

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE D'ALGER

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE

34/83

2ex



PROJET DE FIN D'ETUDES

THESE D'INGENIORAT

SUJET

**ETUDE ET REALISATION: CONTROLEUR
DE DISQUE SOUPLE**

Proposé par : **Mr A. BOURKEB**

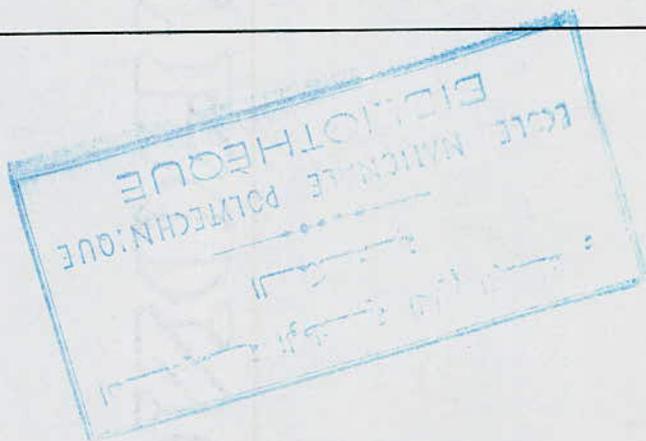
Suivi par : **Mr A. SAIDI**

Realisé par : **Melles KASMI HABIBA
ZEBIRI SAMIRA**

JUIN 83

SUJET

ETUDE ET REALISATION
D'UN CONTROLEUR DE DISQUE SOUPLE



EXTRA STIRPES

DEDICACIES

A NOS PARENTES

A NOS AMIS (ES)

Habiba et Samira

EDISON

WESTER

Remerciements

Qu'il nous soit tout d'abord permis d'exprimer notre entière gratitude à tous les Professeurs qui ont collaboré à notre formation.

Ce travail a été réalisé au service Electronique du Centre aux Energies Nouvelles (CEN).

Nous remercions Monsieur A. BOURGEB pour nous avoir proposé le sujet et pour l'intérêt qu'il a porté à notre projet.

Nos plus vifs remerciements vont particulièrement à Monsieur A. SAIDJ, pour l'attention soutenue, les conseils éclairés, le constant soutien et les encouragements qu'il n'a cessés de nous prodiguer durant tout le semestre.

Nos remerciements vont également à Messieurs J. HERRY et M. HADDAD qui ont bien voulu s'intéresser à notre étude et nous apporter leur aide précieuse.

Nous ne terminerons pas sans citer le précieux concours de Monsieur ZEBIRI Nacer-Eddine pour sa contribution à la mise en forme de ce fascicule.

A STRONG

TEXT

CONTROLEUR DE DISQUE SOUPLE

PLAN

INTRODUCTION

CHAPITRE I.- Présentation du système exorciser.

- I.1.- Description et fonctionnement de l'exorciser.
- I.2.- Signaux d'interface fondamentaux pour le contrôleur de disque souple.

CHAPITRE II.- Le disque souple et son unité d'entraînement.

- II.1.- Le disque souple.
- II.2.- L'unité d'entraînement ou "Drive".
- II.3.- Formatage de la disquette.
- II.4.- Enregistrement de l'information.
- II.5.- Signaux d'interface fondamentaux pour le contrôleur de disque souple.

CHAPITRE III.- Le contrôleur de disque souple.

- III.1.- Etude de l'interface.
 - III.1.1.- Introduction
 - III.1.2.- Description du circuit de récupération des données.
 - III.1.3.- L'interface S.S.D.A.
 - III.1.4.- Opération de lecture.
 - III.1.5.- Détection et correction d'erreurs.
 - III.1.6.- Opération d'écriture.
- III.2.- Réalisation du contrôleur.
 - III.2.1.- Schéma électrique du contrôleur.
 - III.2.2.- Schéma d'implantation des composants.

III.- 3.- Logiciel.

- * Conclusion.
- * Annexe

- A.- Généralités sur le microprocesseur MC 6800
- B.- PIA MC 6820
- C.- LISTING.

INTRODUCTION

Avec l'avènement des microprocesseurs est née une gamme de périphériques dont le plus important est constitué par le "Floppy Disk Driver" (unité de disque souple) ou simplement "Drive".

En effet les principales mémoires de masse utilisées jusqu'alors étaient les dérouleurs de bandes magnétiques ou les disques durs qui présentaient certains inconvénients tels que :

- prix de revient élevé ;
- entretien fréquent et conditions d'installation particulières (climatisation, dépoussiérage etc.....).
- utilisation de supports lourds.

L'apport du "Floppy Disk" ou disque souple a modifié considérablement les paramètres économiques dans les systèmes micro-informatiques.

Le disque souple présente, en effet, un moyen de stockage bon marché (une boîte de 10 disques coûte 20 DA), à grande vitesse (250 K.Bits/S) et une capacité très importante (de 100 K.Octets à 1 mégaoctets).

Par ailleurs, les systèmes à base de microprocesseurs tel que le MC 6800 ont une capacité de la mémoire centrale, limitée (64 K.Bytes) ; il serait donc avantageux de minimiser la taille des programmes résidents en les rangeant dans une mémoire auxiliaire et de les appeler chaque fois que le besoin s'en fait sentir.

Notre travail consiste à doter un outil de développement réalisé au service électronique du CEN, d'un contrôleur de disque souple. Des impératifs liés à un système d'exploitation déjà existant (M DOS) ont orienté notre travail dans le sens de la réalisation d'un interface compatible. A cet fin, nous avons développé notre exposé en trois parties :

1.- la présentation du système EXORCISER :

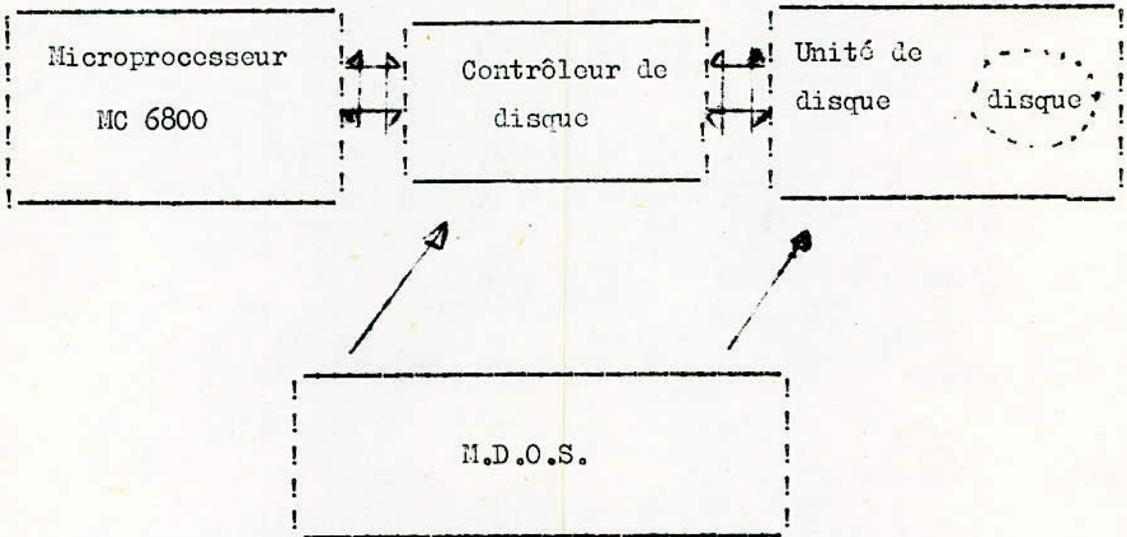
Pour mieux exploiter un système, sa connaissance s'impose, nous ne nous attarderons cependant que sur la présentation de l'ensemble des interconnexions EXORCISER-INTERFACE disque vue son importance pour la compréhension du contrôleur.

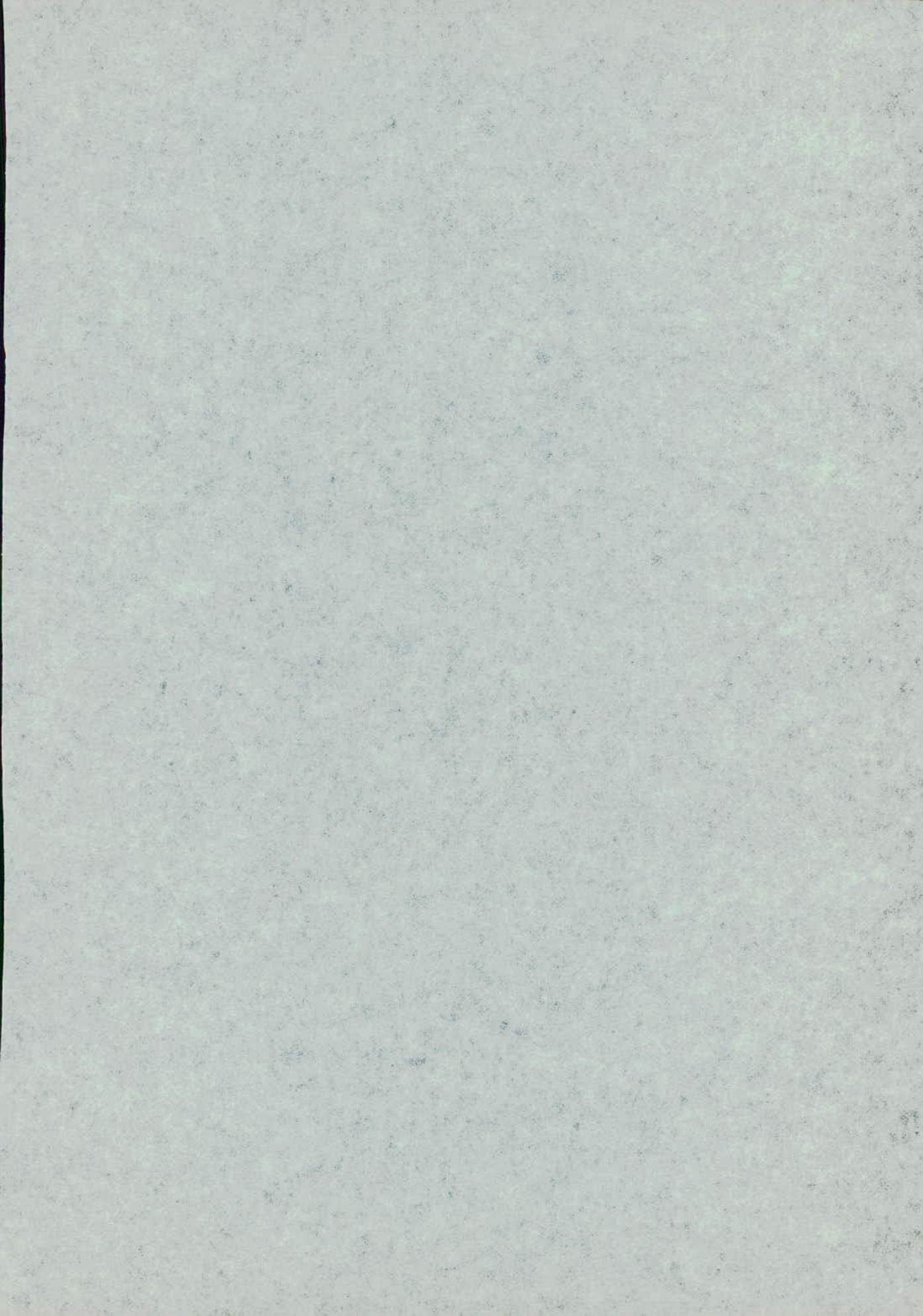
2.- Le disque souple et son unité d'entraînement :

L'étude de cette partie nous permettra de comprendre les signaux d'interface comprenant les commandes et les données envoyées à l'unité, ainsi que les indicateurs d'état et de données envoyées par l'unité au contrôleur.

3.- Le contrôleur de disque :

L'étude du contrôleur est matérialisée par la réalisation d'une carte faisant l'interface entre l'unité du disque et l'EXORCISER comme le montre la figure suivante





I.- PRESENTATION DU SYSTEME EXORCISER :

I.1.- Description sommaire de l'EXORciser :

L'EXORciser M 6800 est un micro-ordinateur de développement, constitué essentiellement par :

- le module M.P.U.
- le module Debug
- les modules mémoire RAM
- Accessoires divers.

I.1.1.- Le module M.P.U. :

Le module M.P.U. comprend deux systèmes essentiels :

- le système horloge
- l'unité centrale MC 6800

a.- Le système horloge : Ce système comprend une horloge de 1 MHz pour les fréquences comprises entre 100 KHZ et 1 MHz, l'utilisateur a la possibilité de sélectionner une horloge externe, par l'intermédiaire d'un interrupteur situé dans le module M.P.U.

Une fois le signal horloge sélectionné, un système de contrôle le convertit en deux phases D1 et Q2 utilisées par l'unité centrale pour les traitements et les échanges avec les circuits d'interface d'entrée/sortie et les mémoires.

b.- l'unité centrale MC 6800 : Cette unité permet l'exécution des instructions du programme de l'utilisateur ainsi que le "Firmware EXBUG" contenu dans le module "DEBUG".

Remarque : la description du microprocesseur MC 6800 est fournie dans la partie Annexe.

I.1.2.- Le module DEBUG :

Ce module permet à l'utilisateur de l'EXORCISER de tester la compatibilité du hardware et du software de sa réalisation.

La mémoire ROM de ce module contient le programme "Firmware EXBUG".

L'opérateur communique avec "EX BUG" au moyen d'un télétype ou d'un terminal similaire.

Lorsque l'EXORciser est initialisé, "EX BUG xx" est imprimé où xx indique la version du programme utilisé, l'utilisateur a alors la possibilité d'accéder à l'une des fonctions suivantes :

(4)

- LOAD : charger son programme dans l'EXORCISER
- VREF : comparer le programme stocké sur bande avec le contenu de la mémoire.
- SRCH : chercher sur bande un fichier particulier.
- PRNT : imprimer une zone mémoire
- PNCH : perforer une zone mémoire sur bande
- MAI D: mettre au point son programme ou système en cours de développement.

Comme pour la mise au point de notre travail, la fonction MAID a été très souvent utilisée, nous précisons ci-dessous les différentes fonctions qu'elle peut réaliser :

- examiner et changer le contenu des mémoires
- calculer le décalage impliqué par le mode d'adressage relatif
- examiner et charger les données dans les registres du M.P.U.
- introduire, afficher et enlever les points d'arrêt dans le programme.
- arrêter l'EXORCISER sur une adresse de mémoire choisie.
- sortir un signal de synchronisation sur une adresse de mémoire choisie.
- exécuter le programme de l'utilisateur
- réaliser des conversions décimal-Octal-Hexadécimal
- rechercher en mémoire une configuration binaire.

II.1.3.- Les modules "mémoire".

Le module MPU de l'EXORCISER peut adresser un ensemble de mémoires "R.A.M" ; leur capacité varie entre 1 K-bytes et 64 K bytes.

I.1.4.- Accessoires divers - le générateur "BAUD-RATE", les alimentations et le châssis :

Le générateur BAUD-RATE fournit à l'EXORCISER une gamme de huit vitesses "BAUD-RATE" ; le microprocesseur peut ainsi communiquer avec les périphériques avec des vitesses de transfert qui varient entre 110 et 9600 bauds (1 baud est l'unité de vitesse de transfert 1 baud $\hat{=}$ 1 byte/seconde).

L'EXOCISER est muni de trois alimentations : + 5 V dc, + 12 V dc et - 5 V dc le fond du châssis appelé "fond de panier" peut supporter 14 modules en fichiers tels que le module MPU, le module Debug, les modules mémoires, les cartes interface comme la carte interface "Floppy-Disk" etc.....

I.2.- Signaux d'interface fondamentaux pour le contrôleur de disque souple :

L'ensemble de ces signaux sont transmis par le connecteur P1 qui relie le contrôleur de disque au module M.P.U.

Ces signaux ont été regroupés dans le tableau suivant :

N° DE LA BROCHE	M némonique des signaux	Nom et description des signaux
A, B, C 1, 2, 3	+ 5 V	+ 5 volts dc : cette alimentation est utilisée par les circuits logiques du contrôleur de disque.
D	IRQ	<u>Interruption masquable</u> : Ce signal provoque une demande d'interruption au M.P.U. quand il passe du niveau haut au niveau bas. Le microprocesseur termine l'exécution en cours avant de prendre en compte la demande d'interruption. La demande d'interruption est prise en compte si le bit masque d'interruption du registre d'état (bit 1) est à zéro.
E	NMI	<u>Interruption non masquable</u> : Quand ce signal est à l'état bas, une demande d'interruption prioritaire est envoyée au M.P.U.
F	VMA	<u>Validation des adresses mémoires</u> : Quand il est à l'état bas ce signal indique au contrôleur qu'il y a une adresse valide sur le bus adresses.
J	∅ 2	Deuxième phase du signal horloge du module M.P.U.
L	MEM CLK	<u>Horloge - mémoire</u> : Ce signal est utilisé par le contrôleur de disque pour générer le signal horloge d'écriture.
M, 11	- 12 V	- 12 volts dc : cette alimentation est utilisée pour générer la tension - 5 V de requise par la PROM.
6 T, 16	+ 12 V	+ 12 volts dc : cette alimentation est utilisée par la PROM.
31	Do	<u>Bus donnée (bit 0)</u> : Quand cette ligne est valide, elle permet un transfert bidirectionnel de donnée entre la carte interface et le module de l'EXOASER.

29	!	D ₁	!	Bus donnée (bit 1)
K	!	J ₂	!	Bus donnée (bit 2)
H	!	D ₃	!	" " 3
32	!	D ₄	!	" " 4
30	!	D ₅	!	" " 5
L	!	D ₆	!	" " 6
J	!	D ₇	!	" " 7
40	!	A ₀	!	Bus Adresse (bit 0) :
	!		!	Une des 16 lignes d'adresses utilisées pour
	!		!	selectionner une position mémoire ou un regist
	!		!	à l'intérieur d'un circuit d'interface d'entré
	!		!	/ sortie.
V	!	A ₁	!	Bus adresse (bit 1)
U	!	A ₂	!	" " 2
39	!	A ₃	!	" " 3
38	!	A ₄	!	" " 4
T	!	A ₅	!	" " 5
S	!	A ₆	!	" " 6
37	!	A ₇	!	" " 7
36	!	A ₈	!	" " 8
R	!	A ₉	!	" " 9
P	!	A ₁₀	!	" " 10
35	!	A ₁₁	!	" " 11
34	!	A ₁₂	!	" " 12
N	!	A ₁₃	!	" " 13
M	!	A ₁₄	!	" " 14
33	!	A ₁₅	!	" " 15

(7)

5	!	RES	!	Reset : Ce signal remet à l'état initial, la
	!		!	carte interface, soit par un bouton extérieur
	!		!	soit par le signal Reset du module MP4 de l'
	!		!	EXORCISER.
6	!	RTW	!	<u>Lecture - écriture</u> :
	!		!	Ce signal détermine si la donnée est à lire
	!		!	(état haut) ou)à écrire (état bas).
41, 42, 43	!	6ND	!	<u>Masse.</u>
W, V, Y	!		!	
	!		!	

Remarque : Toutes les broches non mentionnées sont non utilisées

II.- LE DISQUE SOUPLE ET SON UNITE D'ENTRAINEMENT.

INTRODUCTION

Tout système utilisant des disques souples est composé :

- d'une carte interface réalisant la liaison électrique entre les signaux sortant des lecteurs et le bus micro-ordinateur.

- des lecteurs de disques appelés aussi "drives" ou "lecteurs de Floppy", qui comportent une partie mécanique et une partie électronique.

- d'un programme "système d'exploitation disque", résidant dans le mini-ordinateur après une phase initiale de chargement.

Ce programme appelé souvent D.O.S. (Disk-Operating-System) a pour rôle de gérer tous les accès au disque de manière transparente pour l'utilisateur.

II.1.- Le disque souple normal :

II.1.1.- Structure d'une disquette :

Une disquette est constituée d'un disque mylar recouvert d'oxyde magnétique. Ce disque peut tourner à frottement doux dans une pochette en carton revêtue à l'intérieur d'une couche antistatique facilitant la rotation (Le disque en mylar ne sort jamais de la pochette).

L'ensemble disque plus pochette constitue une disquette ou "Floppy Disk". Trois ouvertures sont aménagées dans la pochette :

- un grand trou central par lequel va passer le système d'entraînement du disque en mylar.

- un petit trou excentré circulaire par lequel le détecteur monté sur le secteur va pouvoir constater le passage d'un trou réalisé sur le disque en mylar ; ce trou est l'index et le trou dans la pochette est le trou d'index : il marque le début du secteur 0.

- une ouverture oblongue donnant accès à toute la largeur du disque en mylar, à quelques millimètres près, par lequel la tête magnétique de lecture va pouvoir entrer en contact avec le disque.

Deux sortes de disquette sont aujourd'hui utilisées :

- les disquettes simple-face : l'information est enregistrée sur la seule face enduite d'oxyde magnétique.

- les disquettes double-face : les deux faces sont enduites d'oxyde et l'enregistrement se fait sur les deux faces.

De plus, suivant la qualité de l'oxyde, on distingue des disquettes de simple densité et des disquettes de double densité.

La notion de densité est liée à la finesse et à la régularité de distribution des particules d'oxyde magnétique.

Remarque : le drive mis à notre disposition utilisè des disquette simple-face, simple densité.

II.1.2.- Spécifications d'un disque souple - (simple-face, simple densité)

- taille disquette : 8 pouces (8 x 25,4 mm)
- format : 76 pistes + la piste d'index (format IBM 3740)
26 secteurs par piste.
- capacité : 128 octets/secteur - 3,3 K octets/piste 253 K octets disquette.
- Densité : 48 pistes par pouce
3268 bits par pouce sur une même piste
- vitesse de rotation : 360 tours/mn + 2 %
- vitesse de transfert : 250 k-bits/s.
- temps : * translation d'une piste à l'autre : 10 à 18 ns
(y compris 8 à 15 ns de stabilisation de la tête)
* recherche maximum : 100 à 768 ms
* engagement de la tête : 40 ms
* accès moyen : 136 à 476 ms.
- Fiabilité : * erreurs lecture (software) : moins de 1 pour 10^{19} bits.
- erreurs lecture (hardware) : moins de 1 pour 10^{12} bits
- erreurs de positionnement : moins de 1 pour 10^6 bits.

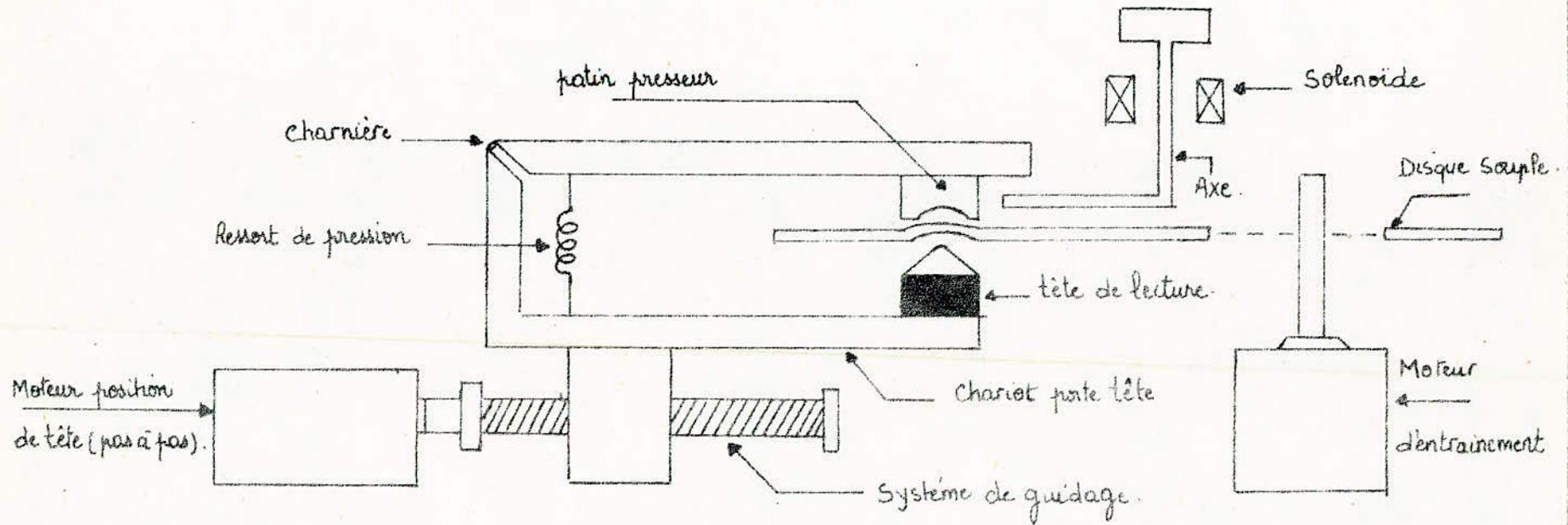
II.2.- L'unité d'entraînement (drive) :

L'unité d'entraînement renferme la mécanique et l'électronique nécessaire pour faire tourner la disquette et accéder aux données.

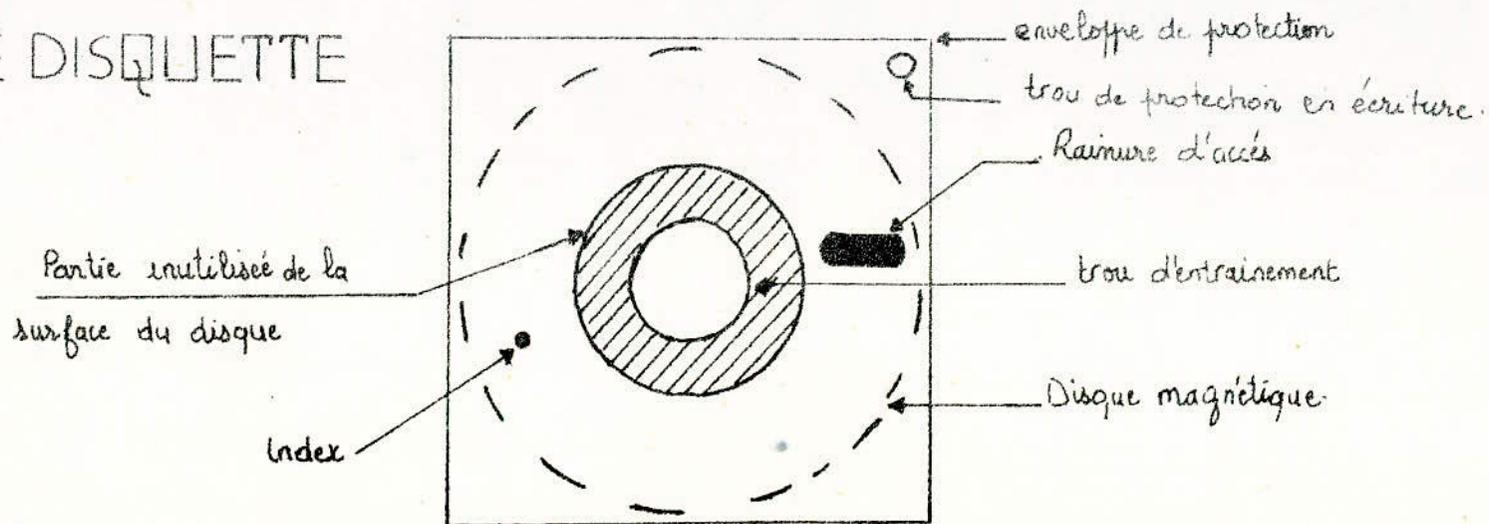
Elle comporte un système d'entraînement de la disquette et un système de positionnement de la tête magnétique sur la disquette.

- un moteur entraine la disquette à une vitesse constante de 360 tours/mn, au moyen d'un presseur constitué par un cône en teflon qui se glisse dans le trou central de la disquette. Ce presseur est solidaire de la porte dont est muni le drive et vient donc se mettre en place, uniquement lorsque l'on ferme celle-ci ; de plus, sa forme cônica assure un centrage de la disquette dans son enveloppe et un positionnement précis dans le drive (voir figure sur page suivante :

Fig II. 1. COUPE SIMPLIFIEE D'UN DRIVE



UNE DISQUETTE



La tête magnétique est solidaire d'un chariot mobile dont un bras muni d'un presseur vient pincer la disquette au niveau de son ouverture oblongue. Pour positionner correctement le chariot, on utilise généralement un moteur pas à pas.

L'électronique de l'unité accomplit quatre fonctions :

- mouvement de la tête vers la piste voulue
- chargement de la tête puis lecture ou écriture
- génération et interprétation des signaux de commande ou des informations d'états.
- commande précise du moteur de rotation.

II.- 3. FORMATAGE DU DISQUE SOUPLE :

La disquette utilisée est formatée suivant la norme IBM 3740 ; elle comprend 77 pistes numérotées de 00 (la plus externe) à 76 (la plus interne). La piste 00 est utilisée comme index, ce qui laisse 76 pistes pour les données.

Chaque piste commence par une impulsion d'index physique correspondant à la détection du trou d'index et est divisée en 26 secteurs de 128 octets (bytes) chacun.

Chaque secteur est précédé d'un identificateur "ou champ ID" ; ce dernier contient 7 bytes répartis comme suit :

Byte 1 : marque d'adresse ID
 Byte 2 : numéro de la piste
 Byte 3 : pas d'information (vide)
 Byte 4 : numéro du secteur
 Byte 5 : pas d'information (vide)
 Byte 6 et 7 : 2 bytes CRC.

Des intervalles sans informations appelés "Gaps" sont utilisés pour que la mise à jour d'un champ se fasse sans affecter les champs adjacents.

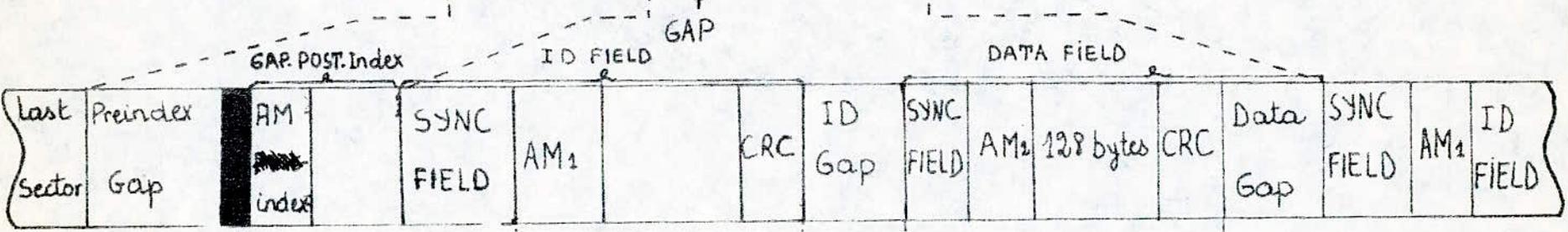
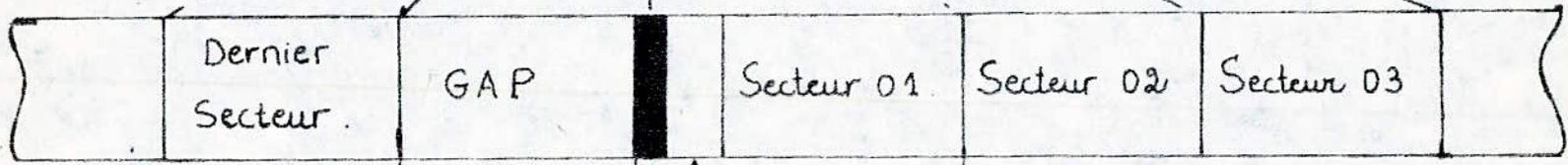
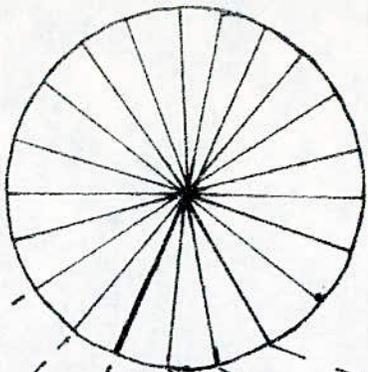
Quatre sortes de gaps sont utilisées (fig. II 3.1) :

- Gap pré-index : Il n'apparaît qu'une fois sur la piste ; il contient 320 bytes et est situé en fin de piste. Il varie légèrement en longueur après la mise à jour d'un champ.

- Gap post-index : Il est formé de 32 bytes et apparaît au début de toute piste entre la marque d'adresse index et la marque d'adresse ID. La longueur du gap ne doit jamais varier.

- Gap ID (identificateur de l'enregistrement) :

Il est formé de 17 bytes et est situé entre le champ ID et le champ de données. La longueur du gap varie légèrement après la mise à jour du champ ID.



INDEX ↑

1	2	3	4	5	6	7
ID Address Mark	Track Address	Zeros	Sector Address	Zeros	CRC Byte 1	CRC Byte 2

1	2 - 129	130	131
Data or deleted Data - Address Mark	128 bytes of user data.	CRC byte 1	CRC byte 2

- Gap de données : Il est formé de 33 bytes et apparait entre le domaine de donnée et le prochain champ ID. La longueur de ce gap varie légèrement après la mise à jour du champ de données.

Les marques d'adresse sont utilisées pour identifier les champs ID et de données et pour synchroniser le S.S.D.A.

Quatre sortes de marques d'adresse sont utilisées :

- marque d'adresse index :

Elle est située au début de chaque piste et est un nombre fixe de bytes.

- marque d'adresse ID :

Elle est située au début de chaque champ identificateur ID

- marque d'adresse de données :

Elle est située au début de chaque champ de données non effacées sur le disque.

- marque d'adresse de données supprimées :

Elle est localisée au début de chaque champ de données effacées sur le disque.

Les profils binaires des marques d'adresse, récapitulés dans le tableau ci-dessous sont représentés par la figure(II 32, a, b, c.d)

	! DONNEES !	! HORLOGE !
! Marque d'adresse Index !	! FC !	! D7 !
! Marque d'adresse ID !	! FE !	! C7 !
! Marque d'adresse données !	! FB !	! C7 !
! Marque d'adresse données !	! !	! !
! supprimées. !	! !	! !

Chaque domaine écrit sur la disquette est augmenté de deux bytes CRC générés à partir d'une permutation cyclique des bits de données qui commence par le bit "Zéro" de la marque d'adresse et finit avec le bit "sept" du dernier byte contenu dans le champ (les bytes CRC exclus).

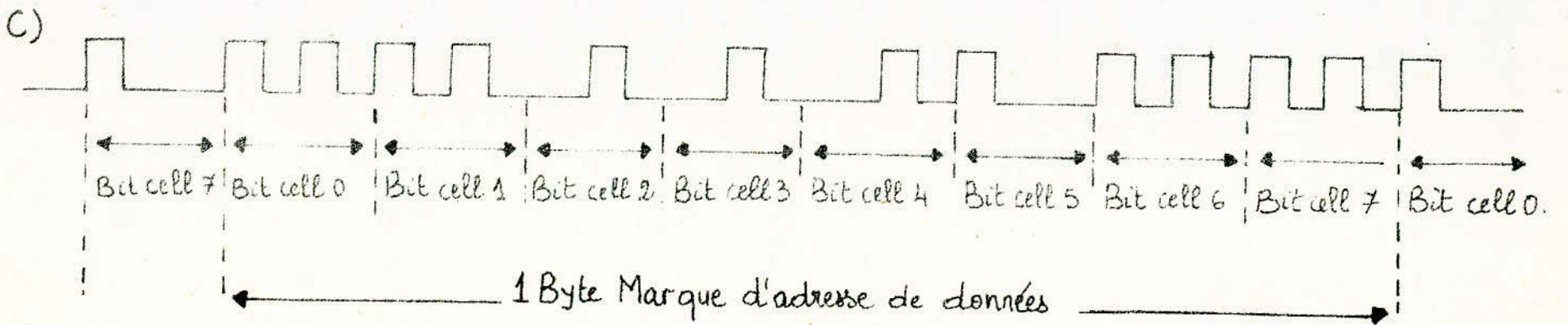
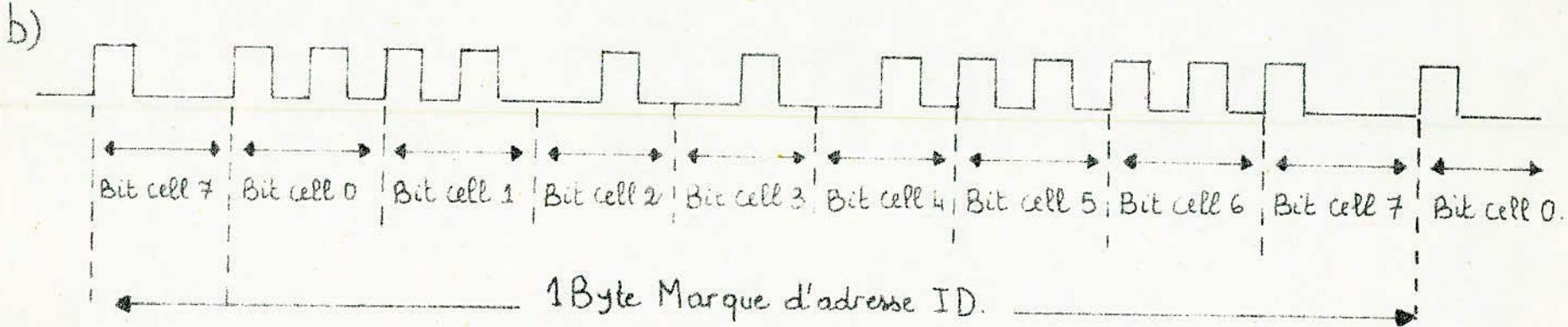
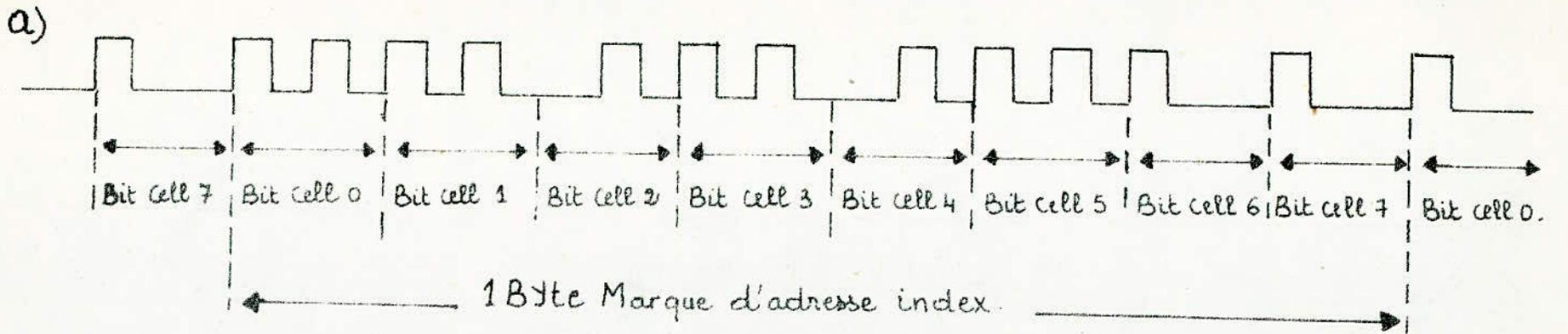


fig II 3.2. PROFIL BINAIRE DES MARQUES D'ADRESSE

II.4.- ENREGISTREMENT DE L'INFORMATION.

L'information est enregistrée sur les pistes d'un disque en format binaire. Pour obtenir ce type d'enregistrement, on utilise une technique NRZ et un codage FM (Modulation de fréquence).

- Technique NRZ (non retour à zéro) :

La position correspondant à chaque bit est magnétisée dans une direction ("0") ou dans la direction opposée ("1"). Il n'y a pas d'état intermédiaire, de magnétisation nulle.

- Codage FM :

Cette technique consiste à insérer un bit de donnée entre deux tops d'horloge successifs.

La période du signal code appelée "cellule-bit" comprend donc un bit d'horloge (toujours à "1", à l'exception de l'enregistrement de la marque d'adresse où le bit d'horloge peut prendre la valeur "0" ~~est marquée par~~ (fig. II.3.2) et un bit de donnée ("0" ou "1"). Une donnée "0" est marquée par l'absence d'impulsion pendant le temps de bit; tandis que la donnée "1" est marquée par l'arrivée d'une impulsion au milieu du temps de bit (fig. II.4.1).
a, b, c

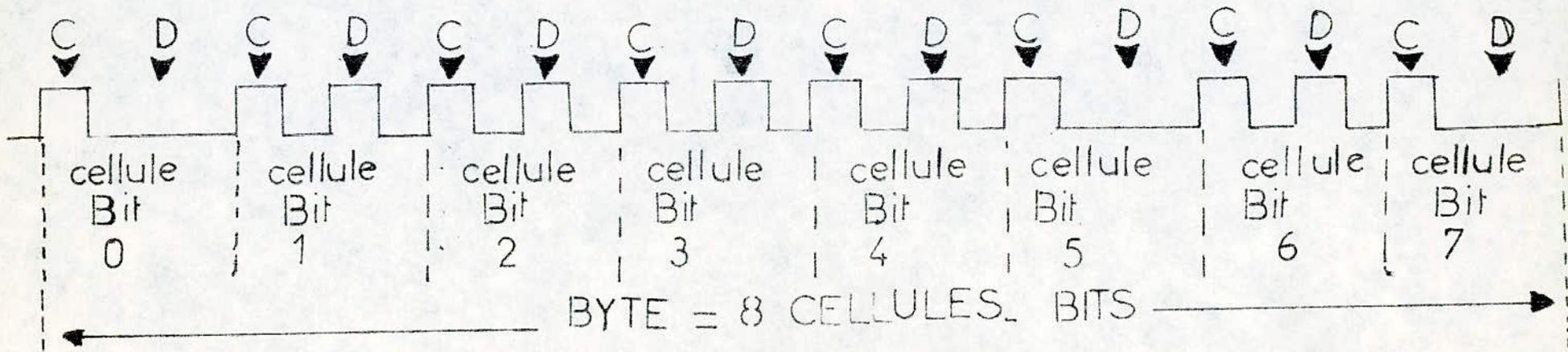
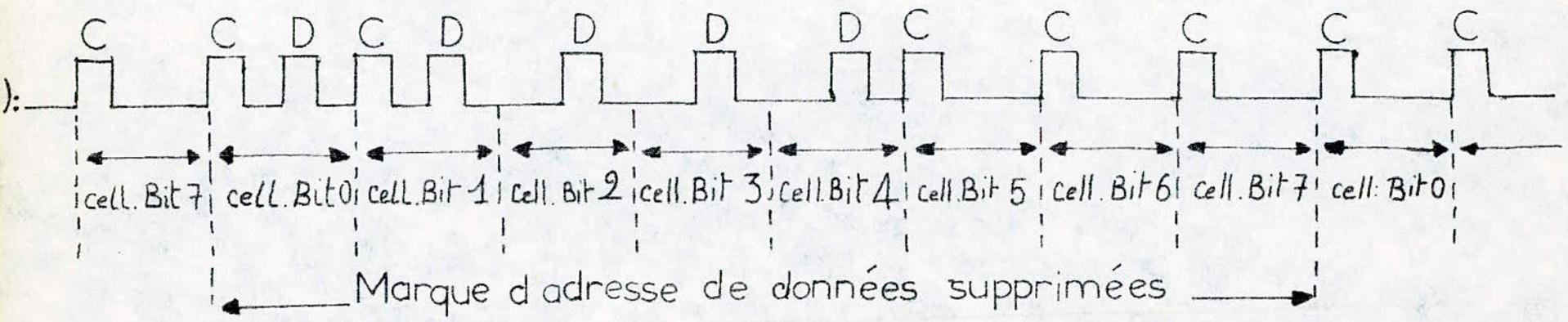
Deux bits consécutifs ne sont jamais absents simultanément. Un temps de bit est l'espace de temps qui sépare deux temps d'horloge successifs.

Ainsi le byte, dont on parlera dans les opérations de lecture ou d'écriture est composé de 8 cellules-bits.

La cellule la plus significative est la cellule-bit "0" et la moins significative est la cellule-bit 7.

Lors d'une opération d'écriture / de lecture, la donnée la plus significative est transmise / reçue en premier.

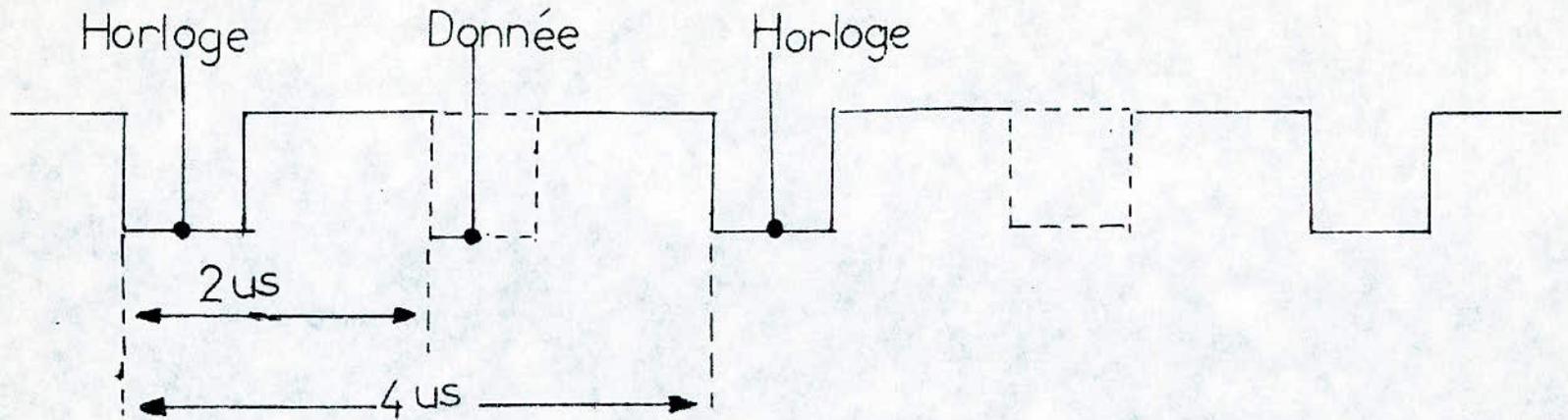
fig II.3.2 PROFIL BINAIRE DES MARQUES D'ADRESSE



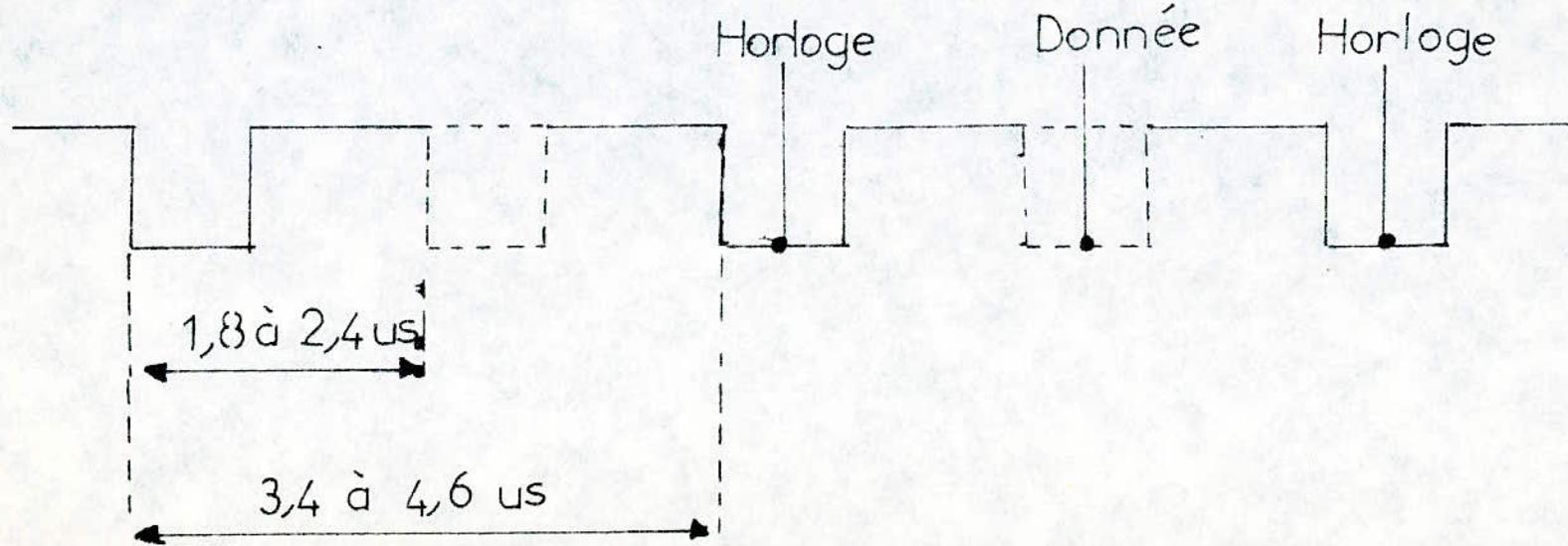
Données : 01111010
 Horloge : 11111111

fig:II 4.1.a Enregistrement de l'information

Diagramme des temps



figII.4.2.5: SIGNAL ECRIT



figII.4.2.6 : SIGNAL LU

II.-5.- Signaux d'interface fondamentaux pour le contrôleur disque souple :

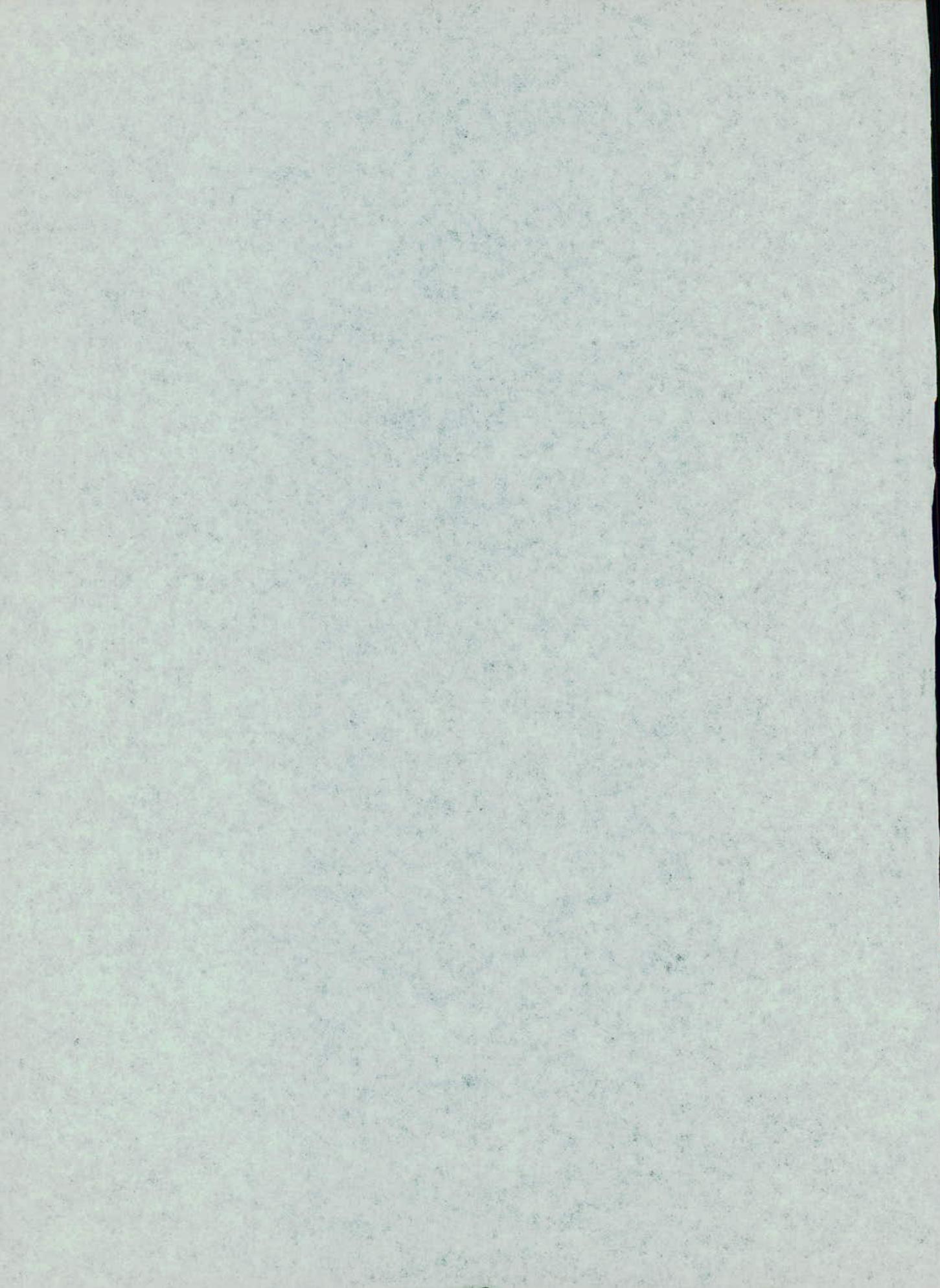
L'ensemble de ces signaux sont transmis par le connecteur P2 qui relie la carte interface au "drive".

Ces signaux sont représentés dans le tableau suivant :

N° DE LA BROCHE	M némonique des signaux	Nom et description des signaux
5	Direction	<p><u>Direction</u> : Ce signal est utilisé avec le signal "Step" pour faire avancer la tête de lecture d'une piste à l'autre.</p> <p>Quand ce signal est un niveau bas, la tête est dirigée vers les pistes situées à l'extérieur par contre quand il est à l'état haut, la tête se dirige vers les pistes intérieures, on rappelle que les pistes sont numérotées de l'extérieur vers l'intérieur.</p>
7	STEP	<p><u>Marche</u> : Ce signal est utilisé avec le signal "direction" pour faire bouger la tête de lecture d'une piste à l'autre.</p> <p>Quand ce signal passe à l'état bas, la tête guidée par le signal "direction" se positionne sur la piste voulue.</p>
15	Select 0	<p><u>Selection 0</u> : Ce signal permet la sélection du drive 0, quand il passe à l'état bas.</p>
39	Select 1	<p><u>Sélection 1</u> : Ce signal permet la sélection du drive 1, quand il est à l'état bas.</p>
23	TRK 43	<p><u>Pistes au dessus de 43</u> : ce signal est utilisé pour contrôler les amplitudes du courant d'écriture dans la tête d'enregistrement, quand l'enregistrement se fait sur les pistes 0 à 43, ce signal est au niveau haut, par contre il est au niveau bas pour les pistes 44 à 76.</p>
25	INDEX	<p><u>Index</u> : Cette impulsion est utilisée pour indiquer le début d'une piste ; cette impulsion arrive une fois par révolution de la disquette.</p>
29	WRITE ENABLED	<p><u>Ecriture autorisée</u> : Quand il est au niveau bas ce signal est utilisé pour indiquer que l'écriture a lieu.</p>

31	WRITE DATA	! Ecriture des données : ce signal est constitué par des impulsions représentant les données à enregistrer sur le disque souple.
33	EN WRITE	! <u>Autorisation d'écriture</u> : Ce signal est utilisé, quand il est à l'état bas, pour permettre l'enregistrement des données sur le disque.
		! Quand ce signal passe à l'état haut, la lecture des données de la disquette est autorisée.
37	RAW-DATA	! <i>Signal lu de la disquette, non décodé.</i>
17	TRK 00	! Piste 00 : A l'état bas, ce signal indique que la tête est positionnée sur la piste 00 du disque.
	HEAD LOAD	! Chargement de la tête : ce signal est utilisé pour positionner le disque contre la tête de lecture.
21	Ready 0	! <u>Drive 0 prêt</u> : ce signal est utilisé pour indiquer que le disque est correctement inséré dans drive 0. Il indique également que le disque a atteint la bonne vitesse de rotation.
	Ready 1	! <u>Drive 1 prêt</u> : Ce signal est identique au signal Ready 0, il nous renseigne sur l'état du drive 1.

Remarque : Toutes les broches qui ne figurent pas dans ce tableau sont non utilisées.



III : Le contrôleur de disque souple :

III.1.- ETUDE DE L'INTERFACE :

III.1.1.- Introduction :

Un bloc diagramme du contrôleur de disque souple est montré sur la figure III.1. ; il comprend :

a.- le circuit de récupération des données :

Ce circuit permet de décoder les informations et de générer un signal d'horloge de synchronisation.

Une séparation donnée / horloge très précise est assurée par un oscillateur à boucle de phase verrouillée (p.L.L. : phase locked Loops) qui élimine le problème de "glissement de bits".

b.- le S.S.D.A. "Synchronous Serial Data Adapter" (MC 6852) :

Le S.S.D.A. réalise la synchronisation des opérations de lecture et d'écriture, la sérialisation des données écrites et la désérialisation des données lues.

c.- Le P.I.A "Peripheral Interface Adapter" (MC 6820) :

Le P.I.A. s'interface avec le "drive" pour :

- contrôler les fonctions du "drive".
- renseigner le M.P.U. (MC 6800) sur les états du drive. D'autre part, quatre lignes d'interface permettent au P.I.A de communiquer avec le contrôleur CRCCG (MC 8506). Le P.I.A fera l'objet d'un paragraphe donné en annexe par la suite.

d.- Le bloc de détection d'erreurs (MC 8506) :

Ce bloc sert à engendrer les deux bytes CRC pendant l'opération d'écriture et à détecter les erreurs pendant l'opération d'écriture.

III. 1.2.- Fonctionnement du circuit de récupération de données :

- le signal provenant du "drive" sort d'une porte T.T.L. à collecteur ouvert dont la résistance de charge se trouve sur le contrôleur disque souple (Cf. Figure III.2.-1).

Le signal reçu passe dans une porte T.T.L. à trigger de schmitt qui élimine les bruits (Cf. figure III-2-2).

Débarrassé des bruits, le signal "Raw-data" se présente à l'entrée de la bascule "Flip - Flop" (1) (u 22) sous forme d'une série d'impulsions de

Figure III-1 : Synoptique du Contrôleur de disque Souple

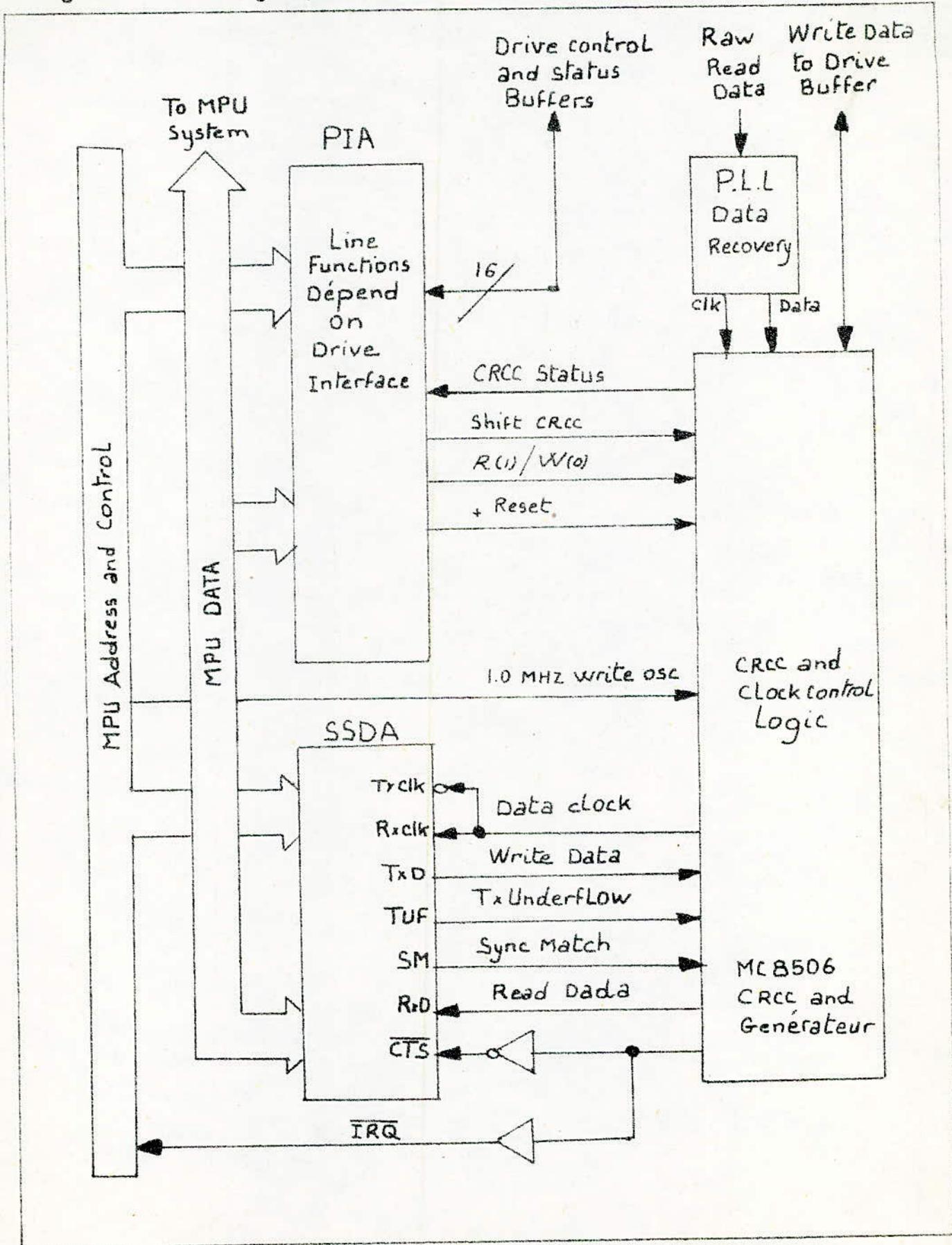
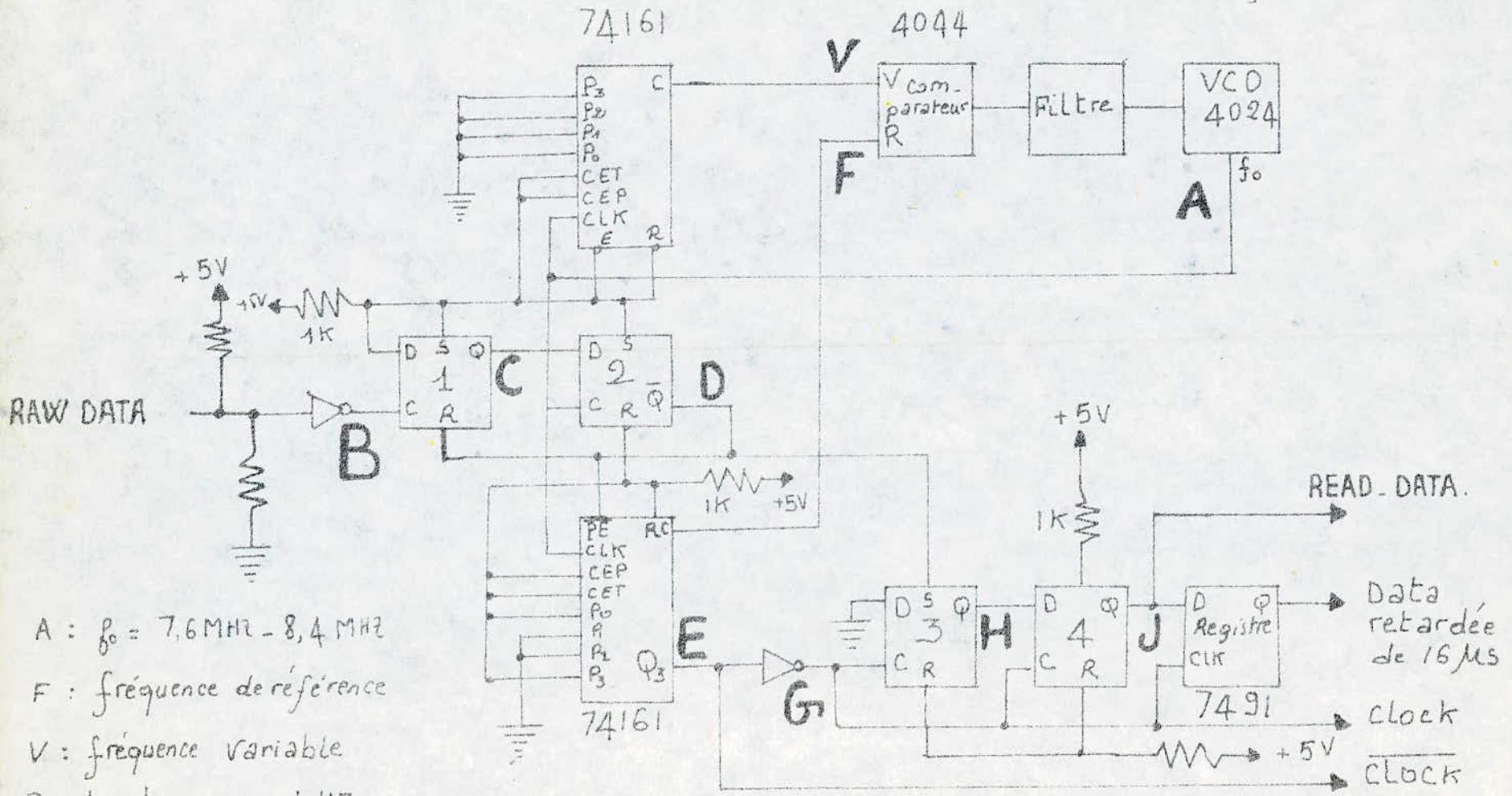


figure III 2.2 : Circuit de Récupération des données



A : $f_0 = 7,6\text{MHz} - 8,4\text{MHz}$

F : fréquence de référence

V : fréquence variable

G : horloge 500 kHz

J : Données Récupérées en Format 14R2.

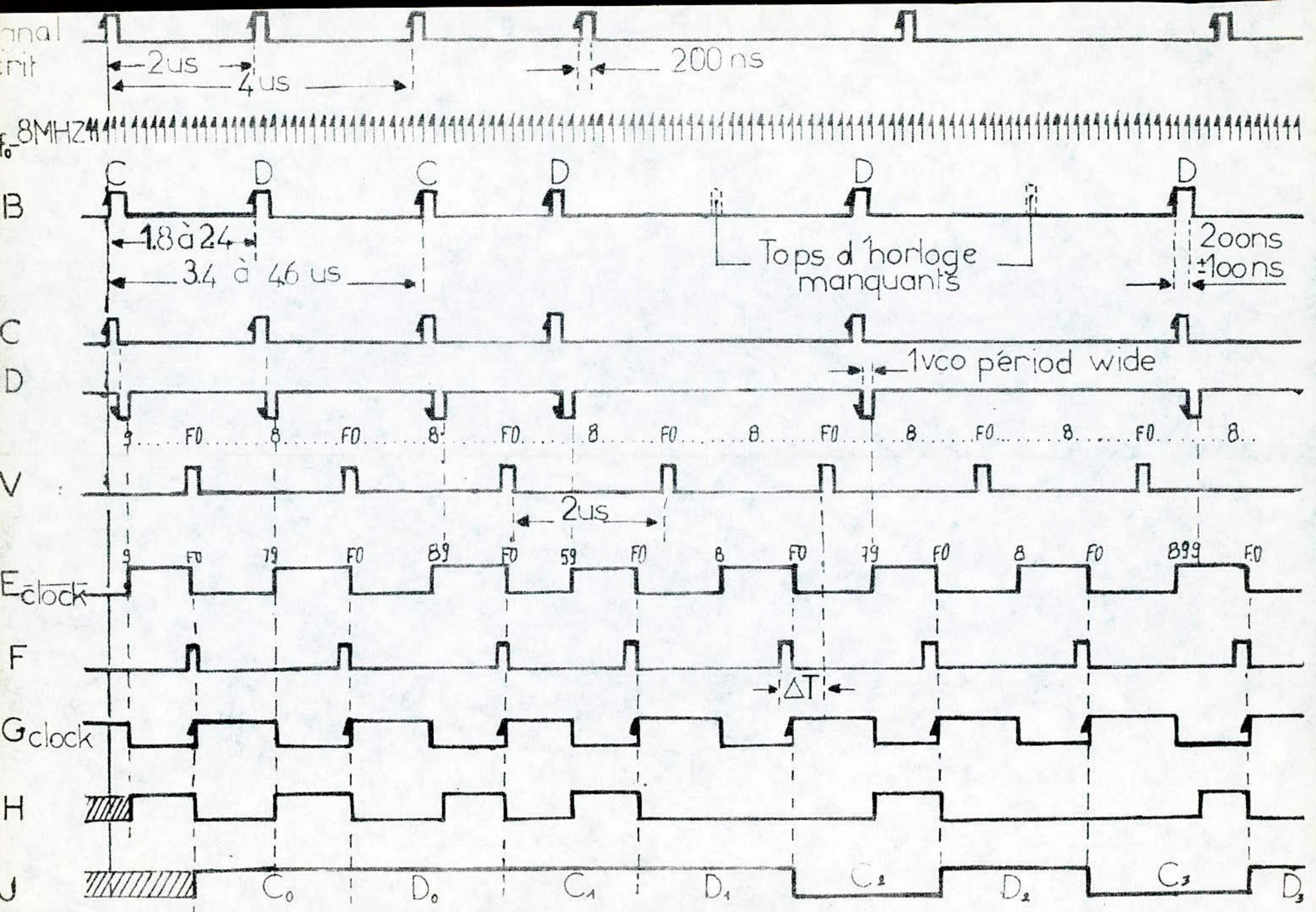
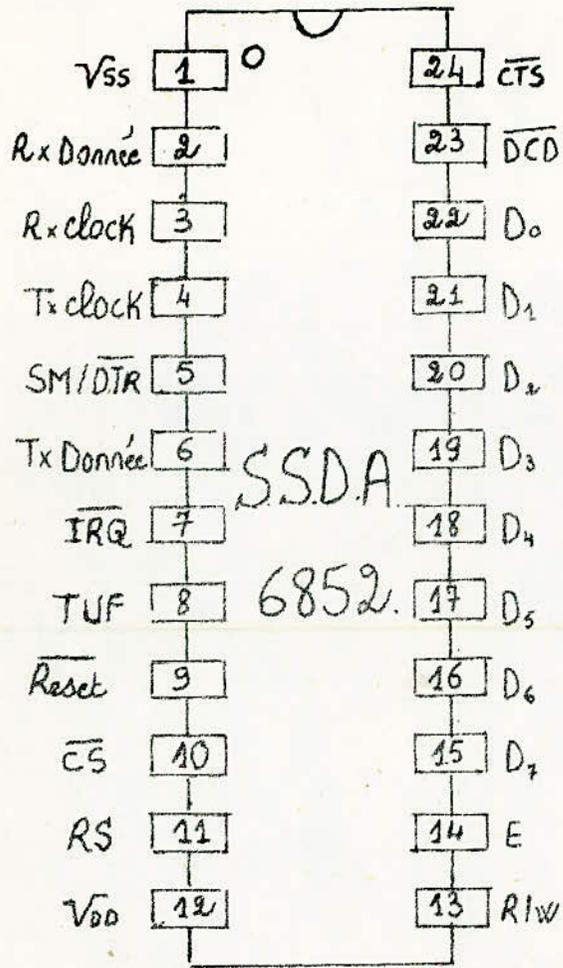


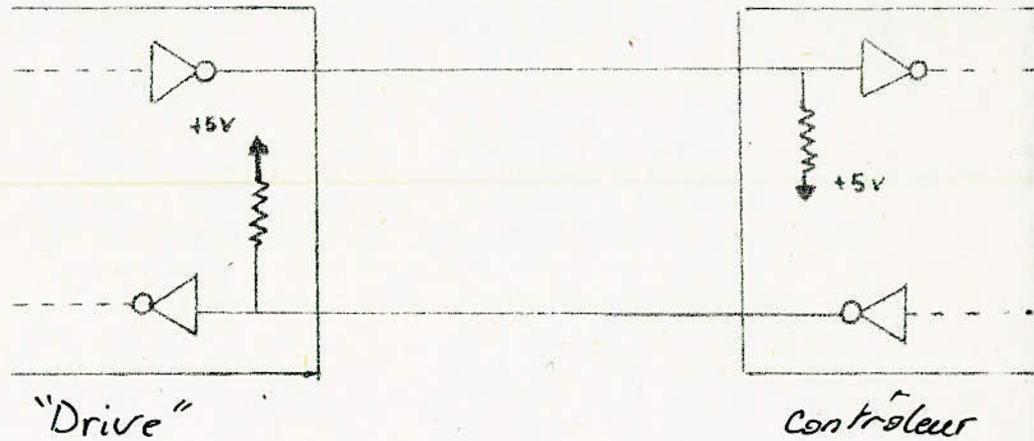
Fig. 2.3. CHRONOGRAMME S. RECUPERATION DES DONNEES



Vue de dessus

Brochage du S.S.D.A.

fig III 2.1 PRINCIPE D'ÉCHANGE DES SIGNAUX
ENTRE LE DRIVE ET L'INTERFACE



g III 3. 1

durée une période VCO ($1/f_0$) (période du signal à la sortie du VCO (4024) : Oscillateur à fréquence contrôlée) chaque impulsion du signal "Raw-data" fait basculer le "Flip-Flop" (1) et forme ainsi un signal "Raw-data" synchronisé avec f_0 . La sortie Q du "Flip-Flop" (1) est reliée à l'entrée D de la bascule (2) activée par f_0 ; à la sortie q, on récupère les impulsions négatives qui activeront la bascule (3), remettront à "0" la bascule (1) et chargeront le compteur de référence (429) par la broche PE. Ce compteur est préchargé par "9" chaque fois qu'une impulsion négative se présente ; Q3 aura une transition positive en 15 périodes (VCO) générant un top d'horloge même si une impulsion de donnée ne se présente pas.

Chaque fois que le compteur compte 16 "VCO" la sortie "Carry-out" génère une impulsion de durée une période "VCO", celle-ci se présente toutes les 2 us ($16/f_0$) fournissant une référence fondamentale pour le comparateur de phase (fig. III.2.2.F, fig. III.2.3-F). D'autre part ~~W~~, la fréquence variable V obtenue est envoyée au comparateur de phase, par la sortie "Carry Out" (fig. III.2., V fig. III.2.3.V). Toute différence (Δf) entre la fréquence de référence F et la fréquence variable V se traduit par l'apparition d'une tension de commande qui, réinjectée sur le "VCO" modifie la fréquence V pour la rendre égale à F.

A la sortie Q3 du compteur de référence, les transitions négatives sont inversées avant de synchroniser le "Flip-Flop" (3) dont la sortie devient basse (fig. III.2-2-H ; fig. III.2-3-G-H). Cet état est maintenu jusqu'à la génération d'une impulsion de donnée par le "Flip-Flop" (2).

Le "Flip-Flop" (4) activé par la sortie Q du "Flip-Flop" (3), et synchronisé par Q3 du compteur de référence, génère le signal "Read-data" (données et horloge en format NRZ) qui sera envoyé vers la logique de lecture (fig. III-2-2-J, fig. III.2-3-J) le signal "Read-data" retardé ~~de~~ $16 \mu s$ par le registre à décalage est injecté au circuit de détection d'erreur. Le signal recueilli à la sortie du compteur de référence est inversé pour obtenir le signal d'horloge utilisé pour la synchronisation des différentes bascules et pour échantillonner les données par la suite.

III.1.3.- L'interface SSDA "MC 6852" (Synchronous Serial Data Adapter) :

Pour mieux comprendre les opérations de lecture et d'écriture traitées dans les chapitres suivants ; il est nécessaire de décrire le circuit interface S.S.D.A.

III.1.3.1.- Description et fonctionnement du SSDA "MC 6852" :

Le circuit SSDA MC 6852 est un interface bidirectionnel qui reçoit les données en série, des périphériques, et les transmet en parallèle au système microprocesseur MC 6800 par le bus données et inversement.

Il se présente dans un boîtier de 24 broches comme le montre la figure III.3.1.

* le deuxième compteur (=16) (428) reçoit la fréquence f du "VCO" et la divise par 16,

Le S.S.D.A. est adressé par le microprocesseur MC 6800 comme deux positions mémoires ; il comprend sept registres internes accessibles par l'utilisateur, répartis comme suit :

- deux registres accessibles à lecture seule.
- le registre d'état
- le registre de réception
- cinq registres accessibles à l'écriture seule.
- le registre de commande 1
- le registre de commande 2
- le registre de commande 3
- le registre caractère de synchronisation
- le registre de transmission.

La configuration interne d'un S.S.D.A. est représenté par la figure III.3.2. La sélection d'un S.S.DA se fait par le signal CS ; les registres internes sont slectés par une entrée de sélection de registre RS, l'entrée lecture / écriture (R/W) et les bits 6 et 7 (AC1 et AC2) du registre de contrôle 1.

Les opérations de base du S.S.DA sont résumées dans le tableau suivant :

CS	RS	RIW	AC2	AC1	TYPE D'OPERATION
0	0	1	X	X	REG D'ETAT - BUS DE DONNEE
0	0	0	X	X	BUS DE DONNEE - REG DE CONTROLE 1
0	1	1	X	X	REG DE RECEPTION - BUS DE DONNEE
0	1	0	0	0	BUS DE DONNEE - REG DE CONTROLE 2
0	1	0	0	1	BUS DE DONNEE - REG DE CONTROLE 3
0	1	0	1	0	BUS DE DONNEE - REG CARACT. SYNC
0	1	0	1	1	BUS DE DONNEE - REG DE TRANSMISSION

La programmation des différents registres est représentée dans le tableau III.3.3.

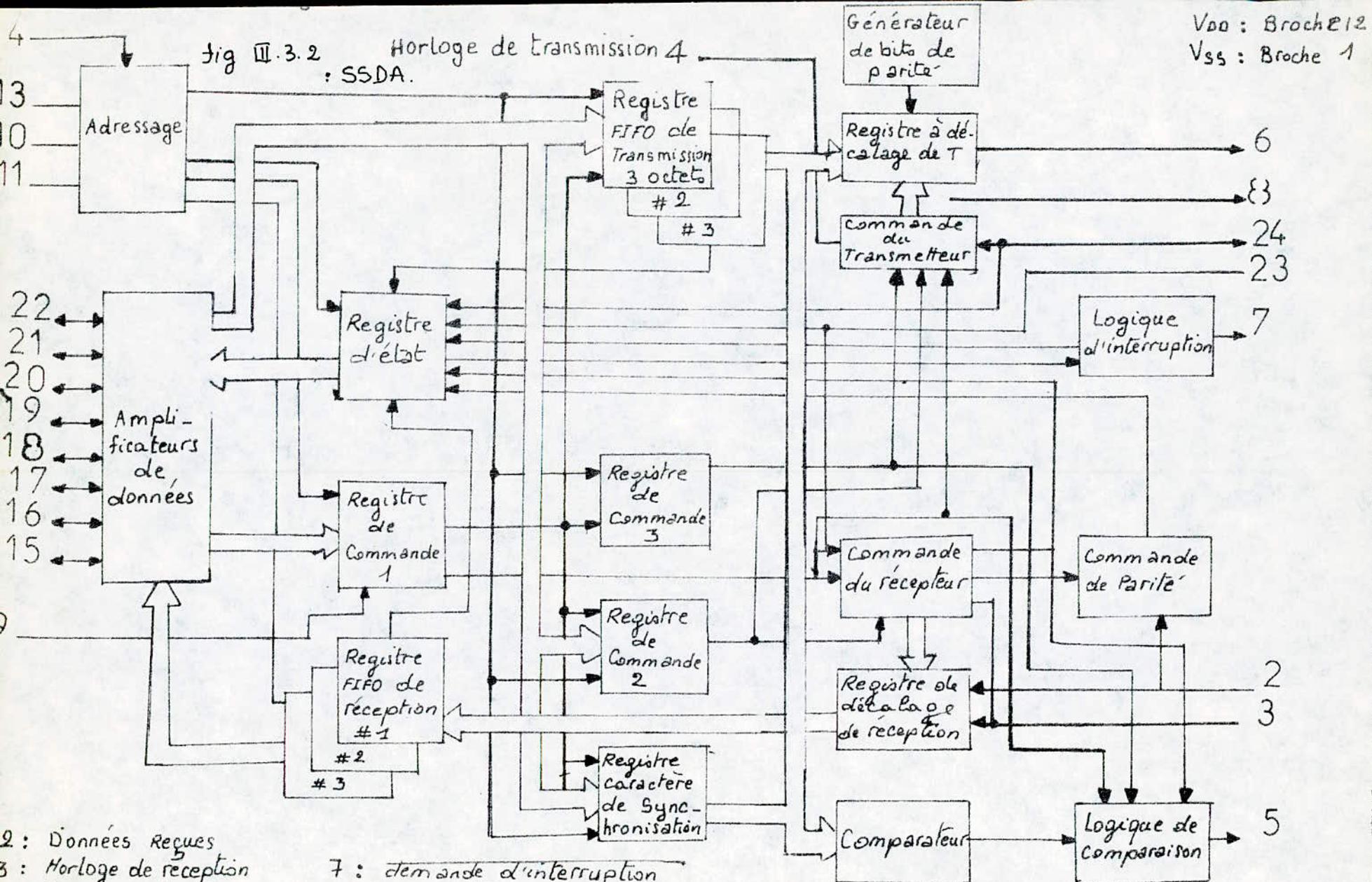


Fig III.3.2 Horloge de transmission 4 : SSDA.

V_{DD} : Broche 12
V_{SS} : Broche 1

- 2 : Données Reçues
- 3 : Horloge de réception
- 5 : Synchronisation / Données prêtes SM / DTR
- 6 : Données transmises

- 7 : demande d'interruption \overline{IRQ}
- 8 : Transmetteur en sous-charge
- 9 : Reset
- 10 : Sélection du boîtier
- 11 : Sélection du registre

- 13 : Ecriture / lecture
- 14 : Activation
- 15 : D₇ ; 18 : D₄
- 16 : D₆ ; 19 : D₅
- 17 : D₅ ; 20 : D₂

- 21 : D₁
- 22 : D₀
- 23 : Présence de la Porteuse
- 24 : CTS

Registre	Caractéristiques de Commande		Commande d'adresse		Contenu du Registre							
	RS	RW	AC2	AC1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Etat (S)	0	1	X	X	Demande d'interruption (IRQ)	Erreur de Parité (PE)	Recepteur en sur-charge (Rx Ovrr)	Transmetteur en sous-charge (T.U.F)	Inhibition du Transmetteur. (CTS)	Présence en Porteur de Données (DCO)	Registre de transmission Disponible (TDRA)	Donnée Reçue Disponible (RDA)
Commande 1 (C1)	0	0	X	X	Commande d'Adresse 2 (AC2)	Commande d'Adresse 1 (AC1)	Autoisation interruptions du récepteur (RIE)	Autoisation interruptions du transmetteur (TIE)	RAZ Sync	Suppression des caractères de Synchronisation Strip Sync	Mise à l'état initial du Transmetteur Tx Reset	Mise à l'état initial du Récepteur Rx Reset
Fifo de Réception	1	1	X	X	D7	D6	D5	D4	D3	D2	D1	D0
Commande 2 (C2)	1	0	0	0	Autoisation Interruption sur erreur (EIE)	Transmission de Caractères de synchronisation en sous-charge (Tx Sync)	Longueur des mots 3 (WS3)	Longueur des mots 2 (WS2)	Longueur de mot 1 (WS1)	Transferts par 1 octet ou 2 octets (1 Octet)	Commande périphérique 2 (PC2) 2 octets	Commande périphérique 1 (PC1)
Commande 3	1	0	0	1	Inutilisé	Inutilisé	Inutilisé	Inutilisé	RAZ Etat de sous-charge du transmetteur (CTUF)	RAZ CTS	Synchronisation sur 1 ou 2 caractères (1sync/2sync)	Synchronisation Externe / Interne (E/I sync)
Caractères de synchro	1	0	1	0	D7	D6	D5	D4	D3	D2	D1	D0
APIFO de Transmissi ON	1	0	1	1	D7	D6	D5	D4	D3	D2	D1	D0

Fig III 33 : Registres programmables du S.S.D.A.

III. 1.3.2.- Signaux d'interface du SSDA Avec le MPU MC 6800 :

Le SSDA communique avec le MC 6800 par :

- les signaux CS et RS issus du bus d'adresse.
- le bus de donnée Do - D7
- par quatre ligne du bus de controle
- le signal d'entrée d'activation E relié à la phase Q2 de l'horloge du MC 6800
- la ligne RIW (selection des registres à lecture ou à écriture)
- l'entrée Reset qui permet la remise à l'état initial du SSDA
- la ligne d'interruption IRQ qui indique au MPU
- la présence d'une donnée dans le registre de reception
- le passage à l'état haut de l'entrée DCD (présence de la porteuse de donnée)
- la surcharge du recepneur
- erreur de parité sur le signal reçu
- le passage à l'état haut de l'entrée CTS (inhibition du transmetteur)
- la sous-charge du transmetteur.

III. 1.3.3.- Signaux d'interface du MC 6852 avec la périphérie :

Le S.S.DA dialogue avec la périphérie par :

- les broches de transmission Tx data et de réception Rx data controlées par les horloges appliquées en Tx et K et R x clk ; ces deux broches sont des lignes de données.

- les lignes de contrôle pour la sortie SM / DTR (synchro Match/Data terminal Ready) ; les entrées CTS (cléar to send) et DCD (data Carrier Detect) et la sortie TUF (Transmitter Under Flow).

III.1.3.4.- Fonctionnement en mode de transmission :

La pile de transmission FIFO (First in - first out) ; comprend trois registres de 8 bits.

Pour initialiser la transmission, le microprocesseur lit le contenu du bit T x Rest "Remise à l'état initial du transmetteur" du registre de commande 1, si ce bit est un "0" la transmission commence sur la première transition positive de l'horloge de transmission, la donnée parallèle est alors transférée du bus de donnée dans la dernière position du registre Fifo.

De la dernière position du registre FIFO, la donnée est envoyée dans le registre à décalage de transmission. A la sortie de ce dernier, nous récupérons un chainage de bits de données série commençant par le bit de poids faible. Quand le registre à décalage devient vide et qu'aucune donnée ne se trouve dans le registre FIFO de transmission, une impulsion indique la sous-charge sur la sortie "TUF" et un caractère de synchronisation est transmis sur le front négatif de l'impulsion de l'horloge de transmission.

Remarque 1 :

Dans les systèmes disque souple, le signal TUF est utilisé pour synchroniser les opérations d'écriture et pour ajouter les deux bytes CRC.

Quand le bit "T x Reset" est mis à 1 le bit d'état "T x DRA" est remis à "0" indiquant que le registre "FIFO" de transmission est remis à "0".

III.1.5.- Fonctionnement en mode de réception :

La pile de réception FIFO comprend trois registres de 8 bits. L'état de ces registres est indiqué par le bit 1 du registre d'état.

Les données et l'horloge de synchronisation sont reçus respectivement par les entrées Rx Data et Rx clock.

Les données séries sont reçues sous forme de flot continu de bits, l'identification des limites des caractères est donc impossible sans une synchronisation préalable.

Dans le SSDA, trois types de synchronisation de caractère sont possibles :

- synchronisation sur un caractère
- synchronisation sur deux caractères
- synchronisation externe.

Le choix de l'un ou de l'autre mode de synchronisation interne se fait par le bit 2 du registre de commande 1.

Remarque 2 :

Les unités de disque souple utilisent la synchronisation à deux caractères qui consiste à chercher bit par bit le premier caractère de synchronisation puis le deuxième caractère successif. Quand la synchronisation est détectée, une impulsion est envoyée sur la sortie "SM" (Synchro-Match).

Une fois que la donnée est disponible dans le FIFO de réception, une interruption IRQ est envoyée au microprocesseur si le bit RIE "Autorisation des interruptions du Récepteur" est à "1", pour que les données soient lues par le microprocesseur. Quand un caractère arrive au registre FIFO, pendant qu'il est plein, le bit "Rx OVRN" du registre d'état indique que le "récepteur est en surcharge" ; le bit "EIE" (autorisation interruption sur erreur) est mis à "1" et une lecture du registre FIFO se fait.

Remarque 3 :

Les systèmes disque souple reçoivent et transmettent les bits de poids fort en premier, il y aura donc inversion des bits de donnée dans tous les tableaux figurant dans ce chapitre.

III.1. A : Opération de lecture :

Les informations enregistrées sur un secteur "X" appartenant à une piste "Y" d'un disque souple, se présentent comme suit :

Marque d'adresse ID	n° piste	n° secteur	CRC	CRC	Marque d'adresse des données	Informations	CRC	CRC
_____ champ ID _____					_____ Champ des données _____			

Pour plus de précision revoir le chapitre II (format IBM).

Pour lire un enregistrement, on lit d'abord l'identificateur du secteur donc le champ ID puis les données.

Les marques d'adresse sont utilisées pour la synchronisation.

L'opération de lecture se fait en deux étapes principales.

1. une étape de synchronisation qui consiste à comparer les deux moitiés du byte de la marque d'adresse (un byte = 8 cellules bits) au contenu du registre caractère de synchronisation du S.S.D.A. pour synchroniser l'opération de lecture.

- une étape de lecture.

a.- Etape de synchronisation.

A la sortie du circuit de récupération des données, la première moitié de la marque d'adresse est introduite dans le registre à décalage de réception du S.S.D.A. ; elle est ensuite envoyée au comparateur pour être superposée au contenu du registre caractère de synchronisation du S.S.D.A. introduit par le logiciel ; si les deux caractères coïncident, le signal "synchro-match" (SM) est envoyé au "synchro-match latch" (bascule JK du circuit de lecture (U31) pour le remettre dans son état de travail. La deuxième moitié de la marque d'adresse est alors envoyée et testée de la même façon que la première moitié, si elle ne concorde pas avec le contenu du registre caractère de synchronisation, la recherche de la première moitié de la marque d'adresse recommence ; dans le cas contraire, l'opération de lecture est initialisée.

b.- Etape de lecture :

Après la synchronisation décrite précédemment, l'opération de lecture est initialisée et la deuxième moitié de la marque d'adresse est stockée dans le registre FIFO de réception du S.S.D.A. ; le signal de l'horloge 250 KHz sera sélectionné et envoyé au S.S.D.A de telle sorte que seuls les bits de données du signal "Read Data" récupéré à la sortie du circuit de séparation des données, soient échantillonnés.

c.- logique de lecture des données :

La logique de lecture des données est représentée par la figure III.4.1. Les chronogrammes correspondant à cette logique sont donnés dans la figure III.4.2.

Dans le format IBM 3740, les marques d'adresse utilisées pour synchroniser les opérations de lecture sont :

"F5 FE" pour la marque d'adresse de ID

"F5 F6" pour la marque d'adresse des données

Pour l'explication de cette logique, le système est supposé initialisé par le software à travers le PIA. En d'autres termes on suppose que :

- la tête a été chargée en activant la ligne PA4 du PIA.
- le signal "Enable Read" (ligne PB2 du PIA) est activé.
- les bascules "Synchro Match Latch" et "Switch clock rate latch" sont mises en position de travail par le passage de l'état de la ligne Reset (PBo du PIA) à un niveau logique "1".

Le circuit de récupération des données fournit à la logique de lecture les données et l'horloge de fréquence 500 Khz nécessaires pour son fonctionnement. Figure III.4.1.A, B et figure III.4.2.A,B.

Quand la première moitié de la marque d'adresse (F5) introduite dans le comparateur du S.S.D.A. coïncide avec le contenu du registre caractère de synchronisation de ce dernier, le signal "SM", de durée une période de l'horloge de lecture (Read Clock), passe à l'état haut (figure III.4.1B et figure III.4.2.B).

La transition négative de l'horloge "Read Clock" qui arrive pendant que le signal "SM" est à l'état haut remet la sortie Q de la bascule "Synchro-Match Latch" à un niveau haut et le compteur diviseur par 16 est alors validé. (figure III.4.1E et figure III.4.2.E).

Une fois testée, la deuxième moitié (7B ou F6) de la marque d'adresse est envoyée dans le registre FIFO de réception et sera considéré comme le premier byte de donnée reçu car la première moitié de la marque d'adresse ne sera pas transférée dans le registre FIFO ; l'opération de lecture est alors initialisée.

La bascule "Switch clock rate latch" activée par la sortie Q3 du compteur diviseur par 16 contrôle le selectionneur (AND/OR) du signal d'horloge. Tant que la sortie Q3 est à un niveau logique "0" ; la "Switch clock rate latch" qui avait sa sortie Q, préalablement, à l'état haut, permet la sélection du signal d'horloge 500 Khz. Le signal ainsi sélectionné est appliqué à l'entrée "Read clock" du S.S.D.A. (figure III.4.1.C et figure III.4.2.C.).

Fig III 41: LOGIQUE DE LECTURE

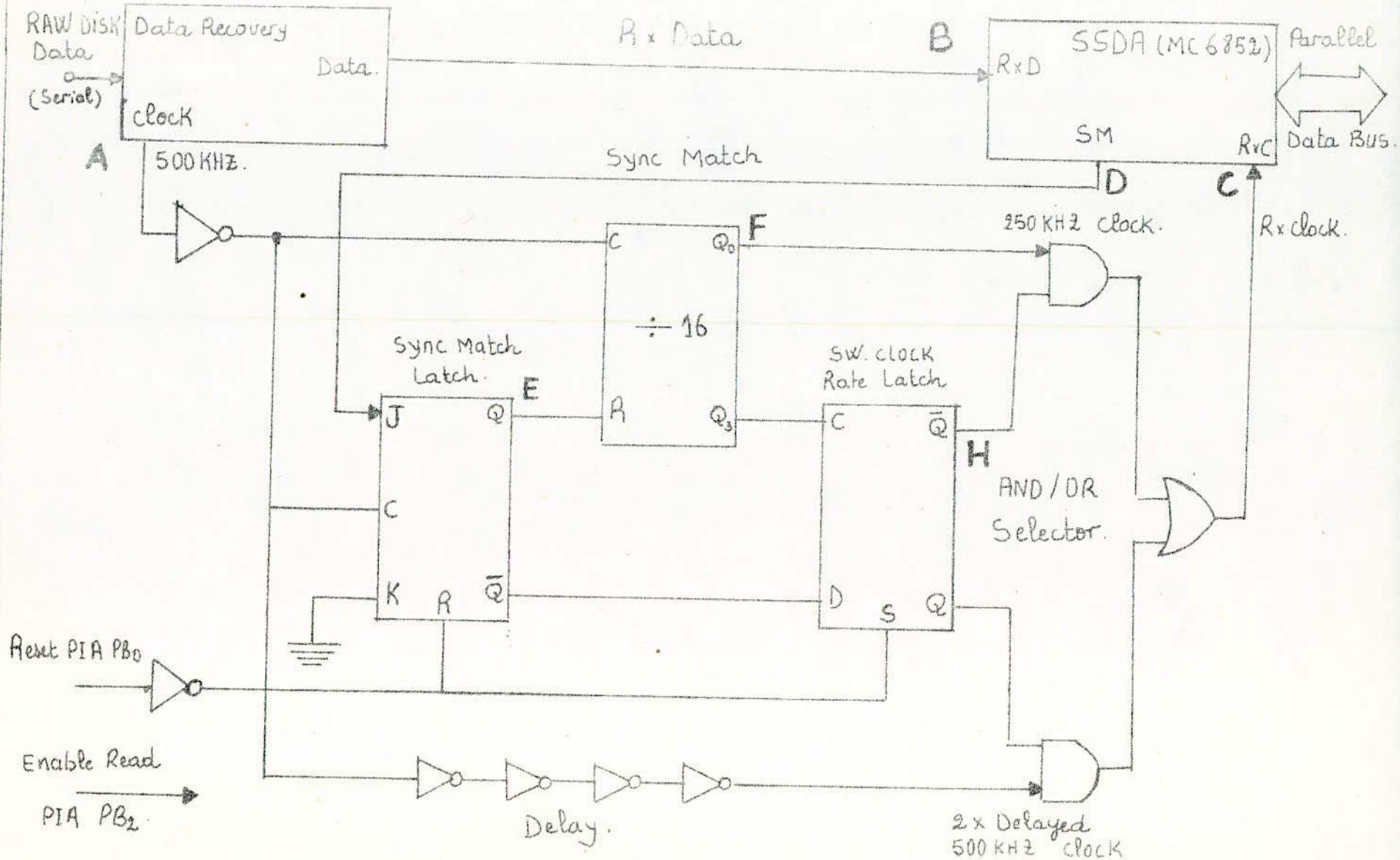
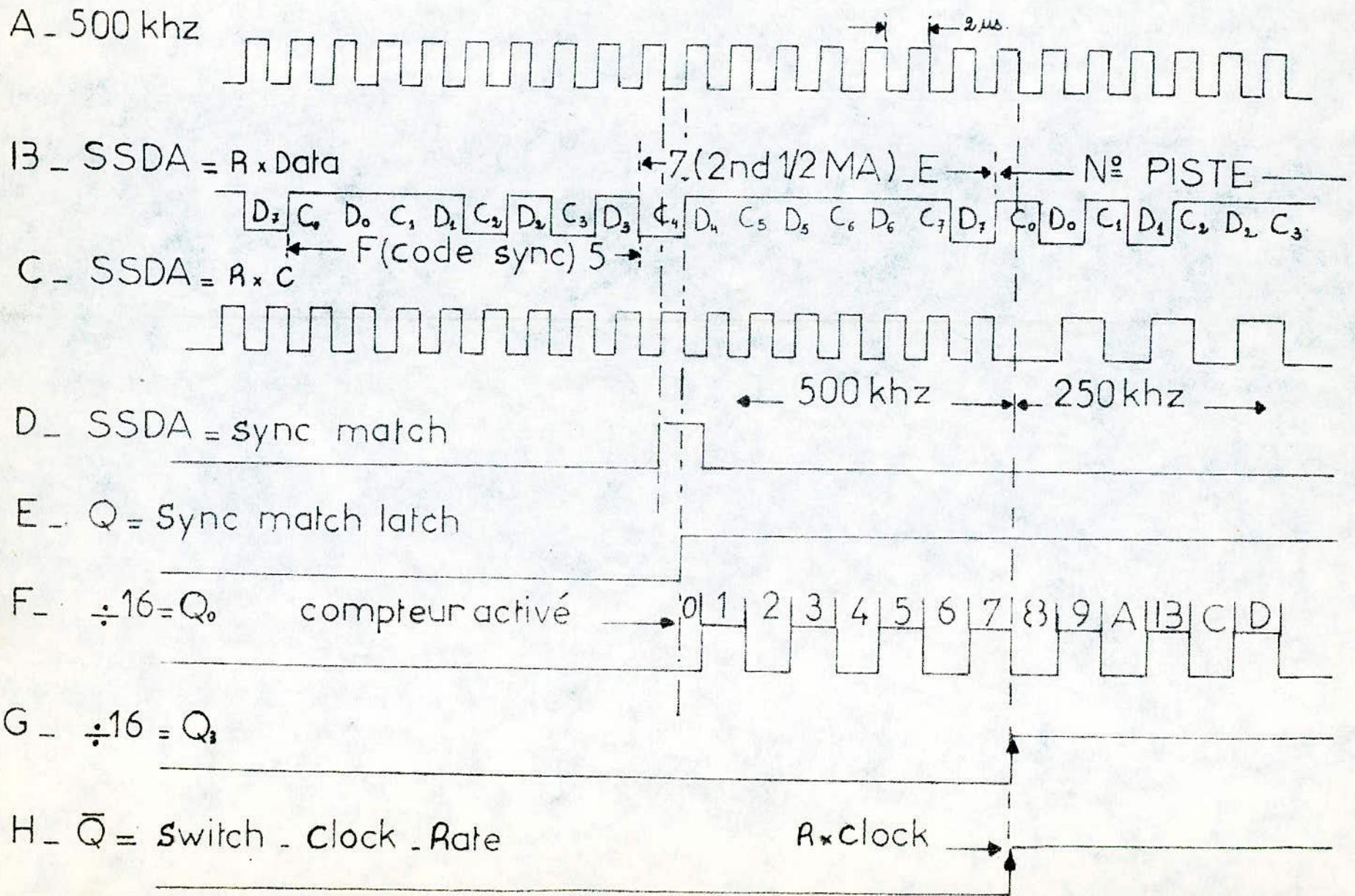


fig III. 4.2 CHRONOGRAMME : OPERATION DE LECTURE .



La première transition positive Q3 sera basculer la "Switch clock latch" dont la sortie Q sélectionnera le signal d'horloge 250 Khz récupéré à la sortie Qo du compteur diviseur par 16. Ce signal est envoyé à l'entrée "RxC" du S.S.D.A. de telle sorte que seuls les bits de données du signal "Read data" soient stockés dans le registre FIFO de réception. Figure III.4.1.F G,H,C et figure III.4.2 F,G;HC.

Comme le compteur diviseur par 16 et la "switch clock rate latch" sont synchronisés par le signal d'horloge 500 Khz, le signal "Read clock" de fréquence 500 Khz est retardé par quatre inverseurs pour que le selectionneur commute l'horloge sur la fréquence 250 Khz avant la prochaine transition positive de 500 Khz.

Quand les deux derniers registres de la pile "Rx FIFO" sont pleins, une interruption IRQ est envoyée au microprocesseur pour qu'une lecture des données parallèles ait lieu.

L'opération de lecture continuera à une fréquence de 250 Khz jusqu'à la remise à zéro indiquée par le logiciel.

Les huit comptages de retard entre le "synchro-match" et le point de commutation de la bascule "switch-clock-rate-latch" permettent à la deuxième moitié de la marque d'adresse d'être envoyée dans le "Tx FIFO" avec une fréquence de 500 Khz.

III.1.5.- DETECTION ET CORRECTION D'ERREURS.

A l'écriture ou à la lecture des données plusieurs erreurs peuvent altérer l'information écrite ou lue. Ces erreurs peuvent-êtré causées par :

- un enregistrement sur une piste endommagée.
- un mauvais positionnement de la tête.
- une variation de la vitesse du moteur d'entraînement du disque.
- une variation de la fréquence du secteur etc.....

Pour vérifier l'intégrité d'un enregistrement, après une opération d'écriture, les données sont relues au prochain tour du disque et si après 10 tentatives, l'erreur persiste, la piste est considérée comme endommagée.

Si l'erreur arrive lors d'une opération de lecture, l'opérateur relit pendant dix fois le secteur ou la piste et si l'erreur n'est pas corrigée elle est irrémédiable.

Pour la détection d'erreur des données enregistrées sur le disque, on effectue un contrôle par redondance cyclique (CRC) réalisé par le générateur de polynôme CRCCG MC 8506 "Cyclic Redondancy check character Generator".

a.- Fonctionnement du MC 8506 : (Figure III.5.1).

Le générateur MC 8506 est caractérisé par son polynôme :

$G(X) = X^{16} + X^{12} + X^5 + 1$ représenté par le caractère binaire : 1000 100000 100001.

Le contrôle d'erreurs par le générateur MC 8506 consiste à diviser les données à enregistrer sur le disque par le polynôme $G(X)$; le reste de la division constituera les deux bytes CRC envoyés à la fin de chaque zone de données enregistrées ; à la lecture les données et les deux bytes CRC seront divisés par le même polynôme $G(X)$, si le reste est égale à zéro aucune erreur n'est détectée.

Le principe de contrôle est le suivant :

Les n bits d'un flot de données reçues sont considérés comme les coefficients d'un polynôme de degré $(n-1)$.

Le motif binaire $B_{n-1} B_{n-2} \dots \dots \dots B_1 B_0$ est interprété comme :

$$F_n(x) = B_{n-1} x^{n-1} + B_{n-2} x^{n-2} + \dots \dots \dots + B_1 x^1 + B_0 x^0$$

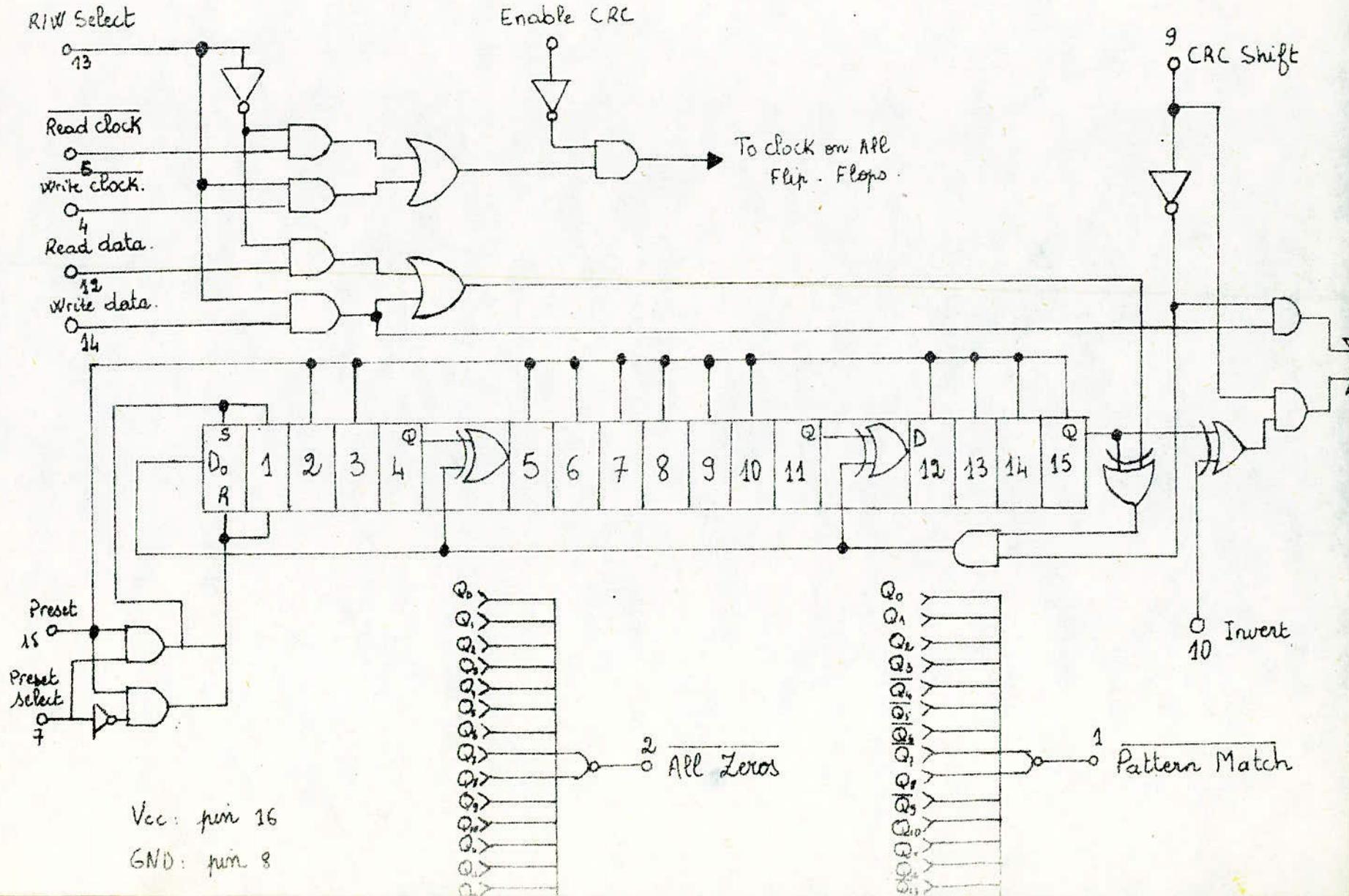
Quand le générateur reçoit pour la première, la chaîne $B_{n-1} B_{n-2} \dots \dots \dots B_1 B_0$, il divise le polynôme $F_n(x)$ par $G(X)$ et on obtient un quotient $Q(X)$ et un reste $R(X)$ tels que :

$$F_n(X) = Q(X) \cdot G(X) + R(X).$$

Le reste $R(X)$ sera juxtaposé à la chaîne $B_{n-1} B_{n-2} \dots \dots \dots B_1 B_0$ sous forme de deux bytes CRC.

Fig III.5.1.

CONFIGURATION INTERNE DU MC 8506.



A la lecture, le générateur recevra toute la zone de données (y compris les deux bytes CRC) ; l'ensemble sera divisé par le même polynôme $G(x)$; si aucune erreur n'a affecté l'enregistrement des données, le reste de cette division sera nul, car en additionnant le caractère de contrôle (reste) à la chaîne des bits de données, on aura rendu le message reçu divisible par le polynôme $G(x)$:

$$\underline{F(x) = R(x) = Q(x) \cdot G(x)}$$

et un signal "ALL ZERO" (AZ) sera envoyé comme le montre la figure III.5.1.

Quand une opération de détection d'erreurs est entamée, la ligne "Préset select" est activée (par le logiciel) pour mettre les bascules formant les trois registres du CRCCG en position travail. Ces bascules avaient, préalablement, leurs états à un niveau logique 1.

Les tableaux, suivants montrent un exemple de calcul et de vérification de deux bytes CRC par le générateur MC8506.

Le motif binaire utilisé pour ces exemples est : 0111100110011000111.

FIGURE 5A - GENERATION OF CRC-COITT
($X^{16} + X^{12} + X^5 + 1$) CHECK CHARACTER

Shift n	Input Data on Shift	Register Contents After nth Shift															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
5	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
6	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
7	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
8	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
9	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
10	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
11	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
12	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
13	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
14	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
15	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
16	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
17	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
18	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
19	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
20	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
21	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
22	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
23	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
24	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
25	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
26	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
27	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
28	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
29	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
30	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
31	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
32	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
33	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
34	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
35	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	

FIGURE 5B - RECEPTION OF CRC-COITT
($X^{16} + X^{12} + X^5 + 1$) CHECK CHARACTER

Shift n	Input Data on Shift	Register Contents After nth Shift															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
5	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
6	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
7	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
8	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
9	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
10	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
11	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
12	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
13	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
14	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
15	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
16	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
17	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
18	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
19	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
20	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
21	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
22	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
23	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
24	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
25	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
26	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
27	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
28	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
29	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
30	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
31	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
32	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
33	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
34	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
35	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	

All Zeros
indication

b.- opération de détection d'erreurs :

Quand un champ est lu du disque, les bits de données (du bit zéro de la marque d'adresse au bit sept du second byte CRC) seront divisés par le polynôme du CRCCG.

Le signal d'horloge de fréquence 250 Khz, récupéré à la sortie Q₀ d'un compteur diviseur par 16 (500/2) après l'arrivée d'un "synchro-match", est appliqué au CRCCG et ne laisse passer que les bits de données, dans les registres à décalage internes du CRCCG. Le signal "Read Data" sera retardé de huit cycles d'horloge de 500 Khz afin que la première moitié de la marque d'adresse arrive à l'instant où le CRCCG est activé.

c.- Logique de contrôle d'erreurs de lecture :

Un diagramme de contrôle d'erreurs de lecture est donné par la figure III.5.2. Les chronogrammes correspondants sont représentés par la figure III.5.3.

L'explication de cette logique suppose que l'opération de lecture est initialisée comme elle a été décrite dans le paragraphe III.1.4.

Les signaux de données et d'horloge (500 Khz) sont fournis à la logique CRC par le circuit de récupération des données (Cf. : figure III.5.2. A,B et figure III.5.3. A,B). Ce signal est identique à celui reçu par la logique de lecture.

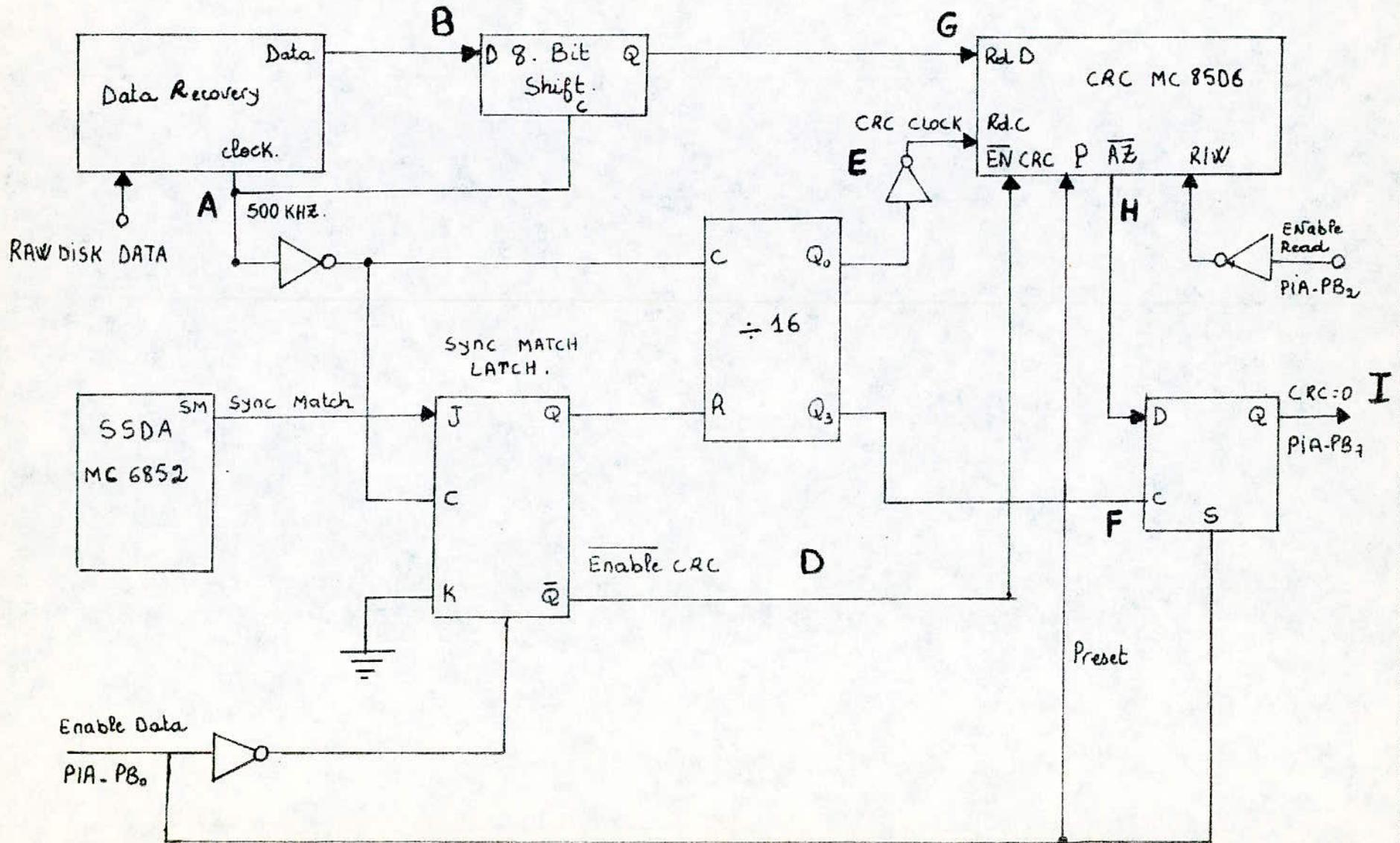
Quand la première moitié de la marque d'adresse envoyée, par le circuit de récupération des données, dans le S.S.D.A concorde avec le contenu du registre caractère de synchronisation, le signal "synchro-match" passe à l'état haut figure III.5.2.C et figure III.5.3.C)

La première transition négative du signal "Read Clock" (signal d'horloge reçu après inversion à la sortie du circuit de récupération), qui arrive pendant que le "synchro-match" est à l'état haut, activera la bascule "synchro-match-latch" dont la sortie Q, générant le signal "ENCRC" (ENABLE CRC, passe à l'état bas et active les registres à décalage internes du CRCCG (MC 8506) (Cf. figure III.5.2.D et figure III.5.3.D).

D'autre part la sortie Q validera le compteur diviseur par 16 (U5) et à sa sortie Q₀, un signal d'horloge de fréquence 250 Khz est récupéré, inversé puis injecté dans le CRCCG par l'entrée (Rdxc) si bien que seuls les bits de données de lecture sont chargés dans le générateur CRCCG. (figure III.5.2.D.E. et figure III.5.3.E). Ce signal est appelé ("CRC Clock").

Le générateur CRCCG reçoit les données retardées, par un registre à décalage de huit bits activé avec un signal d'horloge de 500 Khz. Le retard, de huit cycles d'horloge 500 Khz (quatre bits de données), permet au générateur CRCCG de recevoir la première moitié de la marque d'adresse après sa validation qui ne peut avoir lieu que lorsque le "synchro-match" est envoyé par le S.S.D.A (figure III.5.2.B.G et figure III.5.3.B.G).

fig III.5.2 : LOGIQUE DE DETECTION D'ERREURS



A - Recupération De Données - 500khz

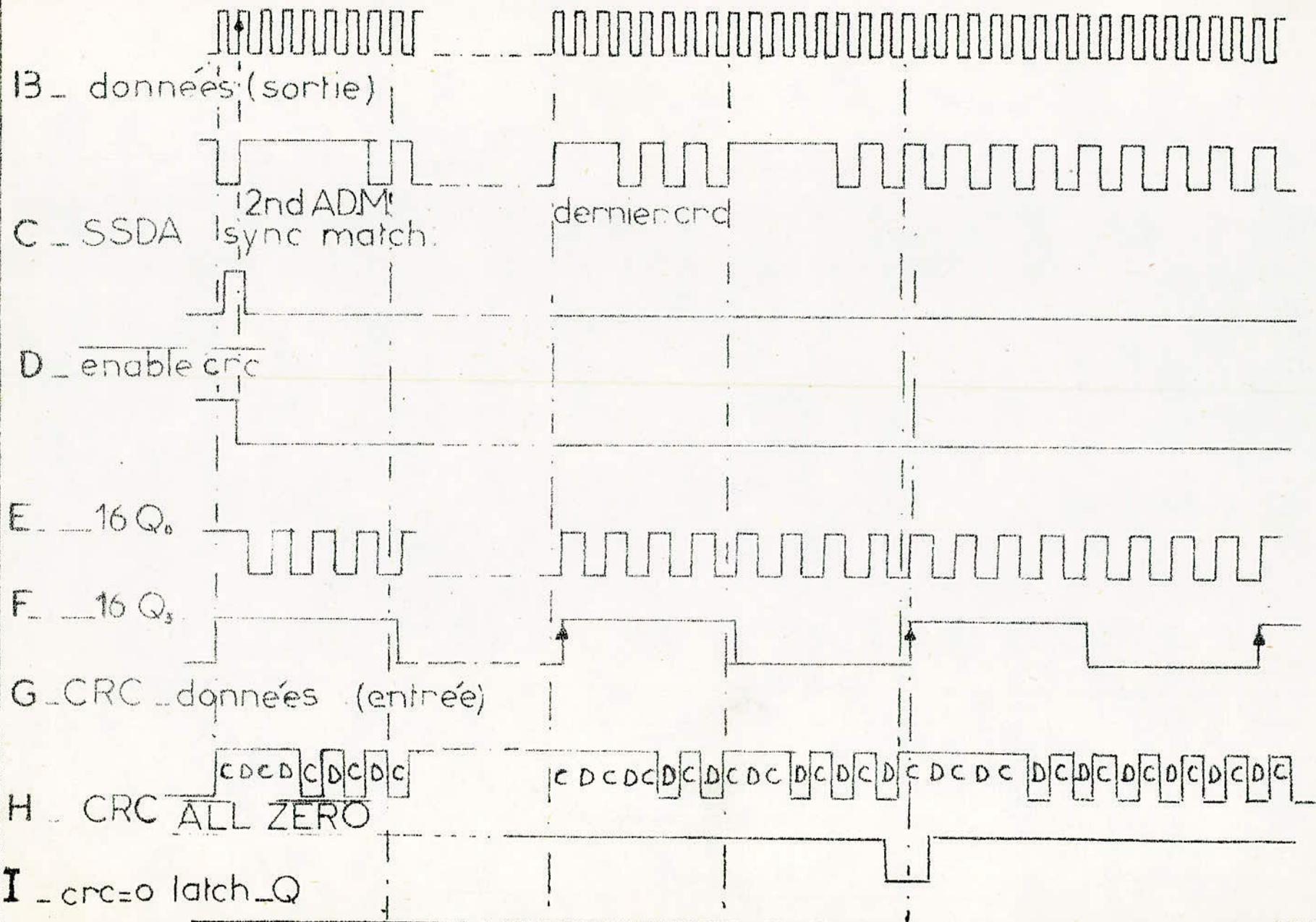


fig III.5.2: CHRONOGRAMME : DETECTION D'ERREURS

Si le champ ID ou le champ de données (y compris les deux bytes CRC) ont été correctement lus le signal \overline{AZ} (ALL ZERO) passera à l'état bas pendant une période "CRC-Clock" (figure III.5.2.H.G. et figure III.5.3.G.H). Le signal \overline{AZ} est appliqué à l'entrée D d'un "Flip-Flop" appelé "CRC = 00 latch", (U 24) qui est activé par la sortie Q3 du compteur diviseur par 16. La première transition positive de Q3, arrivant après passage de \overline{AZ} à l'état bas, bascule le "CRC = 00 latch" dont la sortie devient basse. (figure III.5.2.F.H.I. et figure 5.3.F.H.I.). Le "CRC = 00 latch" gardera son état bas jusqu'à l'arrivée de la prochaine transition positive de Q3, après une durée d'un byte (figure III.5.3.F.I.).

L'état de la sortie Q du "CRC = 00 latch" est lu par software à travers la ligne du PIA : PB7. Le test, par software, d'une erreur doit se faire pendant la durée du byte (~~pas~~ tambule) qui suit immédiatement le dernier byte CRC.

Si une erreur de lecture a été détectée, la ligne \overline{AZ} ne passera pas à l'état bas et la sortie Q du "CRC = 00 latch" restera à l'état pendant le test de l'erreur de durée un byte. ↓
bas

Après avoir terminé le contrôle CRC d'un champ ID ou d'un champ de données, la logique de détection d'erreurs de lecture doit être initialisée avant la lecture du prochain champ par la ligne "Reset" du PIA (ligne PBo).

Remarque :

Les différentes erreurs qui peuvent être détectées par l'interface en adressant une disquette, sont représentées sous forme de code dans un tableau dans le paragraphe III.3.

III.1.6.- Opération d'écriture.

a.- Principe de l'opération d'écriture.

Quand une opération d'écriture est lancée, la sous charge du registre "FIFO" de transmission du S.S.D.A. est indiquée par des impulsions sur la sortie "Transmetteur en sous-charge" ("TUF") du S.S.D.A. et la marque d'adresse index ' (voir format IBM dans le chapitre II) contenue dans le registre caractère de synchronisation est envoyée avec une fréquence de 500 Khz au registre à décalage du S.S.D.A.

A chaque impulsion "TUF", le registre caractère de synchronisation envoie son contenu pour synchroniser l'opération.

Après onze (11) impulsions "TUF", la première moitié de la marque d'adresse (donnée et horloge) désirée sera stockée dans le registre "FIFO" de transmission du S.S.D.A. et cinq et demi (5,5) bytes (un byte = 8 cellules bits) du gap "post-index" seront écrits sur le disque. Quand la première moitié de la marque d'adresse est envoyée avec une fréquence de 500 Khz au registre à décalage de transmission du S.S.D.A., la sortie "TUF" devient inactive, la seconde moitié de la marque d'adresse (donnée et horloge) est alors stockée dans le registre "Tx FIFO" suivie des données à enregistrer sur la disquette. Après transmission de toute la marque d'adresse dans le C.R.C.C.G, l'horloge de transmission du S.S.D.A. est commutée sur la fréquence de 250 Khz.

Quand le transfert des données sur la disquette est achevé, le registre "Tx FIFO" se charge avec 2 bytes fictifs afin que la sortie "TUF" reste inhibée et la ligne "Shift CRC" soit activée par le MPU à travers la ligne PB3 du PIA ; les deux bytes CRC calculés sont alors envoyés pour être écrits à la suite de la chaîne des données sur le disque.

A la fin de la séquence de l'écriture des 2 bytes CRC sur la disquette, une impulsion "TUF" est envoyée par le S.S.D.A. pour indiquer à nouveau une sous charge et le contenu (postambule) du registre caractère de synchronisation est envoyé avec une fréquence de 250 Khz au C.R.C.C.G. puis sur la disquette.

Durant toute l'opération d'écriture, le C.R.C.C.G. est activé avec un signal d'horloge de fréquence 250 Khz de telle sorte que seuls les bits de données (y compris les bits données de la marque d'adresse) transitent par les registres à décalage du C.R.C.C.G.

Le signal des données récupéré à la sortie du C.R.C.C.G. sera codé avant d'être envoyé sur la disquette. Le courant d'écriture, contrôlé par le MPU à travers la ligne PA2 (ABOVE 43) du PIA, est initialisé au début du gap "Post-index" et arrêté après écriture du byte "Postambule".

b.- Logique d'écriture.

La logique d'écriture est représentée par 2 blocs diagrammes donnés en figures (III.6.1. et III.6.5).

Les chronogrammes correspondants sont donnés dans les figures III.6.2, III.6.3, III.6.4, III.6.6, III.6.7.

La synchronisation de l'opération d'écriture est accouplée par la sortie "TUF" du S.S.D.A.

Une impulsion, sur la sortie "Transmetteur en sous charge" ("TUF") survient pendant le transfert de la dernière unité du dernier bit précédent le caractère de synchronisation. La durée qui sépare deux tops "TUF" est égale à la durée d'un byte de synchronisation.

Pendant le transfert de la marque d'adresse index, l'impulsion "TUF" valide le compteur diviseur par 16 ; sa sortie Q3 active les bascules "Switch-clock rate latch" et "Enable CRC".

Après huit comptages, le compteur diviseur par 16, activé par un signal d'horloge d'écriture "Memory clock" divisé par deux de fréquence 500 Khz, est remis à Zéro et sa sortie Q3 garde l'état bas tant que le registre "FIFO" de transmission du S.S.D.A. est vide. figure III.6.1.E et figure III.6.2.E.

A chaque impulsion "TUF", le registre caractère de synchronisation envoie son contenu pour synchroniser l'opération. Ce contenu qui peut être soit une marque d'adresse d'index soit un caractère du gap "Post(index)" sera directement écrit sur la disquette sans participer au calcul des deux bytes CRC car le CRCCG n'est pas encore validé. figure III.5.1.

Après comptage de onze "TUF" (ce qui correspond à l'envoi de cinq et demi bytes du gap "Post-index", la première moitié de la marque d'adresse désirée est stockée dans le registre "Tx FIFO" et le signal "TUF" est inhibé, et à la fin du dernier byte du "Post(index)" reçu le "Flip/Flop" "Enable CRC" est basculé par le passage de la sortie Q3 du compteur diviseur par 16 à l'état haut. La première transition positive de la sortie Q3 active donc le générateur CRCCG par le passage du signal "EN CRC" ("ENABLE CRC") à l'état bas.

Le CRCCG est synchronisé par la fréquence 250 Khz délivré par la sortie Q0 du compteur diviseur par 16, si bien que seule la partie donnée de l'information (y comprise la partie donnée de la marque d'adresse) à transmettre transite par les registres à décalage du CRCCG figure III.6.1.D et figure III.6.2.D.

Quand le dernier byte des données a été transféré du registre "Tx FIFO" au registre à décalage du S.S.D.A., deux bytes fictifs chargeront le "Tx FIFO", pour que la sortie "TUF" reste désactivée, et le HPU valide le signal "SHIFI-CRC" à travers la ligne PB3 du PIA figure III.6.5.K et figure III.6.6.K.

fig III.6.1 LOGIQUE D'ECRIURE

Write Osc : 500KHz

A

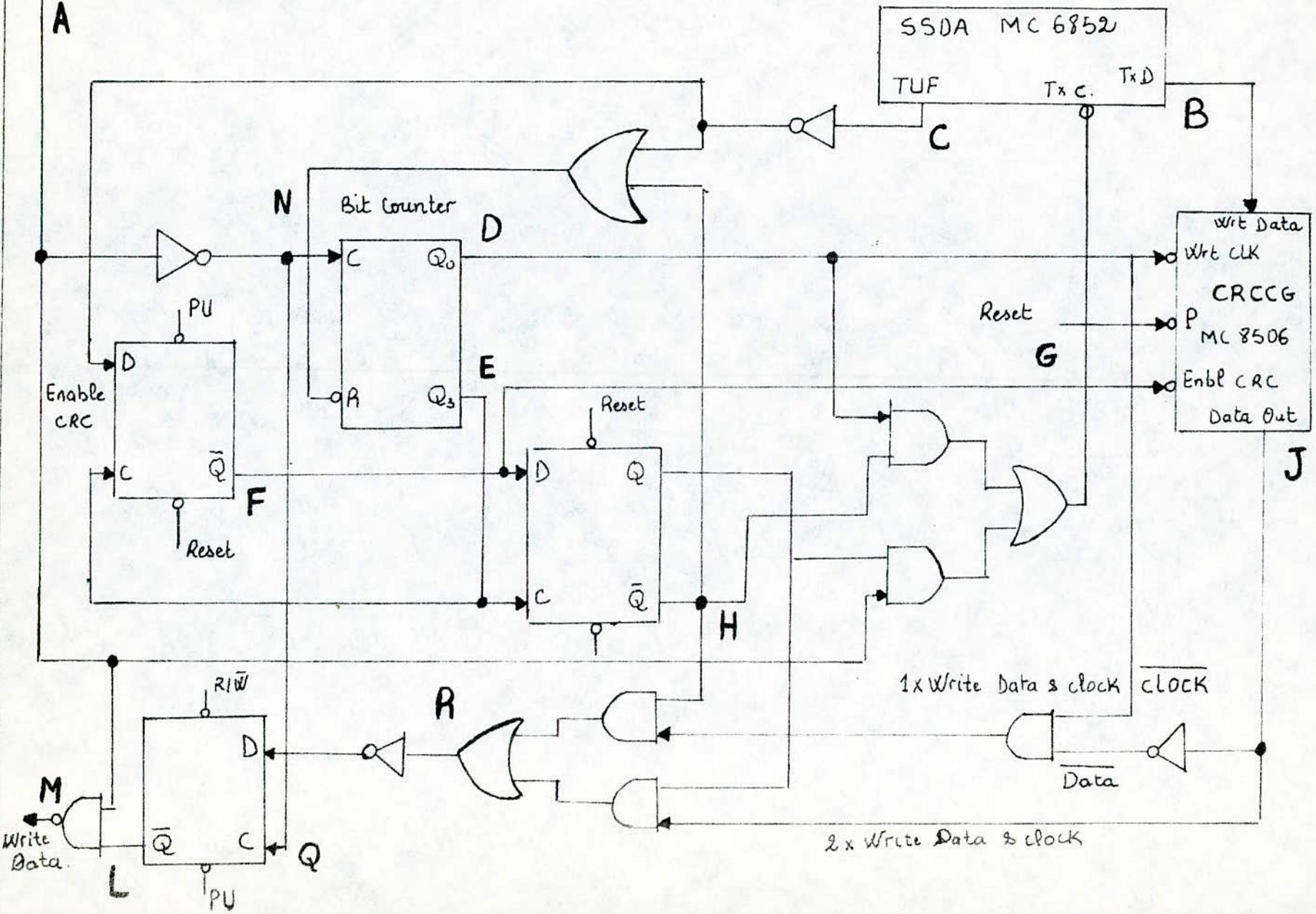


fig III.6.2 : CHRONOGRAMME : OPERATION ÉCRITURE

I - SSDA - tx clk



gap post_index 1^{ère} 1/2 MA

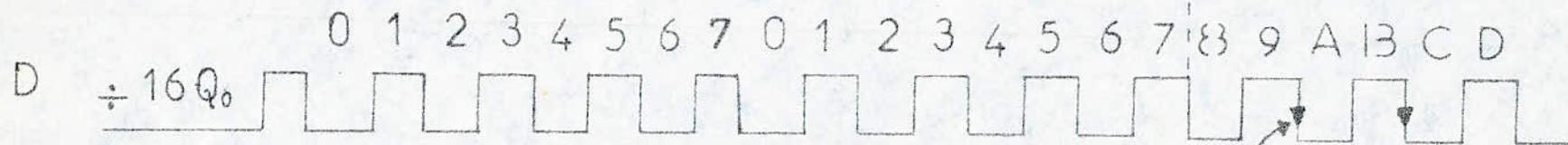
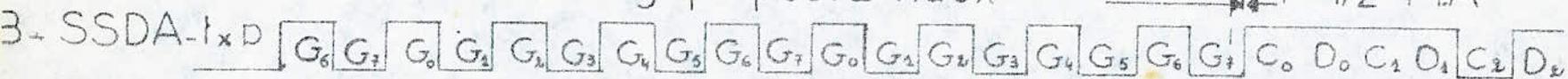


Fig III 63. CHRONOGRAMME - OPERATION D'ECRITURE (switch clock rate)
 A - fréquence d'écriture : 500khz

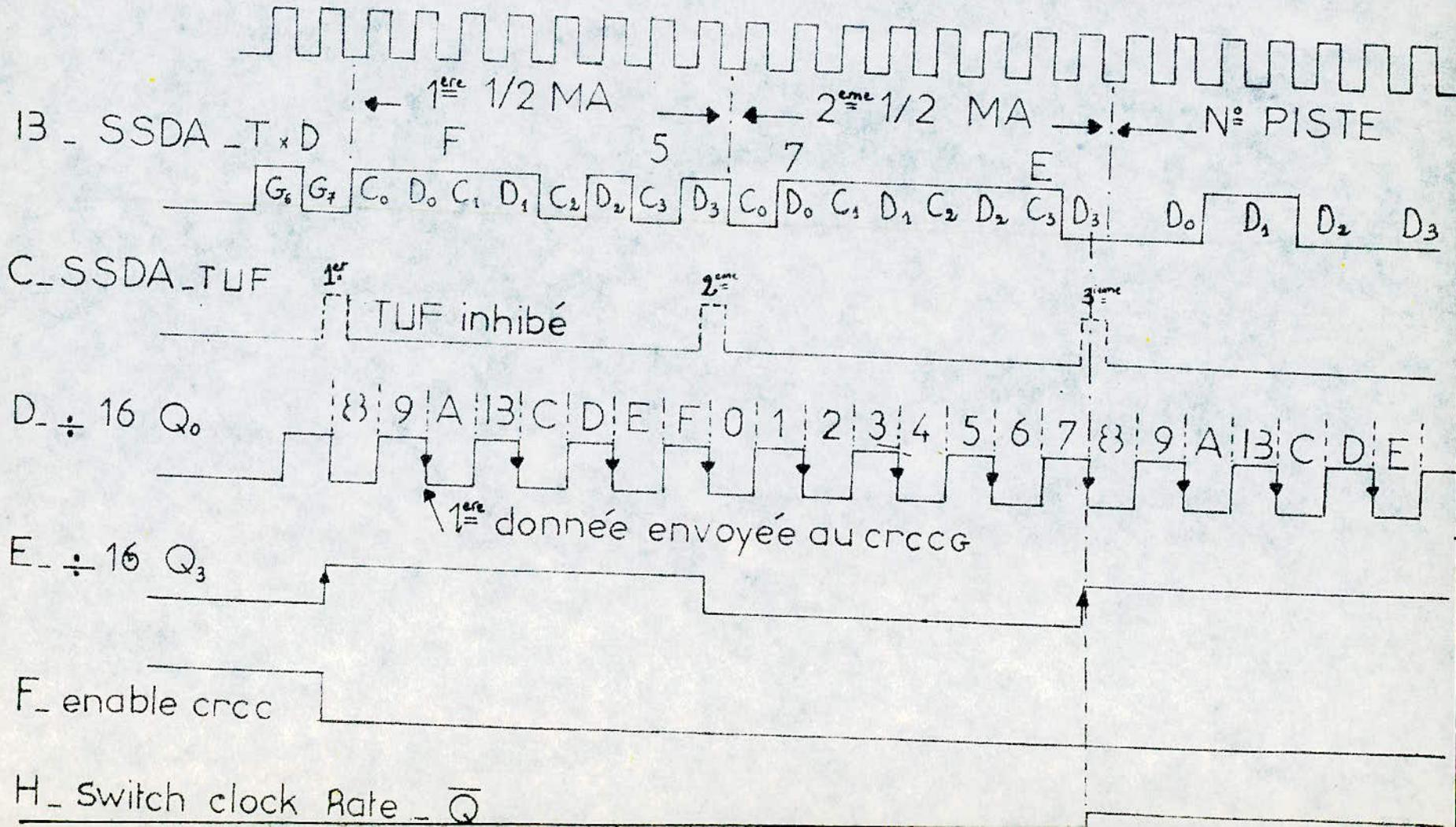
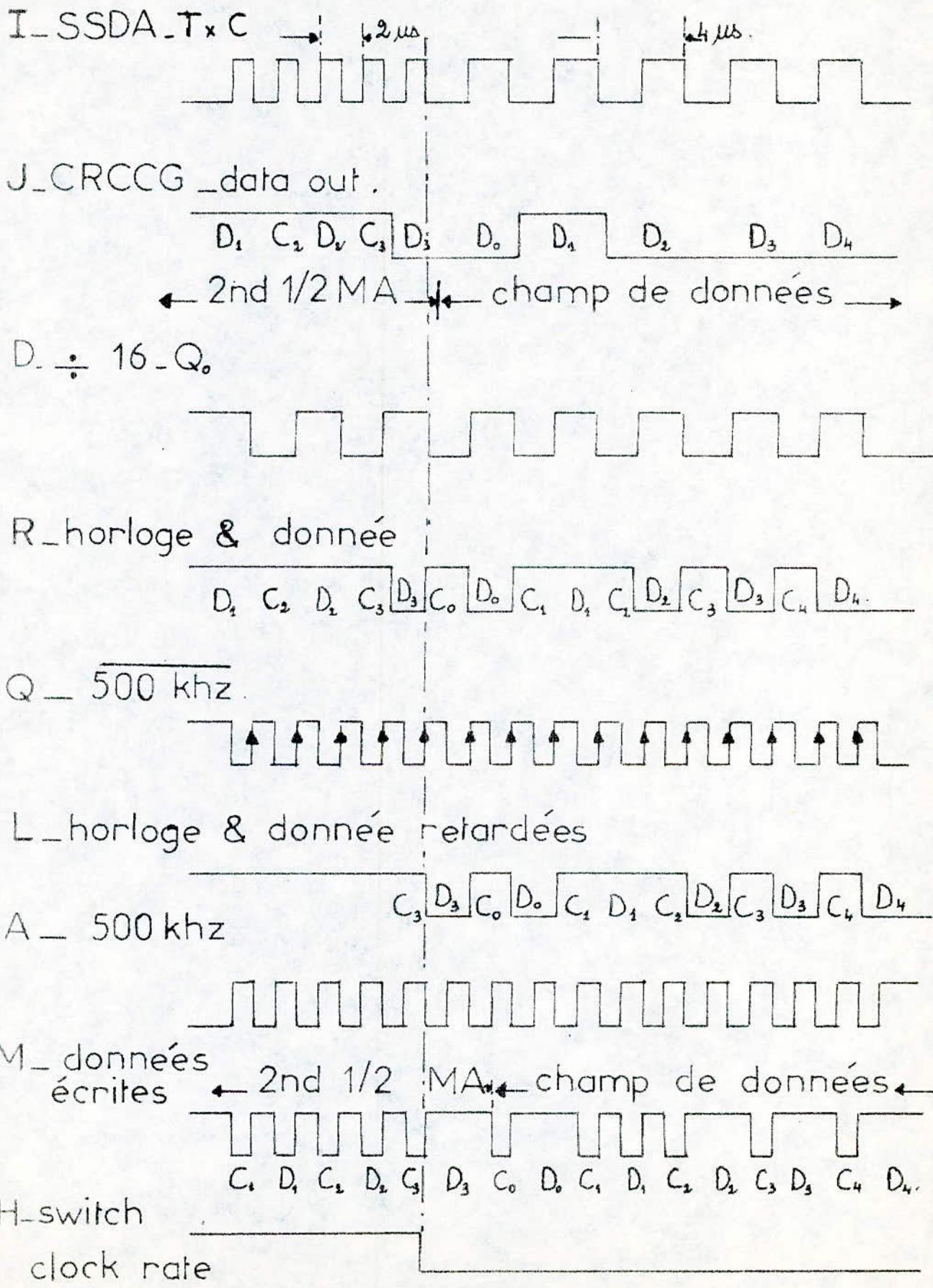


fig III.6.4 CHRONOGRAMME : OPERATION D ECRITURE



Le signal "SHIFT-CRC" est activé par la première transition positive, de la sortie Q3 du compteur diviseur par 16, qui se présente à la fin du dernier byte des données transmises et il est retardé d'une microseconde par le signal d'horloge d'écriture de fréquence 500 Khz pour permettre au dernier bit de donnée d'être transféré au registre du CRCCG avant de commencer le transfert, sur le disque souple, des deux bytes CRC calculés figure III.6.5.P.S et figure III.6.6.P.S.

L'activation de la commande "SHIFT CRC" empêche l'écriture des deux bytes fictifs envoyés par le S.S.D.A. et autorise ainsi l'écriture sur la disquette des deux bytes CRC calculés. figure III.5.1.

Quand le dernier bit du deuxième byte CRC est écrit :

- la commande "SHIFT CRC" est inhibé par le MPU
- la sous charge est indiquée par le S.S.D.A.
- le registre de caractère de synchronisation envoie un postamble au CRCCG. figure III.6.5.P.A. figure III.6.6.P.S.A et figure III.6.7.P.S.A.

La première transition positive de Q3 qui arrive avec la première impulsion "TUF" basculera le "Flip/Flop" "Enable CRC" dont la sortie Q passera à un état haut ; les registres à décalage du CRCCG sont alors désactivés et le postamble sera écrit directement sur le disque souple. figure III.6.4.C.E.F. et figure III.6.8.C.E.F.

La prochaine transition de Q3 du compteur diviseur par 16 commutera l'horloge sur la fréquence 500 Khz ce qui permettra à la seconde impulsion "TUF" de remettre à zéro le compteur diviseur par 16 figure III.6.1.E, figure III.8.E.I.

La séquence d'écriture est ainsi terminée.

Pendant l'écriture du "Postamble", le registre de caractère de synchronisation se charge avec un gap constitué d'un mélange de données et d'horloge avec une fréquence de 500 Khz.

Les caractères de ce gap seront transmis à la logique d'écriture et joints au "postamble" sur la disquette, puisqu'à ce que l'opération d'écriture soit recommencée par l'envoi de la marque d'adresse du prochain enregistrement au registre "FIFO" de transmission du S.S.D.A. ou jusqu'à ce que le transmetteur soit remis à zéro par le logiciel.

La bascule "Switch clock rate latch" permet également le contrôle du sélecteur des données à envoyer au circuit de décodage.

Quand le signal, récupéré à la sortie du générateur CRCCG, est constitué d'un mélange de données et d'horloge (marque d'adresse ou gap) de fréquence 500 Khz, il sera appliqué directement au selectionneur (AND/OR) ; s'il n'est constitué que de données (information) de fréquence 250 Khz, il sera inversé puis appliqué à une porte "AND" avec un signal d'horloge de fréquence 250 Khz (clock) pour obtenir un mélange de données et d'horloge qui sera ensuite envoyé au sélecteur (AND/OR) figure III.6.5. J, H.

Fig III.6.5 LOGIQUE DE LA GENERATION DES BYTES CRC

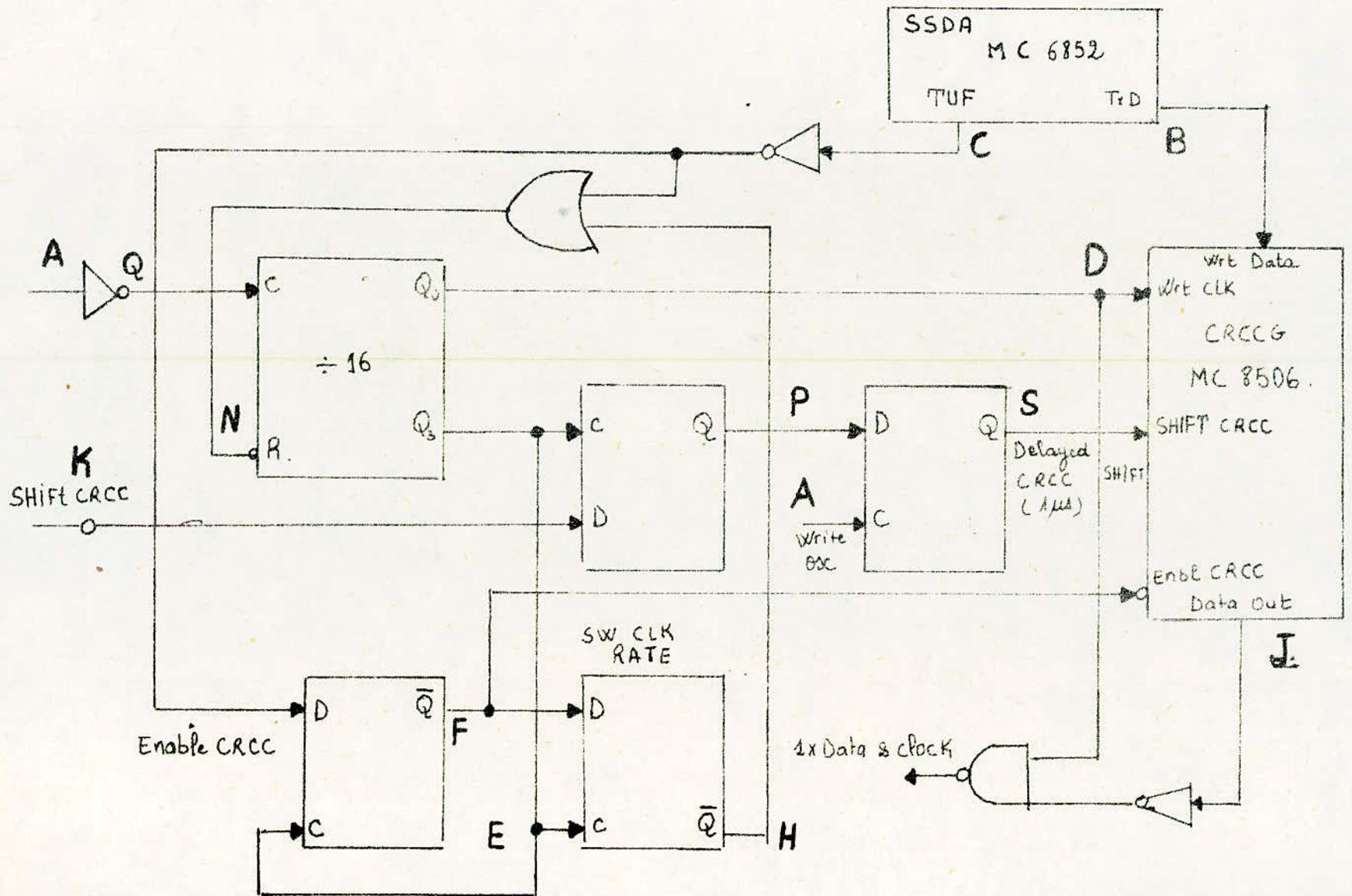


fig III.6.6 : CHRONOGRAMME : GENERATION DES 2 BYTES CRC

fig III.6.6 : CHRONOGRAMME : Génération des 2 bytes CRC.

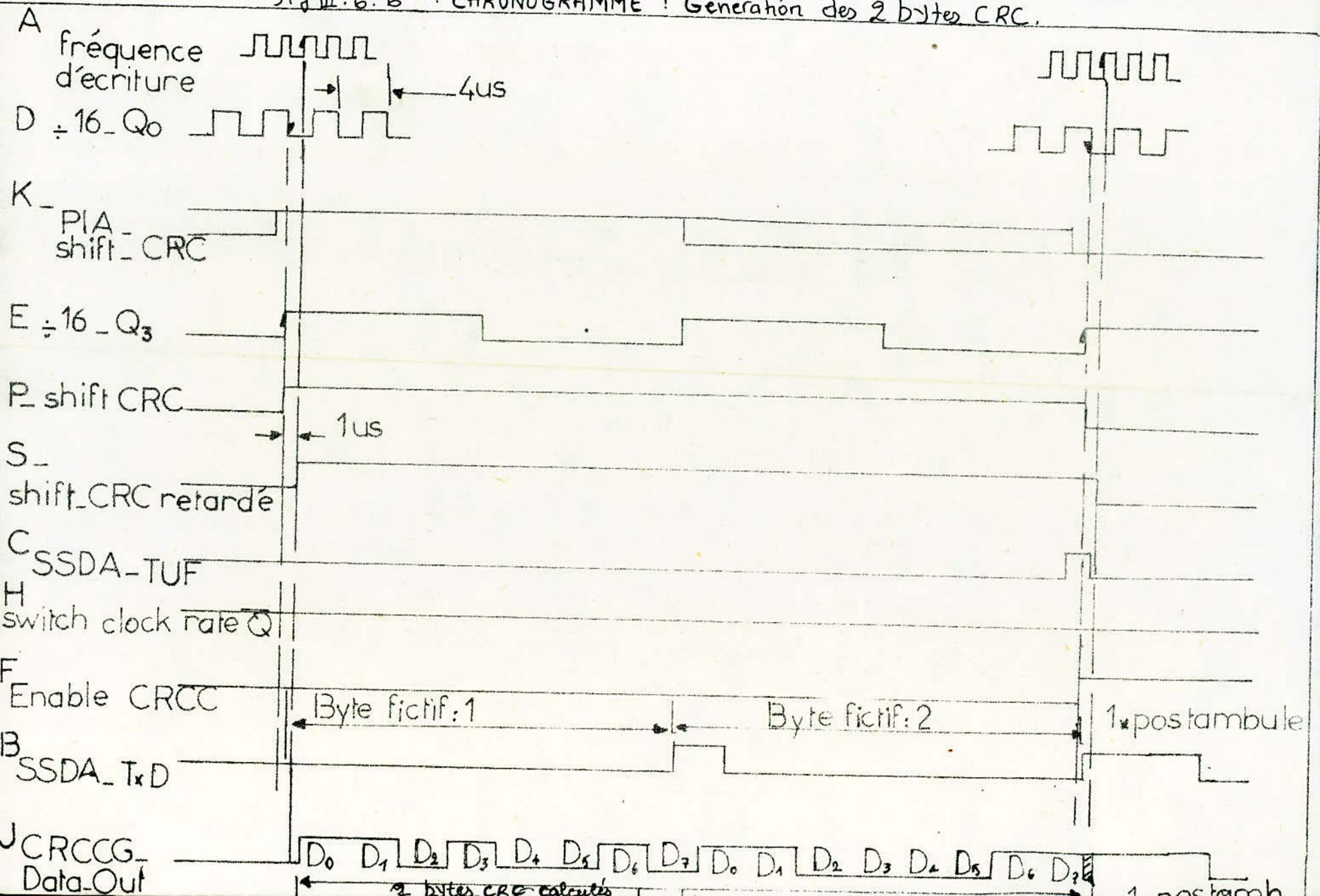
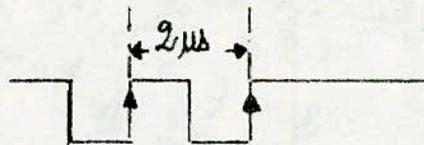


fig III 6.7: CHRONOGRAMME : OPERATION D'ECRITURE

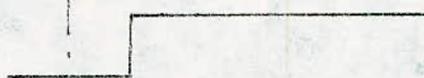
A_ 500 khz



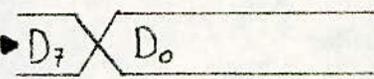
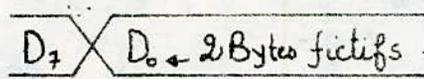
P_ shift crcc activé



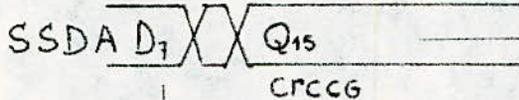
S_ crccg shift



B_ SSDA Tx D



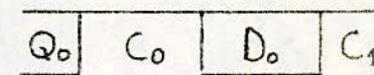
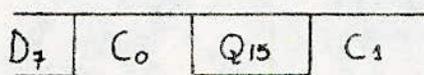
U_ crccg data out



D_ ÷ 16 Q0



R_ donnée & horloge



Q_ 500 khz



donnée &

horloge retardées

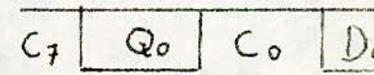
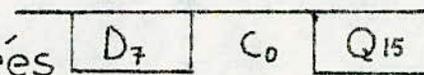
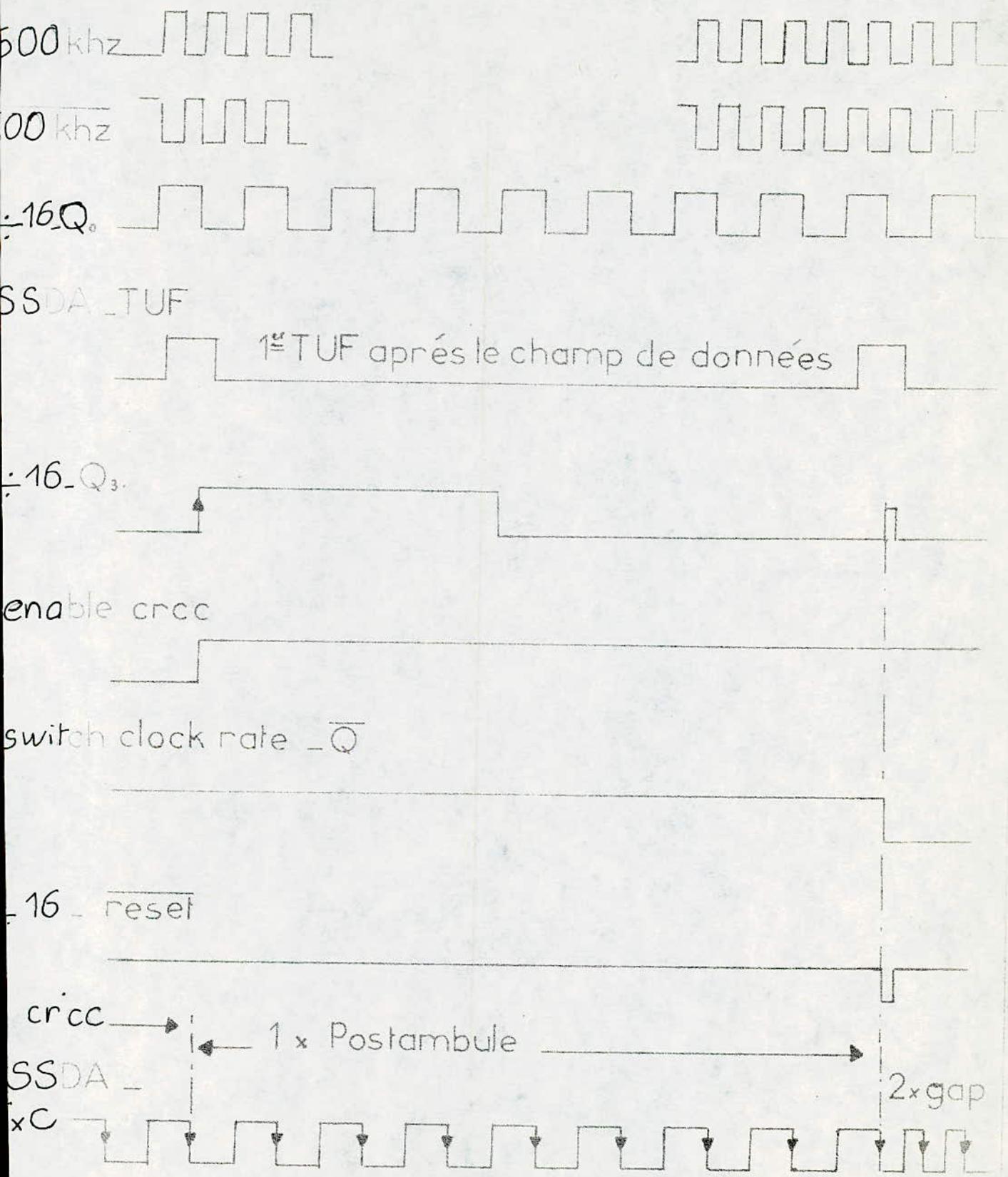


Fig III.6.8 CHRONOGRAMME : OPERATION D'ECRITURE



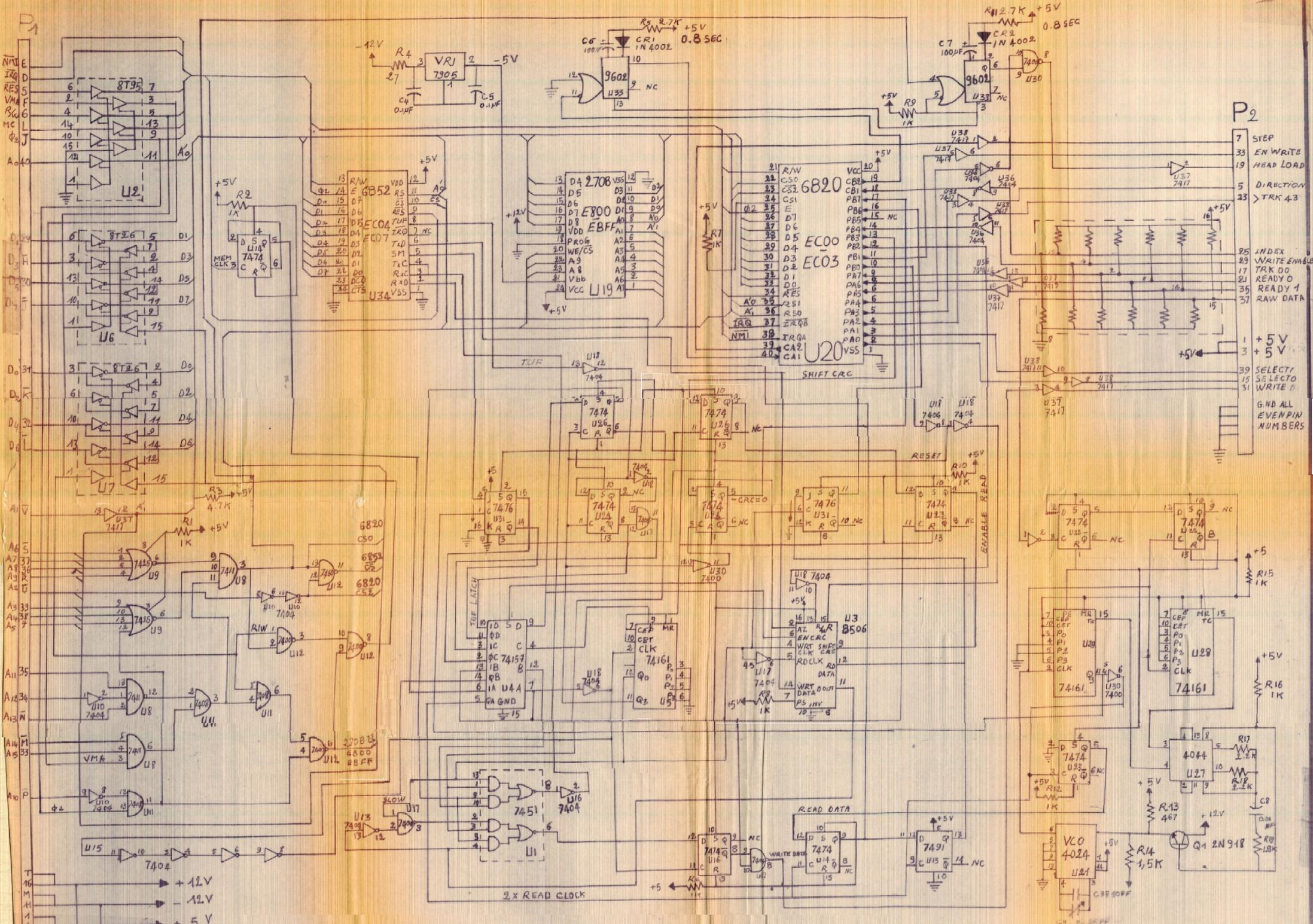
Le signal de données récupéré à la sortie du sélecteur en format NRZ, est inversé puis appliqué à une bascule D, synchronisée par le signal d'horloge d'écriture, pour le retarder d'une demi période du signal de fréquence de 500 Khz afin de supprimer l'effet de tous les retards qui peuvent être produits dans la logique d'écriture figure III.6.1.R, figure III.6.4.R, figure III.6.4.L, figure III.6.1.L.

Le signal ainsi obtenu est appliqué à une porte "AND" avec un signal d'horloge d'écriture de fréquence 500 Khz pour obtenir le signal codé "Write dat." enregistré sur le disque figure 6.1.M et figure III.6.4.H.

III.2.- Réalisation du contrôleur.

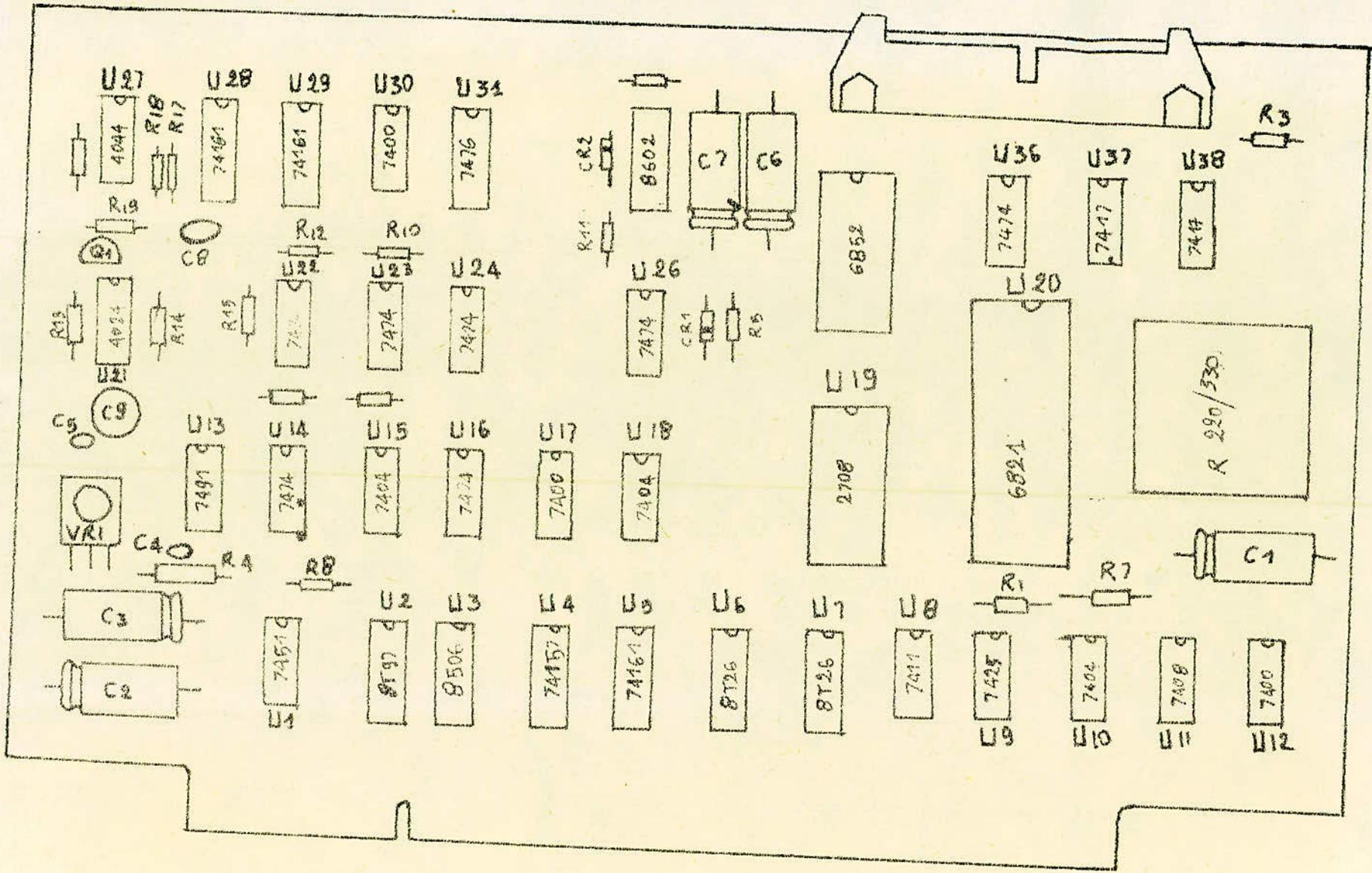
III.2.1.- Schéma électrique.

III.2.2.- Schéma d'implantation.



CONTROLEUR DE DISQUE SOUPLE

- P2
- 7 STEP
 - 33 EN WRITE
 - 19 HEAD LOAD
 - 5 DIRECTION
 - 33 > TRK 4.3
 - 25 INDEX
 - 29 WRITE ENABLE
 - 17 TRK 00
 - 21 READY 0
 - 35 READY 1
 - 37 RAW DATA
 - 1 +5V
 - 3 +5V
 - 39 SELECT
 - 15 SELECT
 - 31 WRITE 0
 - GND ALL EVEN PIN NUMBERS



SCHEMA D'IMPLANTATION

III.3: LOGICIEL :

Le contrôleur de disque souple occupe 1032 positions mémoires réparties comme suit :

- E 800 à EBFF : zone mémoire réservée au programme "Résident Driver Firmware" contenu dans la mémoire "ROM" (2708) et développé par "MOTOROLA", pour contrôler toutes les fonctions "HARDWARE" du drive.

- EC00 à E C03 et EC04 et EC07 ; positions mémoires occupées respectivement par le PIA et le S.S.D.A.

Le contrôleur de disque souple est initialisé par la commande E 800 ; G réalisée sous la fonction "INIT" de l'EXORCISER, si un système d'exploitation disque est disponible sur la disquette ; dans le cas contraire l'utilisateur peut développer son propre programme d'initialisation en chargeant les paramètres représentés dans le tableau suivant :

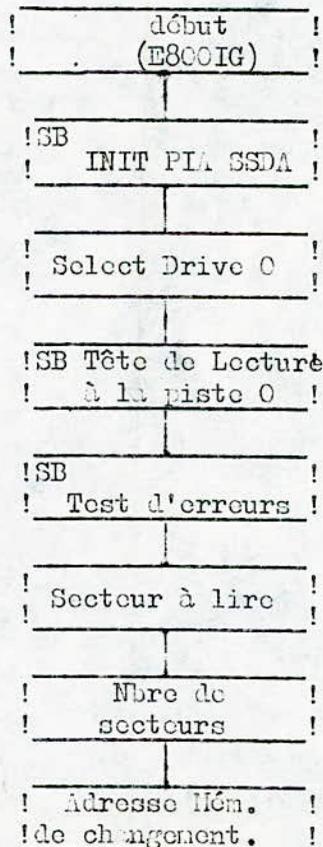
ADRESSE	NON	DEFINITION
00	CURDRV	<u>Drive utilisé</u> : Ce byte contient le numéro du drive sélectionné (0 ou 1).
01	STRSCT	<u>Premier secteur</u> :
02		Ces deux bytes contiennent le numéro du premier secteur utilisé pendant une opération de lecture ou d'écriture. Pour tous les disques de simple densité, ce numéro doit être compris entre 0 et 2001.
03	NUMSCT	<u>Nombre de secteurs</u> :
04		Ces deux bytes contiennent le nombre de secteurs utilisés. Ce nombre ne dépasse pas "2002" (26 x 77) pour les disques de simple densité.
05	LSCTLN	<u>Longueur du dernier secteur</u> : Pendant une opération de lecture, ce byte contient le nombre de bytes à lire à partir du dernier secteur ; ce nombre sera compris entre 1 et 128.
06	CURADR	<u>Adresse en cours</u> :
07		Ces deux bytes contiennent la première adresse vers à partir de laquelle la donnée sera écrite / lue. Le programme "résident driver firmware" indique une incrémentation de "1" du contenu de ces deux bytes après lecture ou écriture d'une donnée.

08 ! FDSTAT ! Etat du disque :
! !
! ! Ce byte contient l'état du disque indiqué par le
! ! programme "résident driver firmware".
! !
! ! Si une erreur a eu lieu pendant une opération sur
! ! disque, le bit "carry" est mis à "1" et le byte
! ! "FDSTAT" contiendra le numéro du byte d'erreurs
! ! (voir tableau "code d'erreurs") si aucune erreur
! ! n'a eu lieu, le bit "carry" sera remis à "0" et le
! ! byte "FDSTAT" contiendra le nombre 30 (en hexadé-
! ! cimal).
! !

Le programme "Résident Driver Firmware" listé en Annexe comprend :

1.- un chargeur élémentaire ou "BOOTSTRAP de chargement" qui permet le transfert en mémoire (à partir de la position 20) du programme "interpréteur de commandes" du système d'exploitation disque (IDOS : MOTOROLA DISK OPERATING SYSTEM) préalablement chargé sur 1. disquette.

"le Bootstrap de chargement" se présente comme suit :



!SB Lecture
! d'un secteur !

!SB Test
! d'erreurs !

! Branchement
! sur IDOS. !

où SB = Subroutine.

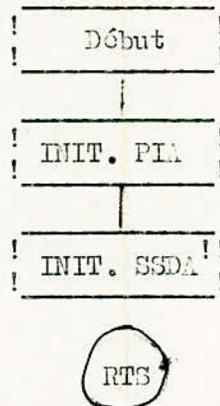
2.- Un ensemble de sous programmes utilisés par le "IDOS" pour exécuter les différents transferts sur disque.

3.- un ensemble de sous-programmes utilisés pour réaliser des sorties sur une imprimante rapide.

Les principaux sous-programmes utilisés sont :

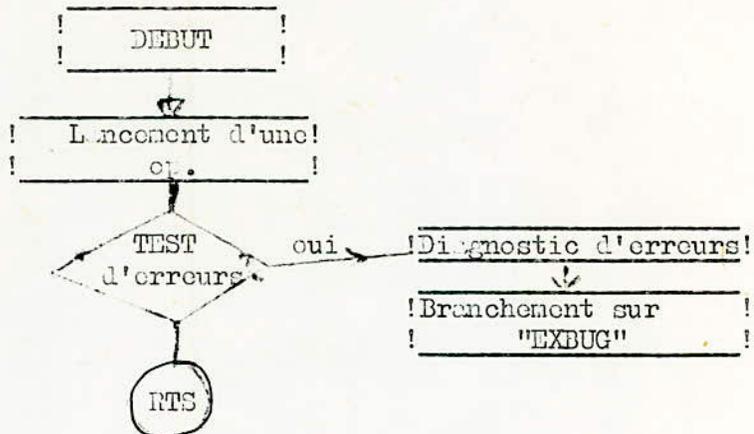
a.- FDINIT (initialisation) "E822" :

Ce sous-programme permet l'initialisation du PLA et du SSDA . nous pouvons l'illustrer par l'organigramme simplifié suivant :



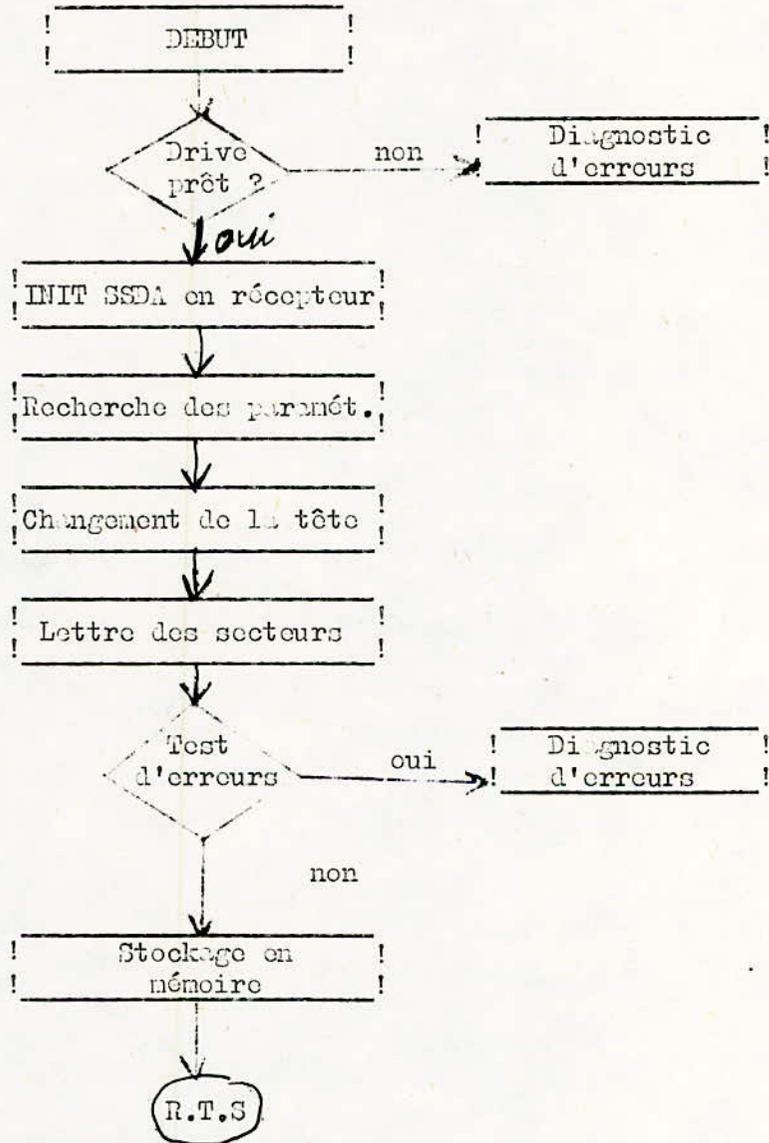
b.- CHKERR (Test d'erreurs) "E853" :

Si une erreur a eu lieu pendant une opération sur le disque, ce sous-programme inscrit le caractère "E" suivi du contenu du byte "FDSTAT" le contrôle est alors mis sous "EXBUG"



READSC (Lecture de secteurs) "E869"

Ce sous-programme permet la lecture d'un nombre de secteurs indiqué par le contenu du paramètre "NUSCT". Après la lecture de chaque secteur, le byte "CURADR" est mis à jour pour indiquer l'adresse de stockage du secteur lu.



WRITSC (Ecriture de secteurs) "E884" :

Ce sous-programme permet l'écriture d'une série de secteurs commençant par le contenu du paramètre STRSCT. Le nombre de secteurs à écrire est indiqué par le paramètre NUMSCT.

L'adresse du premier byte à écrire sur le disque est donnée par le contenu du "CURADR" qui s'incrémente de "2" après l'écriture d'un byte pour donner l'adresse du prochain byte à écrire.

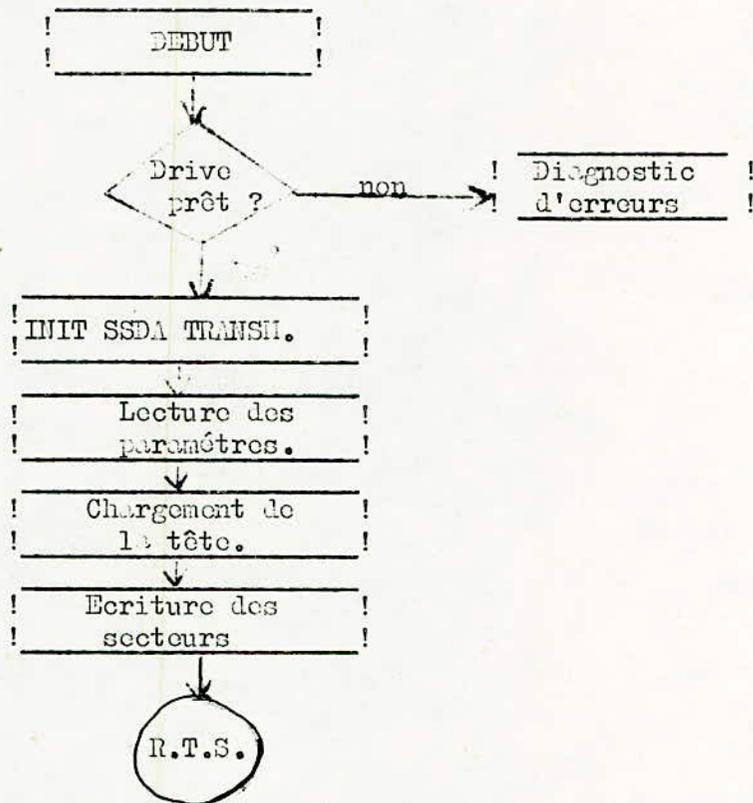


Tableau "code des erreurs" :

FDSTAT	ASC II	BIT "Carry"	Causes possibles
30	0	remis à 0	<p><u>Pas d'erreurs (E.0)</u></p> <p>Cet état est indiqué quand une erreur n'a altéré l'opération sur le disque. Le bit "carry" est alors remis à 0.</p>
31	1	mis à "1"	<p><u>Erreurs sur le CRC du champ de données :</u></p> <p>Pendant une opération de lecture, si les deux bytes CRC sont faux, un message (E.1) est envoyé.</p> <p>- si cette erreur arrive à la lecture de plusieurs secteurs, le numéro du secteur erroné peut être déterminé à l'aide de l'équation donnée à la fin de ce tableau.</p> <p>Cette erreur peut être causée par l'oubli d'écriture et de lecture des données ET/Ou du CRC.</p>
32	2	mis à 1	<p><u>Disque protégé en écriture :</u></p> <p>L'erreur (E2) arrive quand une opération d'écriture est lancée sur un disque protégé en écriture. Pour pouvoir écrire sur un tel disque ; une plaque opaque doit être placée au niveau du trou de protection situé sur l'enveloppe du disque.</p>
33	3	mis à 1	<p><u>Le disque n'est pas prêt :</u></p> <p>L'erreur (E3) arrive quand le disque n'est pas prêt, c'est-à-dire quand la porte du drive est ouverte ou lorsque le disque n'a pas atteint la bonne vitesse de rotation.</p>
34	4	mis à 1	<p>Lecture d'une marque d'adresse de donnée supprimée. L'erreur (E4) arrive à la lecture d'un secteur dont la marque d'adresse de donnée est supprimée. Ce secteur ne sera ni lu ni écrit sur la disquette.</p> <p>L'équation donnée à la fin de ce tableau permet de déterminer le numéro du secteur erroné.</p>

- 35 5 mis à 1 Désynchronisation :
L'erreur (E5) arrive chaque fois que le temps délivré par Hardware (monostable) est insuffisant pour le déroulement d'une opération.
Cette erreur arrive également quand une interruption non-masquable arrive pendant une opération sur le disque.
Une autre cause possible de cette erreur est la lecture ou l'écriture sur une mauvaise piste ou sur un mauvais secteur d'un disque non formaté.
Un reformatage du disque éliminera cette erreur.
Le message (E5) peut être également envoyé quand la tête de lecture se positionne sur une piste fautive.
- 36 6 mis à 1 Adresse non disponible sur le disque :
L'erreur (E6) est envoyée quand la somme des deux bytes STRSCT et NUMST dépasse le nombre des secteurs du disque.
- 37 7 mis à 1 Test d'erreurs :
L'erreur (E7) arrive si la tête de positionnement sur la piste 0 avant que l'opération "Test d'erreurs" n'ait eu lieu.
Cette erreur peut aussi avoir lieu si le drive n'est pas connecté au contrôleur de disque souple.
- 38 8 mis à 1 Marque d'adresse de données erronée :
L'erreur (E8) se présente quand la marque d'adresse de donnée n'est pas reçue.
Le message (E8) sera envoyé avant la lecture du secteur.
En utilisant l'équation écrite à la fin de ce tableau, le secteur erroné peut être déterminé.
- 39 9 mis à 1 CRC de la marque d'adresse erroné :
L'erreur (E9) a lieu avant la lecture ou l'écriture d'un secteur quand le CRC de la marque d'adresse est incorrecte, l'opération en cours est alors arrêtée.
Les causes possibles de cette erreur sont :
- un oubli d'écriture de la marque d'adresse ou de son CRC lors du formatage.
- un oubli de lecture de la marque d'adresse ou de son CRC.
Le numéro du secteur erroné sera déterminé par l'équation donnée à la fin de ce tableau.

EQUATION :

$$\boxed{PSNE = STRSCT + NUMSCT - STCNT - 1}$$

où PSNE est le n° du secteur erroné

STRSCT est le contenu des bytes STRSCT

NUMSCT est le contenu des bytes NUMSCT.

STCNT représente le contenu de 2 bytes situés aux positions "11" et "12" à la suite des paramètres d'initialisation.

Au début de chaque opération, ces deux bytes contiennent le nombre indiqué par le paramètre NUMSCT. Le nombre sera décrémenté au cours du déroulement de l'opération.

C O N C L U S I O N

Afin d'utiliser un "système d'exploitation disque" déjà établi et basé sur un logiciel développé par MOTOROLA ; nous avons étudié et réalisé une carte interface compatible avec ce même logiciel.

Nous remarquerons, cependant, que la taille du logiciel est assez réduite, ce qui a eu pour conséquence la conception d'une carte très riche en composants.

L'intégration à large échelle (LSI) qui incorpore, en un seul boîtier la plupart des fonctions, a rendu simple la conception des interfaces, nous citerons entre autres :

- le FDC MOTOROLA 6843 qui se relie directement au microprocesseur 6800.

- la série western Digital WD 17 xx qui présente les possibilités suivantes :

- * Il exécute des macro-commandes (fonctions qui nécessitent plusieurs actions) et ce sans que le microprocesseur n'ait à intervenir.

- * Il est capable de coder les données et de les décoder quand elles proviennent d'une disquette simple et double densité.

Pour notre part, nous dirons que ce projet nous a permis d'assimiler les techniques d'interface des systèmes à microprocesseurs et de maîtriser les techniques d'enregistrement sur des supports magnétiques.

Enfin, nous espérons que notre modeste travail contribuera à la conception et à la réalisation de contrôleurs de disque souple utilisant des techniques avancées.

ANNEXE

1.- Généralités sur le microprocesseur MC 6800.

1/1/- Présentation :

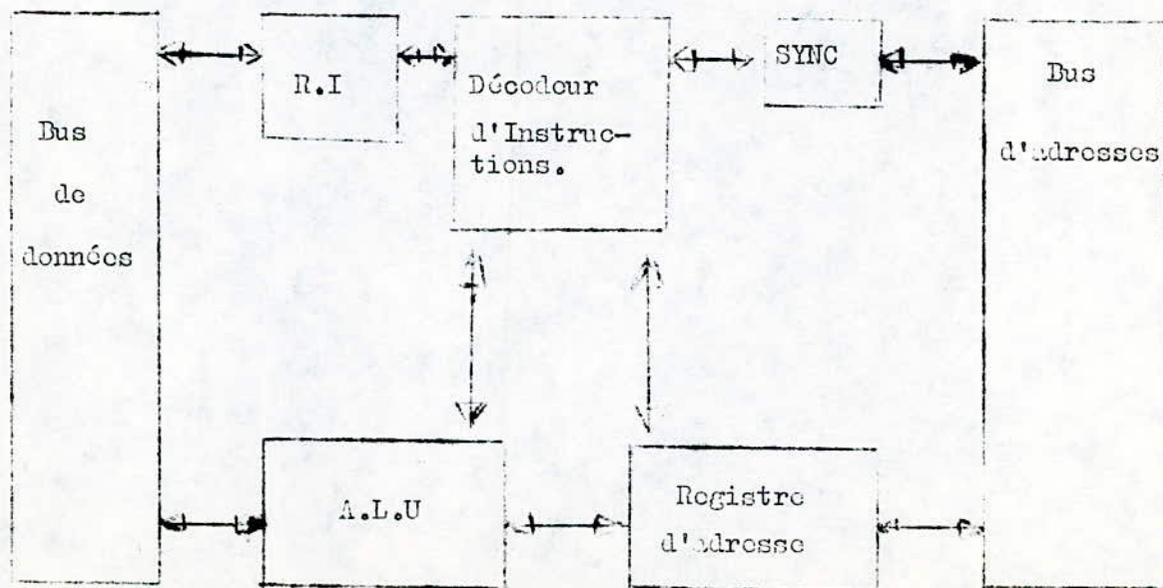
- boîtier de 40 broches
- technologie NMOS à porte au silicium
- piloté par une horloge à deux phases de 1 MHz
- dispose de 72 instructions.

Il présente une caractéristique originale qui est de ~~ne~~ nécessiter qu'une seule alimentation + 5 V.

1.2.- Utilisation de chaque broche :

- Vss = 0V ; Vcc = + 5 + 5 % V
- Halt : signal qui demande l'arrêt du CPU
- Q1 et Q2 : les deux phases de l'horloge
- IRQ : signal qui demande une interruption "masquable"
- VIA : signal qui indique que le circuit périphérique possède une adresse validée sur le bus d'adresses?
- NMI : interruption non masquable
- BA : signal qui indique que le bus d'adresses est disponible.
- A0 - A15 : bus d'adresses
- D0 - D7 : bus de données
- R/W : signal de lecture/écriture
- DBE : signal qui indique que l'on peut accéder au bus de données.
- TSC : possibilités d'interruption (défaut de secteur, mise sous tension).
- Reset : signal de remise à zéro.

1.3.- bloc diagramme du MC 6800



R.I. : Registre d'instruction

Sync : synchronisation.

2.- Eléments internes du microprocesseur :

2.1.- Bus d'entrée / sortie :

Les liaisons avec l'extérieur s'effectuent par l'intermédiaire d'un bus de données bidirectionnel de 8 bits et d'un bus d'adresse de 16 bits en parallèle de façon à adresser les 65 K octets possibles de la mémoire.

2.2.- Registres internes :

Le circuit possède 6 registres internes :

- deux accumulateurs A et B : le fait de disposer de deux accumulateurs permet de stocker un résultat partiel dans l'accumulateur B tout en réalisant un calcul dans l'accumulateur A.

- un registre code de condition : Il est composé de 6 Flip-Flop (interruption, carry, zéro, signe, parité, H) où H est dit "Half carry" qui facilite les calculs décimaux.

- un registre positeur de pile de 16 bits : outre le fait de stocker temporairement des données contenues dans les accumulateurs le pointeur de pile est utilisé pour les opérations suivantes :

* appel d'une sous-routine

* interruption.

- Un registre d'index de 16 bits : Il facilite l'accès à des données séquentielles stockées dans la mémoire. L'adresse de la donnée séquentielle que l'on veut atteindre est obtenue en ajoutant à une base contenant B le contenu variable du registre d'index :

$$\text{adresse} = B + x \text{ (variable)}.$$

- un compteur ordinal :

2/3) l'ALU (unité arithmétique et logique) : effectue toutes les opérations arithmétiques et logique.

3.- modes d'adressage :

Le MC 6800 à 72 instructions de base, il en reconnaît et exécute 1973 ; ceci est dû au fait qu'une même instruction peut avoir plusieurs modes d'adressage différents on en distingue :

- Adressage étendu :

Cet adressage requiert trois octets, un pour l'instruction et deux pour l'adresse. Le microprocesseur a besoin de trois cycles de mémoire pour lire l'instruction et l'adresse et ce n'est qu'au quatrième cycle qu'il exécute l'instruction.

- Adressage absolu ou direct :

Le microprocesseur, après avoir lu l'instruction à la position de mémoire (x), trouve à la position (x + 1) l'adresse de la position mémoire vers laquelle il doit envoyer le mot contenu dans l'accumulateur B.

- Adressage relatif :

La partie "adresse" donnée dans l'instruction est appelée déplacement et correspond à la différence entre la position à atteindre et l'adresse courante (valeur du compteur d'instruction).

- Adressage indexé :

Utilisé comme adresse le contenu de l'un des registres de l'unité arithmétique appelé registre d'index.

- Adressage indirect :

Le mot qui est recherché en mémoire n'est pas considéré comme information, mais comme l'adresse de l'information désirée. Une nouvelle recherche en mémoire doit suivre la première.

- Adressage immédiat :

L'information n'a pas à être recherchée en mémoire, elle fait partie de l'instruction elle-même.

- Adressage implicite :

Ce mode d'adressage n'est pas une réelle méthode d'adressage. Il modifie le contenu d'un registre (8 ou 16 bits). Il se fait avec une instruction de 1 octet.

B.- L'interface PIA MC 6820 :

a.- Description générale et caractéristiques (fig. B.1)

Le MC 6820 fournit un moyen universel d'interface du H.P.U. (MC 6800) avec la périphérie. L'interfaçage est réalisé à travers deux bus de données bidirectionnels (8 bits) et quatre lignes de contrôle, sans aucune logique externe. Le fonctionnement du PIA est programmé par le MPU durant l'initialisation du système ; ainsi chaque ligne de données peut être utilisée soit en entrée soit en sortie et chaque ligne de contrôle peut être programmée pour un des modes de fonctionnement possible.

Les principales caractéristiques du PIA sont :

- interruptions contrôlées et possibilité de masquage d'interruptions.
- possibilités de contrôle de circuits CMOS sur la partie A des lignes vers la périphérie.
- Possibilités de commander deux charges T.T.L. en sortie sur les parties A et B.

b.- Registres internes.

Le PIA possède 6 registres internes accessibles au MPU en écriture et en lecture et répartis en deux groupes de trois registres pour chacun des ports :

- CRA et CRB (contrôle register A/B) :

Ils contiennent les paramètres de fonctionnement.

- DDRA et DDRB (Data Direction Register A/B).

Ils contiennent le mot fixant le sens de transfert (entrée ou sortie) pour chacune des lignes de données.

Etat "1" : ligne en sortie

Etat "0" : ligne en entrée.

- ORA et ORB (out put register, A/B)

Ces registres mémorisent les données en sortie lors d'une écriture. Toutefois, on peut écrire les données présentes en entrée mais elle doivent être mémorisées à l'extérieur.

Enfin on note que deux circuits de commande d'interruption A et B permettent de traiter CA1, CA2, CB1 et CB2. et de générer IRQA et IRQB.

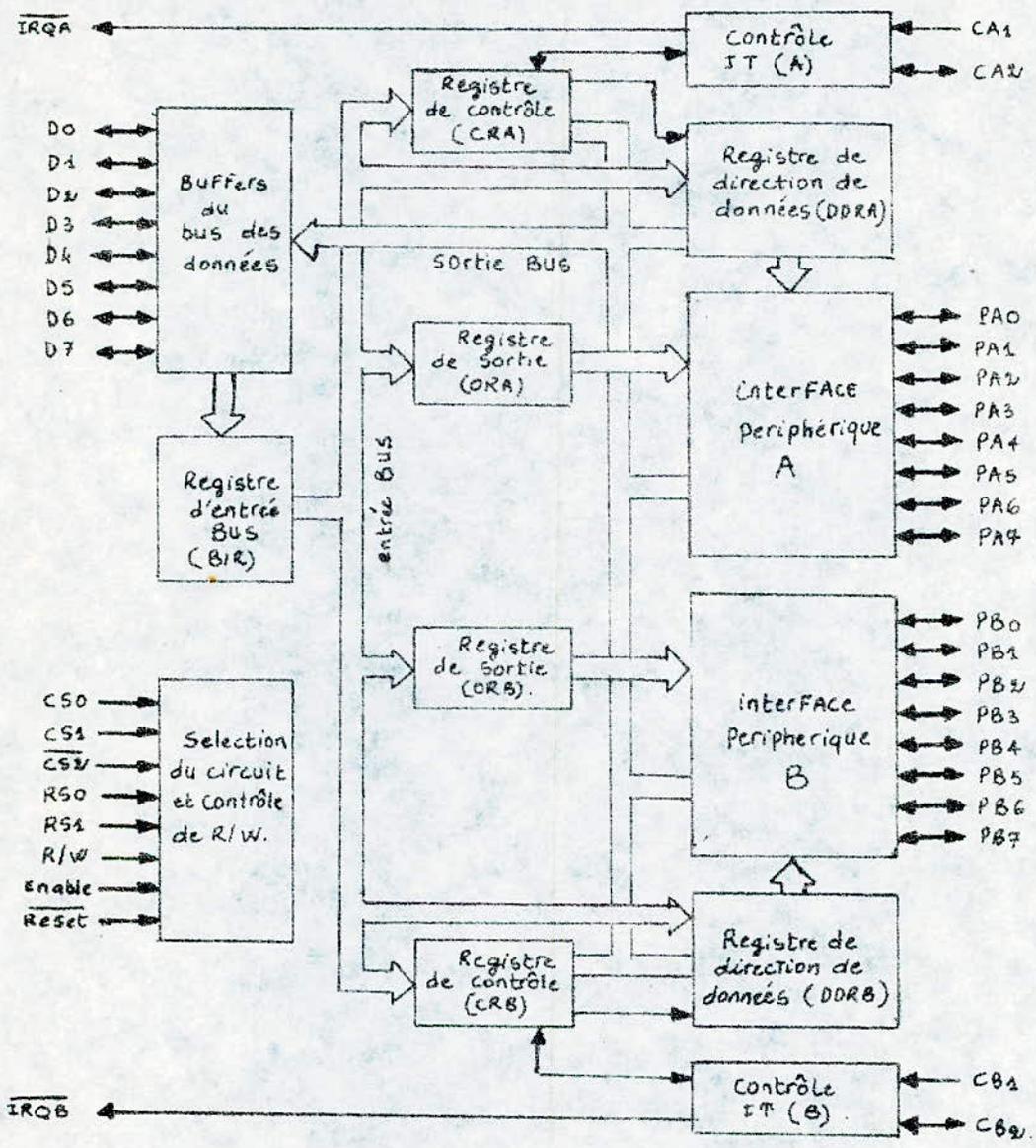


fig B. 1: BLOC DIAGRAMME DU PIA

c.- signaux de liaison avec le MPU.

Un contrôle complet du PIA, par le MPU, peut être aisément réalisé en associant la sortie V1A du IC 6800 avec les lignes suivantes :

- bus de données bidirectionnel (D0-D7).
- ligne d'horloge E (Enable).
- entrée de lecture/écriture (R/W : Read/Write)

R/W = 0 : écriture (MPU vers PIA)

R/W = 1 : lecture (PIA vers MPU)

- entrée de sélection de boîtier CS0, CS1 et CS2 :

ces trois signaux d'entrée sont utilisés pour adresser le PIA, pour cela il faut que :

CS0 = 1

CS1 = 1

CS 2= 0

- entrée de sélection des registres RS0 et RS1 : ces deux lignes permettent de sélectionner les divers registres internes du PIA. Cela se fait en liaison avec les registres de contrôle interne comme le montre le tableau suivant :

RS1	RS0	Bit 2 de CRA (2)	du registre de contrôle	registre sélectionné
0	0	1	x	ORA
0	0	0	x	DDRA
0	1	x	x	CRA
1	0	x	1	ORB
1	0	x	0	DDRB
1	1	x	x	CRB

- lignes de demande d'interruption IRQA et IRQB :

Actives à l'état bas, elles servent à interrompre le MPU. Chacune des lignes IRQA et IRQB est associée à 2 bits indicateurs d'interruption (bits 6 et 7) des registres A et B pour respectivement IRQA et IRQB, chaque indicateur est associé à une des quatre lignes de contrôle d'interruption de la périphérie (CA1, CA2, CB1, CB2).

Enfin le PIA possède quatre d'autorisation d'interruption qui peuvent être utilisés pour masquer les demandes d'interruption.

d.- lignes d'interface avec la périphérie.

L'interface est réalisé avec deux bus de données bidirectionnels de huit bits et quatre lignes de contrôle/interruption.

- lignes de données/Port A (PA0-PA7)
Port B (PB0-PB7)

Elles peuvent être programmée en entrée ou en sortie en mettant respectivement à "0" ou à "1" les bits correspondants à PA0-PA7 (PB0-PB7) au niveau du registre DDRA et DDRB. On aura ainsi accès aux registres ORA et ORB soit en lecture soit en écriture.

- lignes d'interruption (CA1, CB1) :

Ces lignes positionnent les indicateurs d'interruption des registres CRA et CRB.

- lignes de contrôle périphérique CA2, CB2 :

Elles sont programmables pour être utilisées en entrée d'interruption ou en sortie de contrôle.

LISTING

```

00001          NAM    ROM
00002          TTL    MDOS ROM
00003          *
00004          * MOTOROLA DISK OPERATING SYSTEM
00005          * ROM RESIDENT DISK DRIVER
00006          * COPYRIGHT 1977 BY MOTOROLA INC
00007          *
00008          * FOR CALCOMP DRIVE
00009          *

```

```

00011          * FEBRUARY 3, 1977 VERSION 1.1

```

```

00013          E800  A RMSTRT EQU  *E800  START OFF ROM

```

```

00015          * LOW MEMORY STORAGE

```

```

00016A 0000          ORG    0
00017A 0000 0001  A  CURDRV RMB    1  CURRENT DRIVE : 0 OR 1
00018A 0001 0002  A  STRSCT RMB    2  STARTING PHYSICAL SCTR NUMBER
00019A 0003 0002  A  NUMSCT RMB    2  NUMBER OF SECTORS
00020A 0005 0001  A  LSCTLN RMB    1  * BYTES TO LOAD FROM LAST SCTR
00021A 0006 0002  A  CURADR RMB    2  CURRENT LOAD ADDRESS
00022A 0008 0001  A  FDSTAT RMB    1  DISK STATUS UPON RETURN : '0=OK
00023A 0009 0001  A  LOVCNT RMB    1  * OF BYTES TO SKIP IN LAST SCTR
00024A 000A 0001  A  CURSCT RMB    1  CURRENT SECTOR
00025A 000B 0002  A  SCTCNT RMB    2  SECTOR COUNT
00026A 000D 0001  A  SAVCND RMB    1  USER CONDITION CODE STORAGE
00027A 000E 0001  A  FDCMND RMB    1  DISK COMMAND WORD
00028A 000F 0002  A  NMISAV RMB    2  NMI VECTOR STORAGE
00029A 0011 0001  A  CUTDRO RMB    1  CURRENT TRACK OF DRIVE 0
00030A 0012 0001  A  CUTDR1 RMB    1  CURRENT TRACK OF DRIVE 1
00031A 0013 0001  A  CURTRK RMB    1  CURRENT TRACK
00032A 0014 0001  A  TEMP1  RMB    1  TEMPORARY STORAGE LOC 1
00033A 0015 0001  A  TEMP2  RMB    1  TEMPORARY STORAGE LOC 2
00034A 0016 0002  A  USPSAV RMB    2  USER SP STORAGE ON ENTRY
00035A 0018 0001  A  RETRY  RMB    1  READ ERROR RETRY COUNT

```

```

00037          * EQU'S

```

```

00039          * GENERAL

```

```

00040          FF8A  A  STACK  EQU  *FF84  OSLOAD STACK
00041          F56A  A  EXBUG  EQU  *F56A  EXBUG ENTRY
00042          0020  A  BTSTRT EQU  *20   BOOTLOAD START
00043          F018  A  OUTCH  EQU  *F018  PRINT CHAR-PAD NULLS
00044          FFFC  A  NMIVEC EQU  *FFFC  NMI VECTOR ADDRESS
00045          008C  A  SKIP2  EQU  *8C   CPX IMMEDIATE OP CODE
00046          0085  A  SKIP1  EQU  *85   BITA IMMEDIATE OP CODE

```

```

00048          * DISK HARDWARE DEFINITIONS-PIA
00049      EC00 A PIADRO EQU      *EC00    PIA DATA REGISTER A
00050      EC01 A PIADRB EQU      PIADRA+1 PIA DATA REGISTER B
00051      EC02 A PIACRA EQU      PIADRB+1 PIA CONTROL REGISTER A
00052      EC03 A PIACRB EQU      PIACRA+1 PIA CONTROL REGISTER B

00054          * DISK HARDWARE DEFINITIONS-SSDA
00055      EC04 A SSDACR EQU      PIACRB+1 SSDA CONTROL REGISTER
00056      EC04 A SSDASR EQU      SSDACR   SSDA CONTROL REGISTER
00057      EC05 A SSDADR EQU      SSDASR+1 SSDA DATA FIFO'S,CONTROL
00058          *
                                REGISTERS 2&3, AND SYNC CODE

00060          * PIA DATA REGISTER A BIT DEFINITIONS
00061      0001 A PDRAS0 EQU      %00000001 SELECT DRIVE 0(OUT-0 ACT)
00062      0002 A PDRAS1 EQU      %00000010 SELECT DRIVE 1(OUT-0 ACT)
00063      0004 A PDRA43 EQU      %00000100 ABOVE TRACK 43(OUT-0 ACT)
00064      0008 A PDRADI EQU      %00001000 DIRECTION(OUT-1=TOWARD CENTER)
00065      0010 A PDR AHL EQU      %00010000 HEAD LOAD(OUT-0=LOAD)
00066      0020 A NTUSE EQU       %00100000 NOT USED
00067      0040 A PDRADR EQU      %01000000 DRIVE READY(IN-0 ACT)
00068      0080 A PDRATO EQU      %10000000 AT TRACK 0(IN-1 ACT)

00070          * PIA DATA REGISTER B BIT DEFINITIONS
00071      0001 A PDRBED EQU      %00000001 ENABLE DATA(OUT-1 PULSE)
00072      0002 A PDRBEW EQU      %00000010 ENABLE WRITE(OUT-0 ACT)
00073      0004 A PDRBER EQU      %00000100 ENABLE READ(OUT-1 ACT)
00074      0008 A PDRBSC EQU      %00001000 SHIFT CRC(OUT-1 ACT)
00075      0010 A PDRBWE EQU      %00010000 WRITE ENABLED(IN-1 ACT)
00076      0020 A PDRBNU EQU      %00100000 NOT USED
00077      0040 A PDRBRS EQU      %10000000 IN SYNC(IN-1 ACT)
00078      0080 A PDRBC0 EQU      %10000000 CRC=0(IN-1 ACT)

00080          * COMMAND WORD BIT DEFINITIONS
00081      0080 A RDWRBT EQU      %10000000 READ/WRITE FLAG, 1=WRITE
00082          *
                                0=READ, IN COMBINED READ/WRITE
00083          *
                                OPERATIONS, WRITE IS PERFORMED
00084          *
                                FIRST
00085      0040 A VRCRC EQU       %01000000 READ FOR CRC
00086      0020 A NTUSED EQU      %00100000 NOT USED
00087      0010 A SEEKTR EQU      %00010000 SEEK TO TRACK ONLY
00088      0008 A RSTR EQU        %00001000 RESTORE HEAD TO TRACK 0
00089      0004 A NTUSE2 EQU      %00000100 NOT USED
00090      0002 A WRTSTB EQU      %00000010 WRITE TEST, DATA IS (CURADR),+1
00091      0001 A WRDDMB EQU      %00000001 WRITE DELETED DATA ADDRESS MARK

00093          * DISK ERRORS
00094      0031 A EDTCRC EQU       '1          DATA CRC ERROR
00095      0032 A EWRTPR EQU      '2          WRITE PROTECTED
00096      0033 A ENTRDY EQU      '3          DISK NOT READY
00097      0034 A EDDMRK EQU      '4          READ DELETED DATA MARK
00098      0035 A ETIMOU EQU      '5          TIMEOUT ERROR
00099      0036 A EDSKAD EQU      '6          INVALID DISK ADDRESS
00100      0037 A ESEEKE EQU      '7          SEEK ERROR
00101      0038 A EDADMK EQU      '8          DATA MARK ERROR
00102      0039 A EAMCRC EQU      '9          ADDRESS MARK CRC ERROR
    
```

```

00104      *
00105      * MDOS BOOTLOAD ENTRY
00106      *
00107A E800      ORG      RMSTRT
00108      E800 .A      OSLOAD EQU      *
00109A E800 BE FF8A      LDS      *STACK      INITIALIZE STACK
00110A E803 8D ID 3822 BSR      BSR      FDINIT      INITIALIZE PIA, SSDA
00111A E805 7F 0000 A      CLR      CLR      CURDRV      SPECIFY DRIVE 0
00112A E808 8D 6B E875      BSR      BSR      RESTOR      RESTORE HEAD TO TRACK 0
00113A E80A 8D 47 E853      BSR      BSR      CHKERR      CHECK FOR ERROR
00114A E80C CE 0017 A      LDX      LDX      *23      SPECIFY START SECTOR
00115A E80F DF 01 A      STX      STX      STRSCT
00116A E811 CE 0002 A      LDX      LDX      *2      SPECIFY NUMBER OF SECTORS
00117A E814 DF 03 A      STX      STX      NUMSCT
00118A E816 CE 0020 A      LDX      LDX      *BTSTRT      SPECIFY LOAD ADDRESS
00119A E819 DF 06 A      STX      STX      CURADR
00120A E81B 8D 4C 3869      BSR      BSR      READSC      LOAD BOOT
00121A E81D 8D 34 E853      BSR      BSR      CHKERR      CHECK FOR ERROR
00122A E81F 7E 0020 A      JMP      JMP      BTSTRT      START BOOT

00124      *
00125      * MDOS PIA, SSDA INITIALIZATION ROUTINE
00126      *
00127      E822 A      FDINIT EQU      *
00128      * RESET PIA
00129A E822 CE 0000A      LDX      LDX      *0
00130A E825 FF EC02A      STX      STX      PIACRA
00131A E828 FF EC00A      STX      STX      PIADRA
00132      * INITIALIZE SSDA - CLEAR SYNC, RESET TX AND RX,
00133      * DISABLE TIE AND RIE, TRANSFER SYNC CHARACTERS,
00134      * INHIBIT SM, SELECT 2 BYTE TRANSFER, 8 BIT WORDS
00135      * TRANSMIT SYNC ON UNDERFLOW, DISABLE EIE
00136      *
00137      * NOTE : SSDA CONTROL AND STATUS BITS ARE REVERSED
00138      * (LEFT TO RIGHT -) RIGHT TO LEFT) FROM SSDA DATA SHEET
00139      *
00140A E82B CE DCDA A      LDX      LDX      **DODA
00141A E82E FF EC04 A      STX      STX      SSDACR
00142      * SELECT PIA DATA REGISTERS
00143A E831 CE 0404 A      LDX      LDX      **0404
00144A E834 FF EC02 A      STX      STX      PIACRA
00145      * INITIALIZE PIA DATA OUTPUT LINES -
00146      * LIFT HEAD, HEAD DIRECTION TOWARD CENTER,
00147      * BELOW TACK 43, NO DRIVE SELECTED, AND DISABLE
00148      * SHIFT CRC, READ, WRITE, AND DATA
00149A E837 CE 1B02 A      LDX      LDX      **1B02
00150A E83A FF EC06 A      STX      STX      PIADRA
00151      * SET PIA DATA DIRECTION REGISTERS
00152A E830 CE 0000 A      LDX      LDX      *0      SEECT DIRECTION REGISTERS
00153A 3840 FF EC02 A      STX      STX      PIACRA
00154A E848 CE 1F0F A      LDX      LDX      **1F0F      SET DIRECTION REGISTERS
00155A E846 FF EC06 A      STX      STX      PIADRA

```

```

00157      * INITIALIZE PIA CONTROL LINES -
00158      * CA1(TIMEOUT INPUT)      ACTIVE ON NEGATIVE TRAASITION
00159      *                               INTERRUPT DISABLED (FOR NOW)
00160      * CA2(HEAD STEP OUTPUT)    HIGH
00161      * CB1(INDEX)                ACTIVE ON POS TRANS, INTRPT DISAC
00162      * CB2(TIMER RESET OUTPUT)  HIGH
00163A E049 CE 3C3E A  FDINI3 LDX    **3C3E
00164A E04C FF EC02 A      STX      PIACRA
00165      * CLEAR PIA INTERRUPT FLAGS
00166A E04F FE EC00 A      LDX      PIADRA
00167
00168A E052 39      * DONE
RTRN      RTS

```

```

00170
00171      *
00172      * ROUTINE TO CHECK FOR A POSTED ERROR
00173      *
00174A E053 24 FD E052  CHKERR EQU    *
00175A E055 0D 03 E05A      BCC      RTRN    NO ERROR
00176A E057 7E F564 A      BSR      PRNTER  PRINT ERROR NUMBER
                                JMP      EXBUG    GO TO EXBUG-RESETS SP

```

```

00178
00179      *
00180      * ROUTINE TO PRINT DISK ERROR NUMBER
00181      *
00182A E05A 06 45      A PRNTER EQU    *
00183A E05C 0D 08 E066  A LDAA        *'    PRINT E
                                BSR      XOUTCH
00184A E05E 96 00      A LDAA        FDSTAT  PRINT ERROR NUMBER
                                BSR      XOUTCH
00185A E060 0D 04 E066  A LDAA        **20   PRINT TWO SPACES
                                BSR      XOUTCH
00186A E062 06 20      A LDAA        XOUTCH
00187A E064 0D 00 E066  A LDAA        XOUTCH
00188A E066 7E F018      A XOUTCH JMP   XOUTCH

```

```

00190          *
00191          * MDOS DRIVER ENTRY POINTS
00192          *
00193

00194          * READ FULL LAST SECTOR ENTRY
00195          E869 A READSC EQU *
00196A E869 C6 80 A LDAB *128
00197A E86B D7 05 A STAB LSCTLN

00199          * READ PARTIAL LAST SECTOR ENTRY
00200          E86D A READPS EQU *
00201A E86D 5F CLR B
00202A E86E 8C A FCB SKIP2 SKIP NEXT ENTRY

00204          * READ FOR CRC VERIFY CRC VERIFY ENTRY
00205          E86F A RDCRC EQU *
00206A E86F C6 40 A LDAB %01000000
00207A E871 8C A FCB SKIP2

00209          * WRITE TEST THEN READ CRC
00210          B872 A RWTEST EQU *
00211A E872 C6 C2 A LDAB %11000010
00212A E874 8C A FCB SKIP2

00214          *RESTORE HEAD TO TRACK 0 ENTRY
00215          E875 A RESTOR EQU *
00216A E875 C6 00 A LDAB %00001000
00217A E877 8C A FCB SKIP2

00219          * SEEK TO TRACK ENTRY
00220          E878 A SEEK EQU *
00221A E878 C6 10 A LDAB %00010000
00222A E87A 8C A FCB SKIP2

00224          * WRITE TEST ENTRY
00225          E87B A WRTEST EQU *
00226A E87B C6 82 A LDAB %10000010
00227A E87D 8C A FCB SKIP2

00229          * WRITE DELETED DATA ADDRESS MARK ENTRY
00230          E87E A WRDDAM EQU *
00231A E87E C6 81 A LDAB %10000001
00232A E880 8C A FCB SKIP2

00234          * WRITE-VERIFY CRC ENTRY
00235          E881 A WRVERF EQU *
00236A E881 C6 C0 A LDAB %11000000
00237A E883 8C A FCB SKIP2

00239          * WRITE SECTOR ENTRY
00240          E884 A WRITSC EQU *
00241A E884 C6 00 A LDAB %10000000

```

```

00243      *
00244      * COMMON MDOS INITIALIZATION FOR ALL COMMANDS
00245      *
00246      E886 A FDCOMM EQU      *
00247A E886 07      TPA          SAVE USER'S CONDITION CODE
00248A E887 0F      SEI          AND INHIBIT IRQ
00249A E888 97 0D A STAA        SAVCND
00250A E88A D7 0E A STAB        FDCMND  SAVE COMMAND
00251A E88C 86 0D A LDAA        $'0    INITIALIZE STATUS
00252A E88E 97 08 A STAA        FDSTAT
00253A E890 9F 16 A STS         USPSAV  SAVE USER'S SP
00254      *
00255A E092 FE FFFC A LDX        NMIVC   SAVE NMI VECTOR
00256A E095 DF 0F A STX         NMISAV
00257A E097 CE E932 A LDX        $TIMOUT SET UP TIMEOUT (NMI) VECTOR
00258A E09A FF FFFC A STX        NMIVC
00259      *
00260A E09D CE 0011 A LDX        $CUTDRO ASSUME DRIVE 0
00261A E0A0 B6 E000 A LDAA       PIADRA
00262A E0A3 8A 03 A ORAA        $$3
00263A E0A5 4A      DECA
00264A E0A6 7D 0000 A TST        CURDRV  DRIVE 0?
00265A E0A9 27 02 E0AD BEQ        FDCOM3  YES
00266A E0AB 08      INX
00267A E0AC 4A      DECA          CORRECT FOR DRIVE 1
00268A E0AD B7 E000 A FDCOM3 STAA PIADRA  SELECT DRIVE
00269A E0B0 A6 00 A LDAA        0,X    INITIALIZE CURRENT TRACK
00270A E0B2 97 13 A STAA        CURTRK
00271A E0B4 86 40 A LDAA        $PDRADR DRIVE READY?
00272A E0B6 B5 E000 A BITA        PIADRA
00273A E0B9 27 07 E0C2 BEQ        FDCOM5  YES
00274A E0BB C6 33 A LDAB        $ENTRDY  DRIVE NOT READY
00275A E0BD 8C A FCB         SKIP2
00276A E0BE C6 36 A XDSKAD LDAB $EDSKAD DISK ADDRESS ERROR
00277A E0C0 20 77 E939 BRA        POSTER
00279A E0C2 C5 08 A FDCOM5 BITB $RSTR  RESTORE ?
00280A E0C4 27 05 E0CB BEQ        FDCOM4  NO
00281A E0C6 5F      CLRB
00282A E0C7 86 01 A LDAA        $1    YES, STEP IN 1 TRACK
00283A E0C9 20 30 E903 BRA        SEEK01  INIT CURTRK
                                GO RESTORE
00285      *
00286      * DETERMINE TRACK AND SECTOR ADDRESSES
00287      * FIRST CHECK FOR VALID PHYSICAL SECTOR NUMBERS
00288      * STRSCT+NUMSCT(2003
00289      * SKIP THIS PROCESSING IF RESTORE IS REQUESTED
00290      *
00291A E0CB 06 02 A FDCOM4 LDAB  STRSCT+1 STRSCT+NUMSCT(65536?
00292A E0CD 96 01 A LDAA        STRSCT
00293A E0CF 0B 04 A ADDB        NUMSCT+1
00294A E0D1 99 03 A ADCA        NUMSCT
00295A E0D3 25 E9 ERDE BCS        XDSKAD  NO
00296A E0D5 C1 03 A CMPB        $(26*77+1)*256/256 SUMKMAX+1?

```

00297A E8D7 82 07 A SBCA *(26+77+1)/256
 00298A E8D9 24 E3 E8BE BCC XDSKAD NO

00300

* CONVERT PHYSICAL SECTOR NUMBER TO TRACK AND SECTOR ADDRESS.
 **FF INIT TRACK NUMBER

00301A E8DB 86 FF A LDAA **FF
 00302A E8DD 97 0A A STAA CURSCT
 00303A E8DF 96 01 A LDAA STRSCT
 00304A E8E1 D6 02 A LDAB STRSCT+1
 00305A E8E3 7C 000A A GETTR3 INC CURSCT INC TRACK NUMBER/8
 00306A E8E6 C0 D0 A SUBB *26*8 PSNK8 TRACKS?
 00307A E8E8 82 00 A SBCA *0
 00308A E8EA 24 F7 E8E3 BCC GETTR3 NO
 00309A E8EC CB D0 A ADDB *26*8 RESTORE LAST SUBTRACTION
 00310A E8EE 96 0A A LDAA CURSCT GET TRACK NUMBER/8
 00311A E8F0 48 ASLA *8
 00312A E8F1 48 ASLA
 00313A E8F2 48 ASLA
 00314A E8F3 4A DECA
 00315A E8FA 4C GETTR5 INCA CORRECT FOR NEXT LOOP
 00316A E8F5 C0 1A A SUBB *26 INC TRACK NUMBER
 00317A E8F7 24 FB E8F4 BCC GETTR5
 00318A E8F9 CB 1A A ADDB *26 RESTORE LAST SUBTRACTION
 00319A E8FB D7 0A A STAB CURSCT SAVE SECTOR NUMBER

00321

* INITIALIZE SCTCNT

00322A E8FD DE 03 A LDX NUMSCT
 00323A E8FF DF 0B A STX SCTCNT

00325

* SEEK TO TRACK LOGIC

00326 * ENTERED WITH TARTRK IN A
 00327 * A WILL HOLD TRACK DIFFERENCE
 00328 * B WILL HOLD PIA DATA REG A CONTENTS

00329

E901 A SEEKTK EQU *

00330A E901 D6 13 A LDAB CURTRK GET CURRENT TRACK
 00331A E903 97 13 A SEEK01 STAA CURTRK UPDATE CURRENT TRACK
 00332A E905 10 SBA A IS TRACK DIFFERENCE
 00333A E906 F6 EC00 A LDAB PIADRA SET HEAD DIRECTION
 00334A E909 CA 08 A ORAB *PDRADI ASSUME TOWARD CENTER
 00335A E90B 24 03 E910 BCC SEEK03 CORRECT ASSUMPTION
 00336A E90D C4 F7 A ANDB **FF-PDRADI MOVE HEAD TOWARD EDGE
 00337A E90F 40 NEGA CORRECT DIFFERENCE

00338

*

00339

* LOOP TO MOVE HEAD A TRACK AT A TIME

00340

*

00341

* LIFT HEAD IF TRACK DIFFERENCE)4 TRACKS

00342

* OTHERWISE LOAD HEAD

00343A E910 C4 EF A SEEK03 ANDB **FF-PDRAHL ASSUME HEAD LOAD

00344A E912 81 04 A CMPA *4

00345A E914 23 02 E916 BLS SEEK05 DIFFERENCE(=4

00347

* LIFT HEAD HERE

00348A	E916	CA 10	A	ORAB	*PDRAHL	LIFT HEAD
00349A	E918	F7 EC00	A	SEEK05 STAB	PIADRA	
00350A	E91B	4A		DECA		DEC TRACK DIFFERENCE
00351A	E91C	2B 42 E960		BMI	SEEK09	SEEK COMPLETE
00352				* STEP HEAD AND WAIT		
00353A	E91E	8D 1F E93F		BSR	HDSTEP	
00354A	E920	7D EC00	A	TST	PIADRA	AT TRACK 0?
00355A	E923	2A EB E910		BPL	SEEK03	NO-CONTINUE LOOP
00356				* AT TRACK 0		
00357A	E925	AD		TSTA		TRACK DIFFERENCE-0?
00358A	E926	27 38 E960		BEQ	SEEK09	YES, ASSUME SEEK TO TRACK 0
00359A	E928	96 0E	A	LDAA	FDCMND	RESTORE?
00360A	E92A	85 08	A	BITA	*RSTR	
00361A	E92C	27 09 E937		BEQ	XSEEKE	NO-SEEK ERROR
00362A	E92E	8D 22 E952		BSR	STLTIM	YES - LET HFAD SETTLE
00363A	E930	20 5A E98C		BRA	DONE3	
00365				* ENTER HERE IF TIMEOUT OCCURRED		
00366		E932	A	TIMOUT EQU	*	
00367A	E932	9E 16	A	LDS	USPSAV	GET ENTRY SP
00368A	E934	C6 35	A	LDAB	*ETIMOU	GET TIMEOUT ERROR
00370A	E936	8C	A	FCB	SKIP2	
00371A	E937	C6 37	A	XSEEKE LDAB	*ESEEKE	SEEK ERROR
00373				* ENTER HERE TO POST ERROR		
00374		E939	A	POSTER EQU	*	
00375A	E939	D7 08	A	STAB	FDSTAT	POST ERROR
00376A	E93B	8D 4F E98C		BSR	DONE 3	CLEAN UP
00377A	E93D	0D		SEC		SET ERROR INDICATION
00378A	E93E	39		RTS		DONE !
00380				*		
00381				* HEAD STEP ROUTINE-INCLUDES DELAY		
00382				*		
00383		E93FA		HDSTEP EQU	*	
00384A	E93F	37		PSHB		SAVE B
00385A	E940	C6 34	A	LDAB	**34	SET STEP LINE LOW AND
00386A	E942	F7 EC02	A	STAB	PIACRA	CA1 INTERRUPT DISABLED
00387A	E945	C6 3C	A	LDAB	**3C	SET STEP LINE HIGH
00388A	E947	F7 EC02	A	STAB	PIACRA	
00389A	E94A	33		PULB		RESTORE B
00390A	E94B	CE 0350	A	TIM006 LDX	*048	
00391A	E94E	09		TIM0	DEX	
00392A	E94F	26 FD E94E		BNE	TIM0	
00393A	E951	39		RTS		
00395				* HEAD SETTLE DELAY		
00396A	E952	CE 04E2	A	STLTIM LDX	*1250	CALCOMP VALUE
00397A	E955	20 F7 E94E		BRA	TIM0	

```

00300A E957 26 13 A SEEK07 LDAA * CURTRK AT TRACK 0 ?
00400A E959 27 DC E937 BEQ XSEEKE YES, ERROR
00401A E95B 4F CLRA HAVE STEPPED IN
00402A E95C 06 56 A LDAB *86 NOW RESTORE
00403A E95E 20 A3 E903 BRA SEEK01

00405A E960 CE 08CA A SEEK09 LDX *2250
00406A E963 8D E9 E94E BSR TIMO
00407A E965 96 0E A LDAA FDCMND RESTORE COMPLETED?
00408A E967 85 08 A BITA *RSTR
00409A E969 26 EC E957 BNE SEEK07 YVES-MAYBB ERROR
00410A E96B 85 10 A BITA *SEEKTR SEEK TRACK ONLY?
00411A E96D 26 1D E98C BNE DONE 3 YES

```

```

00413 *
00414 * READ/WRITE LOGIC STARTS HERE
00415 *
00416 E96FA RDWR EQU *
00417 * SET UP TO STEP HEAD TOWARD CENTER OF DISK
00418A E96F CA 08 A ORAB *PDRADI
00419A E971 F7 EC00 A STAB PIADRA
00420 * INITIALIZE ADDRESS MARK STORAGE
00421A E974 C6 6F A LDAB **6F ASSUME NOT DD AM
00422A E976 46 RORA WRITE DD AM?
00423A E977 24 02 E97B BCC RDWR02 NO
00424A E979 C6 6A A LDAB **6A YES
00425A E97B D7 14 A RDWR02 STAB TEMP1
00426A E97D 20 52 E9D1 BRA RDWR03

```

```

00428 E97F A DONE EQU *
00429A E97F 96 0E A LDAA FDCMND DONE?
00430A E981 2A 09 E98C BPL DONE3 YES
00431A E983 84 40 A ANDA *%01000000 READ CRC ?
00432A E985 97 0E A STAA FDCMND
00433A E987 27 03 E98C BEQ DONE3 NO
00434A E989 7E E8CB A JMP FDCM4 YES
00435A E98C 06 03 A DONE3 LDAA **03 DISABLE CNTRLR
00436A E98E B7 EC01 A STAA PIADRB
00437A E991 BD E049 A JSR FDINI3 CLEAR TIMER INTRPT
00438 *
00439A E994 DE 0F A LDX NMISAV RESTORE NMI VECTOR
00440A E996 FF FF0C A STX NMIVEC
00441 *
00442A E999 96 13 A LDAA CURTRK UPDATE TRACK INFO
00443A E99B D6 00 A LDAB CURDRV
00444A E99D 27 03 E9A2 BEQ DONE5 DRIVEN
00445A E99F 97 12 A STAA CUTDR1 DRIVE 1
00446A E9A1 0C A FCB SKIP2

```

PAGE 010 ROM MDOS#ROM

00447A E9A2 97 11 A DONE5 CUTDRO
 00448A E9A4 96 0D A LDAA SAVCND RESTORE USER'S CC
 00449A 39A6 06 TAP
 00450A 39A7 0C CLC ASSUME NO ERRORS
 00451A E9A8 39 RTS

00453 *
 00454 * SUBROUTINE TO INIT ELECTRONICS FOR READ
 00455 *

00456 E9A9 A SETRD EQU *
 00457A E9A9 CE D0D8 A LDX *\$D0D8 SET CR1 & CR2
 00458A E9AC FF EC04 A STX SSDACR
 00459A E9AF CE EC00 A LDX \$PIADRA SET X FOR ADDR
 00460A E9B2 86 50 A LDAA \$\$50 ENABLE RX
 00461A E9B4 A7 04 A STAA 4,X SSDACR
 00462A E9B6 86 07 A LDAA \$\$07 STROBE ENABLE DATA
 00463A E9B8 A7 01 A STAA 1,X PIADRB HIGH
 00464A E9BA 6A 01 A DEC 1,X PIADRB LOW
 00456A E9BC 08 INX DELAY
 00466A E9BD 09 DEX
 00467A E9BE 86 40 A LDAA \$\$40 ENABLE SYNC
 00468A E9C0 A7 04 A STAA 4,X SSDACR
 00469A E9C2 86 98 A LDAA \$\$98 ENABLE SM
 00470A E9CA A7 05 A STAA 5,X SSDADR
 00471A E9C6 39 RTS

00473 * UPDATE SECTOR NUMBER AND TRACK NUMBER
 00474A E9C7 97 0A A RDWR11 STAA CURSCT GO TO NEXT TRACK
 00475A E9C9 7C 0013 A INC CURTRK
 00476A E9CC 8D E93F A JSR HDSTEP
 00477A E9CF 8D 81 E952 BSR STLTIM
 00478 E9D1 A RDWR03 EQU *
 00479A E9D1 7C 000A A INC CURSCT
 00480A E9DA 96 0A A LDAA CURSCT
 00481A E9D6 DE 0B A LDX SCTCNT
 00482A E9D8 27 A5 E97F BEQ DONE
 00483A E9DA 80 1B A SUBA \$27 READ SCTR 26?
 00484A E9DC 24 E9 E9C7 BCC RDWR11 YES
 00485A E9DE 86 05 A LDAA \$5 INIT RETRY COUNT
 00486A E9E0 97 18 A STAA RETRY
 00487A E9E2 09 DEX DEC SECTOR COUNT
 00488A E9E3 86 40 A RDWR06 LDAA \$128/2 READ COUNT
 00489A E9EC DF 0B A STX SCTCNT UPDATE SCTCNT
 00490A E9E7 26 07 E9F0 BNE RDWR07 NOT LAST SECTOR
 00491 * CALCULATE NUMBER OF BYTES TO READ IN LAST SECTOR
 00492A E9E9 96 05 A LDAA LSCTLN GET \$ OF BYTES TO READ
 00493A E9EB 8B 07 A ADDA \$7 ROUND UP
 00494A E9ED 44 LSRA DIVIDE BY 2
 00495A E9EE 84 FC A ANDA \$\$FC RESET BITS 0 AND 1
 00496A E9F0 97 15 A RDWR07 STAA TEMP2
 00497A E9F2 40 NEGA
 00498A E9F3 D6 0E A LDAB FDCMND READ FOR CRC?

00499A	E9F5	58			ASLB		
00500A	E9F6	2A	01	E9F9	BPL	RDWR12	NO
00501A	E9F8	4F			CLRA		
00502A	E9F9	8B	40	A	RDWR12	ADDA	\$128/2
00503A	E9FB	97	09	A	STAA	LOVCNT	
00504A	E9FD	86	36	A	LDAA	\$\$36	RESET TIMER
00505A	E9FF	B7	EC03	A	STAA	PIACRB	
00506A	EA02	86	3E	A	LDAA	\$\$3E	
00507A	EA04	B7	EC03	A	STAA	PIACRB	
00508A	EA07	F6	EC00	A	LDAB	PIADRA	CLEAR ANY INTERRUPTS
00509A	EA0A	4A			DECA		ENABLE INTERRUPTS
00510A	EA0B	B7	EC02	A	STAA	PIACRA	
00511A	EA0E	96	13	A	LDAA	CURTRK	SET BEYOND TRK 43 OUTPUT
00512A	EA10	CA	04	A	ORAB	\$\$0000100	ASSUME TRACK(43
00513A	EA12	81	2B	A	CMPA	\$43	
00514A	EA14	23	02	EA18	BLS	RDWR04	
00515A	EA16	CA	FB	A	ANDB	\$\$11111011	TRACK)43
00516A	EA18	F7	EC00	A	RDWR04	STAB	PIADRA
00517					*		
00518					*	SET UP FOR SM	
00519					*		
00520A	EA1B	CE	D270	A	LDX	\$\$D270	
00521A	EA1E	FF	EC04	A	STX	SSDACR	
00522A	EA21	CE	D1F5	A	LDX	\$\$D1F5	SET SM CODE
00523A	EA24	FF	EC04	A	STX	SSDACR	
00524					*		
00525A	EA27	8D	8D	E9A9	RDWR15	BSR	SETRD
00526					*		
00527A	EA29	A6	04	A	RDWR17	LDAA	4,X
00528A	EA2B	2A	FC	EA29	BPL	RDWR17	SSDASR WAIT FOR TWO BYTES READY
00529A	EA2D	A6	05	A	LDAA	5,X	SSDADR GET 2ND HALF AM
00530A	EA2F	81	7E	A	CMPA	\$\$7E	CHECK 2ND HALF AM
00531A	EA31	26	F4	EA27	BNE	RDWR15	WRONG
00532A	EA33	A6	04	A	RDWR19	LDAA	4,X
00533A	EA35	2A	FC	EA33	BPL	RDWR19	SSDASR WAIT
00534A	EA37	A6	05	A	LDAA	5,X	SSDADR TRACK
00535A	EA39	E6	05	A	LDAB	5,X	SSDADR ZEROS
00536A	EA3B	91	13	A	CMPA	CURTRK	CORRECT TRACK?
00537A	EA3D	26	E8	EA27	BNE	RDWR15	NO
00538A	EA3F	A6	04	A	RDWR21	LDAA	4,X
00539A	EA41	2A	FC	EA3F	BPL	RDWR21	SSDASR WAIT
00540A	EA43	A6	05	A	LDAA	5,X	SSDADR SECTOR
00541A	EA45	E6	05	A	LDAB	5,X	SSDADR ZEROS
00542A	EA47	91	0A	A	CMPA	CURSCT	CORRECT SECTOR?
00543A	EA49	26	DC	EA27	BNE	RDWR15	NO
00544A	EA4B	A6	04	A	RDWR23	LDAA	4,X
00545A	EA4D	2A	FC	EA4B	BPL	RDWR23	SSDASR WAIT
00546A	EA4F	A6	05	A	LDAA	5,X	SSDADR CRC1
00547A	EA51	A6	05	A	LDAA	5,X	SSDADR CRC2
00548A	EA53	4C			INCA		6 PULSE DELAY
00549A	EA54	0B			INX		
00550A	EA55	A6	00	A	LDAA	0,X	PIADRB CHECK CRC
00551A	EA57	2B	2A	EA83	BMI	XAMCRC	AM CRC ERROR
00553A	EA59	D6	CE	A	RDWR25	LDAB	FDCMND
							GET COMMAND

00554A	EA5B	2B	73	EAD0		BMI	WRIT	WRITE COMMAND
00555A	EA5D	CE	0029	A		LDX	\$41	WAIT TO START READ
00556A	EA60	09			RDWR27	DEX		
00557A	EA61	26	FD	EA60		BNE	RDWR27	
00558A	EA63	C6	04	A		LDAB	\$4	4 TRIES TO FIND DATA AM
00559A	EA65	BD	E9A9	A	READ	JSR	SETRD	
00560A	EA68	DE	06	A		LDX	CURADR	GET CURRENT ADDR
00561A	EA6A	B6	EC04	A	READ03	LDAA	SSDASR	WAIT FOR 2 BYTES
00562A	EA6D	2A	FB	EA6A		BPL	READ03	
00563A	EA6F	B6	EC05	A		LDAA	SSDADR	GET 2ND 1/2 DATA AM
00564A	EA72	81	6F	A		CMPA	\$\$6F	CORRECT?
00565A	EA74	27	21	EA97		BEQ	READ05	YES
00566A	EA76	81	6A	A		CMPA	\$\$6A	DELETED DATA MARK?
00567A	EA78	27	18	EA92		BEQ	READ07	YES-TELL USER
00568A	EA7A	5A				DECB		DEC TRY COUNT
00569A	EA7B	26	E8	EA65		BNE	READ	HAVEN'T TRIED 4 TIMES
00570								
								* ERROR EXITS
00571A	EA7D	C6	38	A		LDAB	\$EDADMK	DATA MARK ERROR
00572A	EA7F		8C	A		FCB	SKIP2	
00573A	EA80	C6	31	A	ERROR	LDAB	\$EDTCRC	DATA CRC ERROR
00574A	EA82		8C	A		FCB	SKIP2	
00575A	EA83	C6	39	A	EXAMCRC	LDAB	\$EAMCRC	ADDRESS MARK CRC ERROR
00576A	EA85	7A	0018	A		DEC	RETRY	RETRY?
00577A	EA88	27	0A	EA94		BEQ	XOSTER	NO
00578A	EA8A	DE	0B	A		LDX	SCTCNT	YES
00579A	EA8C	7E	E9E3	A		JMP	RDWR06	
00580A	EA8F	C6	32	A	WRIT05	LDAB	\$EWRTPR	WRITE PROTECTED
00581A	EA91		0C	A		FCB	SKIP2	
00582A	EA92	C6	34	A	READ07	LDAB	\$EDDMRK	READ DELETED DATA MARK
00583A	EA94	7E	E939	A	XOSTER	JMP	POSTER	
00584								
								* ERROR EXITS
00585A	EA97	D6	0E	A	READ05	LDAB	FDCMND	GET COMMAND
00586A	EA99	58				ASLB		WHICH READ?
00587A	EA9A	2B	16	EA62		BMI	READ15	READ FOR CRC
00589								
								* READ ALL OR PART OF SECTOR TO MEMORY
00590A	EA9C	D6	15	A		LDAB	TEMP2	
00591A	EA9E	B6	EC04	A	READ13	LDAA	SSDASR	WAIT
00592A	EAA1	2A	FB	EA9E		BPL	READ13	
00593A	EAA3	B6	EC05	A		LDAA	SSDADR	READ TWO BYTES
00594A	EAA6	A7	00	A		STAA	0,X	
00595A	EAA8	B6	EC05	A		LDAA	SSDADR	
00596A	EAA8	A7	01	A		STAA	1,X	
00597A	EAA8	00				INX		
00598A	EAAE	00				INX		
00599A	EAAF	5A				DECB		DONE READING?
00600A	EAB0	26	EC	EA9E		BNE	READ13	NO

00602
 00603A EAB2 B6 EC04 A
 00604A EAB5 2A FB EAB2
 00605A EAD7 B6 EC05 A
 00606A EABA B6 EC05 A
 00607A EABD D6 09 A
 00608A EADF 27 05 EAC6
 00609A EAC1 7A 0009 A
 00601A EAC4 20 EC EAD2
 00611A EAC6 B6 EC01 A
 00612A EAC9 2D B5 EA80
 00613A EACB DF 06 A
 00614A EACD 7E E9D1 A

\$WAIT FOR END OF SECTOR TO CHECK CRC
 READ15 LDAA SSDASR WAIT
 BPL READ15
 LDAA SSDADR GET TWO BYTES
 LDAA SSDADR
 LDAB LOVCNT FINISHED?
 BEQ CHKCRC YES-CHECK CRC
 DEC LOVCNT NO
 BRA READ15
 CHKCRC LDAA PIADRB CRC CORRECT?
 BMI ERROR NO
 STX CURADR YES, UPDATE CURADR
 JMP RDWR03 CONTINUE

00616A EADD 86 04 A
 00617A EAD2 7D EC04 A
 00618A EAD5 2A FB EAD2
 00619A EAD7 B1 EC05 A
 00620A EADA B1 EC05 A
 00621A EADD 4A
 00622A EADE 26 F2 EAD2
 00623A EAE0 CE CODA A
 00624A EAE3 FF EC04 A
 00625A EAE6 CE CIAA A
 00626A EAE9 FF EC04 A
 00627A EAEC CE C270 A
 00628A EAEF FF EC04 A
 00629A EAF2 7C EC01 A
 00630A EAF5 86 82 A
 00631A EAF7 B7 EC01 A
 00632A EAF8 B7 EC04 A
 00633A EAFD 86 10 A
 00634A EAFF B5 EC01 A
 00635A EB02 27 8D EASF
 00636A EB04 56
 00637A EB05 24 01 EB08
 00638A EB07 85 A
 00639A EB08 56
 00640A EB09 01
 00641A EB0A 86 00 A
 00642A EB0C C6 06 A
 00643
 00644A EB0E B7 EC01 A
 00645A EB11 86 08 A
 00646A EB13 CE 8210 A
 00647A EB16 FF EC04 A
 00648A EB19 B5 EC04 A
 00649A EB1C 27 FB EB19
 00650A EB1E FF EC04 A
 00651A EB21 5A
 00652A EB22 26 F5 EB19
 00653A EB24 C6 40 A
 00654A EB26 96 14 A
 00655A EB28 CE 83F5 A

WRIT LDAA \$4 DELAY
 WRIT03 TST SSDASR WAIT FOR 2 BYTES
 BPL WRIT03
 CMPA SSDADR
 CMPA SSDADR
 DECA
 BNE WRIT03
 LDX \$\$CODA SET CR2 FOR WRITE
 STX SSDACR
 LDX \$\$CIAA SET SYNC CODE FOR GAP CLOCK
 STX SSDACR AND DATA PATTERN
 LDX \$\$C270 SET CR3 FOR WRITE
 STX SSDACR
 INC PIADRB TOGGLE CONTROLLER FOR 2X
 LDAA \$\$82 ENABLE CONTROLLER FOR WRITE
 STAA PIADRB
 STAA SSDACR SET CR1 FOR WRITE
 LDAA \$\$10 WRITE INHIBITED?
 BITA PIADRB
 BEQ WRIT05 YES
 RORB WRITE DD AM?
 BCC WRIT07 NO
 FCB SKIP1 YES, DON'T WRITE FROM MEMORY
 WRIT07 RORB CARRY SET=WRITE TEST
 NOP DELAY
 LDAA \$\$0 FOR WRITE ON
 LDAB \$\$6 TUF COUNT
 * GET HERE 11 BYTE TIMES AFTER SECTOR ADDRESS
 STAA PIADRB TURN ON WRITE CURRENT
 LDAA \$0 TUF COMPARE BIT
 LDX \$\$8210 CLEAR TUF
 STX SSDACR
 WRIT10 BITA SSDASR WAIT FOR TUF
 BEQ WRIT10
 STX SSDACR
 DECB DONE?
 BNE WRIT10 NO
 LDAB \$128/2 GET BYTE COUNT
 LDAA TEMP1 GET 2ND HALF AM
 LDX \$\$83F5

```

00556          $ WRITE ADDRESS MARK AFTER 6 BYTES OF 00
0057A EB2B FF EC04 A      STX      SSDACR
0058A EB2E DE 06  A      LDX      CURADR  GET DATA ADDRESS
0059A EB30 B7 EC05 A      STAA     SSDADR  WRITE 2ND HALF AM
0060A EB33 7E EB30 A      JMP      WRIT15

00662A EB36 86 40  A WRIT11 LDAA     $$40    GET TRANSMIT READY MASK
00663A EB38 B5 EC04 A WRIT13 BITA     SSDASR  WAIT TO SEND TWO BYTES
00664A EB3B 27 FB EB38      BEQ     WRIT13
00665A EB30 A6 00  A WRIT15 LDAA     0,X     SEND TWO BYTES
00666A EB3F B7 EC05 A      STAA     SSDADR
00667A EB42 A6 A6 01 A      LDAA     1,X
00668A EB44 B7 EC05 A      STAA     SSDADR
00669A EB47 25 02 EB4B      BCS     WRIT17  DON'T CHANGE CURADR
00670A EB49 08              INX
00671A EB4A 08              INX
00672A EB4B 5A              WRIT17 DECB     DONE?
00673A EB4C 26 E8 EB36      BNE     WRIT11  NO
00674A EB4E DF 06  A      STX      CURADR  UPDATE DATA ADDRESS

00676          $ APPEND CRC
00677A EB50 86 40  A WRRCRC LDAA     $$40
00678A EB52 B5 EC04 A WRRCRC3 BITA     SSDASR  WAIT TO SEND TWO BYTES
00679A EB55 27 FB EB52      BEQ     WRRCRC3
00680A EB57 F7 EC05 A      STAB     SSDADR  STORE DUMMY 1 IN TX FIFO
00681A EB5A B5 EC04 A WRRCRC5 BITA     SSDASR  WAIT TO SEND TWO BYTES
00682A EB5D 27 FB EB5A      BEQ     WRRCRC5
00683          $ LAST DATA IN TEX SHIFTER, DUMMY 1 NEXT
00684A EB5F C6 80  A      LDAB     $$8    ENABLE SHIFT CRC
00685A EB61 F7 EC01 A      STAB     PIADRB
00686A EB64 F7 EC05 A      STAB     SSDADR  STORE DUMMY 2
00687A EB67 B5 EC04 A WRRCRC7 BITA     SSDASR  WAIT TO SEND TWO BYTES
00688A EB6A 27 FB EB67      BEQ     WRRCRC7
00689          $ DUMMY 1 IN TX SHIFTER, DUMMY 2 NEXT
00690A EB6C C6 FF  A      LDAB     $$FF   SET NEXT DATA
00691A EB6E F7 EC05 A      STAB     SSDADR
00692A EB71 F7 EC05 A      STAB     SSDADR
00693A EB74 B5 EC04 A WRRCRC9 BITA     SSDASR  WAIT FOR 2 BYTES
00694A EB77 27 FB EB74      BEQ     WRRCRC9
00695          $ DONE WITH CRC
00696A EB79 7F EC01 A      CLR     PIADRB  DISABLE SHIFT CRC
00697A EB7C EC01  A      INC     PIADRB  DISABLE CONTROLLER AND
00698A EB7F 7C EC01 A      INC     PIADRB  TURN OFF WRITE CURRENT
00699A EB82 7E E9 1  A      JMP     RW03    UPDATE DISK ADDRESS

00701          $ SET DRIVE TYPE AND VERSION FLAGS
00702A EBFE              ORG     RMSTRT+$3FE
00703A EBFE          11  A      FCB     $11    VERSION
00704A EBFF          43  A      FCB     'C     DRIVE TYPE

```

00706

END

AL ERRORS 00000

20 ETSTRT 00042*00118 00122
 6 CHKCR0 00608 00611*
 E853 CHKERR 00113 00121 00173*
 0006 JURADR 00021*00119 00560 00658 00674
 0000 CURDRV 00017*00111 00264 00443
 000A CURSCT 00024*00302 00305 00310 00319 00474 00479 00480 00542
 0013 CURTRK 00031*00270 00330 00331 00399 00442 00475 00511 00536
 0011 CUTDR0 00029*00260 00447
 0012 CUTDR1 00030*00445
 B97F DONE 00428*00482
 E98C DONE3 00363 00376 00411 00430 00433 00435*
 E9A2 DONE5 00444 00447*
 0039 EAMCRC 00102*00575
 0038 EDADMK 00101*00571
 0034 EDDMRK 00097*00582
 0036 EDSKAD 00099*00276
 0031 EDTCRC 00094*00573
 0033 ENRDY 00096*00274
 EA80 ERROR 00573*00612
 0037 ESEEKE 00100*00371
 0035 ETIMBU 00098*00368
 0032 EWRTPR 00095*00580
 F564 EXBUG 00041*00176
 000E FDCMND 00027*00250 00359 00407 00429 00432 00490 00553 00585
 E8AD FDCM3 00265 00268*
 E8CB FDCM4 00280 00291*00434
 E8C2 FDCM5 00273 00279*
 E886 FDCMM 00246*
 E849 BDINI3 00163*00437
 E822 FDINIT 00110 00127*
 0008 FDSTAT 00022*00104 00252 00375
 E8E3 GETTR3 00305*00308
 E8F4 GETTR5 00315*00317
 E93F HDSTEP 00353 00303*00476
 0000 LVCNT 00023*00503 00607 00609
 0005 LSCTLN 00020*00197 00492
 000F NMISAV 00028*00256 00439
 FFFC NMIVEC 00044*00255 00258 00440
 0020 NTUSE 00066*
 0004 NTUSE2 00089*
 0020 NTUSED 00086*
 0003 NUMSCT 00019*00117 00293 000294 00322
 E800 OSLOAD 00100*
 F010 BUTCH 00043*00100
 0040 PDBRIS 00077*
 0004 PDRA43 00063*
 0008 PDRADI 00064*00334 00336 00410
 0040 PDRADR 00067*00271
 0010 PDRAHL 00065*00343 00340
 0001 PDRAS0 00061*
 0002 PDRAS1 00062*
 0000 PDRAT0 00068*
 0000 PDRBC0 00078*

0001 PDRBED 00071*
0004 PDRBER 00073*
0002 PDRBEW 00072*
0020 PDRBNU 00076*
0008 PDRBSC 00074*
0010 PDRBWE 00075*
00138 00138 00052 00138 00144 00153 00164 00386 00388 00510
EC02 PIACRA 00051*00052 00505 00507
EC03 PIACRB 00052*00055 00131 00150
EC00 PIADRA 00049*00050 00131 00150 00155 00166 00268 00272 0033 00349 00354
BC01 PIADRB 00050*00051 00436 00611 00629 00631 00634 00644 00685 00696 00697 006
E939 POSTER 00277 00374*00583
E85A PRINTER 00175 00181*
E86F RDCRC 00205*
E96F RDWR 00416
E97B RDWR2 00423 00425*
00478*00614 00699

SUITE PAGE 016 ROM MDOS ROM

0005 SKIP1 00046*00630
000C SKIP2 00045*00202 00207 00212 00217 00222 00227 00232 00237 00275 00370
00446 00572 00574 00581
EC04 SSDACR 00055*00056 00141 00450 00521 00523 00624 00626 00628 00632 00647
00650 00657
EC05 SSDADR 00057*00563 00593 00595 00605 00606 00619 00620 00659 00666 00668
00680 00686 00691 00692

EC04 SSDASR 00056*00057 00561 00591 00603 00617 00648 00663 00678 00681 00687 006
FF8A STACK 00040*00109
E952 STLTIM 00362 00396*00477
0001 STRSCT 00018*00115 00291 00292 00303 00304
0014 TEMP1 00032*00425 00654
0015 TEMP2 00033*00496 00590
E94B TIM006 00390*
E94E TIM0 00391*00392 00397 00406
E932 TIMOUT 00257 00366*
0016 USPSAV 00034*00253 00367
0040 WRCRC 00085*
E850 WRCRC 00677*
E852 WRCRC3 00678*00679
E85A WRCRC5 00681*00682
E867 WRCRC7 00687*00688
E874 WRCRC9 00693*00694
E87E WRDAM 00238*
E881 WRDAMB 00091*
EAD0 WRIT 00554 00616*
EAD2 WRIT03 00617*00618 00622
EA6F WRIT05 00580*00635
E800 WRIT07 00637 00639*
EB12 WRIT10 00648*00649 00652
EB36 WRIT11 00662*00673
EB30 WRIT13 00663*00664
EB3D WRIT15 00660 00665*
EB40 WRIT17 00669 00672*
E804 WRITSC 00240*
E87B WRTEST 00225*
0002 WRTSTB 00090*
E801 WRVERF 00235*
EAD3 XAMCRC 00551 00575*
EBBE XDSKAD 00276*00295 00290
EA94 XOSTER 00577 00503*
E866 XOUTCH 00183 00185 00187 00188*
E937 XSEEKE 00361 00371*00400

PAGE 002 RONDIA DISCK DIAGNOSTIC

00053					* PROGRAM		
00054A	EB90				ORG	EB90	
00055A	EB90	CE 0014	A	CLRTOP	LDX	\$2*10	\$ BYTES TO CLEAR
00056A	EB93	6F 5F	A	CLRLP	CLR	\$60-1,X	
00057A	EB95	09			DEX		
00058A	EB96	26 FB EB93			BNE	CLRLP	
00059A	EB98	BD E822	A	TOP	JSR	FDINIT	INIT DISK INTERFACE
00060A	EB9B	BD E875	A		JSR	RESTOR	POSITION DISK HEAD
00061A	EB9E	DE 20	A	LOOP	LDX	LDADDR	SET UP BUFFER ADDR
00062A	EBA0	DF 06	A		STX	CURADR	
00063A	EBA2	DE 22	A		LDX	EXADDR	EXECUTE ROUTINE
00064A	EBA4	AD 00	A		JSR	0,X	
00065A	EBA6	96 24	A		LDA	ONECON	DO IT ONCE ?
00066A	EBA8	26 0F EBB9			BNE	ONCE	YES
00067A	EBAA	97 07	A		STAA	FDSTAT-1	UPDATE COUNT
00068A	EBAC	78 0008	A		ASL	FDSTAT	
00069A	EBAF	DE 07	A		LDX	FDSTAT-1	
00070A	EBB1	6C 01	A		INC	1,X	
00071A	EBBE	26 E9 EB9E			BNE	LOOP	
00072A	EBB5	6C 00	A		INC	0,X	
00073A	EBB7	26 E5 EB9E			BNE	LOOP	
00074A	EBB9	7E E855 A ONCE			JMP	CHKERR+2	

00076
TOTAL ERRORS 00000

END

E853 CHKERR 00045*00074
 EB93 CLRLP 00056*00058
 EB90 CLRTOP 00055*
 0006 CURADR 00043*00062
 0022 EXADDR 00050*00063
 E822 FDINIT 00041*00059
 0008 FDSTAT 00044*00067 00068 00069
 0020 LDADDR 00049*00061
 EB9E LOOP 00061*00071 00073
 EBB9 ONCE 00066 00074*
 0024 ONECON 00051*00065
 E875 RESTOR 00042*00060
 EB98 TOP 00059*

00001 NAM LPDRV
 00002 TTL LINE PRINTER DIRVER

00004 *
 00005 * LINE PRINTER DIRVER FOR CENTRONICS TYPE
 00006 * INTERFACE THROUGH A PIA WITH OUTPUT
 00007 * CHARACTER ON A SIDE, INPUT STATUS ON B SIDE
 00008 *
 00009 * VERSION 1.0
 00010 * COPYRIGHT 1977 BY MOTOROLA INC
 00011 *

00013 * EQUATES
 00014 0008 A FDSTAT EQU 8 CHAR TO BE PRINTED
 00015 E85A A PRNTER EQU \$E85A PRINT ERROR MESSAGE
 00016 F015 A INCHNP EQU \$F015 INPUT CHAR

00018 * PIA ADDRESSES
 00019 EC10 A DATA EQU \$EC10
 00020 EC11 A CNTRL1 EQU \$EC11
 00021 EC12 A STAT EQU \$EC12
 00022 EC13 A CNTRL2 EQU \$EC13

00024A EBC0 ORG \$EBC0
 00025 * SUBROUTINE TO INITIALIZE PIA
 00026 EBC0 A LPINIT EQU *
 00027A EBC0 CE FF2E A LDX \$FF2E
 00028A EBC3 FF EC10 A STX DATA
 00029A EBC6 86 BC A LDAA \$3C
 00030A EBC8 B7 EC13 A STAA CNTRL2
 00031A EBCB 39 RTS

00033 * SUBROUTINE TO PRINT CHARACTER FROM A ACC
 00034 * AND CHECK FOR PRINTER ERROR
 00035 * IF ERROR CARRY IS SET ON RETURN
 00036 EBC0 A LIST EQU *
 00037A EBC0 B7 EC10 A STAA DATA
 00038A EBCF B6 EC10 A LDAA DATA
 00039A EBD2 37 LIST3 PSHB
 00040A EBD3 F6 EC12 A LDAB STAT
 00041A EBD6 C4 03 A ANDB \$3
 00042A EBD8 5A DECB
 00043A EBD9 33 PULB
 00044A EBDA 26 06 EBE2 BNE ERROR NO PAPER OR NOT SELECTED
 00045A EBDC 7D EC11 A TST CNTRL1
 00046A EBDT 2A F1 EBD2 CNTRL1 BPL LIST3
 00047A EBE1 39 RTS
 00049A EBE2 0D ERROR SEC
 00050A EBE3 39 RTS

PAGE 000 L222V LINE PRINTER DRIVER

```

00052          * SUBROUTINES TO PRINT STRING AND STRINGS, CR, LF
00053          EBE4 A LDATA EQU *
00054A EBE 4 86 0D A LDAA $$D
00055A EBE6 80 E4 EBCC LDATA7 BSR LIST
00056A EBE8 25 FC EBE6 BCS LDATA7
00057A EBEA 86 0A A LDAA $$A
00058A EBEC 09 DEX
00059A EBED 8D 0D EBCC LDATA3 BSR LIST
00060A EBEB 25 FC EBED BCS LDATA3
00061A EBF1 08 INX
00062A EBF2 A6 09 A LDATA1 LDAA 0,X
00063A EBF4 81 04 A CMPA $4
00064A EBF6 26 F5 EBED BNE LDATA3
00065A EBF8 39 RTS
00066          END
TOTAL ERRORS 00000
    
```

```

EC11 CNTRL1 00020*00045
EC13 CNTRL2 00022*00030
EC19 DATA 00019*00028 00037 00038
EBE2 ERROR 00044 00049*
0008 FDSTAT 00014*
FD15 INCHNP 00016*
EBE4 LDATA 00053*
EBF2 LDATA1 00062*
EBED LDATA3 00059*00060 00064
EBE6 LDATA7 00055*00056
EBCC LIST 00036*00055 00059
EBD2 LIST3 00039*00046
EBCC LPINIT 00026*
E85A PRNTER 00015*
EC12 STAT 00021*00040
    
```

 BIBLIOGRAPHIE.
-ooOoo-ooOoo-ooOoo-ooOoo-

- Techniques d'interface aux microprocesseurs.
RODNAY ZAKS ET AUSTIN LESEA.
- M 6 800 Microprocessor Application Manual.
- Microprocesseurs et mémoires " EFCIS " .

B ROCHURES:

- USER'S GUIDE :
 - EXERCISER .
 - EXOR DISK II .
 - Floppy Disk CONTROLLER Module !
- MOTOROLA Semi conducteurs MC 8 506 P: Polynomial Générateur.

REVUES:

- Haut- Parleur N°166 7 .
- H aut - Parleur N°166 8 .