'ordinateur, la grande préoccupation était le matériel, cette phase est vite passée et depuis lors, peu a été fait dans ce domaine. Preuve de cela, les microprocesseurs construits aujourd'hui ont une architecture qui ressemble beaucoup à celle des précédents. Il existe une évolution technologique du matériel qui n'est pas accompagnée par l'évolution des architectures.

On retrouve très bien ce problème dans les architectures des PBD avec les recherches associatives sur dique. Ces architectures appelées de non-numériques, ont des difficultés à utiliser des UALs communes pour réaliser leurs fonctions, car celles-ci ne sont pas adaptées aux besoins de la "recherche associative". Des travaux sont en cours pour définir une cellule associative adaptée aux PBD. Cela sera un premier pas dans la recherche de nouvelles architectures.

Nous pensons donc, qu'on assistera dans les prochaines années à une révolution dans l'architecture des machines qui devra avoir des effets d'ici une dizaine d'années. Il existe trop de problèmes en informatique d'une manière générale, comme en intelligence artificielle par exemple, qui ne peuvent pas être résolus, ou alors qui le sont partiellement, par les architectures acutelles des machines.

A N N E X E 1

Résumé des NORMES S M D DE CONNEXION D'UNITES DISQUE

1. INTRODUCTION

Ces spécifications concernent les normes de connexion de contrôleurs à des unités de disque de 10 MHz. Ces disques possèdent, préenregistrées dans une de leurs surfaces, les informations sur la position du servomécanisme et sa position rotationnelle.

Ces spécifications décrivent les normes de connexion entre contrôleurs et disques du point de vue fonction, opérations et circuit (ce dernier n'est pas présenté ici).

Les disques employés utilisent normalement la technique d'enregistrement MFM (Modified Frequency Modulated) pour permettre une haute densité de données.

Le servomécanisme utilise une technique connue sout le nom de "suiveuse de piste" qui permet de minimiser les variations dues aux gradient de température entre l'écriture et la lecture et ainsi d'assurer une complète compatibilité de changement des disques d'une unité à l'autre.

La connexion disque/contrôleur est réalisée au travers de deux câbles séparés, l'un concernant les signaux de contrôle (câble A) et l'autre les signaux de lecture/écriture (câble B).

Ces normes sont suivies, à quelques détails près, par plusieurs fabricants et fournissent un standardde connexion entre les disques. Cela assure une complète compatibilité de liaison quel que soit le disque ou le contrôleur. Cependant, comme ces normes sont encore nouvelles, il existe quelques différences d'un fabricant à l'autre.

Le présent document est basé sur une compilation des spécifications des fabricants avec une adaptation aux besoins du projet du PBD-MAGE.

2. SIGNAUX D'INTERFACE

Avec des connexions par câbles plats (flat cables), la distribution des signaux sur le connecteur est la suivante :

Signal	polarité du -	polarité du +	provenance
Sélection de l'unité bit 0	23	53	du contrôleur
Sélection de l'unité bit 1	24	54	du contrôleur
Sélection de l'unité bit 2	26	56	du contrôleur
Sélection de l'unité bit 3	27	57	du contrôleur
Validation d'unité disque	22	52	du contrôleur
Validation d'adresse cylindre	1	31	du contrôleur
Validation d'adresse de tête	2	32	du contrôleur
Validation de contrôle	3	33	du contrôleur
Bus de sortie bit 0	4	34	du contrôleur
Bus de sortie bit 1	5	35	du contrôleur
Bus de sortie bit 2	6	36	du contrôleur
Bus de sortie bit 3	7	37	du contrôleur
Bus de sortie bit 4	8	38	du contrôleur
Bus de sortie bit 5	9	39	du contrôleur
Bus de sortie bit 6	10	40	du contrôleur
Bus de sortie bit 7	11	41	du contrôleur
Bus de sortie bit 8	12	42	du contrôleur
Bus de sortie bit 9	13	43	du contrôleur

Signal	Polarité du -	Polarité du +	Provenance
Détection de liaison ouverte	18	48	du contrôleur
Index	18	48	de l'unité disque
Marque de secteur	20	48	de l'unité disque
Faute	15	45	de l'unité disque
Sur le cylindre	17	47	de l'unité disque
Erreur de recherche	16	46	de l'unité disque
Unité prête	19	49	de l'unité disque
Protection écriture	28	58	de l'unité disque
Marque d'adresse trouve	20	50	de l'unité disque
Séquence de démarrage "pick"	29		du contrôleur
Séquence de démarrage "Hold"	59		du contrôleur
Occupé *	21	51	de l'unité disque
Pas utilisé	30	60	

* pour version à deux canaux.

Les signaux qui traversent le câble B sont :

Signal	Polarité des			Provenance
	-	+	maille	
Ecriture des données	8	20	7	du contrôleur
Horloge d'écriture	6	19	18	du contrôleur
Lecture des données	3	16	15	de l'unité disque
Horloge de lecture	5	17	4	de l'unité disque
Horloge du servo	2	14	1	de l'unité disque
Unité sélectionnée	22	9	21	de l'unité disque
Fin de recherche	10	23	-	de l'unité disque
(réservé index)	12	24	11	
(réservé secteur)	13	26	25	

Le signaux du câble B ne sont pas sélectionnés par les signaux de sélection d'unité.

3. DESCRIPTION DES SIGNAUX

3.1 Le câble A

Signaux provenant du contrôleur

3.1.1 Sélection d'une unité disque

Cinq lignes sont utilisées, qui permettent, par décodage, d'adresser jusqu'à 16 unités de disques.

a) Validation d'unité disque
(select enable / unit select tag / unit select strobe)

Ce signal ira valider les lignes de sélection d'unités permettant d'adresser un des disques. Le front ascendant du signal exécutera la sélection. Celle-ci est faite par la comparaison entre l'adresse de sélection des lignes et l'adresse de la plaque du disque. Elle doit être exécutée dans les 280 ns à 600 ns qui suivent le front ascendant du signal. Les signaux de sélection d'unité ainsi que celui de validation d'unité, doivent rester actifs pendant tout le temps de la sélection.

b) Lignes de sélection d'unités

Quatre lignes codées binaires permettent de choisir une des 16 unités disque branchées au câble de contrôle (câble A). L'adresse de chaque unité disque est déterminée par un "plug" identificateur. Ce "plug" identificateur peut être modifié pour changer l'adresse. (Le numéro logique 15 peut être utilisé comme numéro de maintenance et il apparaîtra toujours lorsque le "plug" sera retiré).

3.1.2 Les lignes de fonction

(Function tags)

Bus de sortie

Ces dix lignes (bus de sortie) auront des fonctions différentes d'adresse ou de contrôle, dépendant des validations. Dans le tableau ci-dessous sont énumérés les trois types de fonctions que les lignes peuvent adopter avec les signaux.

Lignes	Fonction cylindre	Fonction Tête	Fonction de contrôle
Bus de sortie bit 0	. cyl 2^0	. tête 2^0	porte d'écriture
Bus de sortie bit 1	cyl 2^1	tête 2^1	porte de lecture
Bus de sortie bit 2	cyl 2^2	tête 2^2	déplacement en avant
Bus de sortie bit 3	cyl 2^3	tête 2^3	déplacement en arrière
Bus de sortie bit 4	cyl 2^4	tête 2^4	remise à zéro d'une faute
Bus de sortie bit 5	cyl 2^5		recherche de marque d' adresse
Bus de sortie bit 6	cyl 2^6		retour à zéro
Bus de sortie bit 7	cyl 2^7		validation des données anticipées (optionnel)
Bus de sortie bit 8	cyl 2^8		validation des données postérieures (optionnel)
Bus de sortie bit 9	cyl 2^9		-

a) Validation d'adresse cylindre

(set cylinder/ tag 1 / cylinder adresse)

Sous la commande de cette ligne, les lignes de fonctions deviendront des adresses de cylindres. Cette commande exécutera donc le transfert de l'adresse cylindre vers l'unité disque à travers le bus de sortie. Cette validation déclenchera ainsi plusieurs opérations.

L'ordre de validation ne sera accepté que si le signal "sur cylindre" existe déjà. Si accepté, l'unité disque calculera la différence entre la nouvelle adresse cylindre et l'ancienne, gardera la nouvelle adresse dans un registre et un signal de "recherche" sera déclenché si les signaux 'déplacement' ou 'occupé' ne sont pas activés. Le diagramme de temps qui concerne ces opérations est donné dans la figure 1.

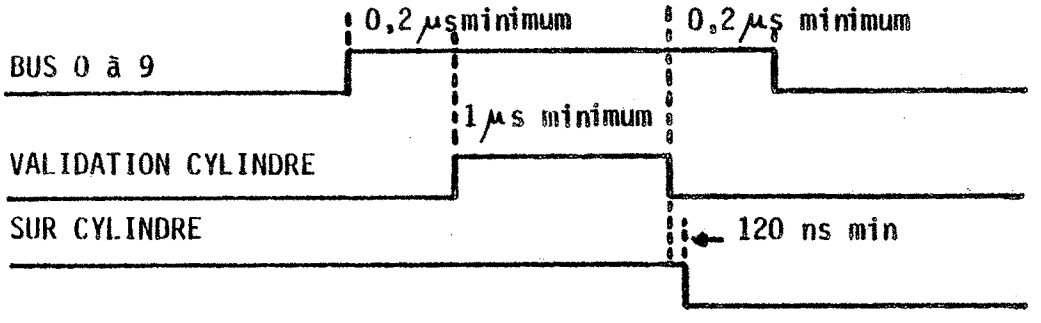


Fig 1 - Diagramme des temps pour la validation d'adresse cylindre

L'unité disque calculera la localisation du cylindre actuel, par rapport à celle désirée (différence) avant de réaliser la recherche d'un nouveau cylindre. Le calcul de cette différence est réalisé en activant les "bus de sortie" du contrôleur et en attendant 1,2 µs pour que le calcul soit réalisé (T1 et T2). Si la nouvelle adresse de cylindre dans le "bus de sortie" est trop grande (hors de la capacité de ce disque), une "faute d'adresse" sera détectée et envoyée au contrôleur des disques, et l'opération prendra fin.

b) Validation d'adresse de tête
(set head/tag 2/head select)
Quand ce signal est activé, les cinq premiers bits du bus de sortie deviennent des adresses de tête pour un disque. Ce signal ira donc spécifier le choix d'une des pistes d'un cylindre pour réaliser les opérations de lecture ou d'écriture.

c) Validation de fonction
(control select/tag 3)
Ce signal activera les neuf lignes "bus de sortie" qui représentent les opérations de contrôle ci-dessous décrites. Ce signal devra rester actif pendant toute l'opération.

C.1) Bus de sortie bit 0/Porte d'écriture (write gate)
Le signal porte d'écriture autorise le circuit d'écriture à convertir l'information digitale de la ligne des données d'écriture, en courant pour les têtes. Ce signal occasionne une condition d'invalidité si les signaux transition d'écriture, courant d'écriture, oscillateur d'écriture, protection d'écriture, déplacement, PLO (phase look oscillator) et piste fine, ne sont pas corrects ou n'arrivent pas au bon moment.

c.2) Bus de sortie bit 1/Porte de lecture
(read gate)
Le signal porte de lecture autorise le circuit de lecture à convertir l'information analogique enregistrée sur un disque en une donnée digitale qui sera transférée vers l'unité de contrôle via la ligne de lecture des données.

c.3) Bus de sortie bit 2/Déplacement en avant
(offset forward/offset plus)
Ce signal actionne le mouvement du positionneur de tête d'un écart en direction de l'axe(l'écart étant 300 µpouces pour des disques de

400 cylindres et 150 µpouces pour ceux de 800 cylindres) par rapport à la position nominale du cylindre. Cette commande désactivera le signal "sur cylindre" pendant 4 ms (2,75) et si elle est active avec le signal "porte d'écriture", elle causera une condition de faute qui désactivera les circuits de lecture et d'écriture.

c.4) Bus de sortie 3/Déplacement en arrière
(offset reverse/offset minus)

Ce signal active le mouvement du positionneur de tête d'un écart par rapport à la position normale du cylindre, en s'éloignant de l'axe. Les observations sur le signal "déplacement en avant" sont aussi valables pour ce signal. Un délai d'environ 4 ms, après le commencement du signal, est nécessaire avant de continuer les opérations de lecture ou d'écriture.

C.5) Bus de sortie 4/Remise à zéro de faute
(fault clear/fault reset)

Le signal remise à zéro de faute élimine la condition de faute si elle n'existe plus. Cette commande doit rester active pendant plus de 100 ns.

c.6) Bus de sortie 5/Recherche de marque d'adresse
(address mark/AM enable)

Ce signal, quand il est utilisé avec la porte d'éciture, permet à l'unité de contrôle d'effacer des portions de piste sans causer de condition d'invalidité due à l'absence de transitions d'écriture. Quand le signal de marque d'adresse est utilisé avec la porte de lecture, les circuits de lecture permettent à l'unité de contrôle de rechercher les zones effacées sur une piste et de retrouver les marques d'adresse. Cette commande permet à l'unité de contrôle d'utiliser des formats d'enregistrement de longueur variable.

c.7) Bus de sortie 6/Retour à zéro
(rezéro/RTZ)

Ce signal commande le retour du positionneur de tête au cylindre zéro. Cette commande désactive le signal "sur cylindre" jusqu'à ce que la tête soit positionnée sur le scylindre zéro. Ce signal doit être activé pendant un minimum de 250 ns et un maximum de 1 ms.

c.8) Bus de sortie 7/Validation des données anticipées
(data strobe early)

Par ce signal, l'horloge des données valide les données de lecture un temps plus tôt que la normale.

c.9) Bus de sortie 8/Validation des données postérieures
(data strobe late)

Par ce signal, l'horloge des données valide un temps plus tard les données lues. Comme le précédent, ce signal est optionnel et n'est normalement inclus que dans l'option NRZ. Ces signaux activent le PLO (Phase Look Oscillator) de séparation des données et permettent de changer l'instant de validation des données.

3.1.3 Détection de liaison ouverte

(Device enable/Open cable detector)

Quand ce signal est activé, il autorise la communication entre l'unité de contrôle et les disques en activant les circuits récepteurs et émetteurs de l'unité disque. L'absence de ce signal met hors de service les circuits récepteurs et émetteurs, ainsi que toutes les fonctions de contrôle incluant l'écriture de l'unité de contrôle.

La génération de ce signal dans l'unité de contrôle peut être faite par une logique de relais (ou un contact dans la clé d'allumage) avec une terminaison statique telle que ce signal soit le

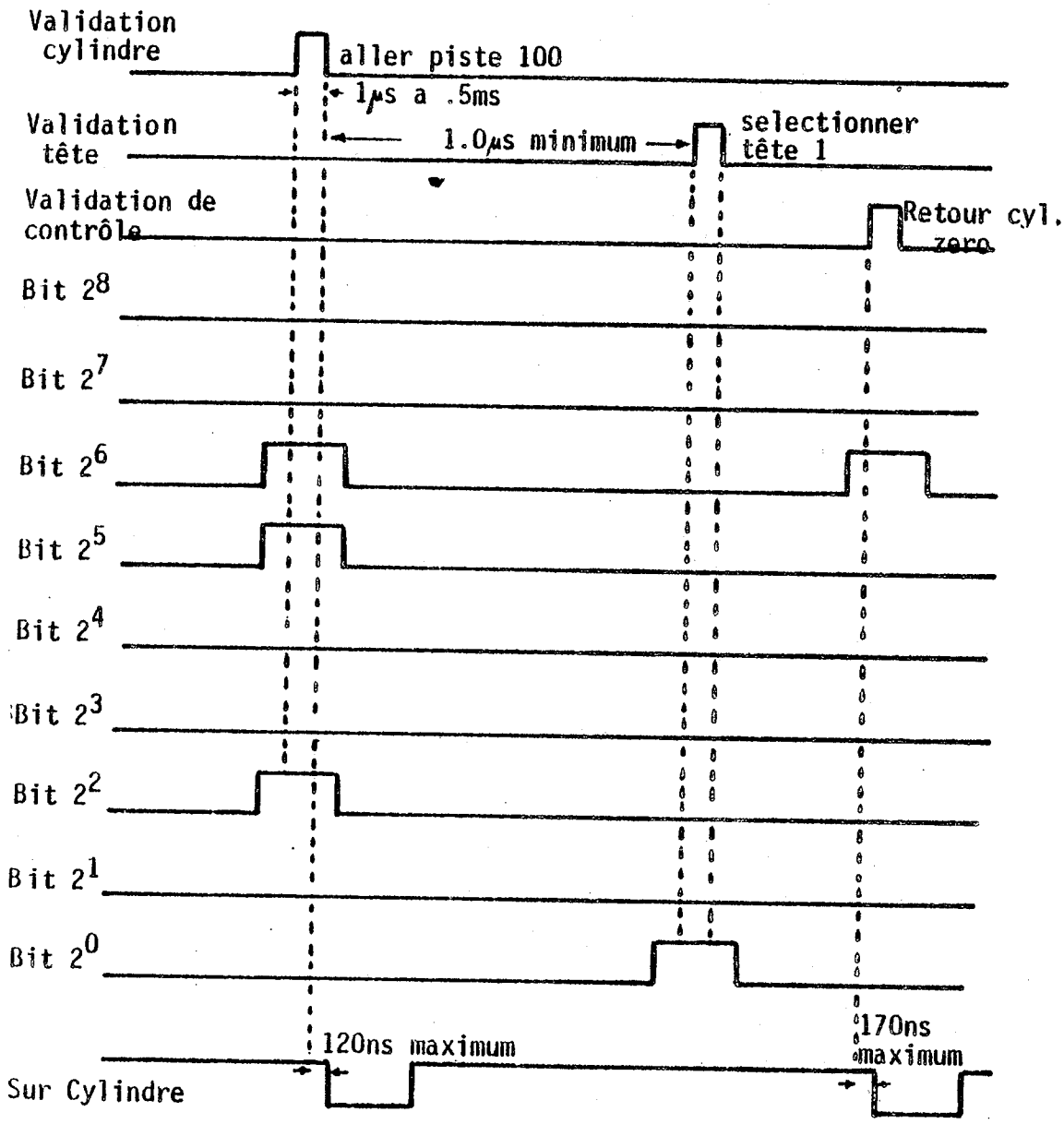


Fig. 2 - Chronogramme des signaux de fonction

dernier activé dans la séquence de mise en route et le premier à tomber lors d'une coupure de courant. Au moment où le câble est retiré, tous les circuits émetteur de l'unité disque sont désactivés.

Signaux provenant des unités disque

3.1.4 Index

La ligne index indique, quand elle est activée, que la zone d'index (le point de référence) passe sous les têtes. Ce point d'index est une pulsation qui apparaît une fois à chaque révolution du disque. Sa durée est de 2,5 µs. Le disque doit être sélectionné pour que cette ligne puisse être activée.

3.1.5 Marque de secteur

(sector mark)

Ce signal établit des points de référence tout au long d'une piste d'une "pile de disques". Chaque piste peut être divisée en 128 secteurs ou plus. Le premier secteur zéro, coïncide avec l'index. La durée d'un signal de marque de secteur est de l'ordre de 1,24 µs. Pour recevoir ce signal, le disque doit être sélectionné. Le nombre de secteurs dans lequel chaque piste est divisée est déterminé par un "plug" ou alors par des clefs.

3.1.6 Faute

(Fault/Read-Write Unsafe)

Le signal de faute activé indique qu'une faute est arrivée à l'intérieur d'une unité disque et donc que l'opération de lecture ou d'écriture est mise en cause. La condition de "faute" désactive les sections de lecture et d'écriture de l'unité disque pour prévenir toute destruction d'un enregistrement de données par inadvertance. Cette condition occasionne également la désactivation du signal "unité prête". Les signaux "sur cylindre" et "fin de recherche" restent activés.

Les conditions qui peuvent occasionner une faute sont les suivantes :

a) Invalidation d'écriture

. écriture avec PLO erronés
. écriture avec clé "d'écriture protégée" activée
. écriture avec déplacement
. écriture et pas de fine piste
. écriture avec lecture.

b) Défaut de transition d'écriture

. écriture avec pas de transition d'écriture
. pas de sélection d'écriture et pas de transition d'écriture.

c) Défaut de courant d'écriture

. écriture et pas de courant d'écriture,
. pas de sélection d'écriture et pas de courant d'écriture,
. écriture et invalidation de courant d'écriture.

d) Coupure de courant (et erreur de selection de tête)

. multiple sélection de têtes,
. conflits de courant de l'unité disque,
. manque de courant.

Le signal de faute est désactivé par :
. une remise à zéro initiale du contrôleur
. la clé de remise à zéro d'erreur du panneau avant de l'unité disque
. la remise à zéro lors du démarrage du système
. la mise en place initiale de la tête lors d'un changement de pile disque ou le retrait et réinsertion du plug d'identification de module.

L'unité disque doit être sélectionnée pour que l'unité de contrôle reçoive ce signal.

3.1.7 Erreur de recherche

(seek error)

Cette ligne indique une des situations suivantes :

a) l'unité disque n'a pas réussi à faire sa recherche en moins de 100 ms ;
b) l'unité disque n'as pas réussi à faire un repositionnement de tête ou une opération de retour au cylindre zéro en 700 ms.
c) une adresse illégale a été reçue par l'unité disque. Le signal d'erreur de recherche est reçu 100 ns après le signal de validation de cylindre, s'il y a erreur d'adresse ;
d) il y a un conflit des alimentations.

La condition d'erreur de recherche force le servomécanisme à être inopérant. Les signaux "fin de recherche" et "sur cylindre" resteront activés.

La condition d'erreur de recherche peut être désactivée par un ordre de "retour à zéro" de l'unité de contrôle ou par toute autre opération qui ne soit pas une activation des têtes (par exemple, redémarrage de l'unité de disque).

L'unité de disque doit être sélectionnée pour que l'unité de contrôle reçoive le signal d'erreur de recherche.

3.1.8 Sur le cylindre

(on cylinder)

Cette ligne d'état indique généralement que le servo est stationnaire, c'est-à-dire positionné sur une piste définie et prêt à exécuter la prochaine opération. Cela est vrai à l'exception du moment

où le signal "sur cylindre" est forcé actif par le signal de recherche incomplet (erreur de recherche). Toute opération de mouvement des têtes occasionne la perte du signal du cylindre, comme par exemple recherche d'un nouveau cylindre zéro, ou "déplacement". Pendant les opérations de "déplacement" le signal sur cylindre sera désactivé pendant 2,75 ms. L'unité disque doit être sélectionnée pour recevoir les signaux.

3.1.9 Unité prête

(Unit ready)

Cette ligne indique que :

a) l'unité disque est sélectionnée,

b) le disque-pack est à la bonne vitesse,

c) les têtes sont positionnées pour initialiser la première recherche,

d) il n'y a pas de condition de faute dans l'unité disque.

3.1.10 Protection d'écriture

(write protected)

Ce signal indique que la clé "read only" (lecture seule) du panneau avant de l'unité disque est active et donc que l'enregistrement des informations dans le disque est inhibée. L'activation du signal "porte d'écriture" occasionnera, quand l'état de la clé est en écriture protégée, une faute.

3.1.11 Marque d'adresse trouvé (option)

(Addresse mark found)

La ligne de marque d'adresse trouvé est activée par la combinaison du signal porte de lecture, la commande de "recherche de marque d' adresse" et l'absence de données pendant à peu près 24 (ou 16) bits de temps. Cette ligne reste activée pendant 10 µs.

Lors de la réception de ce signal, l'unité de contrôle désactivera la ligne "recherche de marque d'adresse" et le données se présenteront valides aux lignes d'entrée-sortie, à la fin du signal "marque d' adresse trouvé ".

3.1.12 Séquence de démarrage (option)
(power sequencing)

Ces signaux peuvent etre utilisés pour le démarrage automatique des unités disque. Cela permet la mise sous tension des unités disques l'une après l'autre à partir du contrôleur disque.

3.2 Le câble B

Signaux provenant du contrôleur

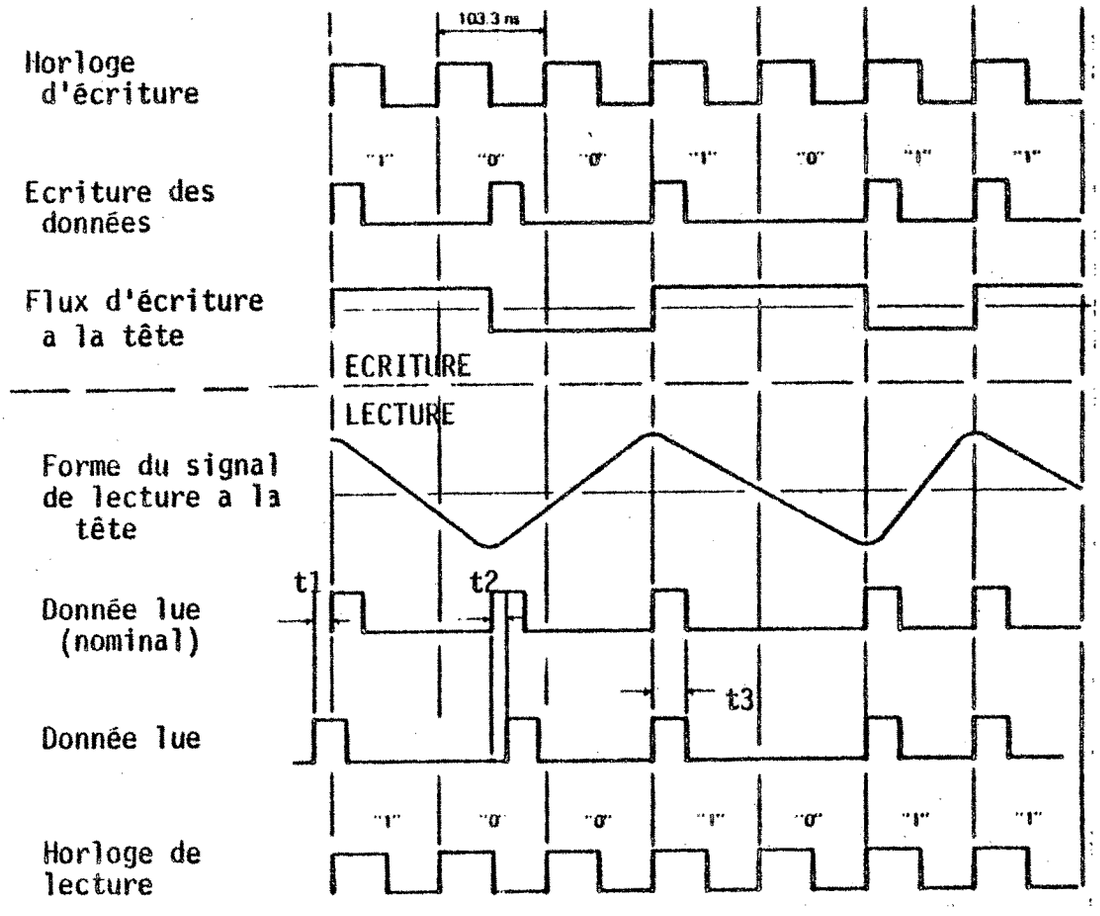
3.2.1. Ecriture des données

La ligne d'écriture des données transmet celles-ci en série vers l'unité disque pour qu'elles soient enregistrées sur les disques. La figure 3 montre un diagramme temporel typique d'un enregistrement de données au format MFM.

Les signaux transmis par le contrôleur de disque peuvent être précompensés pour soulager la plupart des décalages de bits inhérents aux enregistrements sur disques magnétiques. Pendant l'opération de lecture, ce décalage est dû à la combinaison de problèmes physiques d'enregistrement et de la réponse des têtes de lecture. Ce décalage du bit est directement lié à la symétrie entre le bit précédent et le suivant. S'il existe une symétrie (donc les bits précédents et suivants sont zéro ou un), il n'y aura pas de décalage. Cependant, s'il y a assymétrie, le bit en lecture sera décalé. Pour minimiser ce problème, les bits peuvent être intentionnellement prédécalés

(précompensés) dans la direction contraire pendant l'enregistrement au format MFM.

Dans l'unité de contrôle, l'implémentation de la précompensation est faite en réalisant une "fenêtre d'examen", large de deux bits (206,6 ns) qui se centre sur le bit en question pour décider de son décalage.



t1 - décalage avant
t2 - décalage après
t3 - temps de lecture nominal = 50 ± 10ns

Fig 3 - Forme des ondes des signaux en écriture et en lecture MFM

Cette fenêtre centrée sur le bit offre les possibilités suivantes :

a) s'il y a des signaux aux deux extrêmités de la fenêtre, il ne sera pas nécessaire de faire un décalage ;
b) s'il n'y a pas de signaux aux deux extrêmités de la fenêtre, il ne sera pas non plus nécessaire de décaler ;
c) cependant, s'il y a un signal au front ascendant de la fenêtre et pas de signal au front descendant, le bit d'écriture (signal) sera décalé de 6 ± 1 ns plutôt que sa position nominale ;
d) s'il n'y a pas de signal au front ascendant de la fenêtre et qu'il y en a au front descendant, la pulsation d'écriture sera retardée de 6 ± 1 ns de sa position nominale.

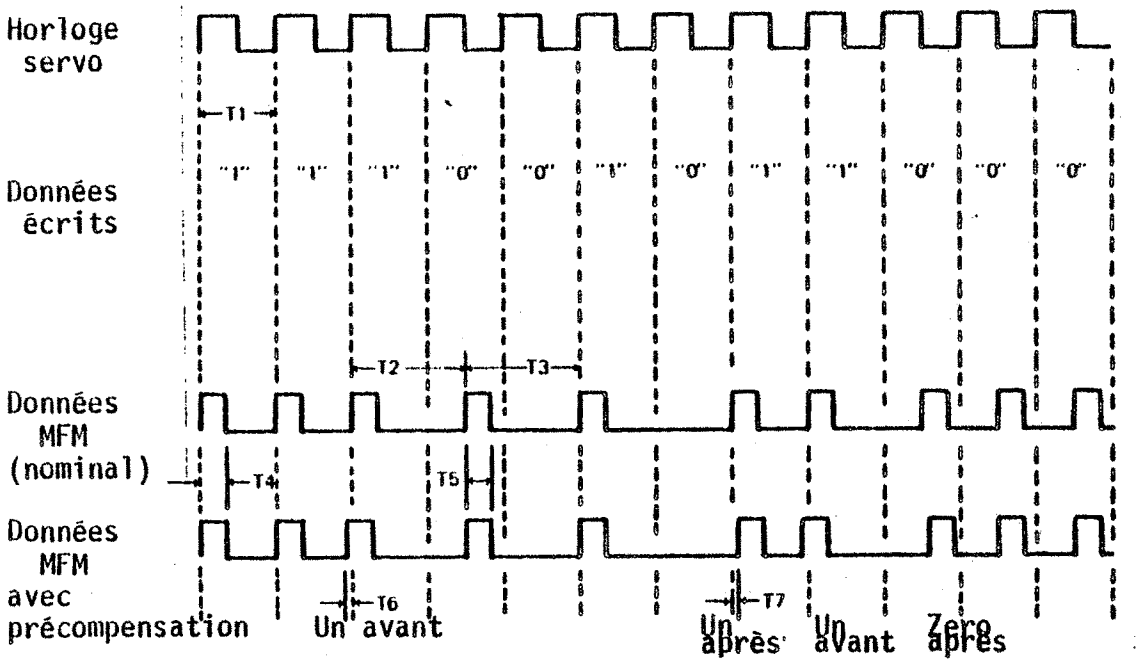
La figure 4 illustre des précompensations.

Les données d'écriture seront transmises pendant que la porte d'écriture est active.

Si l'unité disque utilise l'option NRZ (non retour à zéro) pour l'écriture, la ligne d'écriture des données transmettra les données d'écriture dans le format NRZ synchronisé avec l'horloge d'écriture.

3.2.2. Horloge d'écriture

Le signal "horloge d'écriture" est nécessaire si l'option écriture des données en format NRZ est utilisée. Le signal "horloge d'écriture" est synchronisé avec les données d'écriture, comme indiqué dans la figure 5.



T1 - periode = 103.3 ± 2ns
T2 - Un a Zero = 155 ± 2ns
T3 - Zero a Un = 155 ± 2ns
T4 - largeur du signal Un = 50 ± 10ns
T5 - largeur du signal Zero = 50 ± 10ns
T6 - deplacement avant = 6 ± 1ns
T7 - déplacement après = 6 ± 1ns

Fig. 4 - Précompensation des signaux en écriture MFM

L'"horloge d'écriture" est l'"horloge servo" régénérée par le contrôleur. Elle est utilisée par l'unité disque pour obtenir les données MFM avec compensation d'écriture. Le signal d'horloge d'écriture peut être transmis d'une manière continue, pour éliminer les besoins d'ajustement de la phase de l'oscillateur "phase-locked" d'écriture.

Signaux provenant de l'unité de disque

4.2.3 Lecture des données

Avec des unités de disque standard, les données transmises de l'unité vers le contrôleur consistent en un signal de donnée série lu en MFM à la fréquence du disque. Chaque bit représente la forme des signaux dans le temps. La forme de l'onde possède quelques décalages des bits et variations dans la fréquence, dues aux variations de la vitesse du disque. Ces variations seront compensées dans l'unité de contrôle par le circuit de détection des bits de lecture.

4.2.4 Horloge de lecture

Le signal de lecture d'horloge est utilisé pour synchroniser les données en lecture si l'option NRZ (PLO/data separator) est utilisée. Le front ascendant du signal d'horloge de lecture doit être utilisé par l'unité de contrôle pour valider les données dans le registre de sérialisation. Le diagramme de temps de l'horloge de lecture et des données de lecture est représenté dans la figure 5.

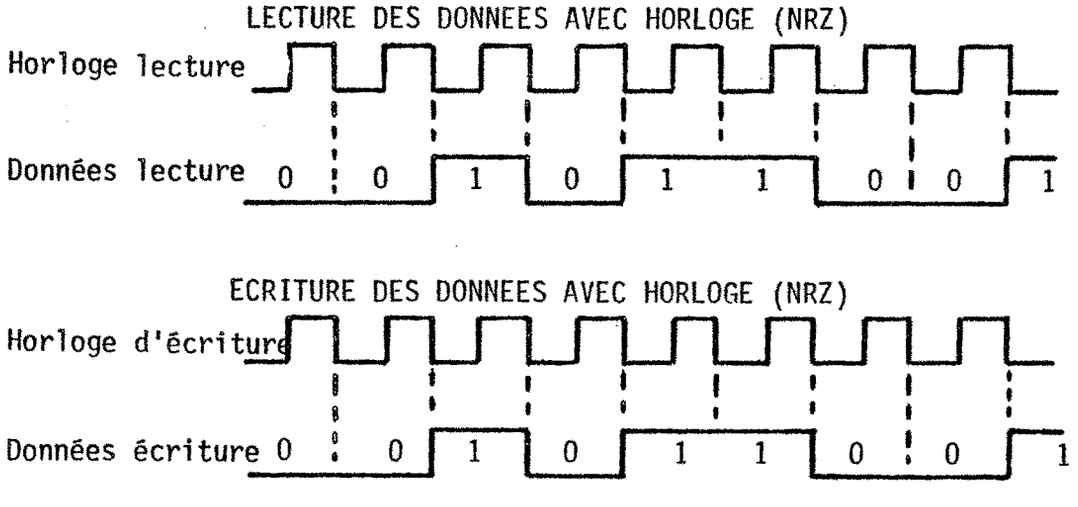


Fig.5 - Horlogerie pour l'écriture et lecture des données

4.2.5 L'horloge du servomécanisme

Le signal d'horloge du servo est généré par l'unité disque en lisant les signaux provenant de la surface servo du disque. La fréquence nominale de l'horloge du servo est de 9,677 MHz. Comme ce signal provient de la lecture du disque, toutes les variations dans la vitesse de rotation se traduisent par des changements de la fréquence de l'horloge du servo. Elle permet donc à l'unité de contrôle de synchroniser le signal des données écrites, à la vitesse du disque. L'horloge du servo est un signal symétriquement carré avec une période nominale de 103,3 ns.

4.2.6 Sélection (option)
(selected/unit selected)

Ce signal est transmis de l'unité disque vers le contrôleur, via le câble B, lorsqu'il y a eu une comparaison réussie entre l'adresse des lignes de sélection d'unité et l'adresse de l'unité disque (établie à travers un "plug", par exemple). Cette ligne sera activée dans les 200 ns qui suivent la réception par l'unité disque du front ascendant du signal de validation de sélection d'unité.

L'unité de contrôle doit examiner cette ligne dans chaque unité disque du système, avant d'activer la ligne de validation de sélection d'unité. De cette manière, le contrôleur peut être sûr que pas plus d'une unité disque répondra à la sélection. Les données de lecture et d'écriture et les horloges peuvent aussi être commandées par cette ligne à travers une porte.

4.2.6. Fin_de_recherche (option)

(seek end)

Le signal de Fin de Recherche est le même que le signal "Sur Cylindre" à la différence qu'il est dans le câble de lecture/écriture (câble B) et qu'il n'est pas commandé par le signal de validation d'unité disque. Son changement de l'état inactif à actif peut être utilisé par l'unité de contrôle comme indicateur. Une interruption indiquera que l'unité disque a accompli une opération et nécessite l'attention du contrôleur. Ce signal est normalement dans l'état actif. Les changements d'état (changement d'actif en inactif) sont dus à :

a) l'initialisation d'une opération de recherche : le signal deviendra actif quand la recherche sera normalement terminée ou par une condition d'erreur de recherche (recherche incomplète) ;
b) l'initialisation d'une opération de retour à zéro : le signal redeviendra actif quand l'opération de retour à zéro sera normalement terminée, ou s'il y a une condition d'erreur de recherche ;
c) des opérations "déplacement" : le signal restera inactif (également le signal sur cylindre) pendant à peu près 4 ms, pendant que la tête bouge jusqu'à sa position ;
d) le signal de fin de recherche restera inactif pendant que les têtes sont rétractées (par exemple pendant le changement d'un disque-pack) ; le signal redeviendra actif quand les têtes seront "sur cylindre" après l'opération de redémarrage du disque ;
e) le signal restera inactif si par exemple on retire le "plug", identificateur de l'unité disque.

Si une adresse de piste plus grande que la capacité du disque est envoyée par le contrôleur à l'unité disque, il n'y aura pas de changement d'état du signal fin de recherche. Cependant le signal erreur de recherche (recherche incomplète) deviendra actif dans les 100 ns. L'unité disque n'a pas besoin d'être sélectionnée pour que cette ligne soit activée.

A N N E X E 2

RAPPORT SUCCINT DE REALISATION DU PROCESSEUR DISQUE

A N N E X E 2

Rapport succint de réalisation du processeur disque

1 IMPLEMENTATION PHYSIQUE

1.1 Le protoype

Le prototype du processeur disque du PBD-MAGE est monté dans un système EXORCISER. Ce système Exorciser utilisé, comporte hors le châssis, une carte processeur avec le microprocesseur 6800, deux cartes de 16 K octets de mémoire dynamique, une carte avec le Système Exbug et le circuit de communication série pour l'interface avec le téléimprimeur et une carte pour l'interface avec des unités de disque souple. Le système comporte deux unités de disque souple.

1.2 Les cartes

Pour le développement du processeur disque deux cartes ont été construite qui concernent la partie contrôleur disque ; le microprocesseur étant l'Exorciser lui-même. La première carte, que nous appelons A, est concernée par les signaux de contrôle des unités disque et a été construite sur une carte standard à câbler (wire wrapping) du kit l'Exorciser. La deuxième carte, que nous appelons B, est concernée par le transfert de données vers ou provenant des unités disques. Elle a été construite sur une carte non standard de 25 x 30 cm, qui fait de l'ordre de deux fois la surface d'une carte standard. Cette carte peut être physiquement placée dans le châssis de l'Exorciser mais elle dépasse en hauteur.

La carte A se connecte directement au bus de l'Exorciser à travers un connecteur de 2 x 43 broches. Elle contient une cinquantaine de circuits intégrés de petite et moyenne echelle. Les connexions entre les circuits sont effectuées au moyen de la technique de "wire wrapping". De la carte A sortent trois connecteurs : a) le câble A, câble plat de 60 broches, qui réalise l'interface des signaux de contrôle entre le contrôleur et les unités disque ; b) un connecteur de 50 broches et un de 24 broches pour des câbles plats (flat cables), qui relie la carte A à la carte B.

La carte B se connecte à l'Exorciser à travers un câble de connexion qui la relie à la carte A. Elle contient une soixantaine de circuits en excluant la mémoire de microprogramme.

Ces circuits sont de petite, moyenne et grande échelle. Des circuits "Schotky" sont utilisés pour la partie du traitement des signaux d'horloge et de données (10 MHz). Cette carte possède deux circuits (2909) séquenceurs en tranche ainsi que deux unités arithmétiques (2901)en tranche. La technique du "wire wraping" a été également utilisée pour réaliser les connexions entre les circuits. La carte B contient plusieurs connecteurs :

a) deux connecteurs de 26 broches pour câbles plats qui permettent l'interface des signaux rapides avec les unités disques. Ce sont les câble B.

b) Six connecteurs de 24 broches pour câbles plats, utilisés pour relier le système de développement MADAM à la carte B. Ces connecteurs permettent de substituer la mémoire de microprogramme du contrôleur par celle de MADAM.

c) un connecteur de 50 broches plus un de 24 broches qui assurent la liaison entre la carte A et la carte B.

2 PROGRAMMES DE TESTS

La technique utilisée tout au long de la construction du prototype a été, d'implanter pas à pas chaque fonction et de la tester avant de passer à la suivante. Les premiers tests ont été réalisés à travers la simulation des ordres et la vérification du comportement des circuits avec un oscilloscope.

Tous les tests du plus simple au plus complexe plassent par le système de l'Exorcicer. L'unique moyen de dialogue avec la carte est ce système. Les programmes de tests sont écrits en "assembleur" et stockés sur disques souples.

Trois types de programmes de test ont été employés. Le premier, le plus simple et le plus primitif, consistait à envoyer des ordres directs à la carte grâce au moniteur MAID. Cela se résumait à écrire et lire les octets dans les registres. Comme les séquences d'activation d'un ordre aux unités disque sont très longues, ce type de test est assez pénible et n'avait d'intérêt que pour les premiers tests (Fig A2.1)

```
EXBUG 1.2 MAID
*0800/20 B7
*0801/03 CF        STAA Adresse CF00
0802/30 F0
0803/EE 7E         programme de chargement du registre 0
0804/04 08
0805/EE 00
*IR
P-E800 X-74FC A-CC B-78 C-E8 S-FF8A
P-E800 0800
X-74FC             Chargement du registre A
A-CC 01
*T1
END ADDR AD0C      Programme pas a pas
*N
P-0803 X-74FC A-01 B-78 C-E0 S-FF8A   éxecution de l'instruction
*N                                    de STAA sur le registre 0
P-0800 X-74FC A-01 B-78 C-E0 S-FF8A
```

Fig.A2.1 - Exemple d'utilisation de MAID pour tests

Le deuxième type de test était d'un niveau plus élevé et consistait à programmer les séquences de commandes nécessaires pour lancer un ordre. Nous avons très rapidement dépassé ce stade pour écrire des programmes paramétrables pour permettre de réaliser tout type de test (Fig A2.2.)

```
 LOAD SPEC;G
STATUS =Ø?                    Etat du disque

STROBE=Ø
DONNEE = Ø1                   Chargement du registre 0
SIGNAL PULSE ? N                   (adresse unité)
STATUS =Ø?

STROBE=1
DONNEE = ØØ                   Chargement du registre 1
SIGNAL PULSE ? N                (registre général - 8 a 9)
STATUS =Ø>
                              ADRESSE CYLINDRE
STROBE=2
DONNEE = ØØ                   Chargement du registre 2
SIGNAL PULSE ? N                (registre général - 0 a 7)
STATUS =Ø>

STROBE=3
DONNEE = Ø1                   Chargement du registre 3
SIGNAL PULSE ? N                (registre spécial)
STATUS =Ø>

STROBE=4
DONNEE = 9Ø                   Chargement du registre 4
SIGNAL PULSE ? OUI              (validatio de l'adresse cylindre)
STATUS =Ø>

STROBE=2
DONNEE = Ø1                   Chargement du registre 2
SIGNAL PULSE ? NON              (registre général - 0 a 7)
STATUS =Ø>                    ADRESSE TETE

STROBE=4
DONNEE = JØ                   Chargement du registre 4
SIGNAL PULSE ? OUI              (validation de l'adresse tête)
STATUS =Ø>
```

Fig.A2.2 - Exemple d'utilisation du programme de tests SPEC parametrable au niveau registre

Ces programmes à paramétrisation directe permettent de donner indépendament les ordres pour chaque registre.

Ces tests se sont montrés fastidieux lors de leur utilisation pour lancer des ordres plus complexes comme l'écriture. Les séquences d'ordres élémentaires devenaient trop longues, ce qui augmentait les erreurs de frappe ou d'ordre et avait comme conséquence de recommencer toute la séquence.

Un troisième type de programme de test a été utilisé, qui permettait de lancer les ordres à un niveau bien plus élevé semblable aux programmes des fonctions primitives utilisées par le processeur disque. A ce niveau, ces programmes de test fournissaient sur le téléimprimeur l'indication des erreurs survenues (Fig A2.3)

```
=LOAD SEEK;G
UNITE DISQUE=1
N. CYLINDRE=000
TETES SUR CYLINDRE TACHE ACCOMPLIE
FIN
=LOAD SEEK;G
UNITE DISQUE=1
N. CYLINDRE=824          Adresse cylindre trop grande
ERREUR DE RECHERCHE
FIN
=LOAD SEEK;G
UNITE DISQUE=1           Unité contrôleur pas branchée
N. CYLINDRE=100
UNITE PAS PRETE
FIN
=
```

Fig.A2.3 - Exemple d'utilisation du programme SEEK de test parametrable au niveau fonction

3 LA CARTE DES SIGNAUX DE CONTROLE - carte A

3.1 Aspects matériels

Le matériel de cette carte est complètement réalisé et testé. Sur cette carteil existe : les circuits amplificateurs différentiels de ligne, plus les résistances de terminaison nécessaires pour le câble A ; les circuits amplificateurs de ligne pour l'interface avec le bus de l'Exorciser ; la logique de décodage d'adresse ; le Registre de Validation ; le Registre Spécial (au niveau du prototype) ; le contôle des signaux du Registre d'Etat du Disque ; le Registre Adresse Mémoire. Hors cela elle possède des circuits pour fournir une alimentation de 5 V pour les circuits amplificateurs différentiels. La mémoire tampon (buffer) de 1 K octets a été implémentée aussi dans cette carte pour des motifs exposés plus loin dans cette annexe (Fig. A2.4).

3.2 Expériences et tests réalisés

Cette carte a été la première réalisée. Les premiers tests ont été fait au niveau électronique. Après que les amplificateurs différentiels de lignes ont été branchés, nous avons lancé des ordres vers les unités de disque. Par l'exécution d'ordres erronnées et l'observation des voyants lumineux du panneau avant des unités disque, il a été possible de vérifier s'il y avait un dialogue effectif. Avec la lecture de l'état du disque, les tests se sont ensuite améliorés.

Avec l'Exorciser et avec les programmes de test il a été possible de simuler les divers types d'ordres vers le contrôleur disque et donc vers les unités de disque.

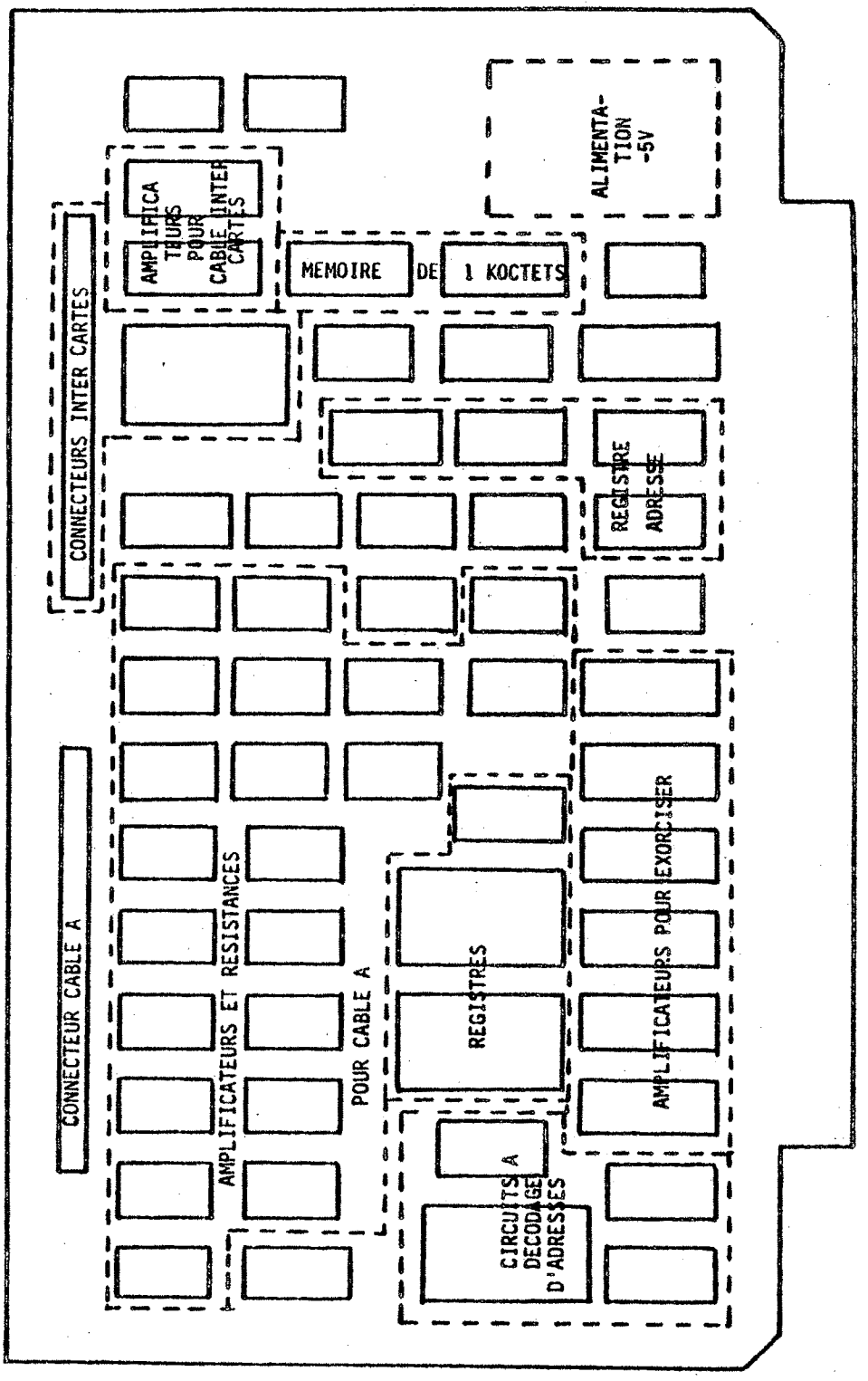


Fig.A2.4 - Vue inferieure de la carte A avec son implantation

4 LA CARTE DU TRANSFERT DES DONNEES - CARTE B

4.1 Aspect matériel

Le matériel de cette carte est complètement réalisé. Elle est responsable plus particulièrement du transfert des données entre la mémoire tampon (buffer) et les unités disque. Au niveau du prototype, elle a été construite pour être reliée à deux unités de disque. Sur la carte il existe : l'Unité de Séquencement et de Contrôle réalisée avec processeurs en tranche 2909 ; l'Unité Arithmétique et Logique réalisée avec les processeurs en tranche 2901 ; les circuits pour la Reconnaissance Secteur et le Registre Secteur ; les circuits amplificateurs différentiels de ligne pour communiquer avec les unités disque par l'intermédiaire du câble B ; les circuits amplificateurs de ligne pour la connexion avec la plaque A ; les circuits pour la Reconnaissance de Synchronisme ; le circuit à décalage et de CRC de l'Unité de Sérialisation et de Désérialisation ; le Registre de Sorties données ; le Registre d'Etat Contôleur ; le Registre de Commande ; ainsi que les circuits pour Horlogerie, pour le registre interne de microconditions, le registre de microinstructions, le registre de micro adresse et d'autres (Fig A2.5). Cette carte possède aussi des circuits pour générer une alimentation de - 5V pour les amplificateurs différentiels.

4.2 Expériences et tests réalisés

Le test de la plaque B a été bien plus complexe que celui de la plaque A. Pour le test des microprogrammes des processeurs en trnches on a utilisé MADAM. Cet outil de développement se substitue à la mémoire de microprogrammes permettant de modifier les microinstructions. Trois microprogrammes ont été écrits pour les tests : le formatage, l'écriture et la lecture.

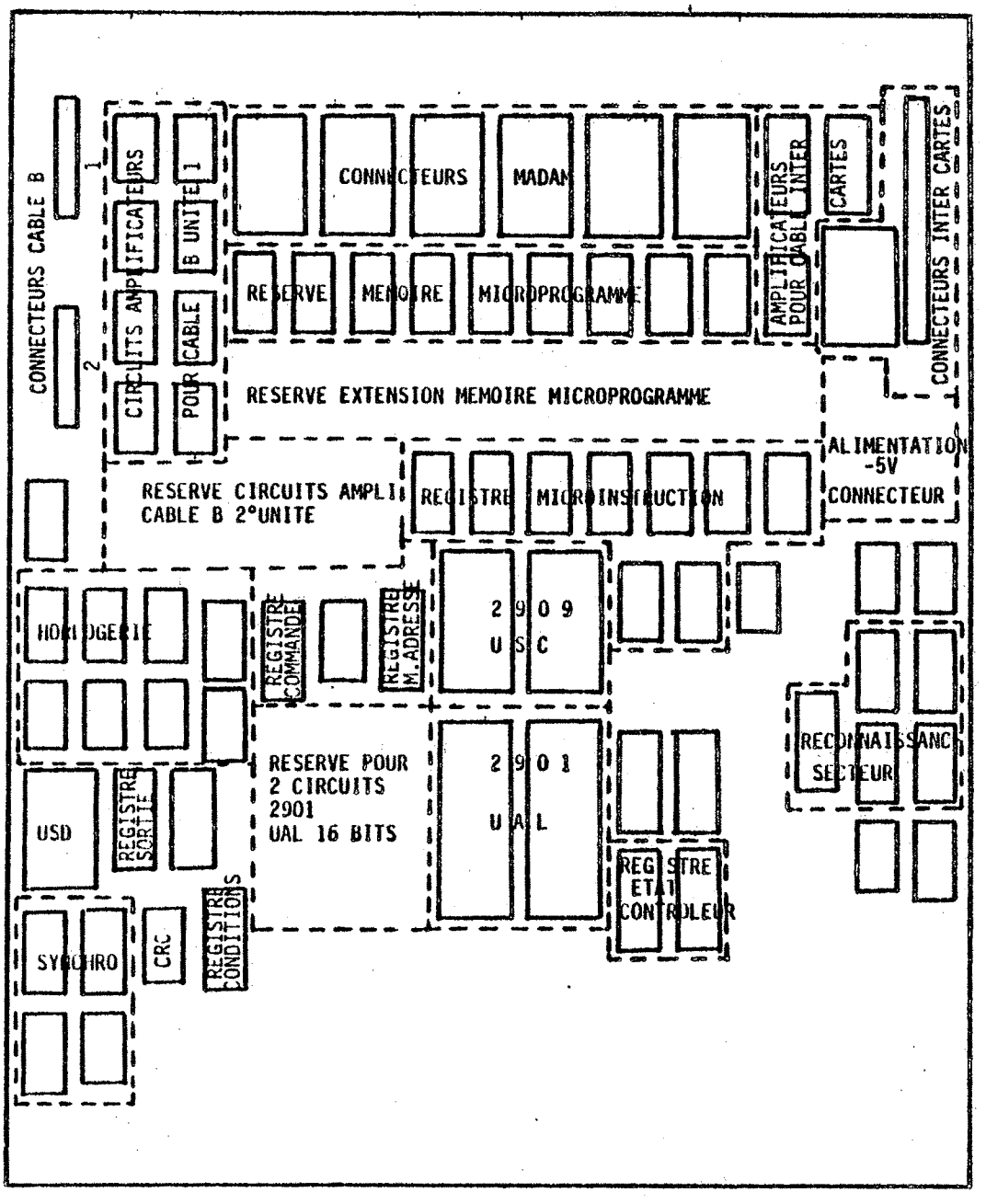


Fig.A2.5 - Vue inferieure de la carte B avec son implantation

La microinstruction des processeurs en tranche possède 44 bits. Pour réaliser les microprogrammes nous avons utilisé des fiches sur lesquelles était réalisé le codage au niveau du bit. Les microinstructions étaient après, concaténées de 8 en 8 bits pour créer une micro-instruction de 6 octets. Ces octets étaient utilisés pour la rentrée directe des microprogrammes sur MADAM (Fig A2.6).

De la même manière qu'avec les programmes de test, nous avons utilisé MADAM pour réaliser les tests des microprogrammes en pas à pas. Chaque fonction a été testée pour vérifier son activation à partir de l'USC.

Plusieurs autres tests électroniques ont été faits pour déterminer les temps de traversée des diverses unités du contrôleur disque. Cela a permis de s'assurer du bon fonctionnement des processeurs en tranche ainsi que de la partie d'horlogerie et du circuit de décalage (l'USD).

4.3 Description de l'outil de développement de microprogrammes MADAM

MADAM [SCH.78] est un outil de développement portable. Sa fonction principale est de remplacer la mémoire morte (ROM) de microprogramme d'une machine, par une mémoire vive RAM chargée à travers un microprocesseur. Il permet de réaliser la mise au point des microgrammes au niveau du prototype.

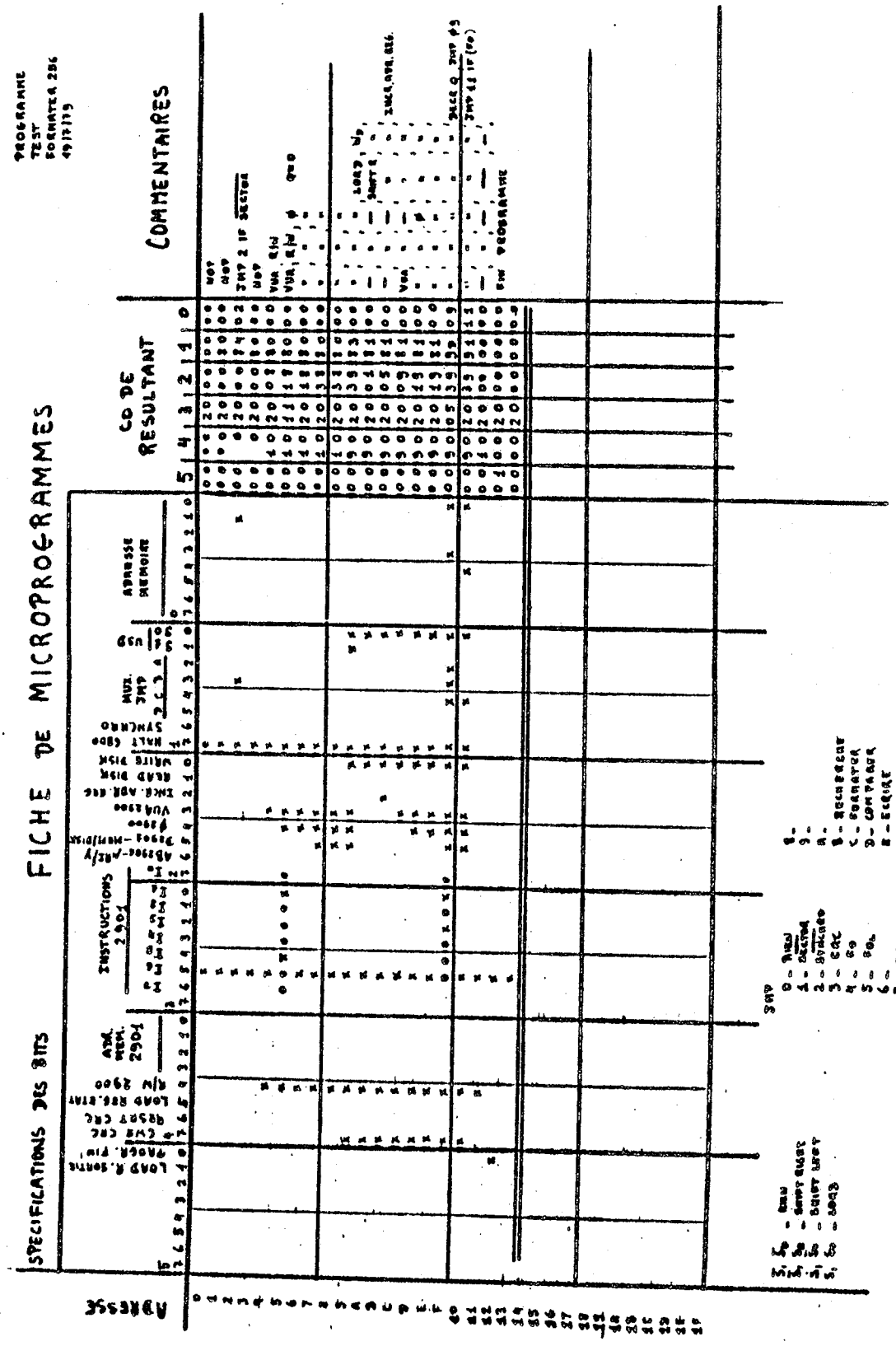
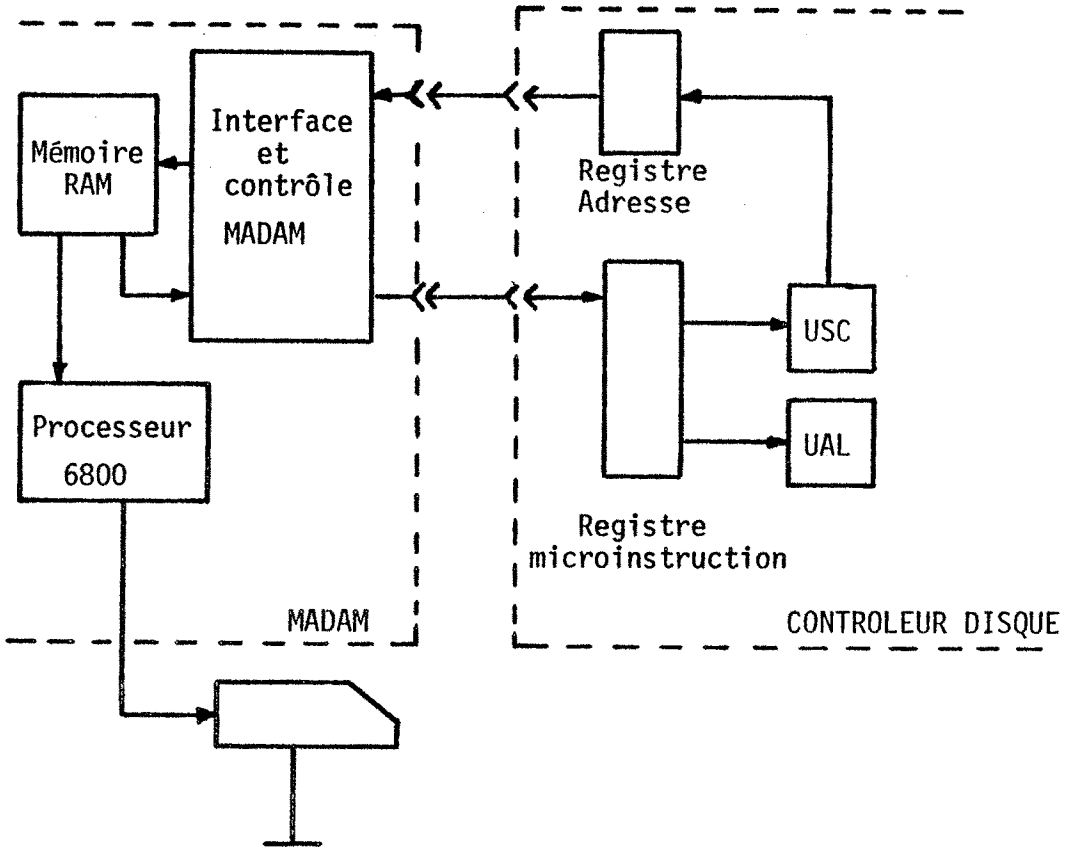


Fig.A2.6 - Fiche de microprogramme. Exemple d'un microprogramme de test de la fonctio "formatage"

Le système MADAM est contrôlé par un microprocesseur (6800) qui interprète le language de requêtes émises par l'utilisateur sur le clavier d'un téléimprimeur. Ce language de requêtes permet la définition de formats de micro-instructions (jusqu'à 21 champs de longueur maximum de 8 bits), le chargement ou la modification du microprogramme, ainsi que le contrôle de son exécution par la machine microprogrammée [SCH.78]. MADAM permet aussi de contrôler l'exécution d'un programme en pas-à-pas (Fig A2.7)



Fif A2.7 - L'outil MADAM et de CD

5 LA MEMOIRE TAMPON

La mémoire tampon (buffer) prévue premièrement à l'extérieur de la carte contrôleur disque a due cependant être constuite sur la plaque A pour la maquette. Cela est dû au fait que l'Exorciser n'admet pas l'accès direct à la mémoire sur son bus et que le transfert des données entre la mémoire et les unités disque doit être fait par DMA. Cette mémoire est de 1 K octets et représente deux buffers de 512 octets. L'USC réalise le contrôle du transfert des données en activant et désactivant les portes nécessaires pour la communication avec la mémoire. L'USC met le microprocesseur (Exorciser) en "Halt" désactive les portes de communication avec le bus Exorciser et active celles pour les données provenant de l'USD. A ce moment tout est prêt pour le transfert en DMA. L'allocation de la mémoire est sous le contrôle direct de l'USC.

A N N E X E 3

EXEMPLE DE PROGRAMME D'UTILISATION DU CONTROLEUR DISQUE :

PROGRAMME DE TEST DE FORMATAGE

```
PAGE  001   DEFOCO

00001                   * * * * * * * * * * * * * * * * * * * * * * * * *
00002                   *
00003                   *       PROGRAMME  FORMATER                     *
00004                   *      TEST DEMONSTRATION                       *
00005                   *                                               *
00006                   * * * * * * * * * * * * * * * * * * * * * * * * *
00007                   *
00008                   *  DONNEES DE ENTREE : N. UNITE  N. CYLINDRE
00009                   *        s00   N. SECTEUR
00010                   *
00011                           NAM    DEFOCO
00012                           OPT    REL,PAGE=60
00013           CFF0  A ADR     EQU    $CFF0     * ADRESSE INITIALE DES REG
00014           C000  A TAMPON  EQU    $C000
00015D 0000                     DSCT
00016D 0000     00    A CYLUP   FCB    $00
00017D 0001     00    A CYLDOW  FCB    $00
00018D 0002     00    A TETE    FCB    $00
00019D 0003     00    A SECT    FCB    $00
00020D 0004     0005  A BUF     RMB    5
00021D 0009     0005  A BUF1    RMB    5
00022D 000E     0005  A BUFSEC  RMB    5
00023D 0013     0005  A BUFTET  RMB    5
00024D 0018     55    A MSG     FCC    /UNITE DISQUE (X)= /
00025D 002A     04    A         FCB    $04
00026D 002B     4E    A MSG1    FCC    /N. CYLINDRE (XXX)= /
00027D 003E     04    A         FCB    $04
00028D 003F     55    A MSG3    FCC    /UNITE PAS PRETE/
00029D 004E     0D    A         FCB    $0D
00030D 004F     45    A MSG4    FCC    /ERREUR DE RECHERCHE/
00031D 0062     0D    A         FCB    $0D
00032D 0063     46    A MSG5    FCC    /FAUTE/
00033D 0068     0D    A         FCB    $0D
00034D 0069     54    A MSG6    FCC    /TETES SUR CYLINDRE /
00035D 007C     0D    A         FCB    $0D
00036D 007D     45    A MSG7    FCC    /ERREUR   PAS SUR CYLINDRE/
00037D 0096     0D    A         FCB    $0D
00038D 0097     53    A MSG8    FCC    /STATUS DE L UNITE DISQUE ANORMAL/
00039D 00B7     0D    A         FCB    $0D
00040D 00B8     46    A MSG9    FCC    /FIN/
00041D 00BB     0D    A         FCB    $0D
00042D 00BC     4E    A MSG10   FCC    /N. TETE (XX)= /
00043D 00CA     04    A         FCB    $04
00044D 00CB     45    A MSG11   FCC    /ERREUR TETE/
00045D 00D6     0D    A         FCB    $0D
00046D 00D7     54    A MSG12   FCC    /TETE PRETE/
00047D 00E1     0D    A         FCB    $0D
00048D 00E2     4E    A MSG13   FCC    /N. SECTEUR (XX)= /
00049D 00F3     04    A         FCB    $04
00050D 00F4     43    A MSG14   FCC    /CONTROLEUR PAS ACTIVE/
00051D 0109     0D    A         FCB    $0D
00052D 010A     43    A MSG15   FCC    /CONTROLEUR OCCUPE/
00053D 011B     0D    A         FCB    $0D
00054D 011C     41    A MSG16   FCC    /ANOMALIE CONTROLEUR/
00055D 012F     0D    A         FCB    $0D
00056D 0130     41    A MSG17   FCC    /ACTION FORMATAGE/
00057D 0143     0D    A         FCB    $0D
00058                   *
00059P 0000                     PSCT
00060P 0000 8E 0231   P FORM    LDS    #POIN     * INITIALISATION DU STACK
```

```
PAGE  002   DEF0C0

00061                    *
00062P 0003 CE 0018  D           LDX    #MSG      *
00063P 0006 3F                   SWI              *  ECRIRE MESSAGE UNITE DI
00064P 0007    0C    A           FCB    $0C       *  .DSPLZ
00065                    *
00066P 0008 C6 01    A           LDAB   #1        *
00067P 000A CE 0004  D           LDX    #BUF      *  LIRE N. UNITE
00068P 000D 3F                   SWI              *
00069P 000E    09    A           FCB    $09       *  .KEYIN
00070                    *
00071                    *
00072P 000F CE 002B  D           LDX    #MSG1     *
00073P 0012 3F                   SWI              *  ECRIRE N. CYLINDRE
00074P 0013    0C    A           FCB    $0C       * .DSPLZ
00075                    *
00076P 0014 C6 03    A           LDAB   #3        *
00077P 0016 CE 0009  D           LDX    #BUF1     *  LIRE N. CYLINDRE
00078P 0019 3F                   SWI              *
00079P 001A    09    A           FCB    $09       *  .KEYIN
00080                    *
00081                    *
00082P 001B B6 000B  D           LDAA   BUF1+2    *
00083P 001E 84 0F    A           ANDA   #$0F      * L UNITE
00084                    *
00085P 0020 5F                   CLRB
00086P 0021 3F                   SWI              *
00087P 0022    25    A           FCB    $25       *  B,A DANS X
00088                    *
00089P 0023 F6 000A  D           LDAB   BUF1+1    *
00090P 0026 C4 0F    A           ANDB   #$0F      *  DIZAINE
00091                    *
00092P 0028 27 08 0032 CYL3      BEQ    CYL2
00093P 002A 86 0A    A           LDAA   #$0A
00094                    *
00095P 002C 3F                   SWI              *
00096P 002D    28    A           FCB    $28       *  A+X DANS X
00097                    *
00098P 002E 5A                   DECB
00099P 002F 7E 0028  P           JMP    CYL3
00100                    *
00101P 0032 F6 0009  D CYL2      LDAB   BUF1      *  CENTAINE
00102                    *
00103P 0035 C4 0F    A           ANDB   #$0F
00104P 0037 27 08 0041 CYL5      BEQ    CYL4
00105P 0039 86 64    A           LDAA   #$64
00106                    *
00107P 003B 3F                   SWI              *
00108P 003C    28    A           FCB    $28       *  A+X DANS X
00109                    *
00110P 003D 5A                   DECB
00111P 003E 7E 0037  P           JMP    CYL5
00112                    *
00113P 0041 3F           CYL4    SWI              *
00114P 0042    24    A           FCB    $24       *  X DANS  B,A
00115                    *
00116P 0043 F7 0000  D           STAB   CYLUP
00117P 0046 B7 0001  D           STAA   CYLDOW
00118                    *
00119P 0049 B6 CFFD  A           LDAA   ADR+13    * LIRE STATUS
00120                    *                         CONTROLEUR
```

```
PAGE  003   DEFOCO

00121P 004C 48                ASLA
00122P 004D 25 38 0057        BCS     STAT
00123P 004F CE 00F4  D        LDX     #MSG14   *
00124P 0052 3F                SWI              *  ECRIRE CONTROLEUR
00125P 0053    0A    A        FCB     $0A      *  PAS ACTIVE
00126P 0054 7E 01C2  P        JMP     FIN
00127P 0057 48          STAT  ASLA
00128P 0058 24 08 0062        BCC     STAT2
00129P 005A CE 010A  D        LDX     #MSG15   *
00130P 005D 3F                SWI              * ECRIRE CONTROLEUR
00131P 005E    0A    A        FCB     $0A      *  OCCUPE
00132P 005F 7E 01C2  P        JMP     FIN
00133                  *
00134P 0062 B6 0004  D STAT2  LDAA    BUF      *
00135P 0065 84 0F    A        ANDA    #$0F     * LECTURE DU N. UNITE
00136P 0067 B7 CFF8  A        STAA    ADR      * ET CHARGEMENT DU REGISTR
00137                  *
00138P 006A 86 01    A        LDAA    #$01     *
00139P 006C B7 CFF3  A        STAA    ADR+3    * ACTIVER OPEN CABLE
00140                  *
00141P 006F 86 80    A        LDAA    #$80     *
00142P 0071 B7 CFF4  A        STAA    ADR+4    *  VALIDATION DE L UNITE
00143                  *
00144P 0074 B6 CFF7  A        LDAA    ADR+7    *LIRE STATUS
00145                  *
00146P 0077 84 20    A        ANDA    #$20     *  MASQUE
00147                  *
00148P 0079 2E 11 008C        BGT     CYLIN    * SAUT SI PLUS GRAND QUE Z
00149                  *
00150P 007B CE 003F  D        LDX     #MSG3    *
00151P 007E 3F                SWI              *  ECRIRE NOT READY
00152P 007F    0A    A        FCB     $0A      *
00153                  *
00154P 0080 7E 01C2  P        JMP     FIN
00155                  *
00156P 0083 7E 01A0  P SANOM  JMP     ANOM
00157P 0086 7E 01B8  P SANOM2 JMP     ANOM2
00158P 0089 7E 01AB  P SERSEK JMP     ERSEK
00159P 008C B6 CFF7  A CYLIN  LDAA    ADR+7    *LIRE STATUS DU DISQUE
00160                  *
00161P 008F 84 C0    A        ANDA    #$C0     *  ANORMAL
00162P 0091 2E F0 0083        BGT     SANOM    * SAUT SI ZERO (SEEK ERROR
00163                  *
00164P 0093 B6 CFF7  A        LDAA    ADR+7    *
00165                  *
00166P 0096 84 01    A        ANDA    #01      *  ANORMAL , NOT ON CYLIND
00167P 0098 27 EC 0086        BEQ     SANOM2   * SAUT SI ZERO
00168                  *
00169P 009A B6 0001  D        LDAA    CYLDOW   *
00170P 009D F6 0000  D        LDAB    CYLUP    *
00171P 00A0 B7 CFF2  A        STAA    ADR+2    * A DANS REG. 2
00172P 00A3 F7 CFF1  A        STAB    ADR+1    * B DANS REG 1
00173                  *
00174P 00A6 86 90    A        LDAA    #$90     *
00175P 00A8 B7 CFF4  A        STAA    ADR+4    *  VALIDATION
00176P 00AB 86 00    A        LDAA    #$00     *  ADRESSE
00177P 00AD B7 CFF4  A        STAA    ADR+4    *  CYLINDRE
00178                  *
00179P 00B0 B6 CFF7  A CYL6   LDAA    ADR+7    *
00180P 00B3 48                ASLA             *  TESTER SEEK ERROR
```

```
PAGE  004   DEFOCO

00181P 00B4 25 D3 0089        BCS    SERSEK   *
00182                  *
00183P 00B6 84 02    A        ANDA   #$02     *
00184P 00BB 27 F6 00B0        BEQ    CYL6     *  TESTER SI ON CYLINDER
00185                  *
00186P 00BA CE 0069  D        LDX    #MSG6    *
00187P 00BD 3F                SWI             *   ECRIRE TETES
00188P 00BE    0A    A        FCB    $0A      *   SUR CYLINDRE
00189                  *
00190P 00BF CE 00BC  D        LDX    #MSG10   *
00191P 00C2 3F                SWI             *  ECRIRE
00192P 00C3    0C    A        FCB    $0C      *  N.TETE
00193                  *
00194P 00C4 C6 02    A        LDAB   #2,      *
00195P 00C6 CE 0013  D        LDX    #BUFTET  *  LIRE
00196P 00C9 3F                SWI             *  N. TETE
00197P 00CA    09    A        FCB    $09      *
00198                  *
00199P 00CB B6 0014  D        LDAA   BUFTET+1 *
00200P 00CE 84 0F    A        ANDA   #$0F     *  UNITE
00201                  *
00202P 00D0 F6 0013  D        LDAB   BUFTET   *
00203P 00D3 C4 0F    A        ANDB   #$0F     *  DIZAINE
00204                  *
00205P 00D5 27 06 00DD TET3   BEQ    TET2
00206P 00D7 8B 0A    A        ADDA   #$0A     * A+$A=A
00207P 00D9 5A                DECB
00208P 00DA 7E 00D5  P        JMP    TET3
00209                  *
00210P 00DD B7 CFF2  A TET2   STAA   ADR+2
00211P 00E0 B7 0002  D        STAA   TETE
00212P 00E3 4F                CLRA
00213P 00E4 B7 CFF1  A        STAA   ADR+1
00214                  *
00215P 00E7 86 A0    A        LDAA   #$A0     *
00216P 00E9 B7 CFF4  A        STAA   ADR+4    *  VALIDATION
00217P 00EC 86 80    A        LDAA   #$80     *   TETE
00218P 00EE B7 CFF4  A        STAA   ADR+4    *
00219                  *
00220P 00F1 B6 CFF7  A        LDAA   ADR+7
00221P 00F4 84 40    A        ANDA   #$40
00222P 00F6 27 0B 0100        BEQ    TET4
00223                  *
00224P 00F8 CE 00CB  D        LDX    #MSG11   *
00225P 00FB 3F                SWI             * ECRIRE
00226P 00FC    0A    A        FCB    $0A      * ERREUR TETE
00227P 00FD 7E 01C2  P        JMP    FIN
00228                  *
00229P 0100 CE 00D7  D TET4   LDX    #MSG12   *
00230P 0103 3F                SWI             *  ECRIRE TETE
00231P 0104    0A    A        FCB    $0A      * SUR PLACE
00232                  *
00233P 0105 CE 00E2  D        LDX    #MSG13   *
00234P 0108 3F                SWI             *  ECRIRE
00235P 0109    0C    A        FCB    $0C      *  N.SECTEUR
00236                  *
00237P 010A C6 02    A        LDAB   #2       *
00238P 010C CE 000E  D        LDX    #BUFSEC  *  LIRE
00239P 010F 3F                SWI             *  SECTEUR
00240P 0110    09    A        FCB    $09      *
```

```
PAGE  005   DEFOCD

00241                    *
00242P 0111 B6 000F  D        LDAA   BUFSEC+1 *
00243P 0114 84 0F    A        ANDA   #$0F     *  UNITE
00244                    *
00245P 0116 F6 000E  D        LDAB   BUFSEC   *
00246P 0119 C4 0F    A        ANDB   #$0F     *  DIZAINE
00247                    *
00248P 011B 27 06 0123 SEC2   BEQ    SEC1     *
00249P 011D 8B 0A    A        ADDA   #$0A     *  A+0A=A
00250P 011F 5A                DECB
00251P 0120 7E 011B  P        JMP    SEC2
00252                    *
00253P 0123 B7 CFF8  A SEC1   STAA   ADR+8     *CHARGER REG ADR SEC
00254P 0126 B7 0003  D        STAA   SECT
00255                    *
00256P 0129 CE 0130  D        LDX    #MSG17   *
00257P 012C 3F                SWI             *  ECRIRE
00258P 012D    0A    A        FCB    $0A      * ACTION FORMATAGE
00259                    *
00260P 012E CE C000  A        LDX    #TAMPON
00261P 0131 86 44    A        LDAA   #$44
00262P 0133 A7 00    A        STAA   0,X
00263P 0135 08                INX
00264                    *
00265P 0136 C6 1A    A        LDAB   #26
00266P 0138 4F                CLRA
00267P 0139 A7 00    A TAMP   STAA   0,X      *
00268P 013B 08                INX             *  26 ZEROS
00269P 013C 5A                DECB            *
00270P 013D 26 FA 0139        BNE    TAMP     *
00271                    *
00272P 013F 43                COMA            *
00273P 0140 A7 00    A        STAA   0,X      *  1 ONE
00274                    *
00275P 0142 4F                CLRA            *
00276P 0143 08                INX             * STATUS
00277P 0144 A7 00    A        STAA   0,X      * SECTEUR
00278                    *
00279P 0146 08                INX             *
00280P 0147 B6 0000  D        LDAA   CYLUP    * CYLUP
00281P 014A A7 00    A        STAA   0,X      *
00282                    *
00283P 014C B6 0001  D        LDAA   CYLDOW   *
00284P 014F 08                INX             *  CYLDOW
00285P 0150 A7 00    A        STAA   0,X      *
00286                    *
00287P 0152 B6 0002  D        LDAA   TETE     *
00288P 0155 08                INX             *  TETE
00289P 0156 A7 00    A        STAA   0,X      *
00290                    *
00291P 0158 B6 0003  D        LDAA   SECT     *
00292P 015B 08                INX             *  SECTEUR
00293P 015C A7 00    A        STAA   0,X      *
00294                    *
00295P 015E 4F                CLRA            *
00296P 015F 08                INX             *
00297P 0160 A7 00    A        STAA   0,X      *  CRC
00298P 0162 08                INX             *
00299P 0163 A7 00    A        STAA   0,X      *
00300                    *
```

```
PAGE  006   DEFOCO

00301P 0165 C6 0C    A        LDAB   #12
00302P 0167 08                INX
00303P 0168 4F                CLRA
00304P 0169 A7 00    A TAP2   STAA   0,X      *
00305P 016B 08                INX              *
00306P 016C 5A                DECB             * 12 ZEROS
00307P 016D 26 FA 0169        BNE    TAP2      *
00308                 *
00309P 016F C6 FF    A        LDAB   #255
00310P 0171 43                COMA
00311P 0172 A7 00    A TAP3   STAA   0,X      *
00312P 0174 08                INX              *
00313P 0175 5A                DECB             *  256 ZEROS
00314P 0176 26 FA 0172        BNE    TAP3      *
00315                 *
00316P 0178 4F                CLRA             *
00317P 0179 A7 00    A        STAA   0,X       *  2 CRC
00318P 017B 08                INX              *
00319P 017C A7 00    A        STAA   0,X       *
00320                 *
00321P 017E 08                INX              *
00322P 017F A7 00    A        STAA   0,X       * 1 ZERO
00323                 *
00324P 0181 86 C0    A        LDAA   #$C0
00325P 0183 B7 CFFF  A        STAA   ADR+15
00326P 0186 86 00    A        LDAA   #$00
00327P 0188 B7 CFFE  A        STAA   ADR+14
00328P 018B 4F                CLRA
00329P 018C B7 CFF1  A        STAA   ADR+1
00330P 018F B7 CFF2  A        STAA   ADR+2
00331P 0192 86 C0    A        LDAA   #$C0
00332P 0194 B7 CFF4  A        STAA   ADR+4
00333                 *
00334P 0197 B7 CFFA  A        STAA   ADR+10  *ACTIVER CONTROLEUR
00335P 019A 01                NOP
00336P 019B 01                NOP
00337P 019C 01                NOP
00338                 *
00339P 019D 7E 01C2  P        JMP    FIN
00340                 *
00341P 01A0 CE 0097  D ANOM   LDX    #MSG8    *
00342P 01A3 3F                SWI             *   ECRIRE STATUS DU
00343P 01A4    0A    A        FCB    $0A      *   DISQUE ANORMAL
00344                 *
00345P 01A5 48                ASLA             *  TESTER BIT
00346P 01A6 24 05 01AD        BCC    FAULT     *  ERREUR DE RECHERCHE
00347                 *
00348P 01A8 CE 004F  D ERSEK  LDX    #MSG4    *
00349P 01AB 3F                SWI             *  ECRIRE ERREUR DE
00350P 01AC    0A    A        FCB    $0A      *   RECHERCHE
00351                 *
00352P 01AD 48        FAULT   ASLA             *
00353P 01AE 24 12 01C2        BCC    FIN       *  TESTER BIT DE FAUTE
00354                 *
00355P 01B0 CE 0063  D        LDX    #MSG5    *
00356P 01B3 3F                SWI             * ECRIRE FAUTE
00357P 01B4    0A    A        FCB    $0A      *
00358                 *
00359P 01B5 7E 01C2  P        JMP    FIN
00360                 *
```

```
PAGE  ##7   DEFOCO

##361P #188 CE ##97  D ANOM2  LDX    #MSGB    *
##362P #18B 3F               SWI              * ECRIRE STATUS DISQUE
##363P #18C    #A    A       FCB    $0A       * ANORMAL
##364                 *
##365P #18D CE ##7D  D       LDX    #MSG7     *
##366P #1C# 3F               SWI              *  ECRIRE PAS SUR
##367P #1C1    #A    A       FCB    $0A       *   CYLINDRE
##368                 *
##369P #1C2 4F        FIN    CLRA             *
##370P #1C3 B7 CFF4  A .     STAA   ADR+4     * DESACTIVER UNITE DISQUE
##371                 *
##372P #1C6 CE ##B8  D       LDX    #MSG9     *
##373P #1C9 3F               SWI              *  ECRIRE FIN
##374P #1CA    #A    A       FCB    $0A       *
##375                 *
##376P #1CB 3F               SWI              *
##377P #1CC    1A    A       FCB    $1A       *  RETOUR A L EXOR
##378                 *
##379P #1CD    ##64  A       RMB    100       *
#w380P #231    ##    A POIN  FCB    $00       *  MEMOIRE POUR STACK
##381                 *
##382          ####  P       END    FORM
TOTAL FNRORS #####
```

REFERENCES

[ABR.70] ABRIAL, BAS, BEAUME, HENNERON, MORIN, VIGLIANO
"Projet SOCRATE : Spécifications générales", Université de Grenoble, Août 1970.

[ANC.76] F. ANCEAU
"Architecture des ordinateurs - Tome 1 - Architecture des unités de traitement classiques", ENSIMAG, Avril 1976, 3ème version, Cours.

[ANC.77] F. ANCEAU, N.J. HADJIDAKIS
"Organisation d'un mécanisme de communication pour un système transactionnel réparti de manière fonctionnelle", Rapport de Recherche n° 96, ENSIMAG, Novembre 1977.

[ANC.78] F. ANCEAU
"Systèmes fonctionnellement distribués", Congrès AFCET, Paris, Novembre 1978, p. 85-94.

[AND.75] G.A. ANDERSON, E.D. JENSEN
"Computer inter-connection structures : taxonomy, characteristics and exemples", Computer Surveys, Vol. 7, n° 4, Décembre 1975, p. 197-213.

[AND.76] D.R. ANDERSON
"Data Base processor technology", AFIPS Conf., Proc. 76 NCC, Vol. 45, Juin 1976, p. 811-818.

[BAB.79] E. BABB
"Implementing a Relational Database by Means of Specialized Hardware", ACM Trans. DB Syst., Vol. 4, n° 1, Mars 1979, p. 1-29.

[BAU.76] R.I. BAUM, D.K. HSIAO
"Data base computers - A step toward data utilities", IEEE Trans. on Comp., Vol. C-25, n° 12, Décembre 1976, p. 1254-1259.

[BEL.71] BELL & NEWELL
"Computer Structures : Readings and Examples", Mc Graw Hill, 1971.

[BER.77] G. BERGER-SABBATEL, M. SARRE
"MAGE : Matériel Adapté à la Gestion d'Entités", Journées AFCET, Processeur Base de données, Paris, Mars 1977, p. 1-23.

[BER1.78] G. BERGER-SABBATEL
"Etude fonctionnelle d'un processeur de base de données hiérarchiques", Thèse docteur 3ème Cycle, ENSIMAG, Grenoble, Juin 1978.

[BER2.78] G. BERGER-SABBATEL, Ph. NAVAUX, M. SARRE
"Conception d'un processeur base de données adapté à des petits systèmes", Congrès AFCET, Paris, Novembre 1978, p. 48-58.

[BHA.79] D.P. BHANDARKAR, J.B. BARTON, A.F. TASCH Jr.
"CCD-Charge-Coupled Device Memories : A Perspective", Computer IEEE, Janvier 1979, p. 16-24.

[BOO.76] GRAYCE M. BOOTH
"Distributed information systems", AFIPS-NCC 76, p. 789-794.

[CAS.78] CASWELL, BROWN, CARTER, and all
"Basic Technology-Oregon Report", IEEE Computer, Septembre 1978. p. 10-19

[CAN.74] R.H. CANADAY, HARRISON, INIE, RYDER, WEHR
"A back-end computer for data management", CACM, Vol. 17, n° 10, Octobre 1974, p. 575-582.

[COP.73] G.P. COPELAND, G.J. LIPOVSKI, S.Y. SU
"The Architecture of CASSM; A Cellular System for non-numeric processing", Proc. 1th An Symposium on Comp. Arch., Décembre 1973, p. 121-128.

[COU.79] B. COURTOIS
"Some results about the efficiency of simple mechanisms for the detection of microcomputer malfunctions", 9th Fault-Tolerant Computing Symposium, Juin 1979, USA, p. 71-74.

[ENS.78] Philip H. ENSLOW Jr.
"What is a Distributed Data Processing System ?", Computer, Janvier 1978, p. 13-21.

[FIO.73] C.R. DE FIORE, P.B. BERRA
"A data management system utilizing an associative memory", AFIPS-NCC, Vol. 42, Juin 1973, p. 181-185.

[FRY.76] J.P. FRY, E.H. SIBLEY
"Evolution of date base management systems", Computer Surveys, Vol. 8, n°1, Mars 1976, p. 7-42.

[HAD.79] N. HADJIDAKIS, Ph. NAVAUX
"Architecture logicielle et matérielle d'un processeur fonctionel dans le cadre du système réparti CORAIL V2-Processeur base de données", Journées BIGRE, Nancy, Janvier 1979, p. 140-163.

[HAU.75] K.E. HAUGHTON
"An overview of Disk Storage Systems", Proc. of IEEE, Vol. 63, n° 8, Août 1975, p. 1148-1152.

[HSI.77] D.K. HSIAO, K. KANNAN
"The Architecture of a Database Computer-A Summary", 3th Workshop on Comp. Arch. for Non-Numeric Proc., Mai 1977, p. 31-34.

[HSI.79] David K. HSIAO
"Data Base Machines are Coming", Guest Editor's Introduction, IEEE Computer, Mars 1979, p. 7-9.

[JEN.76] E. DOUGLAS JENSEN
"A Distributed Function Computer for Real-Time Control", 2nd An. Symp. on Comp. Arch., 1975, p. 176-180.

[KAN.78] K. KANNAN
"The design of a Mass Memory for a data base Computer", 5th An. Symp. on Comp. Arch., 1978, p. 44-51.

[LAU.89] S.Y. LAU, M. JANAK
"To cut controler costs, replace random logic with microcoded bit slices", Electronic Design, n2, 18 Janvier, p. 90-97.

[LIN.76] C.S. LIN, D.C.P. SMITH, J.M. SMITH
"The design of a Rotating Associative Memory for Relational Database Applications", ACM Trans. on DB Syst., Vol. 1, n°1, Mars 1976, p. 53-65.

[LIP.76] G.J. LIPOVSKI
"Non-Numeric Architecture", Rapport interne du Dep. of Electrical Engineerign, University of Texas, 1976-1977.

[LIP.78] G.J. LIPOVSKI
"Architectural Features of CASSM : A Context Adressed Segment Sequential Memory", 5th An. Symp. on Computer Architecture, 1978, p. 31-38.

[MAD.75] S.E. MADNICK
"INFOPLEX - Hierarchical Decomposition of a large Information Management System Using a Microprocessor Complex", AFIPS Conf. Proc. 1975 NCC, Vol. 44, Mai 1975, p. 581-586.

[MAR.75] T. MARILL, D. STERN
"The Data Computer - A network data utility", AFIPS Conf. Proc. 1975 NCC, Vol. 44, Mai 1975, p. 389-395.

[MAR.78] F.J. MARYANSKI
"A survey of developments in distributed data base management systems", Computer, Vol. 11, n°2, Février 1978, p. 28-38.

[MAS.78] A. EL MASRI, J. ROHMER, D. TUSERA
"A Machine for Information Retrieval", 4th Workshop on Comp. Arch. for Non-Numeric Proc., Août 1978, p. 117-120.

[MCG.76] Mc GREGOR, THOMPSON, DAWSON
"High Performance Hardware for Data Base Systems", Systems for Large DB, Lockemann and Newfold edit, North Holland Publishing Co., 1976, p. 103-116.

[MUK.78] Amar MUKHOPADHYAY
"Hardware Algorithms for Nonumeric Computation", 5th An. Symp. on Comp. Arch., 1978, p. 8-16.

[NAV.77] Ph. NAVAUX
"Processeur base de données MAGE - étude du matériel", Rapport final contrat SAGEM n° 852 068, ENSIMAG, Grenoble, Juin 1977.

[NAV.79] Ph. NAVAUX, G. BERGER-SABBATEL
"A data base processor : hardware design and implementation based on MAGE", Euromicro Symposium, Août 1979, p. 91-98.

[NEU.66] J. VON NEUMANN, edited by A.W. BURKS
"Theory of Self-Reproducing Automata", University of Illinois Press, 1966.

[OZK.75] E.A. OZKARAHAN, S.A. SCHUSTER, K.C. SMITH
"RAP - An associative processor for data base management", NCC-AFIPS 1975, Vol. 44, p. 379-387.

[OZK.77] E.A. OZKARAHAN, K.C. SEVCIK
"Analysis of Architectural Features for Enhancing the Performance of a Database Machine", ACM Trans. on Database Syst., Vol. 2, n° 4, Décembre 1977, p. 297-316.

[PIR.79] S.V. PIRES, VATTON
"MARIUS - Moniteur d'Applications Réparties sur une Interconnexion d'Unités Spécialisées", Rapport interne SAGEM-ENSIMAG, Mars 1979.

[POU.77] G.H. POUJOULAT
"Architecture of the CORAIL building block system", 4th Annual Symp. on Comp. Arch., 1977, p. 201-204.

[QUA.77] H.H. QUANG, J. SUCHARD, J. VIDALIN
"Un processeur associatif itéractif et modulaire de reprise de données", Journées AFCET-Processeurs BD, Mars 1977, p. 25-42.

[SAD.78] P.J. SADOWSKI, S.A. SCHUSTER
"Exploiting Parallelism in a Relational Associative Processor", 4th Workshop on Comp. Arch. for Non-Num. Proc., AoQt 1978, p. 99-10

[SAG.76] SAGEM-ENSIMAG
"Etude et évaluation sur maquette de l'implantation hardware d'algorithmes de gestion de fichiers adpatés aux petits systèmes", Contrat IRIA n° 74.136, Rapport final, Juillet 1976.

[SCH.76] S.A. SCHUSTER, E.A. OZKARAKAN, K.C. SMITH
"A virtual memory system for a relational associative processor", NCC-AFIPS 1976, p. 855-862.

[SCH.78] S.A. SCHUSTER, H.B. NGUYEN, E.A. OZKARAHAN, K.C. SMITH
"RAP.2 - An Associative Processor for Data Bases", 5th An. Symp. on Computer Architecture 1978, p. 52-59.

[SCH.79] F. SCHWAAB
"Contribution à l'étude d'un processeur de consultation de données - Les communications et l'évaluation", Thèse, Université de Nancy I, Juin 1979.

[SLA.78] M. SLANA, G. DUMAS
"Workshop Report : The new and the not so new", IEEE Computer, Mars 1978, p. 47-51.

[SMI.78] A.J. SMITH
"On the Effectiveness of Buffered and Multiple Arm Disks", 5th An. Symp. on Comp. Arch. 1978, ACM SIGARCH, p. 242-248.

[SMI.79] Diane C.P. SMITH, John Miles SMITH
"Relational Data Base Machines", IEEE Computer, Mars 1979, p. 28-38.

[STR.79] E. STRITTER, T. GUNTER
"A Microprocessor Architecture for changing World : The Motorola 68.000", IEEE Computer, Février 1979, p. 43-52.

[SU.79] Stanley Y.W. SU
"Cellular-Logic Devices : Concepts and Applications", IEEE Computer, Mars 1979, p. 11-25.

[THU.78] Kenneth J. THURBER
"Computer Communication Techniques", Computer Architecture News, ACM-SIGRAPH, Vol. 7, n°3, Octobre 1978, p. 7-16.

[TOO.78] D. TOOMBS
"CCD and bubble memories : system implications", IEEE Spectrum, Mai 1978, p. 36-39.

[TOO1.78] Dean TOOMBS
"An update : CCD an bubble memories", IEEE Spectrum, Avril 1978, p. 22-30.

[UNG.77] L. UNGARO
"Processeur associatif pour base de données relationnelle", Journées AFCET-Processeur Base de Données, Mars 1977, p. 43-60.

[VRA.76] Zvonko G. VRANESIC, Safwat G. ZAKY
"Nonnumeric Applications of Microprocessors", Proc. IEEE, Vol. 64, n°6, Juin 1976, p. 954-959.

[YAO.78] S.B. YAO, P.A. BERNSTEIN, N. GOODMAN, S.A. SCHUSTER, D.W. SHIPMAN, D.C.P. SMITH
"Data Base Systems", IEEE Computer, Septembre 1978, p. 46-60.

[YPM.75] John E. YPMA
"Bubble domain memory systems", NCC 1975, AFIPS, p. 523-528.

[ZAK.77] S.G. ZAKY
"Microprocessors for Non-Numeric Processing", 3th Workshop on Comp. Arch. for Non-Numéric Processing, Mai 1977, p. 23-30.

REFERENCES TECHNIQUES

[T.AMD.78] Advanced Micro Devices
The AM 2900 Family Data Book, 1978.

[T.AMP.77] AMPEX
"Disk storage Drives-Engineering Specifications DM940, DM980", Juillet 1977.

[T.CDC.77] Control Data Corporation
Product Specification for the 9760 drive family, Janvier 1977.

[T.CDC1.77] Control Data Corporation
Product Specification for the Mini-module drive family 9730, Juin 1977.

[T.INT.75] Intel
Intel Series 3000 Reference Manual, 1975.

[T.MOT.75] Motorola
M 6800. Exorciser User's Guide, Benchmark Family for Microcomputer Systems, 1975, second edition.

[T.MOT.76] Motorola
M 6800, Microcomputer system design data, 1976.

[T.MOT.77] Motorola
Motorola Microsystems Family-Micromodules (plusieurs), 1977.

[T.NAV.78] NAVAUX
Spécifications de connexion entre disque pack de 10MHz et contrôleurs - Normes SMD, Version 1, Communication interne SAGEM-ENSIMAG, Janvier 1978.

[T.NAV1.78] NAVAUX
Unité de contrôle de disque - version 1, Communication interne SAGEM-ENSIMAG, Février 1978.

[T.SAG.74] SAGEM-SEREL
"SCRIB-Saisie et Contrôle de Résultats et d'informations Biologiques - Manuel d'Exploitation", 4 manuels, 1974.

[T.SCH.78] J.P. SCHOELLKOPF
Présentation du Système MADAM : Matériel D'Aide au Développement D'Applications Microprogrammées, Communication Interne, ENSIMAG, Mai 1978.

[T.TEX.77] Texas Instruments
The TTL Data Book for Design Engineers, 1977.

Si c'était à refaire . . . .

je le ferai autrement . . .

dernière page de la thèse

AUTORISATION DE SOUTENANCE

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

VU les rapports de présentation de Messieurs :

- F. ANCEAU, Professeur à l'Institut National Polytechnique de GRENOBLE
- J. ROHMER, Chercheur à l'I.R.I.A. - LE CHESNAY -

Monsieur Philippe NAVAUX

est autorisé à présenter une thèse en soutenance pour l'obtention du diplôme de DOCTEUR-INGENIEUR, spécialité "Génie Informatique".

Grenoble, le 12 Novembre 1979

Le Président de l'I.N.P.G.

Ph. TRAYNARD
Président
de l'Institut National Polytechnique