

ECOLE NATIONALE POLYTECHNIQUE

Aed

DEPARTEMENT ELECTRONIQUE

المدرسة الوطنية للعلوم الهندسية
 المكتبة

 ECOLE NATIONALE POLYTECHNIQUE
 BIBLIOTHEQUE

FLOPPY DISK DRIVE

المدرسة الوطنية للعلوم الهندسية
 المكتبة

 ECOLE NATIONALE POLYTECHNIQUE
 BIBLIOTHEQUE

PROJET DE FIN D'ETUDES

Proposé par :

M^r Tedjini H

Etudié par :

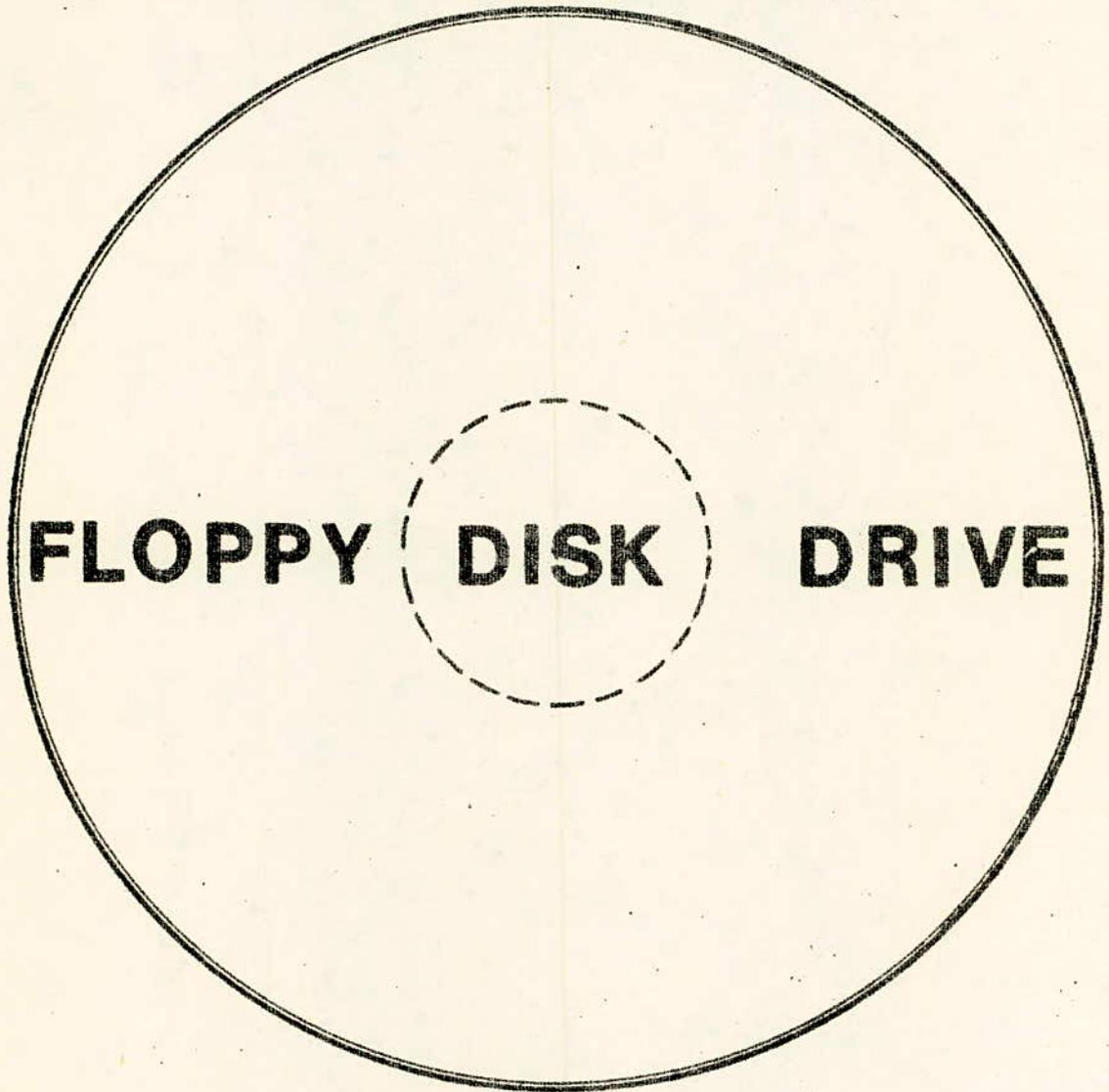
M^r Mouhamou

M^r Ngili S

UNIVERSITE D'ALGER

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT ELECTRONIQUE



PROJET DE FIN D'ETUDES

Proposé par :
M^r Tedjini H

Etudié par :
M^r Mouhamou B
M^r Naïli S

— Janvier 79 —

III-) III A III) EMOIRE

--0o0- DE NOTRE CHER ET REGRETTE -0o0-

-----0o0-----

III) RESIDENT

--oo-- III) O U A R I III) OUMEDIENE --oo--

  ON  ERE

A MA MERE
A MES FRERES & SOEURS
A TOUS MES AMIS

 AILI. S

  ON  RAND-PERE

A MON PERE
A MA MERE
A MES FRERES ET SOEURS
A TOUS MES AMIS (es)

 OUHAMOU B.

- 2°) système de positionnement
- 3°) Système de lecture/ écriture

V) OPERATION D'ECRITURE

- 1°) Interface de l'opération d'écriture
- 2°) Logique d'écriture du " Formatter "
- 3°) Logique de detection d'erreurs du Formatter

CHAPITRE V : PROPOSITIONS DES INTERFACES.

I) INTERFACE MICROPROCESSEUR/ FORMATTER

- 1°) Constitution de l'interface MPU/ FORMATTER.
- 2°) Interface proposé

II) INTERFACE FORMATTER/ UNITE DE DISQUES

CHAPITRE VI : ORGANIGRAMMES.

- 1°) select
- 2°) Read status
- 3°) READ
 - a) organigramme
 - b) commentaire
- 4°) WRITE.

CONCLUSION GENERALE.

BIBLIOGRAPHIE.

LEXIQUE ET ABREVIATIONS

INTRODUCTION

L'objet de notre projet est l'étude et la mise en ligne des unités d'entraînement du disque souple par laquelle nous contribuons à un projet: " SIMULATION D'UN REACTEUR NUCLEAIRE " en cours de réalisation au service SIMULATION ET CONTROLE DE l'institut nucléaire.

Notre étude a pour but donc, la mise en état d'exploitation des unités d'entraînement du disque souple d'appellation Anglo - Saxone Floppy disk drive ou drive tout court. Ces derniers, sont destinés à être utilisés par un mini-ordinateur de la série MULTI 20 D'INTERTECHNIQUE après un essai concluant avec le microprocesseur MC 6800 de MOTOROLA.

Le disque souple est vite devenu un important de stockage d'information. La promesse d'un coût peu élevé, l'accès direct et le temps d'accès amélioré ont encouragé les utilisateurs de mini-ordinateurs à choisir le disque souple pour le besoin de stockage de grande capacité (1.943.552 Bits)

Le système de floppy disk utilisé (CALCOMP 140) est normalisé (NORME IBM 3740) et enregistré les données de façon série avec une vitesse de 250 K Bits/ Seconde.

.../...

Notre travail consiste à relier ces unités d'entraînement du disque souple du MC 6800 et préparer le système à être utilisé avec efficacité. Pour cela, deux méthodes seront présentées : Une première méthode consiste en une gestion directe des unités d'entraînement du disque par le microprocesseur. Quant à la deuxième méthode, les unités de disque ne sont plus gérées directement par le microprocesseur mais par une logique auxiliaire appelée Formatter. Ainsi pour ne pas occuper l'unité centrale du mini-ordinateur, un formatter contrôle les unités d'entraînement du disque souple.

Le formatter de disquettes est un sous-système chargé de gérer les commandes et la " mise au format " des disques souples. "réalise douze commandes bien étudiées pour faciliter la gestion de transfert avec le microprocesseur (MPU)" est aussi utilisé pour la conversion série - parallèle ou parallèle - série, la récupération de données et leur contrôle.

Deux PIA (périphéral Interface Adapter) sont aussi nécessaires pour assurer un dialogue efficace et contrôlé entre le microprocesseur et le formatter d'une part ou le microprocesseur et les unités d'entraînement du disque souple d'autre part (selon la méthode utilisée).

Le MC.6800 constitue l'unité centrale pour la réalisation de microordinateurs. Les microprocesseurs de performance, en amélioration continue permettent de traiter les informations venant des unités périphériques, à partir d'une programmation adéquate et de circuits additifs prévus par les constructeurs (PIA, ACIA, RAM, ROM, ...); le MPU sera alors d'une souplesse remarquable et pourra s'adapter à l'utilisation du périphérique désiré.

PRESENTATION DU MPU. (MC.6800) Voir fig.1

Le MC.6800 est un circuit bidirectionnel basé sur la structure Bus, avec 8 Bits de données et 16 Bits d'adresse.

I/ Structure Bus: elle est constituée de 3 Bus:

1°/ DATA BUS (D₀ - D₇):

C'est un bus à 8 lignes bidirectionnelles permettant le transfert de données entre les organes de la famille du MC.6800

2°/ Bus Adress (A₀ - A₁₅):

C'est un bus d'adresses à 16 lignes sortantes du MPU avec possibilité "THREE STATE" (0,1,OFF) l'état "OFF" correspond à un circuit ouvert ce qui permet de faire des accès mémoire direct (DMA).

3°/ Bus Contrôle

a) R/W Direction du transfert de données

R/W = "1" pour lecture

R/W = "0" pour écriture

b) VMA (Valid Memory adress)

Valide l'adresse qui se présente sur le Bus d'adresses

c) RESSET: Cette entrée au MPU, est utilisée pour redémarrer ou remettre à zéro le MPU après une coupure de secteur ou pour une initialisation.

.../...

pour une routine de "RESTART" les locations FFFE, FFFF sont utilisées pour charger le programme adressé par le programme countete.

- d/ Horloge: Ø 1 et Ø 2 ce sont les deux phases d'une **même horloge**
Ø 1: prévue pour etiver le MPU de fréquence comprise entre 100KHZ
Ø 2: prévue pour activer les **autres éléments** et 1 KHZ
du MPU
- e/ D.B.E DATA bus Enable:
cette ligne nous renseigne sur l'état des données.
D.B.E ="1" le Data est à l'état "0" ou "1"
D.B.E ="0" le data est à l'état OFF des accès directs à la mémoire
peuvent alors être accomplis par la logique extérieure
- f/ B.A BUS AVAILABLE
Quand B.A est à "1", il indique que le MPU est stoppé (HALT);
dans ce cas le data Bus et l'adress Bus à l'état "OFF" seulement
après une entrée HALT.
- g/ T.S.C THREE State Contrôl
Son rôle est de mettre les lignes et R/W à l'état OFF; le Bus
adress est alors disponible pour un autre MPU par utilisation du
mode DMA pendant un court instant. Ce signal n'a pas d'effet sur le
data Bus.
- h/ HALT: Quand ce signal est l'état bas, le MPU cesse toute exécution (i.e)
les signaux "THREE STATE" sont l'état OFF, le signal B.A est à "0"
et VMA est à "0" . le MPU est alors pratiquement deconnecté des autres éléments.
- i/ N.M.I (Non Masquable Interrupt)
Lors d'une interruption NMI, le MPU sauvegarde le contenu de tous ses
registres internes (Accumulateurs A et B, P.C, C.CR) dans les adresses
indiquées par le SP (STACK pointe) met à 1 l'interrupt Mask (Im)
afin d'éviter qu'une autre interruption simultanée arrive, charge
son PC avec le contenu des adresses FFFC (PC High et FFFD (PC low)
puis commence à exécuter la routine d'interruption dont la première
instruction se trouve à l'adresse présente dans le PC.

.../...

j/ IRQ Interrupt Request :

Indique la présence ou l'absence de demande d'interruption de la part d'un des interfaces (PIA # 1, PIA # Z, ...). Cette interruption n'est prise en compte que si l'interrupt MASK, Im = "0".

Avant de reconnaître l'interruption le MPU termine l'instruction en cours, ensuite teste le bit Im du CCR. Si Im = 0 le MPU commence l'exécution d'une séquence de l'interruption en sauvegardant les contenus de (A, B; PC et CCR) dans la pile, puis chargera le PC avec le contenu des locations FFF 8 et FFF9 qui permettent au MPU de se brancher à un sous programme d'interruption.

// Structure Interne du MPU:

Le MPU est constitué de:

- 1°/ A.L.U (Unité Arithmétique et logique) où s'effectuent toutes les opérations.
- 2°/ Deux accumulateurs A et B de 8 bits chacun qui ont rôle principal de maintenir les opérandes et les résultats de l'A.L.U.
- 3°/ INX : Registre Index de 16 bits. Ce registre est très utilisé pour le stockage d'une donnée et pour l'adressage indexé.
- 4°/ Stack pointer: ou pointeur de pile; c'est un registre de 16 bits, qui contient l'adresse de la location disponible dans la pile mémoire. Cette pile (Stack) est utilisée pour charger et sauvegarder les registres ou autres zones mémoires.
- 5°/ Program. Counter (PC): ou compteur ordinal.
c'est un registre de 16 bits qui contient l'adresse courante du programme en exécution. Il est incrémenté chaque fois qu'on passe à l'instruction suivante.
- 6°/ Code condition register (CCR): c'est un registre de 8 Bits permettant de contrôler le déroulement du programme pendant une opération quelconque.
Ses Bits sont affectés par des instructions, les résultats des instructions ou des interruptions.

.../...

La signification des bits du CCR est la suivante :

- a) H. HALT CARRY utilisé quand le MPU travaille en BCD (Decimal code Binaire) H est mis à "1" s'il y a une retenue passant du bit b3 et b4 lors d'une condition par des accumulateurs A et B par exemple.
- b) Im: Interrupt Mask Mis à "1" après une interruption hardware ou software
Im = "0" la demande d'interruption hardware IRQ est acceptée
IM = "1" elle est rejetée
- c) N: Negative : La mise à "1" de ce bit indique que le bit b7 du resultat est "1" c'est à dire que le resultat est négatif.
- d) Z: Zéro : Ce bit est mis à "1" si le resultat est nul.
- e) V. OVERFLOW : nous renseigne sur le depassement de capacité.
- f) C: CARRY : Ce bit est mis à "1" pour indiquer l'existence d'une retenue au niveau de b7 du resultat.

III/ JEU D'INSTRUCTIONS:

Le MC.6800 comprend 72 instructions: instructions logiques, arithmétiques, de rotation, de déplacement, chargement, rangement, branchement conditionnels et inconditionnels et des instructions d'interruption et de pointage.

Elles sont classées en 4 catégories, selon qu'elles agissent sur:

- 1°/ les accumulateurs et la memoire
- 2°/ les registres: Index et pointer
- 3°/ le registre d'état (CCR)
- 4°/ les instructions de branchement et de saut

le minimum des cycles mémoires pour une instruction est 2 (L,DA) et le maximum est 12 (SWI). (Voir fig 1.2 jeu d'instructions)

IV/ Les modes d'adressages du M.6800

- 1°/ Adressage Inherent = instruction à 1 Octet, agissant sur un registre interne du 6800 exemple INCA, INX, DECA, ABA,...
- 2°/ Adressage immédiat: instruction à 2 ou 3 bytes selon qu'elle s'adresse aux accumulateurs A et B ou aux registres et servant au chargement, la comparaison, addition.

.../...

ACCUMULATOR AND MEMORY		ADDRESSING MODES															BOOLEAN/ARITHMETIC OPERATION (All register labels refer to contents)	COND. CODE REG.								
		IMMED			DIRECT			INDEX			EXTND			INNER				5	4	3	2	1	0			
OPERATIONS	MEMORIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#	H	I	N	Z	V	C	
Add	ADDA	89	2	2	9B	3	2	AB	5	2	BB	4	3				A ← M → A									
	ADDB	C8	2	2	D8	3	2	E8	5	2	FB	4	3				B ← M → B									
Add Acmltrs	ABA															1B	2	1	A ← B → A							
Add with Carry	ADCA	89	2	2	99	3	2	A9	5	2	B9	4	3				A ← M ← C → A									
	ADCB	C8	2	2	D9	3	2	E9	5	2	F9	4	3				B ← M ← C → B									
And	ANDA	84	2	2	94	3	2	A4	5	2	B4	4	3				A ← M → A									
	ANDB	C4	2	2	D4	3	2	E4	5	2	F4	4	3				B ← M → B									
Bit Test	BITA	85	2	2	95	3	2	A5	5	2	B5	4	3				A ← M									
	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3				B ← M									
Clear	CLR							GF	7	2	7F	6	3				00 → M									
	CLRA															4F	2	1	00 → A							
	CLRB															5F	2	1	00 → B							
Compare	CMPA	B1	2	2	91	3	2	A1	5	2	B1	4	3				A ← M									
	CMPB	C1	2	2	D1	3	2	E1	5	2	F1	4	3				B ← M									
Compare Acmltrs	CBA															11	2	1	A ← B							
Complement, 1's	COM							63	7	2	73	6	3				M → M									
	COMA															43	2	1	A → A							
	COMB															53	2	1	B → B							
Complement, 2's (Negate)	NEG							60	7	2	70	6	3				00 ← M → M									
	NEGA															40	2	1	00 ← A → A							
	NEGB															50	2	1	00 ← B → B							
Decimal Adjust, A	DAA															19	2	1	Converts Binary Add. of BCD Characters into BCD Format							
Decrement	DEC							5A	7	2	7A	6	3				M ← 1 → M									
	DECA															4A	2	1	A ← 1 → A							
	DECB															5A	2	1	B ← 1 → B							
Exclusive OR	ESRA	88	2	2	98	3	2	AB	5	2	BB	4	3				A ⊕ M → A									
	EORB	C8	2	2	D8	3	2	EB	5	2	FB	4	3				B ⊕ M → B									
Increment	INC							6C	7	2	7C	6	3				M ← 1 → M									
	INCA															4C	2	1	A ← 1 → A							
	INCB															5C	2	1	B ← 1 → B							
Load Acmltr	LDAA	88	2	2	96	3	2	A6	5	2	B6	4	3				M → A									
	LDAB	C8	2	2	D6	3	2	E6	5	2	F6	4	3				M → B									
Or, Inclusive	ORAA	9A	2	2	9A	3	2	AA	5	2	BA	4	3				A ← M → A									
	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3				B ← M → B									
Push Data	PSHA															36	4	1	A ← M _{SP} , SP ← 1 → SP							
	PSHB															37	4	1	B ← M _{SP} , SP ← 1 → SP							
Pull Data	PULA															32	4	1	SP ← 1 → SP, M _{SP} → A							
	PULB															33	4	1	SP ← 1 → SP, M _{SP} → B							
Rotate Left	ROL							69	7	2	79	6	3				M									
	ROLA															49	2	1	A							
	ROLB															59	2	1	B							
Rotate Right	ROR							66	7	2	76	6	3				M									
	RORA															46	2	1	A							
	RORB															56	2	1	B							
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M									
	ASLA															48	2	1	A							
	ASLB															58	2	1	B							
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M									
	ASRA															47	2	1	A							
	ASRB															57	2	1	B							
Shift Right, Logic	LSR							64	7	2	74	6	3				M									
	LSRA															44	2	1	A							
	LSRB															54	2	1	B							
Store Acmltr	STAA				97	4	2	A7	6	2	B7	5	3				A → M									
	STAB				D7	4	2	E7	6	2	F7	5	3				B → M									
Subtract	SUBA	80	2	2	90	3	2	A0	5	2	B0	4	3				A ← M → A									
	SUBB	C0	2	2	D0	3	2	E0	5	2	F0	4	3				B ← M → B									
Subtract Acmltrs	SBA															1B	2	1	A ← B → A							
Subtr. with Carry	SBCA	82	2	2	92	3	2	A2	5	2	B2	4	3				A ← M ← C → A									
	SBCB	C2	2	2	D2	3	2	E2	5	2	F2	4	3				B ← M ← C → B									
Transfer Acmltrs	TAB															16	2	1	A → B							
	TBA															17	2	1	B → A							
Test, Zero or Minus	TST							60	7	2	70	6	3				M ← 00									
	TSTA															40	2	1	A ← 00							
	TSTB															50	2	1	B ← 00							

FIGURE 1-3-1 MC8800 Instruction Set

a) 2 bytes: s'adressant aux accus A et B.

Ex: LDA A immédiat 25 (I) consiste à charger l'accumulateur A par
86 25 même instruction que (I) écrite en code²⁵
opération.

b) 3Bytes: s'adressant aux STACK POINTER et registre Index. de 16 bits
chacun.

Ex: LDX A001 signifit: charger le registre index X immédiatement
par A001.

3°/ ADRESSAGE DIRECT : utilise 2 Bytes et ainsi ne peut atteindre que les
adresses 225 l'instruction est suivie d'un mot de
1 Octet qui re présente l'adresse où aller chercher la
donnée à traiter.

Ex: STA A 52 ---- signifiant: stocker le contenu de l'accumula-
A à la location mémoire 0052. teur
ou en code opération: 97 52

4°/ ADRESSAGE ETENDU: C'est une extention de l'adressage direct. l'instruction
est suivie de deux mots de 1 Octet chacun et qui repré-
sentent l'adresse où aller chercher la donnée à traiter.
Ce mode d'adressage permet donc de balayer toute la
mémoire de 0000 à FFFF.

5°/ ADRESSAGE RELATIF: Utilant deux bytes dont le 2ème contient l'adresse ou
plus précisément le déplacement que doit effectuer le PC
pour que le MPU puisse continuer son programme.
le déplacement se fait entre : 125 et 129 ainsi cette
adressage est utilisé dans les branchements. Quand le
PC rencontre un "BRANCH", alors il lit le déplacement
à effectuer, puis s'incrémente automatiquement de 2 pour
passer à l'instruction suivante dont l'adresse sera
prise comme référence à laquelle on ajoutera le déplace-
ment. Deux cas se présentent :

1er Cas déplacement vers l'avant

PC sera chargé par (PC) + 2 + déplacement (addition
en hexadecimal)

2ème Cas: Déplacement vers l'arrière: Moins simple que le précédent
car il ne s'agit plus ici d'une addition. Il faut faire
une soustraction :

.../...

Exemple: supposons que lorsque PC pointe sur BNE (BRANCH if not equal zéro) il contient 000A et que le déplacement à effectuer en arrière soit FA = 1111 1010 étant un nombre négatif car $b_7 = 1$ au lieu de faire: $(PC) + 2 + FA$ pour trouver l'adresse de retour. on effectue une complémentation à 2 de FA comme suit:

FA = 1111 1010 on le rend positif : 0111 1010
on change les bits par 1000 0101
leurs opposées
et on ajoute "1" +0000 0001
on trouve le complément à
2 de FA : 1000 0110

ce complément à 2 s'écrit alors en hiéxadécimal - 0006
et la bonne opération à effectuer pour pointer le PC au retour
désiré est $PC + 2 - 0006 = 000A + 0002 - 0006 = 0006$

6°/ ADRESSAGE INDEXE: instruction utilisant deux bytes dont le 2^{ème} représente le déplacement à ajouter à l'index X pour fournir l'adresse où aller chercher la donnée à traiter.

Ce complément étant code sur 8 Bits, son amplitude se trouve limitée à 255 en valeur absolue ce qui n'est pas un inconvénient puisqu'en jouant sur X, on peut décrire facilement toute la mémoire.

Exemple: supposons que dans le registre Index on a la quantité 2 A et que l'instruction à exécuter est :

! LDA A 35, X cela signifie, charger le
!- contenu de l'adresse
! ou en code opération A6 35: X + 35 = 2A + 35 = 5 F dans
! l'accumulateur A

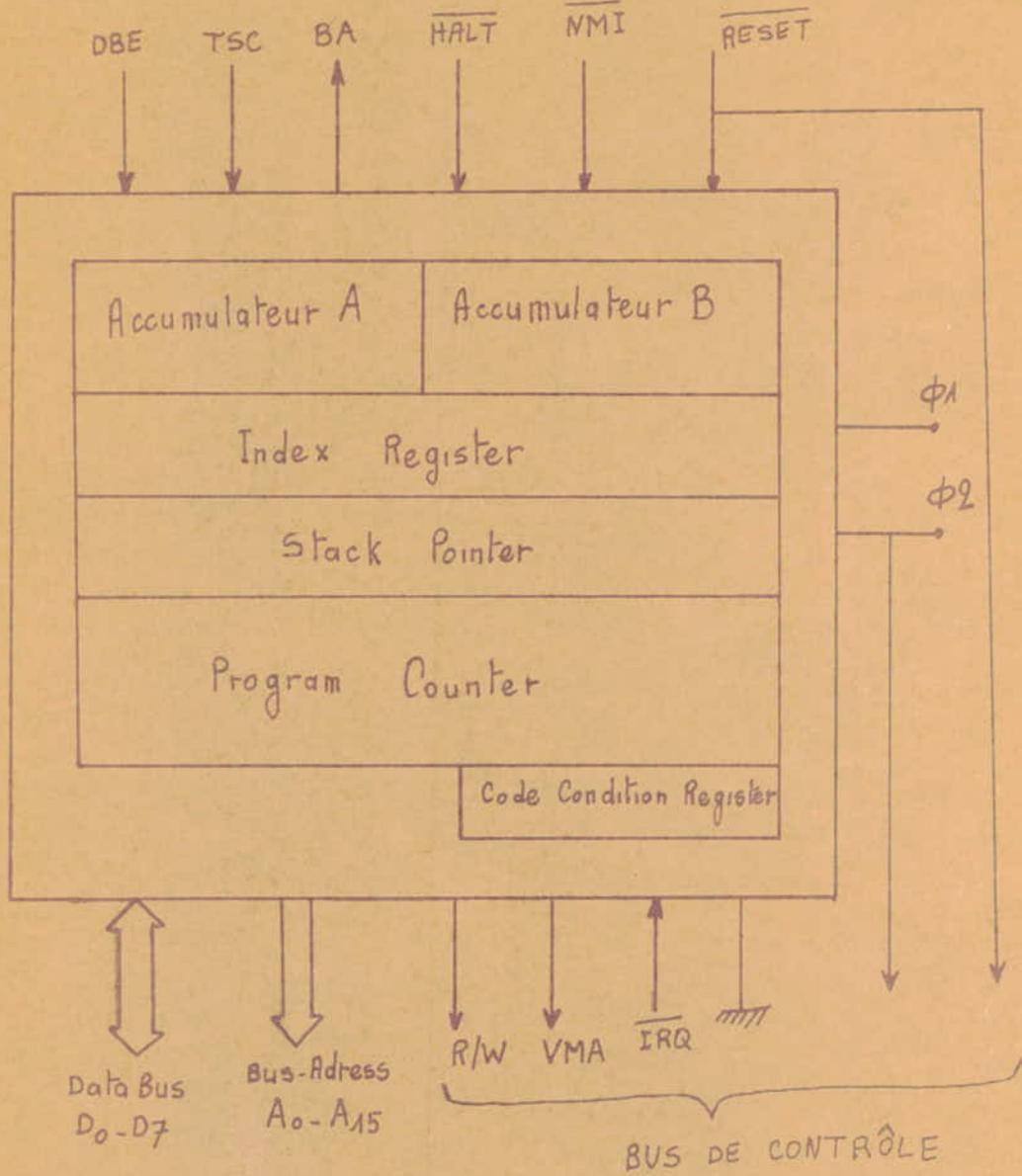


Fig I-1. Organisation interne et externe du MPU

CH 2 LES INTERFACES DU PIA

Le PIA MC.6820 assure une méthode souple d'échange d'informations (données, signaux de contrôle, d'états, d'interruption ...) entre le MPU et les unités extérieures. Bien que relativement, complexe lui-même le PIA permet au MPU de manipuler une large variété de types d'équipements avec un minimum de logique supplémentaire et une simple programmation.

Quant au périphérique que nous utilisons, en l'occurrence les Unités d'entraînement de disque souples et le formater et que nous aborderons, par la suite, le grand nombre de signaux mis en jeu nous amène à les "interfacer" c'est à dire les relier au MPU par l'intermédiaire de deux PIA (Peripheral Interface Adapter).

Avant d'aborder les différents signaux et leur utilité; ainsi que leur organisation dans les deux interfaces; nous préférons pour être plus clairs dans notre exposé, d'écrire un PIA avec ses différents registres et l'ensemble de ses entrées/ sorties

I/ PIA (Peripheral Interface Adapter)

1°/ Organisation extérieure d'un PIA (Voir Fig. II 1)

du côté du MPU on distingue les lignes de signaux suivantes :

Do - D7: Ce sont les lignes bidirectionnelles permettant le transfert de données entre le PIA et MPU.

E (ENABLE) : c'est un signal derivant généralement de $\emptyset 2$ et qui valide le transfert en controlant les interruptions venant de CA, CA 2, CB 1 et CB 2.

R/W : Signal d'écriture - lecture donnant le sens de transfert des données

RESET : signal servant à effacer tous les registres du PIA (remise à zéro)

CHIP SELECT CS0 - CS.1 CS.2 : Ces lignes d'adresse permettant de selectionner le PIA elles sont reliées à 3 Lignes du bus (Ao - A15) et doivent rester stables pendant la durée de l'impulsion E.

REGISTER SELECT: (RS0 - RS1) : permettent la selection des 6 registres internes du PIA.

INTERRUPT REQUEST (IRQA, IRQB) : Ces lignes agissent comme demandes d'interruption au MPU. A chaque IRQ correspondent, dans le registre de contrôle du PIA, 2 Bits "flag" b7, b6 qui peuvent désactiver la demande d'interruption, ce qui permet d'inhiber une interruption.

.../...

PA0 - PA7 : (Peripheral Interface A)

Ces lignes peuvent individuellement être programmées soit en entrée soit en sortie. Programmées comme ~~sorties~~ le transfert des données se fait du MPU vers le peripherique. Programmées comme entrées: le MPU peut lire les données fournies par le peripherique.

LIGNES PBO - PB 7:

Même fonctionnement que PA0 - PA.7 sauf qu'elles possèdent trois états (0,1,OFF) et des buffers de sortie.

CAI :

Cette entrée signale au MPU la présence d'une information à l'entrée du PIA. La transition active de CA 1 (front montant ou descendant selon le choix) mettra à "1" le bit b7 du registre de controle (CRA) et placera un "0" sur **IRQA**

CB 1:

Si les lignes PBO - PB 7 sont programmées en sortie; l'entrée CB 1 signale au MPU qu'un périphérique demande une information.

CA 2 :

La ligne CA 2 utilisée en sortie est remise à "0", après une opération de lecture de l'information présente à l'entrée A du PIA, par le MPU.

CB 2:

CB2 utilisée en sortie est remise à "0" après une opération d'écriture dans le PIA par le MPU.

2°/ ORGANISATION INTERNE DU PIA (Voir fig. II 1)

Le PIA est divisé en deux configuration symétriques indépendantes. Chaque moitié à 3 registres: 1 output register, 1 control register et 1 data register.

Le MPU traite ces registres comme des locations mémoires, il peut y lire ou écrire. Les output register et les DDR sont sélectionnés à l'aide de RSO, RS 1 et le bit deux "du control register" (CR 2, A ou B)

.../...

Le tableau suivant nous montre comment peut être sélectionné un des 6 registres du PIA.

TABLEAU : adressage interne

RS 1	RS 0	Registre de controle		registre sélectionné
		bit b 2		
		CRA	CRB 2	
0	0	1	x	PIRA (périphéral Interface register A)
0	0	0	x	DDR A
0	1	x	x	CRA
1	0	x	1	PIRB (périphéral Interface register B)
1	0	x	0	DDR B
1	1	x	x	CRB

DDR (DATA DIRECTION REGISTER):

Son rôle est d'établir le sens des lignes périphériques PA et PB. Pour que ces lignes soient sortantes on écrit des "1" sur les DDR, ou pour les avoir entrantes on écrit des "0" sur les DDR.

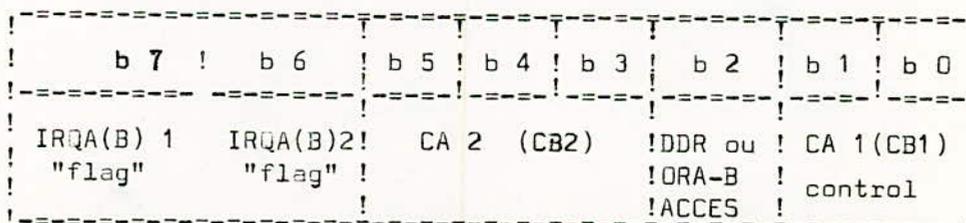
OR : OUT PUT REGISTER :

Quand ils sont adressés ces registres rangent les données présentes au DATA BUS du MPU, pendant une opération d'écriture (sur registre). Ces données vont aussi, apparaître sur les lignes périphériques PA, PB programmées en sortie.

CR: CONTROL REGISTER :

Chaque registre de controle permet d'une part d'adresser DDRA, DDRB et ORA, ORB (CRA,B étant directement sélectionnées par RS0 - RS 1) et d'autres part de contrôler le transfert des données et le lancement des interruptions. Il sert aussi à programmer CA 2 et CB 2 en entrée ou sortie.

LA CONFIGURATION DU REGISTRE DE CONTROLE EST LA SUIVANTE :



Les bits b 0 - B 5 peuvent être écrits et lus par le MPU mais les bits b6 - B7 peuvent seulement être lus et sont modifiés uniquement par des interruptions externes arrivant sur les lignes de contrôle CA.1, CB.1, CA2 et CB.2

II/ SYSTEM HARDWARE / SOFTWARE INTERFACE :

La fig. II.2) est un block diagramme fonctionnel qui montre toutes les lignes de signaux entre le systeme du floppy et les PIA.

1°/ PIA # 1

- Le périphéral Interface ou encore le registre périphérique A est un registre Buffer de signaux de mande d'écriture pour les opérations d'écriture et une porte de données à lire pour les opérations de lecture.
- Le registre périphérique B est un Buffer de données à écrire pour les opérations d'écritures et est utilisé aussi comme porte pour les signaux d'états pour les opérations de lecture.
- Les 4 signaux d'états de lecture qui sont acheminés au PIA # 1 via le registre périphérique B sont :

PB0 + 1ST BIT

PB 3 - CRC = 00

PB 6 - index CLOCK MARK

PB 7 - ID/ DATA MARK

à noter que: + signifie actif au niveau haut

- actif au niveau bas

PB0 + 1ST BIT

C'est un signal de lecture qui se déclenche quand un bit de donnée "1" a été détecté dans une donnée à série à lire. Le déclenchement de 1ST bit est validé et effacé par le signal de controle Enable CRC

PB.3 - CRC = 00

CRC = 00 est utilisé dans le contrôle d'erreur pour valider la donnée à lire.

CRC = 00 passe à "0" pendant la période de un byte après qu'un enregistrement ait été lu, s'il n'y avait pas d'erreurs de lecture détectées.

.../...

PB.6 - INDEX CLOCK MARK

Index CLOCK MARK passe à "0" pendant le temps d'un byte durant une opération de lecture quand la chaîne de données séries contient $\$$ D7 dans les bits d'horloge. Ce profil unique est enregistré après le signal Index.

PB.7 - ID/ DATA CLOCK MARK

ID/DATA CLOCK MARK passe à "0" pendant un byte durant une opération lecture quand un profil d'horloge spécial $\$$ C.7 est détecté. Quand ce signal à lieu, cela signifie que le signal BYTE READY suivant (CA.1) correspondra au premier byte de données de l'enregistrement.

Les registres A et B sont utilisées comme deux entrées (opération de lecture) et sorties (opération d'écriture). Parceque toutes les opérations d'écriture de champs de données doivent être précédées par une opération de lecture d'un champ ID (Identificateur), les registres de direction doivent être remis tous à zéro en préparation d'une opération de lecture CA1 et CA2 du PIA # 1 sont utilisés pour synchroniser la durée du programme au débit de données durant des opérations d'écriture ou de lecture.

CA 1

Est une ligne de mande de service. Dans une opération de lecture une transition positive de la ligne CA 1 indique qu'un Byte complet a été assemblé et est présent au registre périphérique A.

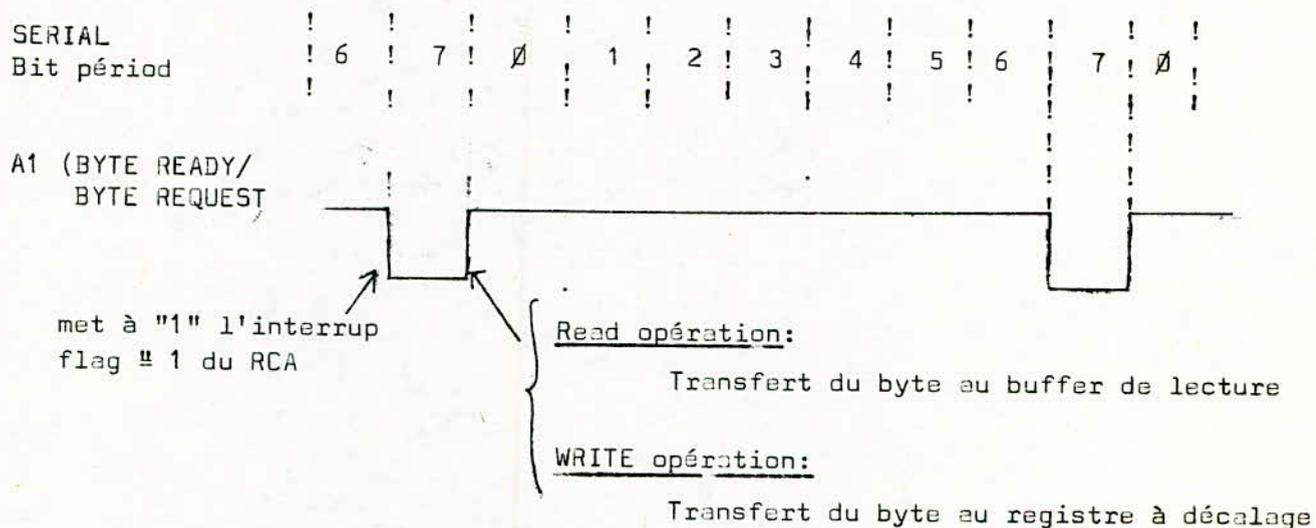
Dans une opération d'écriture une transition positive de CA 1 signifie que les profils de données et d'horloge dans les registres périphériques A et B ont été transférés aux registres à décalage parallèle/ série et les registres périphériques A et B peuvent alors être mis à jour pour d'autres données à transférer.

La synchronisation des données est controlée par exploration du bit b 7 du (indicateur d'interruption) du registre de contrôle.

Une transition active de la ligne CA 1 met à "1" le bit (b7) du RCA. Ceci prend au MPU un minimum de 8 cycles pour reconnaître que l'indicateur a été mis à 1 (si le PIA) est situé dans le rang d'adresse étendue de $\$$ 0100 à $\$$ FFFF) prés que le flag soit reconnu par le MPU, une donnée à lire ou à écrire est transférée du PIA ou vers le PIA. A une fréquence de base de 1MHZ du MPU il y a un retard minimum de 8 entre la reconnaissance de la demande de service et l'établissement du service du PIA.

LE SIGNAL CA.1 (BYTE READY /BYTE REQUEST/

Dure le temps d'un bit. Il passe à "0" au commencement du dernier bit d'un bit série, puis repasse à "1" au commencement du Byte série suivant (voir fig. ci-dessous)



A cause du retard du programme de 8 us ; l'interrupt flag (b7) pourra se mettre à "1" au commencement du dernier bit **quoique** les données ne seront pas encore prêtes avant la fin de ce bit.

CA.2 :

La ligne CA.2 est la réponse du MPU au système de disque indiquant que la demande de service à CA 1 a été acceptée. Elle est utilisée pour signaler des erreurs de dépassement de capacité dans une opération de lecture et un dépassement de capacité négatif dans une opération d'écriture.

Le mode d'opération de CA 1/CA 2 du PIA # A est l'établissement de liaison (Handshaking) dans lequel CA2 est à "1" d'une transition active de CA1 et passe à "0" par une instruction de lecture du MPU (LDA, BIT, ...)

CB 2 :

Est un signal de controle appelé "ABOVE TRACK A3" qui est utilisé uniquement dans les opérations d'écriture. Quant les données sont à enregistrer sur une piste disk de N° supérieur à 43 , cette ligne est mise à "1". Ce signal est utilisé par l'unité de disque CAL COMP 140 pour controler l'amplitude du courant d'écriture sur les pistes afin de réduire les influences magnétiques qui peuvent occasionner un phénomène appelé Bit Shift.

2°/ PIA * 2

Ce PIA est utilisé comme interface - de controle
- des états
- d'interruption

a) SIGNAUX DE CONTROLE :

PA0 :	- WRITF	GATE
PA1 :	- ENABLE	WRITE
PA2 :	- ENABLE	CRC
PA3 :	- SHIFT	CRC
PA 4:	- DISK	SELECT
PA5 :	- FILE	INOP RESET
PA6 :	+ READ	ENABLE
PB5 :	- HEAD	LOAD
CA2 :	-	STEP

PA0: - WRITE GATE (sortie)

Met en marche le courant d'écriture de l'unité d'entraînement du disque

PA1 : - ENABLE WRITE (sortie)

Commute les circuits du formater (Write logie et error detect) pour le mode d'écriture.

PA2: - ENABLE CRC (sortie)

Opération d'écriture :

Enable CRC met en marche le Générateur Polynomial CRC.

Ce signal est à "0" durant le byte dans lequel la marque d'adresse (Adress Mark) est transférée au au Buffer d'écriture.

Opération de lecture :

Quand Enable CRC est à "0" l'opération de lecture commence, le Générateur ponymial est déclenché automobile au 1er bit. Le retour à "1" de Enable CRC remet à zéro les circuits de lecture du formater.

.../...

PA.3 -SHIFT CRC (sortie)

SHIFT CRC est utilisé durant une opération d'écriture. Ce signal à zéro pendant le byte qui vient après que le dernier byte de données soit transféré au buffer d'écriture. Cela fait que les bytes CRC accumulés soient annexés à la chaîne des données à écrire en série. SHIFT CRC doit être maintenu à "0" pendant deux bytes.

PA4 : - DISK SELECT (sortie)

Ce signal agit sur l'unité de disque pour permettre son utilisation.

PA5: - FILE INOPERABLE (sortie)

C'est la mise à "0" de l'unité de disque non opérationnelle. File inopérable reset efface n'importe quelles conditions d'erreurs verrouillées (dans un circuit appelé latch). Si la condition d'erreur est hardware (i-e) présente au moment de la mise à "0" la bascule de verrouillage d'erreur ne se met pas à zéro.

PA6: + READ NABLE (sortie)

Il permet la mise à "1" de "NOT IN SYNE ERRAR LATCH" dans le cas d'une perte de synchronisation de données series en lecture. READ ENABLE est mis à "1" durant une opération de lecture après que la marque d'adresse désirée ait été reconnue par le programme.

PB4 : + FORWARD Direction (sortie)

Quant c'est à "1" , FORWARD Direction fait mouvoir la tête R/W vers le centre du disque (dans le sens des adresses des pistes croissantes) avec une impulsion de saut (STEP).

Quant c'est à "0", la tête R/W s'éloigne du centre du disque vers la piste 00 (vers le bord extérieur du disque) avec une impulsion de saut (s tp).

PB5 : - HEAD LOAD (sortie)

Ce signal passant à "0" fait que la tête R/W touche la surface d'enregistrement du disque.

CA2 : - STEP (sortie)

C'est un signal sortant de controle de sauts. Durant une opération de recherche, la tête R/W est déplacée d'une piste pour chaque impulsion du signal STEP (CA2). Ceci quant l'impulsion est générée par lecture du registre périphérique A de PIA * 2 dans une opération de recherche. Si cela n'est pas dans une opération de recherche, le signal STEP est maintenu à "1".

b) SIGNAUX D'ETATS :

Ils nous renseignent sur l'état de l'unité de disque.

PA7 : + TRACK 00
PBO : + NOT IN SYNC
PB1 : + DISK SYSTEM INOPERABLE
PB2 : + UNDERFLOW/ OVERRUN
PB3 : + WRITE PROTECTED
PB6 : • READY
PB7 : + INDEX

PA7: + TRACK 00 (entrée)

C'est un signal d'état de l'unité de disque , qui se met à "1" quand la tête de R/W est à la piste 00.

PBO : + NOT IN SYNC (entrée)

C'est un signal de verrouillage d'erreur qui se met à "1" quand un bit "0" est détecté dans le profil de bits d'horloge en train d'être lus. Le signal de verrouillage d'erreur ne peut pas se mettre à "1" quand READ ENABLE est bas.

PB1 : + DISK SYSTEM INOPERABLE (entrée)

C'est le "OU LOGIQUE" de file INOPERABLE venant de l'unité de disque et de ENABLE WRITE si l'état WRITE PROTECTED de l'unité de disque est actif. Ce signal est la sortie d'une bascule de verrouillage d'erreurs qui signifie que le système du disque est impropre à utiliser (les données enregistrées peuvent être détruites).

PB2 : + UNDER FLOW/ OVERRUN (entrée)

C'est un signal de bascule de verrouillage qui est mis à "1" quand le MPU ne répond pas à temps au signal (Byte Request/Byte Ready) venant du système de disque. Cela signifie qu'une donnée a été perdue. Le mode d'opération pour établissement de liaison, de CA1 et CA2 du PAI * 1 gère le "timing" de cette condition d'erreur.

PB3 : + WRITE PROTECTED (entrée)

C'est un signal d'état de l'unité de disque qui signifie qu'une opération d'écriture ne peut pas être accomplie. Un état d'erreur aura lieu si ENABLE WRITE ou WRITE GATE est rendu actif pendant que l'unité de disque est protégée contre l'écriture (WRITE PROTECTED).

PB6 : - READY (entrée)

C'est un signal d'état de l'unité de disque qui signifie que l'unité de disque est prête à l'utilisation.

PB7: + INDEX (entrée)

C'est une impulsion de durée approximative 450 us qui arrive une fois à chaque révolution du disque. Index indique le commencement d'une piste.

C) SIGNAUX D'INTERRUPTIONS

CA1 : - READY

CB1 : + INDEX

CB2 : + R/W ERROR.

CA1: - READY (entrée)

CA1 est utilisée pour générer une interruption du MPU quand l'unité de disque passe de l'état NOT READY à l'état READY.

CB1 : + INDEX (entrée)

(b0 = 0 et ~~1~~) du FP2CRB

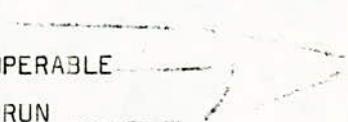
Durant une opération d'écriture de donnée, le front d'attaque de Index indique WRITE ERROR.

Durant l'opération d'écriture du format, le front d'attaque de Index est la synchronisation de référence initiale. L'interruption Index est invalidée dans l'écriture de format sinon dans une opération d'écriture de données.

CB 2 : + R/W ERROR (entrée)

L'interruption R/W ERROR est validée ou permise durant une opération de transfert de données. R/W ERROR est le "OU LOGIQUE" de:

NOT IN SYNC
DISK SYSTEM INOPERABLE
UNDERFLOW/UNDERRUN



R/W ERROR

Durant les opérations de transfert de données, une transition au front montant sur CB2 génère une interruption du MPU. Le programme d'interruption alors suspend l'opération de transfert de données.

.../....

3°/ LES REGISTRES DE CONTROLE CRA ET CRB DU PIA # 2

3.1) - FP 2 CRA PIA # 2 CONTROL REGISTER A, ADRESSE § 8042

b 7	b 6	b 5	b 4	b 3	b 2	b 1	b 0	
						0	1	a
						1	1	b
						1	0	c
		1	0	1				d
		1	1	1				e
				1				f
				0				g

- a) : CA1 est programmé pour générer une interruption du MPU quand l'unité de disque passe de l'état NOT READY à l'état READY durant l'inutilisation.
- b) : Dans une opération de transfert de données, CA 1 est programmée pour générer une interruption d'erreur au MPU si le drive n'est pas prêt.
- c) : Quand on est pas dans une opération de transfert de données, l'opération READY n'est pas validée.
- d) :-CA 2 est le signal de controle (sortant) STEP. Durant une opération de recherche, la tête R/W est déplacée d'une piste par pulsation du signal STEP (CA 2)
- Qand dans une opération de recherche, l'impulsion STEP est **générée** par lecture du registre périphérique A (FP 2 PR A)
- e) : Quand il ne s'agit pas d'opération de recherche, le signal STEP est maintenu à "1"
- f) : Pour la sélection du registre périphérique A de PIA # 2
- g) : Pour la sélection du registre de direction A de PIA # 2

.../...

3.2) FP.2 CRB PTA * 2, REGISTRE DE CONTROLE B; ADRESSE § 8043

b 7	b 6	b 5	b 4	b 3	b 2	b 1	b 0	
!	!	!	!	!	!	1	1	a
!	!	!	!	!	!	1	0	b
!	!	0	0	1	!	!	!	c
!	!	0	0	0	!	!	!	d
!	!	!	!	!	1	!	!	e
!	!	!	!	!	0	!	!	f

- a): Durant les opérations d'écriture de données, le premier front INDEX implique une erreur d'écriture (WRITE ERROR).
- b): Durant une opération d'écriture de format, le premier front de INDEX est la synchronisation initiale de référence. L'interruption INDEX est invalidée dans l'écriture de format, sinon dans une opération d'écriture de données.
- c): R/W ERROR; Cette interruption est validée durant l'opération de transfert de données.
R/W ERROR est le "OU" logique de:
NOT IN SYNC, DISK SYSTEM INOPERABLE et UNDER FLOW/OVERRUN.
- Durant les opérations de transfert de données, une transition (front montant) sur CB2 génère une interruption du MPU. de programme d'interruption suspend alors l'opération de transfert de données.
- d): Quand l'interruption est invalidée moins pas dans une opération de transfert de données.
- e): Sélection du registre périphérique B du PIA * 2.
- f): Sélection du registre de direction B du PIA * 2.

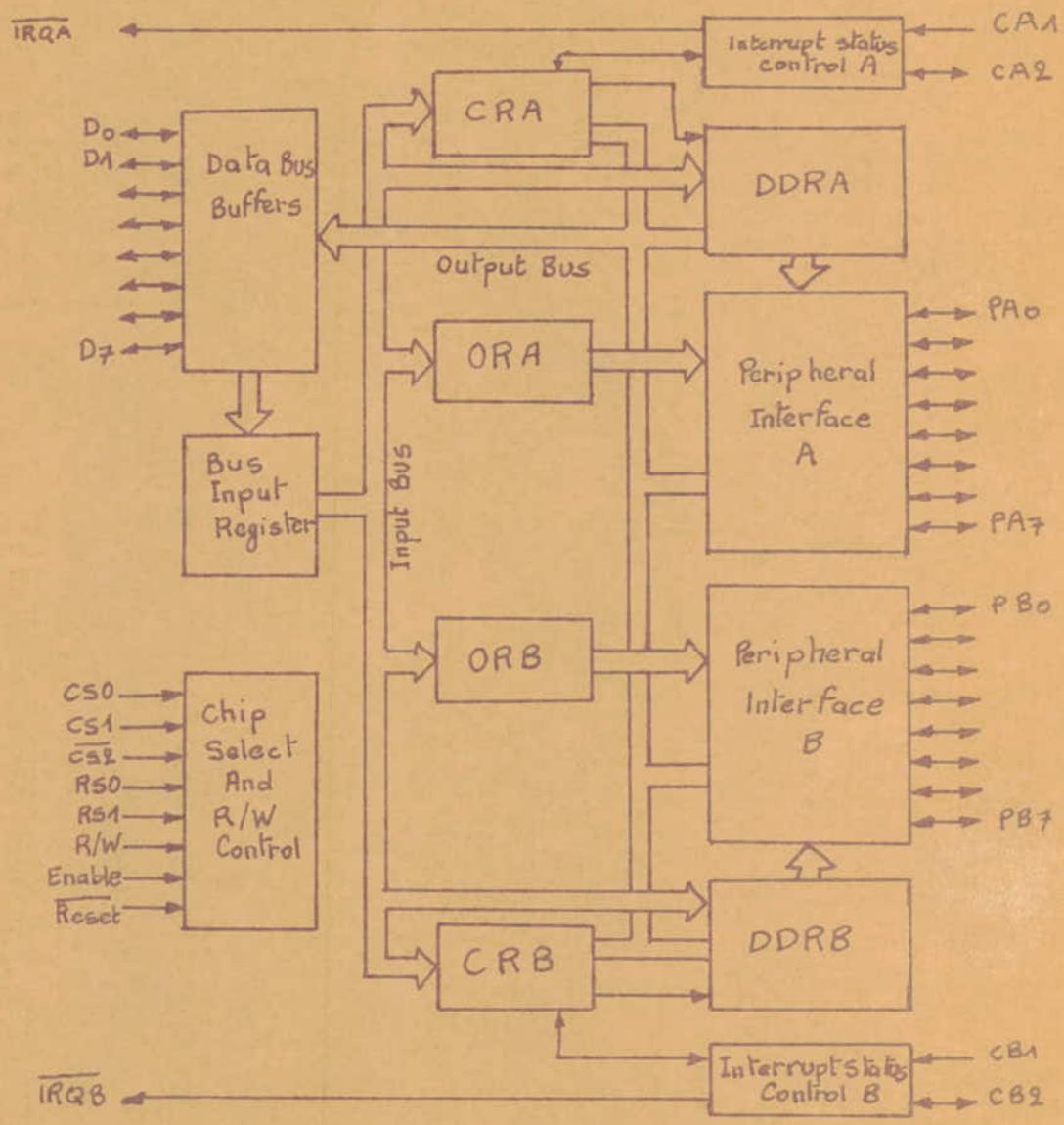


Figure II-1. Diagramme fonctionnel du PIA

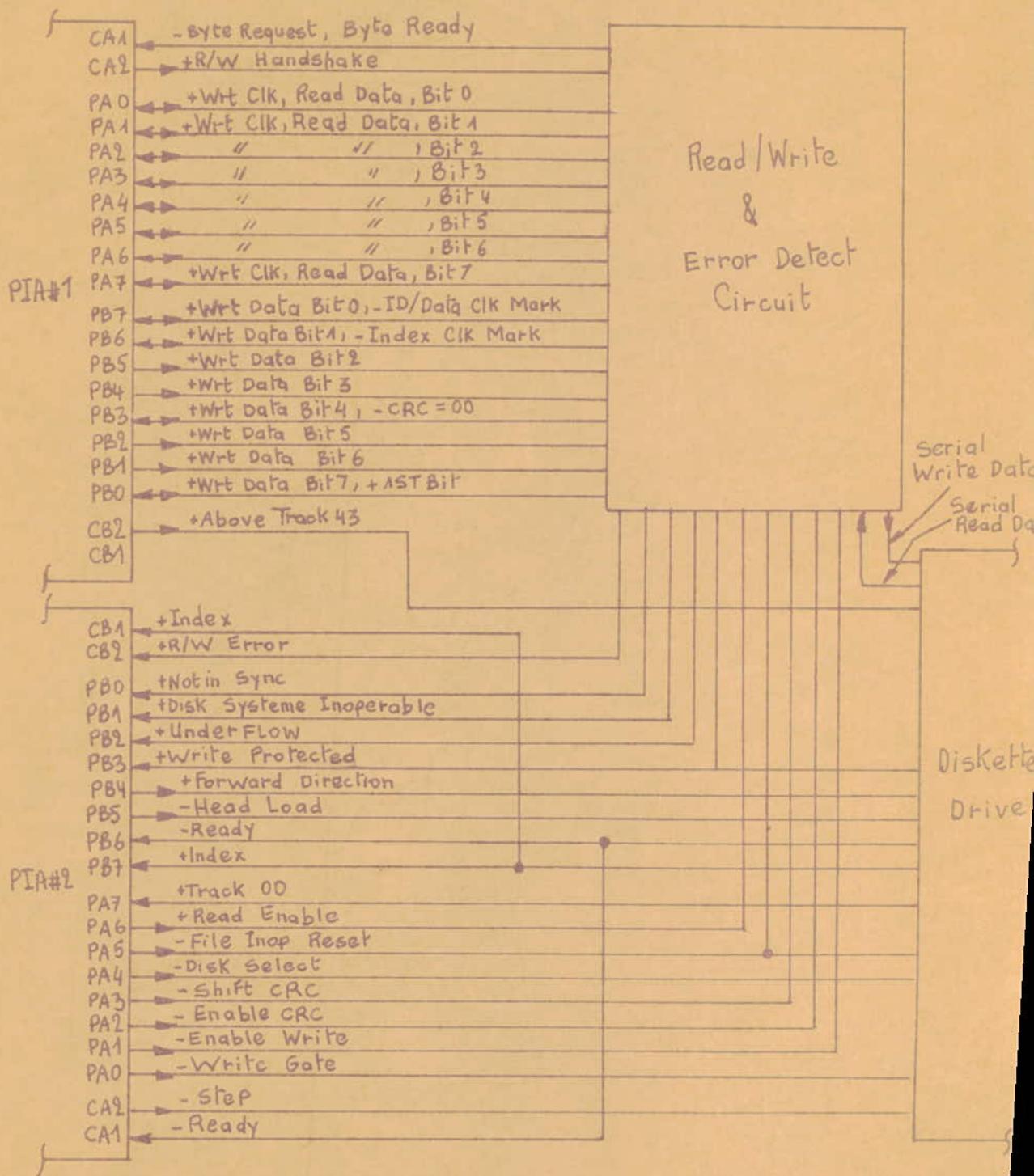


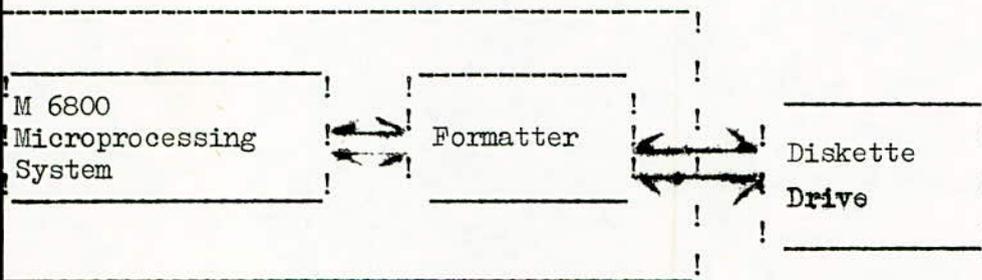
Figure 11.2. Interface du Floppy Disk

A cause de l'enregistrement série des données sur le floppy et du taux élevé de données séries, il est nécessaire d'utiliser une logique auxiliaire pour:

- la conversion parallèle - série et série - parallèle.
- la récupération des données
- le contrôle des erreurs dans les données

Cette logique auxiliaire qui est utilisée comme interface entre le floppy disk et le système M 6800 (MPV + 2 PIA) et qui accomplit les fonctions nommées, est appelée FORMATTER. Le formatter sert aussi comme buffer entre le système 6800 et les unités de disque.

Le but du système 6800 et du formatter est de contrôler les unités d'entraînement du disque. Ce système M 6800 / formatter est appelé contrôleur du disque souple.



Sous-Système M 6800 pour FLOPPY disk.

Les commandes du formatter (Calcomp 1140 FLOPPY disk Formatter) permettent de déterminer les causes d'une erreur et de récupérer une donnée d'une piste défectueuse.

1°) COMMANDES FONCTIONNELLES. (voir fig.III. 1 et III 2)

12 commandes de 16-bits chacune sont prévues pour contrôler le système 1140/140. Ces commandes sont étudiées pour faciliter la gestion de transfert entre le MPU et le Formatter. Chaque commande, sauf READ STATUS et SECTOR LENGTH possède une partie réservée au numéro de l'unité de disque à laquelle elle appartient. Ceci permet d'envoyer les différentes commandes à plusieurs unités (pas plus que quatre (4)).

SELECT : Cette commande accomplit la sélection de l'unité de disque et présente les états suivants s'il y a lieu.

- 1°) Drive not ready
- 2°) Write Protect.

L'unité choisie ne serait pas disponible si un disque n'est pas convenablement placé ou bien l'unité de disque n'a pas atteint sa vitesse opérationnelle.

L'état Write Protect signifie que le disque ne peut qu'être lu seulement.

../...

RESTORE : La commande RESTORE fait que l'unité de disque accomplisse 76 sauts vers le bord extérieur du disque et teste le signal TRACKOO. Si après 76 sauts, TRACKOO n'est pas détecté, un état RECALIBRATE ERROR se présente.

SEEK : Passé dans la commande SEEK, le numéro de piste est automatiquement localisé et vérifié une opération de recherche (SEEK) réussie est indiquée par la comparaison de l'adresse de piste et CRC CHECK, et l'état SEEK COMPETE se présente. Une erreur de CRC mettra à "s" l'état ID DATA CHECK après quatre tentatives de vérification.

Quatre tentatives de vérification du numéro de piste sont faites avant incrémentation vers la piste suivante dans la même direction. Cette opération est répétée deux (2) fois afin de tenir compte de deux pistes défectueuses avant que l'état SEEK ERRDR se présente.

SECTOR LENGTH: Cette commande définit la longueur du secteur formaté (en bytes) toutes les commandes après une coupure du secteur ou remise à zéro du système seraient exécutées avec la longueur de secteur implicite de 128 bytes. Une fois que la longueur du secteur est émise, elle reste en effet jusqu'à réémission ou que remise à zéro soit faite.

READ : La commande READ est lancée après une commande SEEK ou RESTORE. L'unité de disque s'oriente vers le numéro d'enregistrement écrit dans la commande. La donnée est lue dans cette enregistrement (SECTOR) et dans ceux qui lui succèdent jusqu'à ce que un de ce qui suit ait lieu:

1°) La ligne READ devient inactive. Si moins d'un secteur entier est lue quant la commande READ termine, un contrôle CRC (CRC Check) du secteur est accomplie. Si une erreur est trouvée, l'état DATA CHECK se présente.

2°) une erreur est détectée.

3°) UN "NO AM FOUND" a lieu si un numéro de secteur non autorisé est accédé, si l'identification d'enregistrement (ID) manque d'être comparé sans qu'il y ait erreur ID DATA CHECK après deux (2) révolutions du disque, l'opération de lecture est terminée et l'état ID NOT FOUND sera présentera. Plusieurs lectures de secteurs sont automatiques si la ligne READ reste active.

READ ID : Cette commande lit les quatre (4) bytes de l'information piste et secteur au commencement de chaque secteur. Plusieurs lectures auront lieu si la ligne READ reste active. Un CRC CHECK est accomplie sur chaque champ identificateur (ID) et l'opération se termine par un état ID DATA CHECK si une erreur en résulte.

READ STATUS : Cette commande permet l'accès aux états supplémentaires du système. Le champ STATUS code de la commande est utilisé pour choisir une information des états spécifiques via le DATA BUS. Les codes des états 08,09 et 0A sont valables seulement si une erreur a lieu.

<u>SATUS CODE (HEX)</u>	<u>DATA BUS (Bits 7-0)</u>
00	adresse de piste de l'unité disque 0
01	" " " " 1
02	" " " " 2
03	" " " " 3
04- 07	disponible
08	dernier secteur exploité
09	calcul de byte restant (Résidual Byte Count).
0A	Nombre de secteurs lus / écrits
0B-FF	non affectés.

READ FORMAT : Cette commande assure l'accès au champ d'enregistrement protégé par un ID Field défectueux ou une adresse MARCK (AM) de l'enregistrement. Elle est aussi utilisée pour vérifier le format de piste. Quand un index est détecté, une lecture de toute information : DATA , GAPS , CRC et AM (exclusivement pour les 24 PREMIERS BYTES) est effectuée jusqu'à ce que la ligne READ devient inactive. Les 24 Premiers bytes peuvent être lus à la révolution suivante.

WRITE : Cette commande est lancée après une commande SEEK ou RESTORE. L'unité de disque s'oriente vers le numéro d'enregistrement écrit dans la commande et un profil de données FB indiquant la marque d'adresse de l'enregistrement est écrit de façon à précéder le champ d'enregistrement de données. L'opération d'écriture continue d'écrire les enregistrements (SECTORS) jusqu'à ce qu'un de ce qui suit a lieu.

1°) La ligne Write devient inactive. Si le secteur entier n'était pas écrit il est automatiquement rempli de ZEROS avant que le CRC soit écrit.

2°) une erreur est détectée

3°) un " NO AM FOUND " a lieu si numero de secteur non autorisé est accédé si l'identification d'enregistrement (ID) manque d'être comparé sans qu'il y ait erreur ID DATA CHECK après deux révolution du disque, l'opération d'écriture est terminée et l'état NO AM FOUND se présentera si la marque d'adresse de l'identificateur (ID) n'est pas détectée dans deux (2) révolutions du disque.

WRITE DELETED RECORD :

Identique à la commande Write sauf que : une marque d'adresse F 8 est écrite à la place de FB (voir la suite pour ces codes).

WRITE AND VERIFY.

Les données sont écrites comme déjà, d'écrites dans la commande Write. Après la fin de l'opération écriture, l'unité de disque s'oriente au début du secteur de cette commande et accomplit un controle CRC sur tous les secteurs écrits par cette commande. L'opération de vérification est terminée par le premier secteur avec une erreur DATA CHECK, ou après vérification du dernier secteur écrit.

Si un NO AM FOUND est détecté durant l'écriture, l'exécution de l'opération de vérification n'a pas lieu.

WRITE FORMAT:

La commande Write Format est utilisée pour formater le disque selon le choix de l'utilisateur. Cette option requiert, de l'analyste, un choix de format identique à celui de IBM 3740 sauf pour le nombre de bytes de données dans le secteur et le nombre de secteurs; permettant ainsi un choix de 1(un) à 80 secteurs par piste.

Toutes les données à écrire, les GAPS, AM et CRC inclus sont transférés par le MPU. Quand une AM est en cours d'écriture, il est nécessaire que la ligne de commande soit mise à "1" durant une demande de donnée (DATA REQUEST). Ceci permet au formater de générer un profil binaire d'horloge approprié. L'écriture sur la piste est terminée par la detection de INDEX. Une fin normale a lieu quand l'état TRACK OVERFLOW se présente durant l'écriture du postamble.

.../..

Les pistes formatées peuvent être vérifiées par utilisation de la commande READ FORMAT. Après utilisation de la commande WRITE FORMAT, la commande RESTORE doit être émise avant la commande SEEK sinon la vérification ne se fera pas.

Les données de marque d'adresse codées qui suivent peuvent être utilisées.

FB	marque d'adresse de l'enregistrement effacé
FB	marque d'adresse de l'enregistrement
FC	" " " INDEX
FE	" " " ID (identificateur)

II) CALCOMP 1140 FORMATS (Formats du Formatter)

1°) FORMAT de l'IBM 3740

La (fig. (III.3)) montre le format de piste qu'on trouve sur les disques compatibles avec IBM 3740.

Le formatter permet à l'utilisateur d'écrire et lire des secteurs à longueurs unique ou multiple sans se soucier du positionnement des secteurs ou la détermination du nombre de secteurs désiré.

Deux (2) types d'enregistrements peuvent être faits sur le disque: un enregistrement normal et une écriture d'effacement (DELETED RECORD). Le formatter assure ces deux (2) types sans restrictions. Un effacement peut être écrit comme un enregistrement vraiment effacé ou bien la marque d'adresse "FB" d'effacement peut être utilisée pour simplement indiquer l'état et la position des enregistrements à considérer comme effacés. Quand un enregistrement effacé est lu par le formatter la lecture est terminée à la fin du secteur contenant la marque d'adresse de l'effacement. Le MPU est averti par la ligne d'état DELETED RECORD devenant active. Plusieurs lectures de longueur de secteur des effacements ne sont pas possible.

2°) REMARQUES: Si l'utilisation désire créer un format avec un nombre unique de bytes de données par secteur, il le peut.

La logique du Formatter permet à l'utilisateur de modifier le nombre de secteur et/ou le nombre de bytes par secteur; le Format devant rester identique à celui de IBM sauf pour le nombre de secteur et de bytes de données dans le secteur.

../...

../...

Il est possible de "FORMATTER" chaque piste par un(1) ou 80 secteurs un secteur unique par piste accroitra la capacité de stockage du floppy disk de 45%. Tous bytes en excés sont à introduire dans le postambule.

III) DESCRIPTION DE L'INTERFACE MPU/ FORMATTER.

Les lignes d'interfaces du système MPU/FORMATTER, sont constituées de:

- 16 lignes bi-directionnelles du DATA Bus
- 10 lignes du Formatter au MPU
- 5 Lignes du MPU au Formatter.

1°) Les lignes bi-directionnelles du DATA Bus.

Les lignes du DATA Bus de 0 à 15 sont bidirectionnelles pour les mots de commandes et le transfert de mot de données utilisant des données à niveau bas.

2°) Lignes du formatter au MPU.

T/R : (transmit/receive)

Cette ligne d'emission/reception indique la direction de transfert de données. Une fois la direction établie, elle reste ainsi pendant la durée de l'exécution du transfert de la commande présente.

- un niveau haut de (T/R) indique une émission de données du formatter.
- un niveau bas pour reception de données venant du MPU.

DATA IN : Quand la ligne T/R est "haute", ce signal devenant actif à niveau bas signifie qu'un mot de données venant du formatter est pret.

Quand la ligne T/R est "basse" , ce signal devenant actif à niveau bas **indiqué** une demande de données, du Formatter au MPU.

B U S Y : Ce signal est actif à niveau bas indiquant qu'un traitement de commande ou une remise à zéro continue.

STATUS FLAG : Ce signal est actif à niveau bas indiquant que les 16 bytes de l'état automatique sont différents de zéro.

../...

.../...

OVERRUN : Ce signal à niveau bas indiquant que l'interface des entrées/Sorties manqua d'émettre ou de recevoir les données durant la période de demande de communications de données. Cette situation termine une opération d'écriture ou de lecture. Les opérations de lecture terminent le transfert de données au MPU mais le Formatter reste occupé (BUSY) jusqu'à ce que le contrôle CRC soit accompli. Les opérations d'écriture mettent des zéros jusqu'à la fin du secteur et écrivent un CRC DATA.

DELETED RECORD : c'est un signal actif à niveau bas qui indique qu'une marque d'adresse (F8) d'un effacement a été rencontrée durant une opération de lecture. L'opération de lecture se terminerait à la fin de ce secteur (enregistrement effacé).

DISK CHANGE 0-3 : Quatre lignes actives à niveau bas, indiquant chacune un changement de disque d'une unité de disque bien spécifiée.

Ces niveaux sont remis à "1" après installations de l'unité de disque par une commande RESTORE.

3°) LES LIGNES DU MPU AU FORMATTER :

COMMAND : C'est un signal actif à niveau bas indiquant que le DATA BUS contient le "mot commande" durant un transfert de commande. Ce niveau est maintenu jusqu'à ce que BUSY devienne actif. Durant l'écriture de format (format Write), la ligne "COMMAND" indique qu'un des bytes devant être transférés contient une donnée AM (AM DATA).

READ : C'est un signal actif à niveau bas indiquant qu'une opération de lecture va être exécutée et continuée jusqu'à ce signal devient haut ou un INDEX a lieu.

WRITE : C'est un signal actif à niveau bas indiquant qu'une opération d'écriture va être exécutée et continuée jusqu'à ce que ce signal devient haut ou un INDEX a lieu.

DATA OUT : Quand la ligne (T/R) est "haute", ce signal devenant actif à niveau bas indique que MPU a accepté le mot de données.

Quand la ligne (T/R) est "basse", ce signal (DATA OUT) devenant actif à l'état bas indique que le MPU a un mot de données disponible.

../...

SYSTEM RESET: C'est une impulsion active à niveau bas, de largeur ininémale 100ms qui fait que BUSY soit établi et les microdiagnostics exécutés. Toutes les unités de disque correspondantes sont remises à la piste 00 (TRACK00), accordant un minimum de 500 μ s par unité de disque avant l'émission des commandes qui suivent la remise à zéro du système.

COMMAND TIMING: Ce que nous allons décrire, ce sont les "INTERFACE TIMING DIAGRAMES" pour les diverses commandes. Certains figures peuvent regrouper plus d'une commande si le; diagramme des durées est le même.

1°) SEEK/SELECT/RESTORE/SECTOR LENGTH:

Ces commandes (voir fig.III.4) ne lancent aucun transfert de données sauf la commande elle-meme. Elles ont lieu dans le cas d'état automatique disponible pour le MPU après achèvement.

Aussitot que la commande est placée sur le DATA BUS, la ligne de commande peut être établie (mise à 0V). Le formater repondra en établissant le signal BUSY (à 0V). Le temps maximum entre l'établissement de la commande par le MPU et celui de BUSY par le formater va être de 20 μ s. AUTOMATIC STATUS est placé sur le DATA BUS en un minimum de deux (2) microscondes avant que la ligne BUSY soit mise à un (1) (desactivée).

2°) READ STATUS:

Quand une information états supplémentaires après AUTOMATIC STATUS est requise, cette commande (fig.III.5) retrouve les états sur le bytes de poids faible (bits 0-7) du DATA BUS. Le MPU initialise cette commande comme suit:

- 1) établissement la ligne READ (à 0V)
- 2°) place le mot commande sur le DATA BUS
- 3°) établit la ligne COMMAND.

Dans 20 μ s au maximum, le formater repond par établissement de la ligne BUSY en un minimum de 2 μ s avant d'informer le MPU que le mot de données est pret pour le transfert, le formater s'assure que la ligne (T/R) est "haute" ce qui indique que le transfert arrivera du formater au MPU. Après que l'information des états est placée soit placée dans le byte de poids faible sur le DATA BUS, la ligne DATA IN est établie (à 0V).

Cela veut dire pour le MPU qu'un mot de données est présent sur le DATA BUS. Après que le MPU ait accepté ce mot, la ligne DATA OUT doit être mis à "1" dans 30 μ s. 2ms après la mise à "1" de DATA OUT, AUTOMATIC STATUS sera placé sur DATA BUS. BUSY sera mis à "1" (RESET) 2ms après.

.. / ...

WRITE :

Le MPU initialise cette commande (fig.III.6) comme suit:

- 1°) établit la ligne d'écriture (WRITE) (à 0V)
- 2°) Place le mot de commandes sur le DATA BUS
- 3°) établit la ligne COMMAND (à 0V).

En réponse le Formatter met $24 \mu s$ au maximum pour activer la ligne BUSY. En $2 \mu s$ au minimum avant d'envoyer une demande de données au MPU, la ligne T/R est mise à zéro, ce qui détermine qu'un transfert arrivera du MPU au Formatter. La ligne DATA IN est mise à zéro(0) pour demander des données du MPU. Ce dernier place les données sur le DATA Bus puis met à zéro (0) la ligne DATA OUT indiquant ainsi qu'un mot de données est disponibles.

DATA OUT doit être mise à zéro(0) en $24 \mu s$ au maximum après que la ligne DATA IN soit devenue active. Le formatter extrait les données du DATA BUS et continue la séquence d'établissement de liaison DATA IN /DATA OUT jusqu'à ce que le transfert de la dernière donnée arrive. Pour éviter l'encombrement des données, DATA OUT doit arriver en $24 \mu s$ au maximum après que DATA IN ait été mise à zéro(0). DATA OUT doit être remise à 1 en $30 \mu s$ après réception de DATA IN. La ligne WRITE peut être remise à 1 dès la mise à 0 de DATA IN pour le transfert de la dernière donnée.

T/R peut changer d'état en un minimum de $2 \mu s$ après la dernière transition de remise à 1 de DATA OUT. Après que l'état automatique ait été placé sur le DATA BUS, le formateur remet BUSY à 1 en $2 \mu s$.

4°) READ : Cette commande (fig.III.7) est initialisée comme suit:

- 1°) mettre à zéro la ligne READ
- 2°) placer le mot des commandes sur le DATA BUS
- 3°) Remettre à zéro la ligne COMMAND.

Le formatter répond, en $20 \mu s$ au maximum, en activant la ligne BUSY.

Avant d'informer le MPU qu'un mot de données est prêt pour le transfert, le formatter assure, en $2 \mu s$ au minimum la mise à 1(un) de la ligne T/R pour indiquer qu'un transfert aura lieu du formatter au MPU. Le formatter place les données sur le DATA BUS puis après $2 \mu s$ au moins mettra la ligne DATA IN à zéro. Ceci signifiera pour le MPU qu'un mot de données est présent sur le DATA BUS.

Après que le MPU ait accepté ce mot de données, la ligne DATA OUT serait mise à zéro. DATA OUT doit être mise à zéro(0) en $24 \mu s$ pour éviter que les données ne débordent, puis remise à 1(un) après $30 \mu s$. une fois DATA OUT remise à un(1) et après $2 \mu s$ au minimum le formatter extrait les données du DATA BUS. L'établissement de la liaison entre DATA OUT cessera quant le dernier mot de données est transféré.

.. / ..

L'état automatique étant placé sur le DATA BUS, 2 μ s après, le formater remet la ligne BUSY à 1.

5°) WRITE FORMAT: Cette commande (fig.III.8) écrit format désiré sur la piste choisie en commençant par l'index.

Le MPU doit transmettre toute l'information sur les GAPS, les marques d'adresses et les mots de données la commande est initialisée comme suit:

- 1°) mettre à 0 la ligne WRITE
- 2°) placer le mot de commande sur le DATA BUS
- 3°) mettre à 0 la ligne COMMAND.

Le formater répond en 20 μ s en activant la ligne BUSY. Deux(2) microscondes au minimum avant d'envoyer une demande de données au MPU, la ligne T/R est mise à un état bas ce qui indique qu'il y aura un transfert du MPU au formater. DATA IN est mise à 0, pour demander des données du MPU. Ce dernier, place les données sur le DATA BUS puis met la ligne DATA OUT à zéro pour indiquer qu'un mot de données est disponibles. Les données sont alors extraites du DATA BUS par le formater.

La ligne WRITE doit être remise à 1 après la mise à zéro de la ligne DATA IN pour le dernier transfert. Le formater continue l'écriture du format jusqu'à ce qu'il rencontre une impulsion d'index. L'établissement de liaison DATA IN/DATA OUT est identique à celle de la commande WRITE sauf quand la marque d'adresse, est transférée quand la marque d'adresse est placée sur le DATA BUS, la ligne COMMAND est mise à 0 au lieu de la ligne DATA OUT. Ceci indique au formater que la donnée présentée est une marque qui demande une combinaison de bits manquants d'adresse.

6°) READ ID: TIMING indentique à celui de la commande READ. Pour chaque secteur Les quatre bytes de l'information identificateur piste/secteur (TRACK/SECTOR ID) sont présentés au MPU aussi longtemps que la ligne READ reste active.

7°) WRITE AND VERIFY. Sont timing est identique à celui de la commande WRITE. Le retard entre le dernier mot données transféré et la présentation de l'état automatique est plus grand parce qu'avant de présenter l'état automatique, le formater fait un autre tour du disque pour lire et contrôler le CRC.

8° READ FORMAT: son timing identique à celui de la commande READ. Cette commande lit la piste entière en commençant par la marque d'adresse de l'index ou bien lit jusqu'à ce que la ligne READ soit remise à 1 (un).

9°) AUTOMATIC STATUS : Quand un état automatique (Automatic Status) différent de zéro est placé sur le DATA BUS (fig.III.9) la ligne STATUS FLAG est mise à zéro. Après 2 μ s au moins, la ligne BUSY est remise à 1 (un).

Permettant au MPU de prendre l'information d'état du DATA BUS avant d'envoyer une autre commande utilisant le même niveau que la ligne READ ou WRITE, il est nécessaire de mettre à 1(un) puis remettre à 0(zéro). Ceci doit se passer 2 μ s au moins avant la mise à "0" de la ligne COMMAND. La ligne STATUS FLAG est maintenu à "0" par le formater un maximum de 4 μ s après que la ligne COMMAND soit établie (OV).BUSY sera mise à "0" seulement après un 2 μ s au moins après que STATUS FLAG ait été remis à 1(un).

SELECT	15	13	12	11	10	8	7	0						
	0	0	0	0	0	DEV. NO.	X	X	X	X	X	X	X	X
SEEK	15	13	12	11	10	8	7	0						
	0	0	1	0	0	DEV. NO.	TRACK NUMBER							
RESTORE	15	13	12	11	10	8	7	0						
	0	1	0	0	0	DEV. NO.	X	X	X	X	X	X	X	X
SECTOR LENGTH	15	13	12	0										
	1	1	1	SECTOR LENGTH										
READ	15	13	12	11	10	8	7	0						
	0	0	0	1	0	DEV. NO.	INITIAL RECORD NO.							
READ ID	15	13	12	11	10	8	7	0						
	0	0	1	1	0	DEV. NO.	X	X	X	X	X	X	X	X
READ STATUS	15	13	12	11	10	8	7	0						
	0	1	0	1	0	X	X	X	STATUS CODE					
READ FORMAT	15	13	12	11	10	8	7	0						
	1	0	0	1	0	DEV. NO.	X	X	X	X	X	X	X	X
WRITE	15	13	12	11	10	8	7	0						
	0	0	0	0	1	DEV. NO.	INITIAL RECORD NO.							
WRITE DELETED RECORD	15	13	12	11	10	8	7	0						
	0	0	1	0	1	DEV. NO.	INITIAL RECORD NO.							
WRITE AND VERIFY	15	13	12	11	10	8	7	0						
	0	1	0	0	1	DEV. NO.	INITIAL RECORD NO.							
WRITE FORMAT	15	13	12	11	10	8	7	0						
	1	0	0	0	1	DEV. NO.	X	X	X	X	X	X	X	X

Figure III System Command Format

Bit Position	DESCRIPTOR	DEFINITION
0	SEEK COMPLETE	• COMMANDE DE RECHERCHE Accomplie AVEC SUCCES.
1	ID DATA CHECK	• CRC NON COMPARE dans ID Field
2	PROGRAM ERROR	• COMMANDE NON-VALABLE.
3	SYSTEM ERROR	• Lignes Ecriture et lecture actives simultanément.
4	DATA CHECK	• le controle CRC echoua sur le champ de données.
5	ID NOT FOUND	• la recherche de ID Field a echoué après deux revolutions
6	EQUIPEMENT CHECK	• Les microdiagnostics internes ont echoué.
7	(NOT USED)	(Ce bit n'est pas utilisé)
8	WRITE PROTECT	• le Drive selectionné est protégé contre l'écriture.
9	NO AM FOUND	• un des 3 types d'address marks (ID, INDEX, or DATA) n'a pas été trouvé.
10	TRACK OVERFLOW	• INDEX a été detecté pendant que la ligne Write ou READ est active.
11	DRIVE NOT READY	• le Drive choisi n'est pas prêt.
12	SEEK ERROR	• après l'achevement de la commande Seek, l'adresse de piste n'a pas été comparée.
13	RECALIBRATE ERROR	• Apres la commande Restore, le signal TRACK 00 n'est pas detecté.
14	DEVICE CHECK	• TRACK 00 non detecté. Durant une commande Write format, le code d'état seul indique une vitesse assez faible du disque. S'il arrive avec TRACK OVERF. il indique une vitesse assez élevée.
15	WRITE ERROR	• INDEX a été detecté pendant que le courant d'écriture a été envoyé.

Figure III 2 Automatic Status Error Code

Bit Position	DESCRIPTOR	DEFINITION
0	SEEK COMPLETE	• COMMANDE DE RECHERCHE Accomplie AVEC SUCCES.
1	ID DATA CHECK	• CRC NON COMPARE dans ID Field
2	PROGRAM ERROR	• COMMANDE NON-VALABLE.
3	SYSTEM ERROR	• Lignes Ecriture et Lecture actives simultanément.
4	DATA CHECK	• le controle CRC echoua sur le champ de données.
5	ID NOT FOUND	• la recherche de ID Field a echoué après deux revolutions
6	EQUIPEMENT CHECK	• Les microdiagnostics internes ont echoué
7	(NOT USED)	• Ce bit n'est pas utilisé
8	WRITE PROTECT	• le Drive selectionné est protégé contre l'écriture.
9	NO AM FOUND	• un des 3 types d'address marks (ID, INDEX, or DATA) n'a pas été trouvé.
10	TRACK OVERFLOW	• INDEX a été detecté pendant que la ligne Write ou READ est active.
11	DRIVE NOT READY	• le Drive choisi n'est pas prêt.
12	SEEK ERROR	• après l'achèvement de la commande Seek, l'adresse de piste n'a pas été comparée.
13	RECALIBRATE ERROR	• Apres la commande Restore, le signal TRACK 00 n'est pas detecté.
14	DEVICE CHECK	• TRACK 00 non detecté. Durant une commande Write Format, le code d'état seul indique une vitesse assez faible du disque. s'il arrive avec TRACK OVERF. il indique une vitesse assez élevée.
15	WRITE ERROR	• INDEX a été detecté pendant que le courant d'écriture a été envoyé.

Figure III 2 Automatic Status Error Code

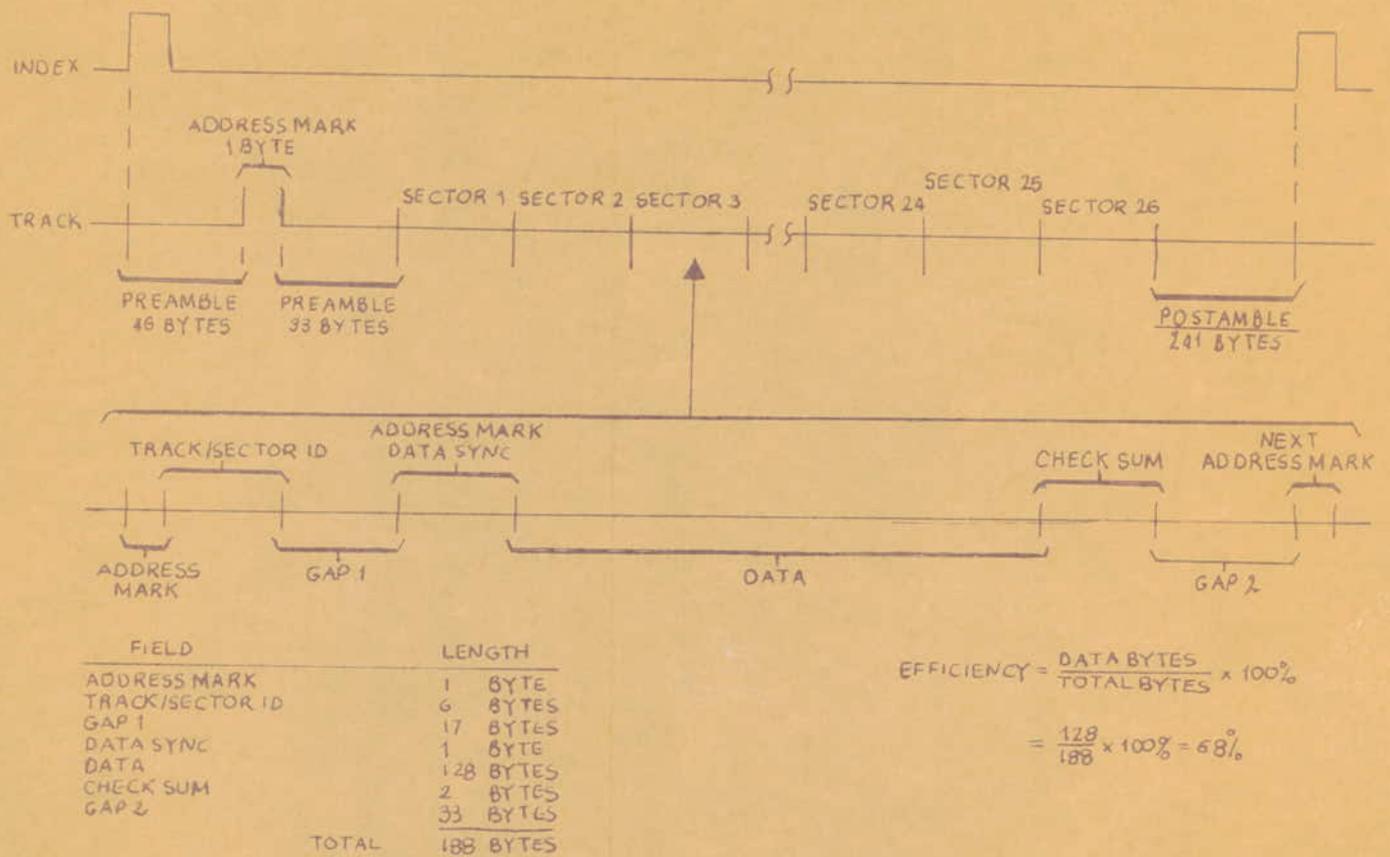


Figure III-3 IBM Track Format

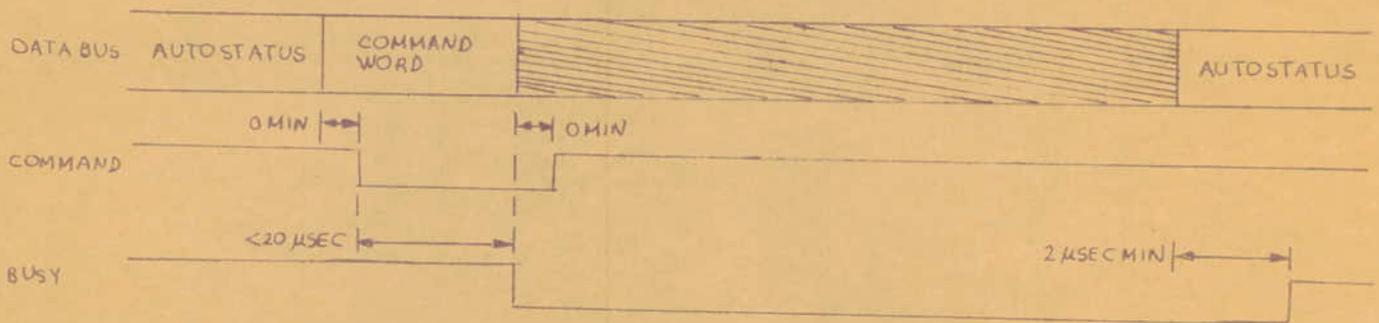


Figure III-4 Select/Seek/Restore/Sector Length Timing

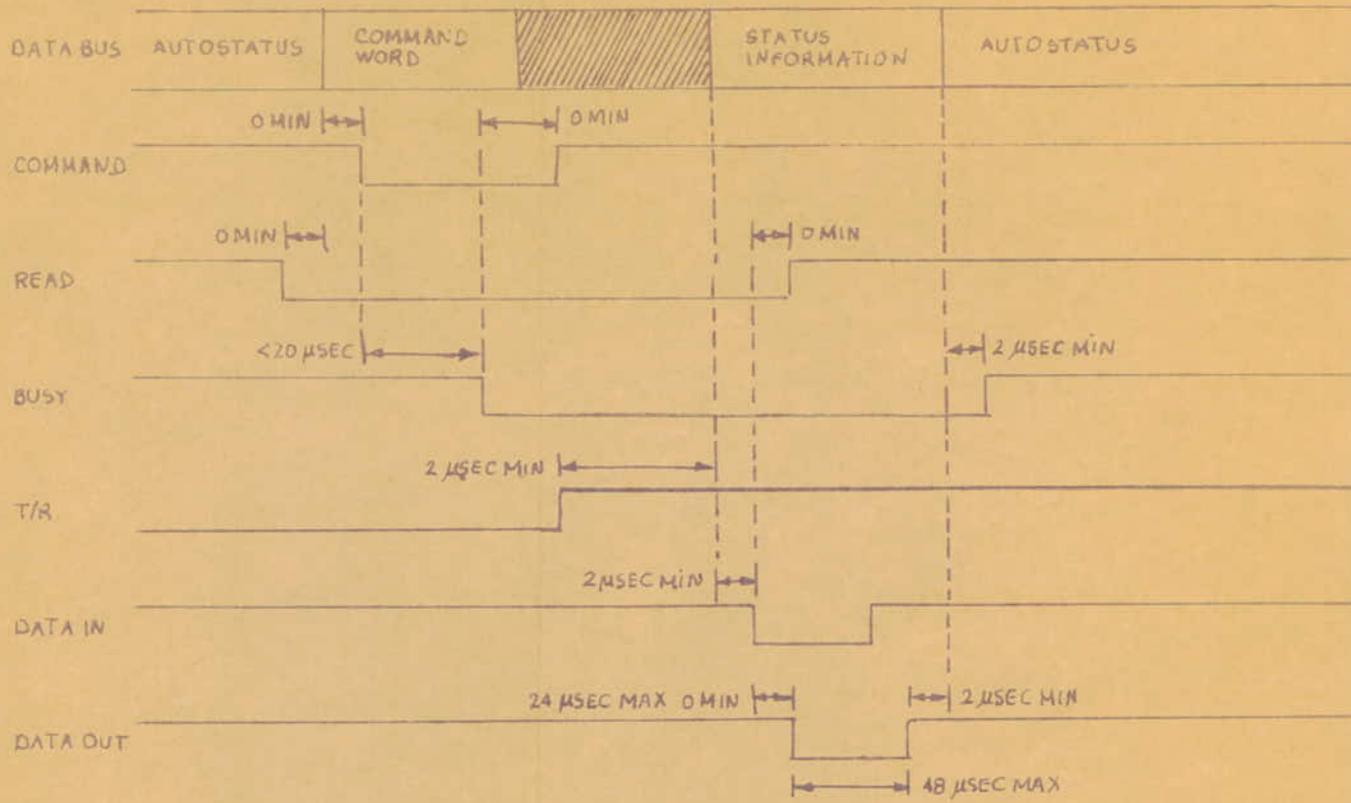


Figure III 5 Read status Timing

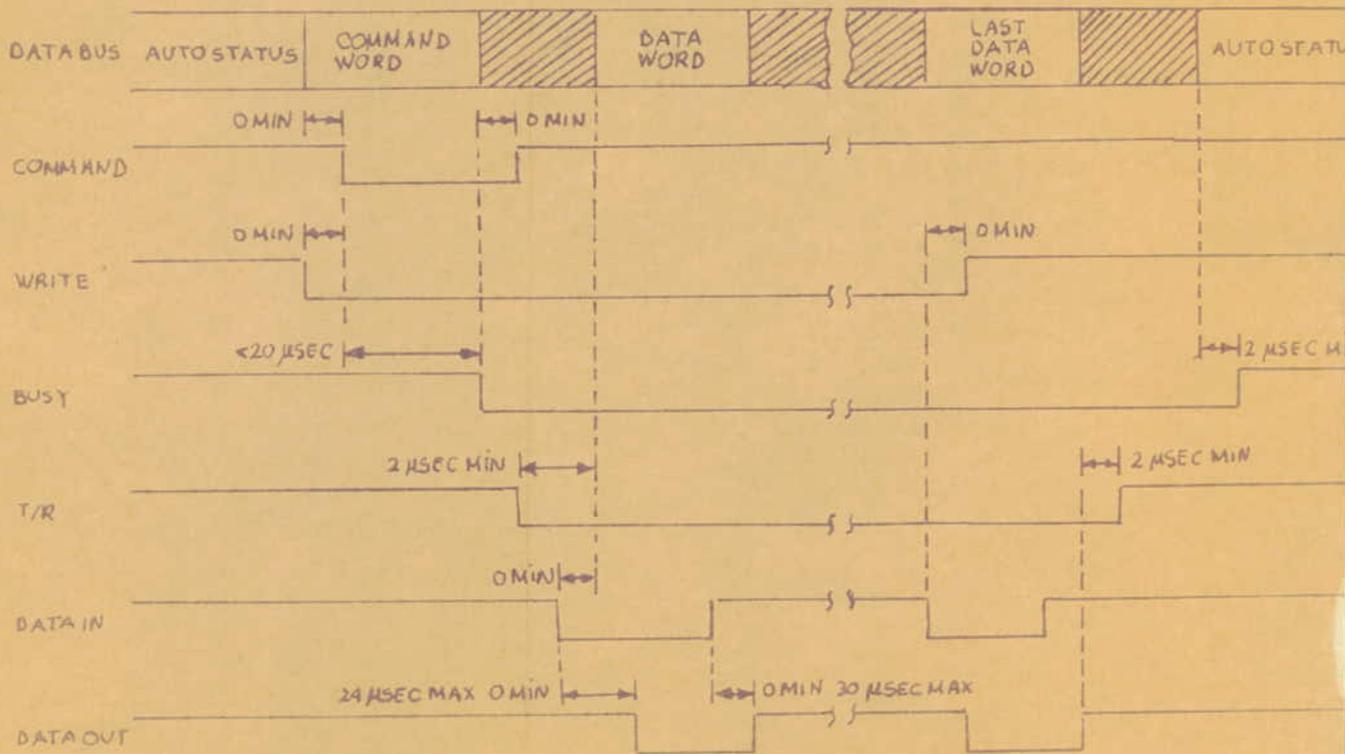


Figure III 6 Write Timing

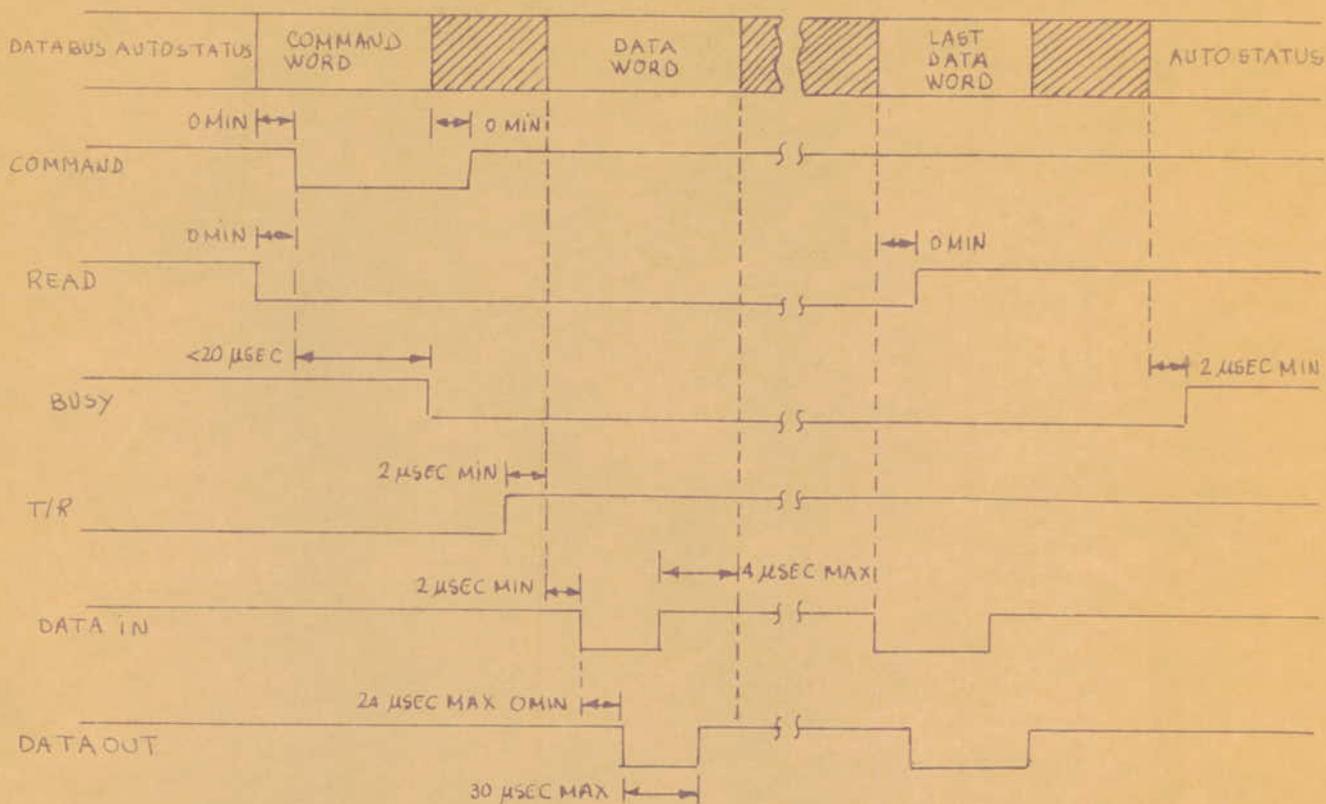


Figure III 7 Read Timing

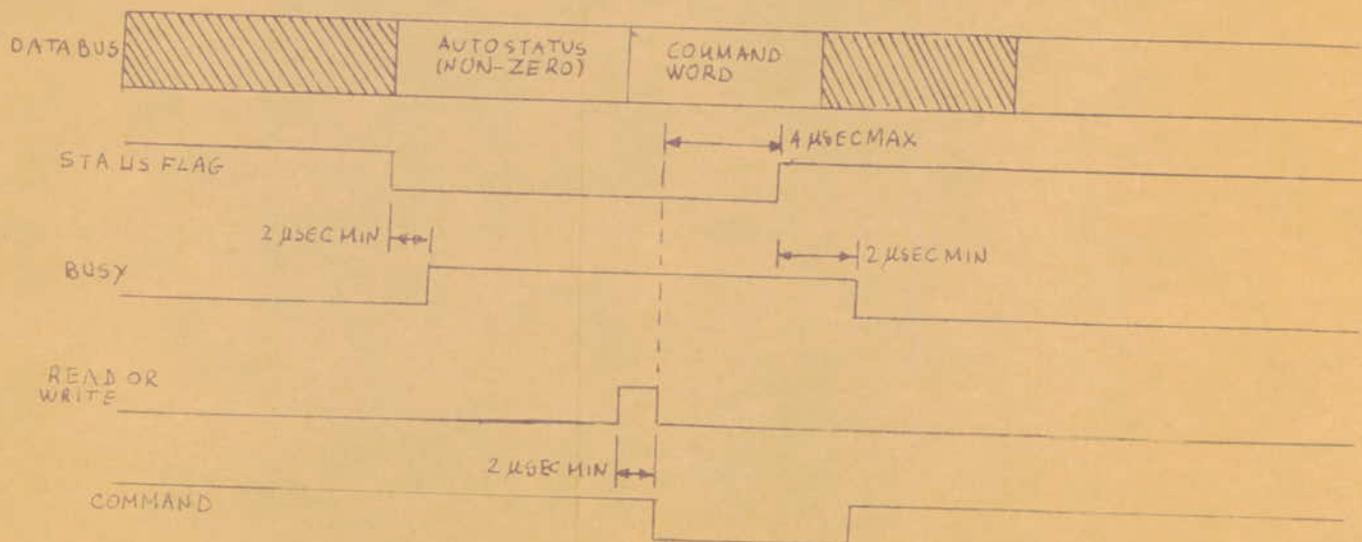


Figure III 9 Automatic status Timing

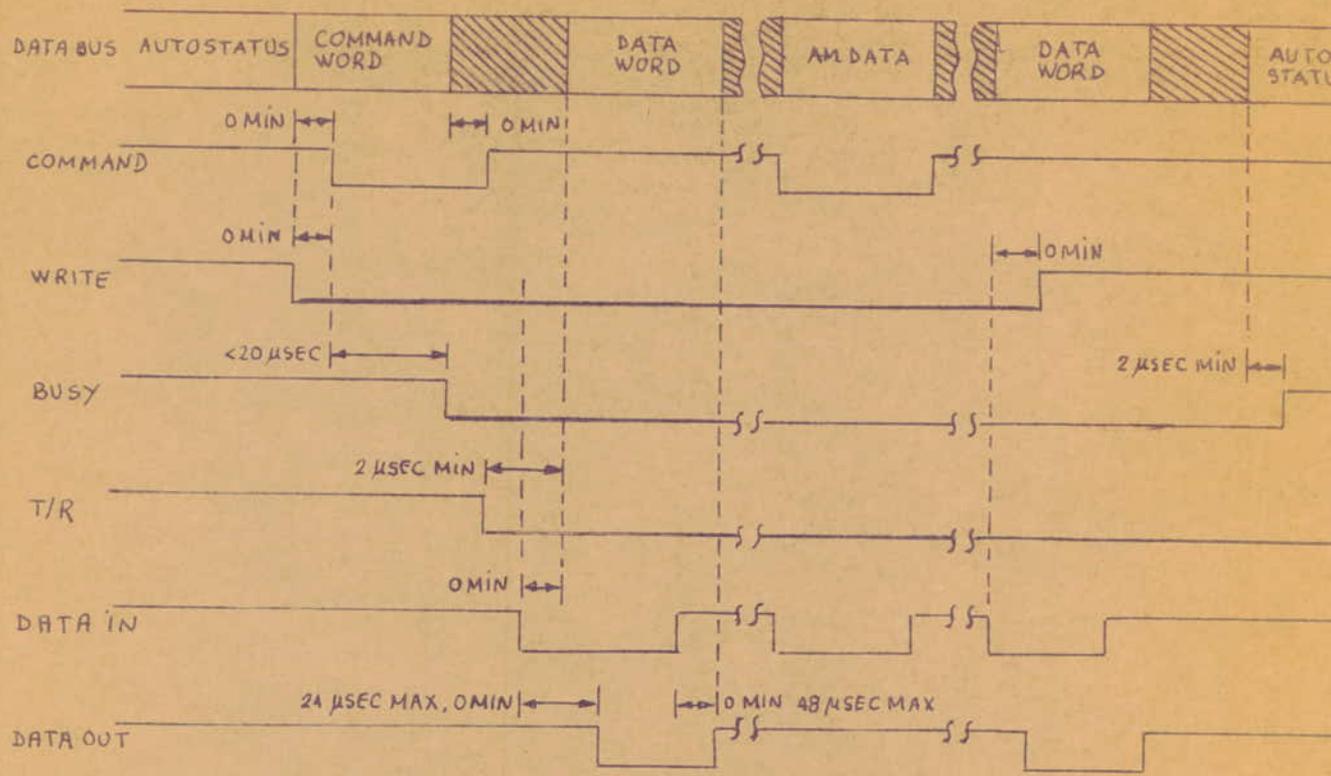


Figure III B Write Format Timing

CHAPITRE IV.FLOPPY DISK DRIVE

L'unité d'entraînement du disque est un dispositif avec mémoire rapide et à accès aléatoire utilisé pour le stockage des données dans un système de traitement de données (MPU).

Chaque unité d'entraînement du disque du type CALCOMP 140 DISKETTE DRIVE possède une interface qui permet l'écoulement des signaux d'états, de contrôle et de lecture/écriture entre l'unité de disque et le contrôleur. Le débit de 250 KHZ permet un échange rapide de données entre l'unité de disque et le formater. Ce dernier peut être utilisé pour "interfacer" jusqu'à quatre unités de disques soit en radical soit en daisy-chain. Pour notre cas, nous avons opté pour la liaison en daisy-chain (voir fig.IV.0) car elle nécessite moins de HARDWARE.

I DESCRIPTION DU DISQUE SOUPLE.

1°) Définition: C'est un disque flexible ayant une face seulement revêtue d'oxyde magnétique et utilisée comme support d'enregistrement dans les unités d'entraînement de disque.

2°) CONSTITUTION : Le disque souple (fig.IV.1) est protégée en permanence par une pochette en plastique qui comporte les couvertures suivantes:

a) Ouverture de guidage : ouverture circulaire au centre qui permet au cône de centrage d'y pénétrer pour centrer et bien saisir le ^{pivot} ~~axe~~ contre le disque afin de le faire tourner à l'intérieur de la pochette (pour écriture ou lecture)

b) Ouverture R/W: quand le disque est mis en charge, la tête de lecture/écriture (R/W) est mise en contact avec le disque par cette ouverture.

c) Trou Index/secteur: Ces trous (fig.B) existent sur le disque et sont détectés comme Index ou Secteur.

Impulsion Index: développé par le trou Index et est utilisée pour indiquer la vitesse de rotation du disque et marque le début de piste.

Impulsions Secteurs: 32 par piste, elles sont développées par les trous secteurs. Les trous sont espacés d'une manière égale dans le disque et sont utilisés pour diviser chaque piste en portions égales pour le format de l'enregistrement (fig.C) - Trou de protection contre l'écriture.

La donnée est enregistrée en magnétisant les bits digitaux dans les cercles concentriques (pistes) sur la surface revêtue d'oxyde magnétique. La surface d'enregistrement (fig.C) comprend 77 pistes espacées de 0,020 INCH l'une de l'autre, et numérotées de 00 à 76 en commençant de l'extérieur.

../. .

Plus de 3,2 millions de bits de données peuvent être stockés sur une surface unique du disque souple. Quant à l'utilisation de la présentation de données relative à l'IBM 3740, 1,94 millions de bits de données peuvent être enregistrés.

II. ORGANISATION DE L'INFORMATION:

L'enregistrement sur le disque souple se fait de façon série dont le formater assure la conversion parallèle/série ou l'inverse pour une lecture.

1°) FORMAT D'enregistrement: Le format de l'information enregistrée sur le disque est totalement fonction du contrôleur.

Le mode d'enregistrement utilisé est par modulation de fréquence c'est à dire, à chaque bit de donnée enregistré sur disque est associé un bit d'horloge enregistré avec lui.

2°) BIT CELL. Les bits d'horloge et de données sont intercalés. Par définition un bit Cell est la période entre le front montant d'un bit d'horloge et le front montant du bit d'horloge suivant:

3°) BYTE : Quand nous parlons de données série (étant écrites sur ou lues du disque), un byte est défini comme (huit) 8 bits cells consécutifs. Le bit cell le plus significatif est défini comme étant le bit cell 0 et le moins significatif le bit cell 7.

Durant une opération d'écriture. Le bit cell 0 de chaque byte est transféré en premier à l'unité de disque puis le bit cell 7 en dernier. De même pour les bytes de données, le plus significatif est transféré en premier et le moins significatif en dernier. Pour une lecture, le bit cell 0 de chaque byte sera transféré en premier, et en dernier le bit cell 7. De même pour les bytes. La figure IV.2. illustre le rapport des bytes pour la lecture et l'écriture des données.

4°) FORMAT DE PISTE: Les pistes peuvent être formatées de plusieurs manières et selon l'utilisateur (voir CH.III).

5°) FORMAT D'ENREGISTREMENT DE L'INDEX: Dans ce format, l'utilisateur doit enregistrer un long et unique enregistrement ou plusieurs petits enregistrements. Chaque piste commence par une impulsion d'index physique et chaque enregistrement est précédé par un seul identificateur (ID) enregistré.

6°) FORMAT D'ENREGISTREMENT SECTEUR: L'utilisateur peut enregistrer jusqu'à 32 secteurs par piste. Chaque piste est commencée par une impulsion index et chaque secteur par une impulsion secteur.

../. .

.../...

7°) LES GAPS : Chaque champ sur la piste est séparé des champs adjacents par un nombre de bytes ne contenant pas de bits de données. Ces espaces sont appelés des GAPS et servent à mettre à jour un champ sans affecter les champs adjacents.

8°) LES MARQUES D'ADRESSES (ADDRESS MARKS):

a) Index adresse MARK : (fig.IV.3) Elle est située au début de chaque piste et est un nombre fixe de bytes.

b) ID ADDRESS MARK: Le byte de la marque d'adresse de l'identificateur est situé au début de chaque champ identificateur sur le disque (voir fig.IV.4.)

c) DATA ADDRESS MARK: Elle est située au commencement de chaque champ de données non effacées sur le disque.

Le profil binaire pour cette marque d'adresse est représenté en figure IV.5.

d) DELETED DATA ADDRESS MARK: Le byte de la marque d'adresse de données effacées est localisé au début de chaque champ de données effacées sur le disque son profil binaire est montré. en figure IV.6.

9°) CYCLIC REDUNDANCY CHECK (CRC): chaque champ écrit sur le disk est augmenté de deux (2) bytes (CRC). Les deux bytes sont générés à partir d'une permutation cyclique des bits de données en commençant par le bit zéro de la marque d'adresse et finissant avec le bit sept du dernier byte contenus dans le champ (les bytes du CRC exclus). Quand un champ est relu d'un disque, les bits de données (du bit zéro de la marque d'adresse au bit sept du second byte du CRC) sont divisés par le meme générateur polynomial.

Un reste différent de 0 indique qu'il y a une erreur dans la relecture des données du disque, tandis qu'un reste égal à zéro indique que les données ont été correctement relues du disque.

III. POSITIONNEMENT DE LA TETE DE LECTURE/ECRITURE:

La logique de positionnement de la tete de lecture/ecréture place cel de tete vers une piste comme réponse au controleur en emettant trois signaux:

- DIRECTION : Ce signal donne la direction du mouvement relati à la position de la piste actuelle.

- STEP: Ces impulsions déterminent, grace au numéro de piste reçu, le nombre de pistes.

- SELECT: Ce signal met l'unité de disque en liaison directe avec le controleur .

La vitesse de saut est de 167 pistes/seconde (6ms/piste).Et il faut ajouter 30 ms après le dernier saut pour la stabilisation du système de positionnement.

..//..

../...

Aussi le temps d'accès est exprimé par six millisecondes(6ms) fois le nombre de piste, plus 30 ms. L'impulsion STEP peut avoir une largeur d'impulsion de 200 nanosecondes au minimum (2ms au maximum) tandis que le signal DIRECTION doit être stable pendant au moins 100 nanosecondes avant la réception de la première impulsion STEP.

IV ESPACES FONCTIONNELS DU DRIVE:

Chaque unité de disque contient trois espaces fonctionnels permanents:

- Le système de contrôle
- Le système de positionnement
- Le système de lecture/écritures.

1°) LE SYSTEME DE CONTROLE: Il constitue l'ensemble des circuits d'interface entre le contrôleur et le floppy disk drive. Ce système surveille continuellement les états d'exploitation de l'unité du disque. Le contrôleur accède à l'unité de disque pour un traitement " en direct" avec le MPU en activant une seule ligne SELECT. Les commandes sont ensuite reçues et exécutées par l'unité de disque choisie.

a) EXECUTION DE LA COMMANDE:

Les commandes sont reçues par l'unité de disque sous forme d'un ou de plusieurs signaux d'interface qui désignent une opérations suivantes:

SELECT: met l'unité de disque en liaison directe avec le contrôleur.

STEP: déplace la tête de lecture/écriture d'une piste à une autre (dans les deux sens) selon l'état logique de DIRECTION.

DIRECTION: Détermine la direction du mouvement de la tête quand l'impulsion STEP est reçue.

DIRECTION : "1" déplacement de la tête vers le centre

DIRECTION : "0" déplacement de la tête vers l'extérieur.

HEAD LOAD: place la surface d'enregistrement du disk en contact avec la tête de lecture/écriture.

WRITE : Valide le circuit d'écriture et bloque la sortie de lecture

READ : Valide les sorties de lecture et empêche l'écriture.

../...

..//...

Les instructions SELECT et HEAD LOAD doivent précéder une opération de lecture ou d'écriture. SELECT valide les portes entrée/Sortie, tandis que HEAD LOAD permet l'écriture ou la lecture sur le floppy disk. STEP et DIRECTION déplacent la tête de lecture/écriture vers ou bien l'une plus haute ou une plus basse position de piste. La progression peut arriver à une vitesse de 6 ms par piste ou plus lentement. Dès le positionnement de la piste relative, le contrôleur maintient la position de la piste actuelle et génère le nombre d'impulsions nécessaires pour atteindre à une nouvelle position de piste. Une fois positionné, le contrôleur désigne une opération de lecture ou d'écriture.

Dans une opération d'écriture, l'unité de disque enregistre les données présentées par le contrôleur. Dans une opération de lecture, la donnée est décodée par l'unité de disque puis acheminée vers le contrôleur.

b) Analyse des états:

Cinq signaux des états de l'unité de disque sont immédiatement envoyés aux lignes I/O quand le contrôleur choisit une unité de disque.

WRITE PROTECTED (Facultatif)

TRACK 00 : la tête de lecture /écriture est placée sur la piste 00.

INDEX : commencement d'une piste.

READY : Non envoyé avec SELECT et signifie que l'unité de disque est opérationnelle

SECTOR (facultatif): début d'un secteur.

READY et WRITE PROTECTED sont des niveaux fixes.

READY indique qu'une unité de disque est chargée et qu'elle est prête pour opérer.

WRITE PROTECT signifie que les données à écrire ne peuvent pas être enregistrées sur le floppy disk.

Une impulsion d'index arrive une seule fois par tour du disque.

L'état TRACK 00 est disponible pour initialiser le registre des adresses de piste du contrôleur.

2°) SYSTEME DE POSITIONNEMENT: Le système de positionnement répond aux impulsions de progression.

(STEP COMMAND) reçues du contrôleur en déplaçant la tête de lecture/écriture d'une position de piste par impulsion.

..//...

.../...

3°) SYSTEME DE LECTURE/ECRITURE: Le système de lecture/écriture enregistre des données codées durant une opération d'écriture et récupère ces données durant une opération de lecture.

Le signal WRITE ENABLE du contrôleur désigne une opération de lecture quand il est au niveau haut ou une opération d'écriture quand il est au niveau bas.

Pour terminer ce chapitre sur les unités d'entraînement du disque, nous donnons un exemple d'opération d'écriture.

V. OPERATION D'ECRITURE.

1- INTERFACE DE L'OPERATION D'ECRITURE.

La fig.(IV.7) est un bloc-diagramme qui montre les principales fonction du formater utilisées dans une opération d'écriture.

Dans une opération d'écriture, le PIA 1 est programmé comme une poste de sortie. Le profil binaire d'horloge ainsi que celui des données sont tous les deux fournis par le système MC 6800.

Avant de commencer l'opération d'écriture, le signal ABOVE TRK 43 est mis à "1" à "0" par le programme. Si le programme fixe la location piste plus grande que 43, le signal est mis à "1" ABOVE TRK 43 est utilisé par quelques modèles d'unités de disque pour contrôler le courant d'écriture dans les pistes.

Une opération d'écriture commence par l'envoi du signal ENABLE WRT actif au formater. Ce signal prépare les circuits du formater pour la réception des informations d'horloges et de données venant du PIA ~~1~~ 1. Il permet aussi au signal BYTE REQUEST de déclencher l'entrée CA1 du PIA ++1 une fois tous les 8 bits.

La donnée série est envoyée à l'unité de disque quand le signal WRITE GATE est rendu actif.

Dans une opération d'écriture, un gap (les bits d'un gap sont tous égaux à zéro) est écrit selon le format IBM 3740. Puis une marque d'adresse est écrite avec un profil binaire d'horloge spécial. La marque d'adresse est suivie de 128 bytes de données (DATA FIELD) ou d'un 4 bytes de données (ID Field) et de 2 bytes du CRC.

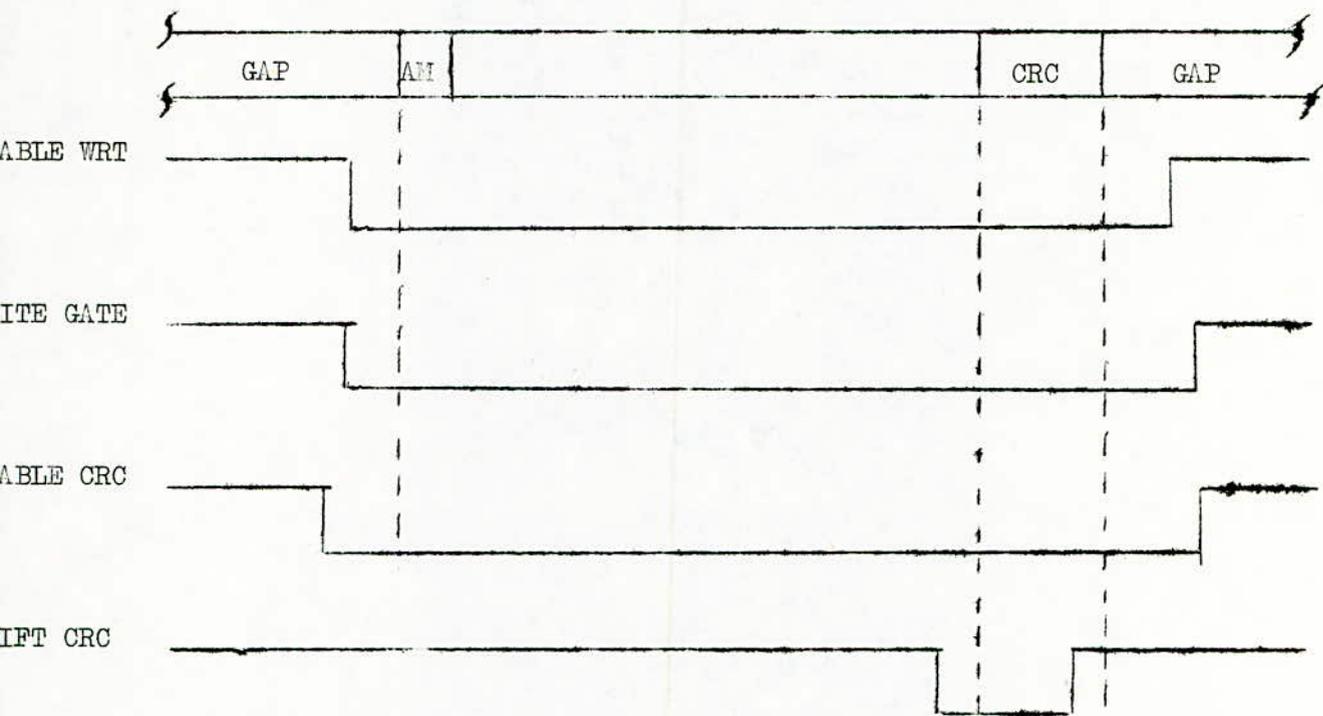
ENABLE CRC est rendu actif par le programme au moment de la marque d'adresse.

Après que le dernier byte des données ait été transféré par le programme, le signal SHIFT CRC est rendu actif. Puis il est remis à "1" le temps de deux(2) bytes après qu'il ait été mis à "0" Ceci fait que les deux(2) bytes du code CRC soient annexés aux données enregistrés.

.../...

../...

Le temps d'un byte après le second byte du CRC, toutes les lignes de controle d'écriture changent de niveau.



D'autres signaux de controle demandés à etre actif pendant une opération d'écriture sont SELECT et HEAD LOAD.

Le signal FILE INOP reset est utilisé pour effacer les états d'erreur détectés par la logique.

Le fonctionnement de CA1 et CA2 du PIA ++1 (BYTE REQUEST et R/W HANDSHAKE) est décrit au chapitre III.

Le signaux d'états qui décrivent la condition d'exploitation du système d'unités de disques durant l'opération d'écriture sont aussi expliqués dans le chapitre III.

2°) LOGIQUE D'ECRIURE DU FORMATTER.

La logique d'écriture du formatter est représentée en fig.IV.8, les diagrammes du timing pour le commencement et la fin d'une opération d'écriture sur les figures IV.9. etIV.10.

ENABLE WRT valide la logique d'écriture la remise à zéro du compteur binaire, en permettant la charge parallele aux registres à décalage d'horloge et de données, et en supprimant la remise à zéro du fli-flop de données séries.

../...

..//...
L'oscillateur d'écriture fait progresser le compteur binaire un décalage aux registres à décalage toutes les 4ms. Au moment où le compteur binaire est à 7 les registres à décalage sont chargés avec les données du PIA ++ 1. Notons que les registres d'horloge et de données sont déphasés de II. Le registre à décalage des données est chargé à la fin du temps du bit 7 Alors que celui de l'horloge est chargé à la mi-temps du bit 7.

La donnée série en format de modulation de fréquence digitale est générée par le flip-flop de données séries et une porte NOR. Le flip-flop est chronométré à deux fois le débit (en bits). Les profils de données et d'horloge séries sont introduits dans le flip-flop par demi-alternance de bits cells.

Après que le programme ait introduit les adresses dans le PIA++1, le signal ENABLE CRC est rendu actif-puis à la fin du temps du prochain bit 7, le générateur polynomial MC 8503 est validé. Un polynome de controle CRC est généré jusqu'à ce que le signal SHIFT CRC soit activé par le programme. A la fin du temps du bit 7, le caractère du CRC cumulé est envoyé en série dans le flip-flop de données séries à la place du profil binaire de données. Seize bits du CRC sont joints à la fin des données avant que le programme ne remet à zéro le signal SHIFT CRC. Le temps d'un byte après que le profil binaire du CRC ait été enregistré le programme remet à 1 ENABLE WRITE et WRITE GATE.

3°) LOGIQUE DE DETECTION D'ERREURS DU FORMATTER:

La logique de détection d'erreurs(fig.IV.11) interrompt les défaillances logiques qui pourraient surgir dans l'unité de disque ou dans la logique du formatter. Quand une condition d'erreur est interrompue, une interruption(R/W ERROR) est rendue actives vers le système MC 6800 via le PIA ++2. Les conditions sont:

- a) OVERRUN/UNDERFLOW
- b) DISK SYSTEM IN PERABLE
- c) NOT IN SYNC.

Le flip flop OVERRUN/UNDERFLOW est mis à "1" quand le programme d'écriture ne répond pas au dernier service demandé par BYTE REQUEST avant le prochain BYTE REQUEST (i.e. la donnée n'est pas regenerés dans le PIA++1.)

.../....

../...

Le signal DISK SYSTEM INOPERABLE est actif le signal d'états de l'unité de disque FILE OPERABLE ou quand WRITE ENABLE et WRITE PROTECTED déclenchent le loquet des erreurs (voir fig.IV.11).

La logique de détection d'erreurs est destinée telle que quand n'importe quelle condition d'erreur est détectée le signal R/W ERROR empêche le positionnement d'autres bascules à verouillage d'erreurs.

Quand une condition d'erreur est détectée, l'unité de disque vérifié que le signal WRITE GATE est invalidé pour empêcher tout nouveau manque de données.

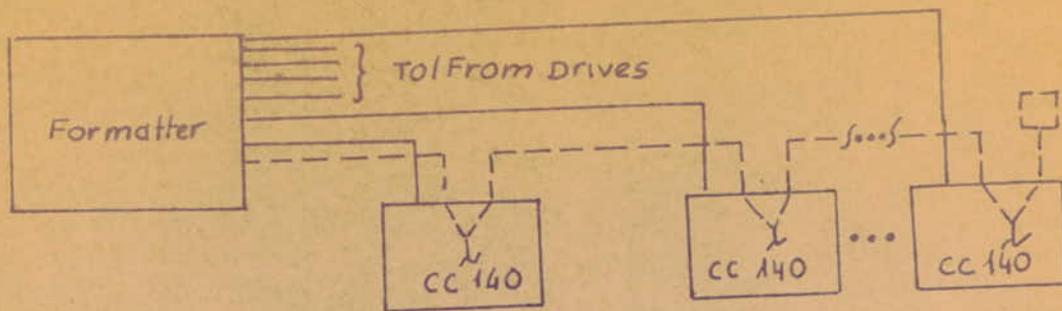


fig IV-0 Daisy-chain Interface

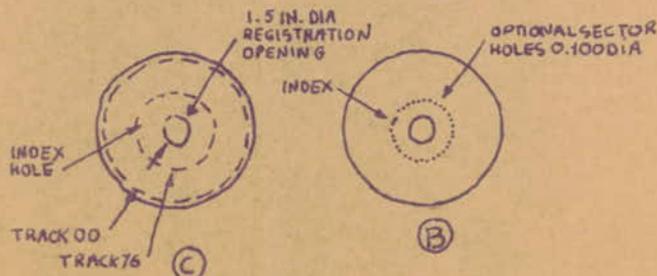
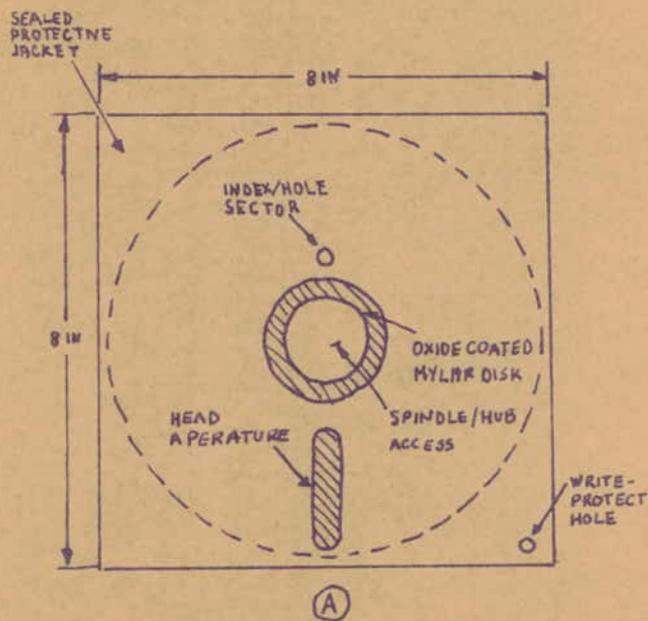


Figure IV-1 Flexible Disk Construction

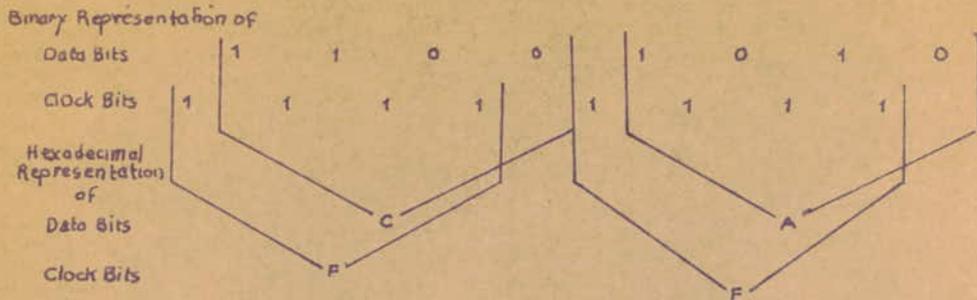
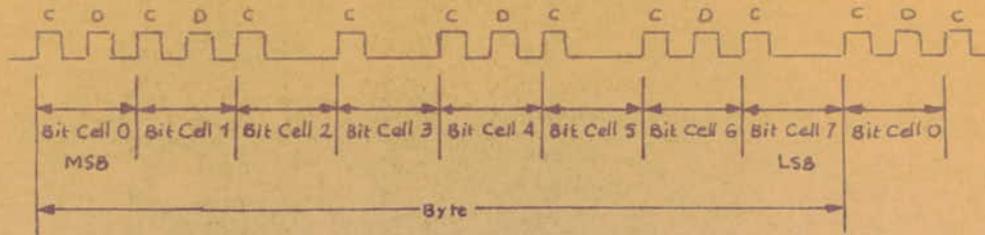


FIGURE IV 2 Data Bytes

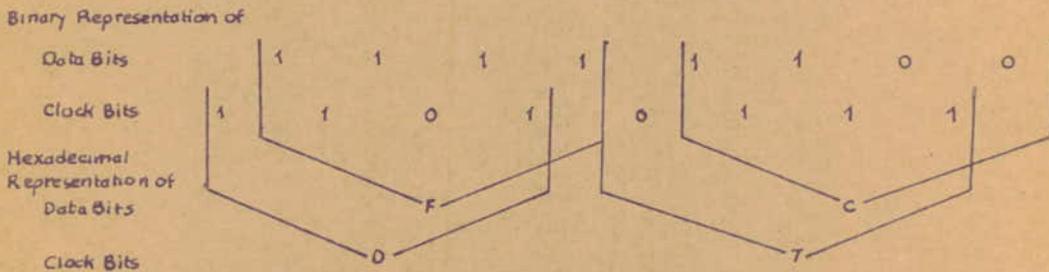
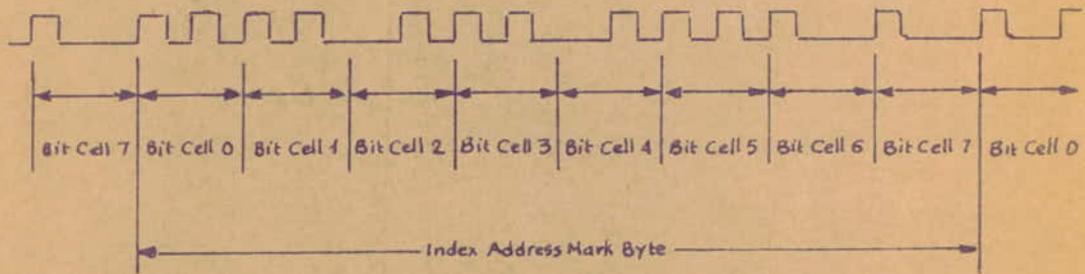


FIGURE IV 3 Index Address Mark

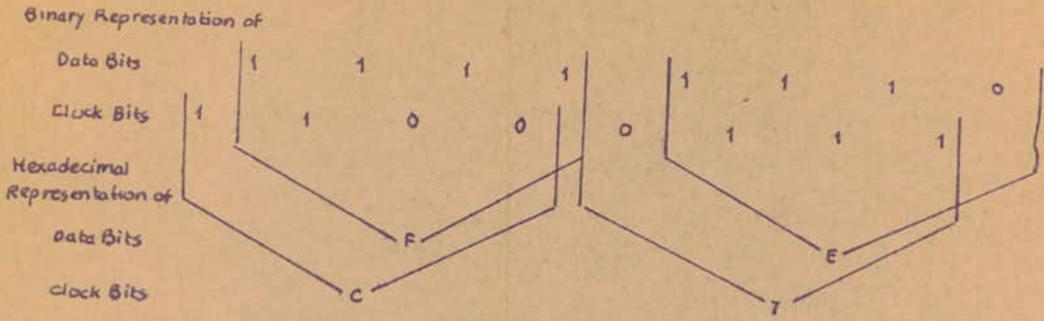
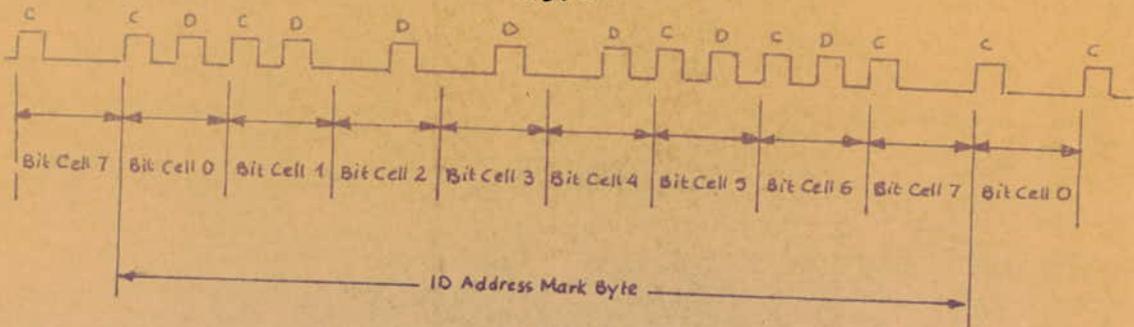
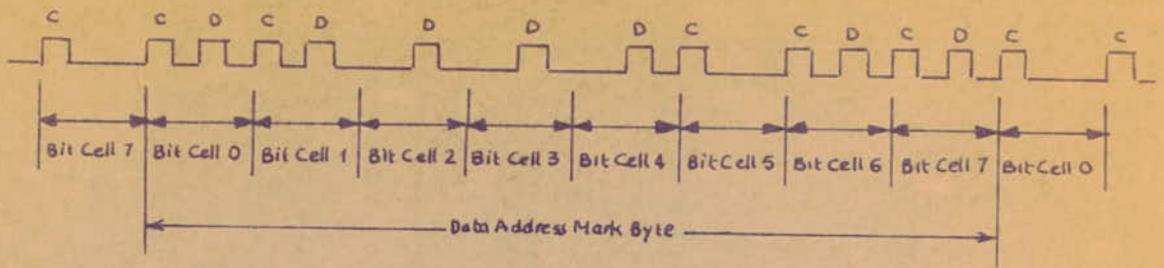


FIGURE IV 4 ID Address Mark



Binary Representations of

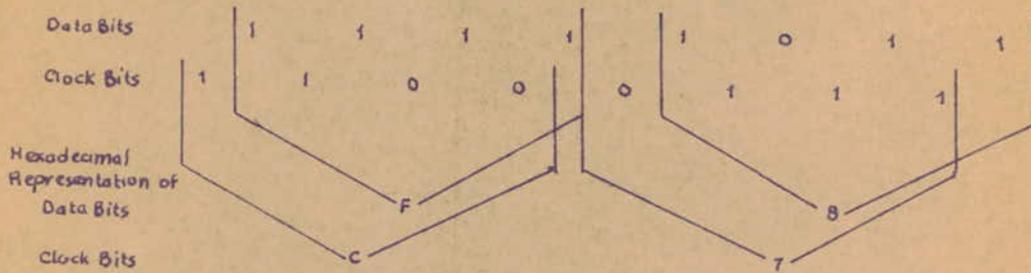
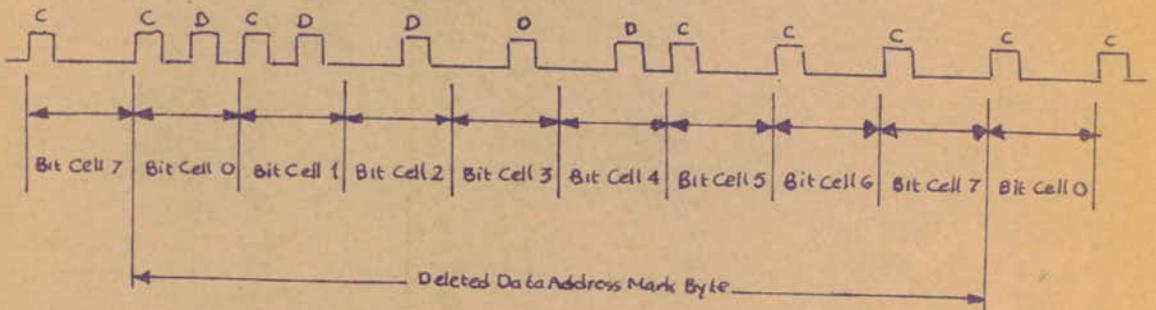


FIGURE IV 5 Data Address Mark



Binary Representation of

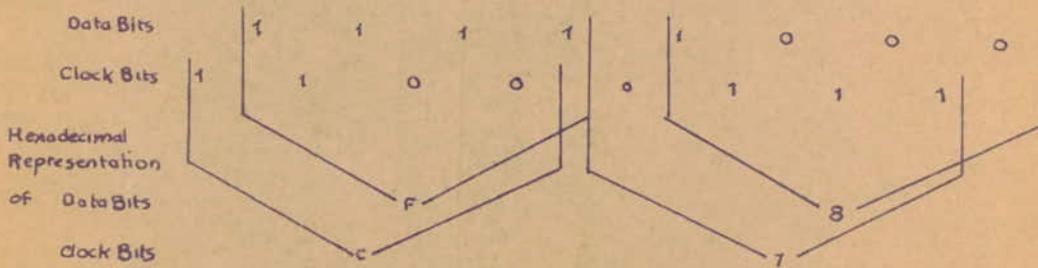


FIGURE IV 6 Deleted Data Address Mark

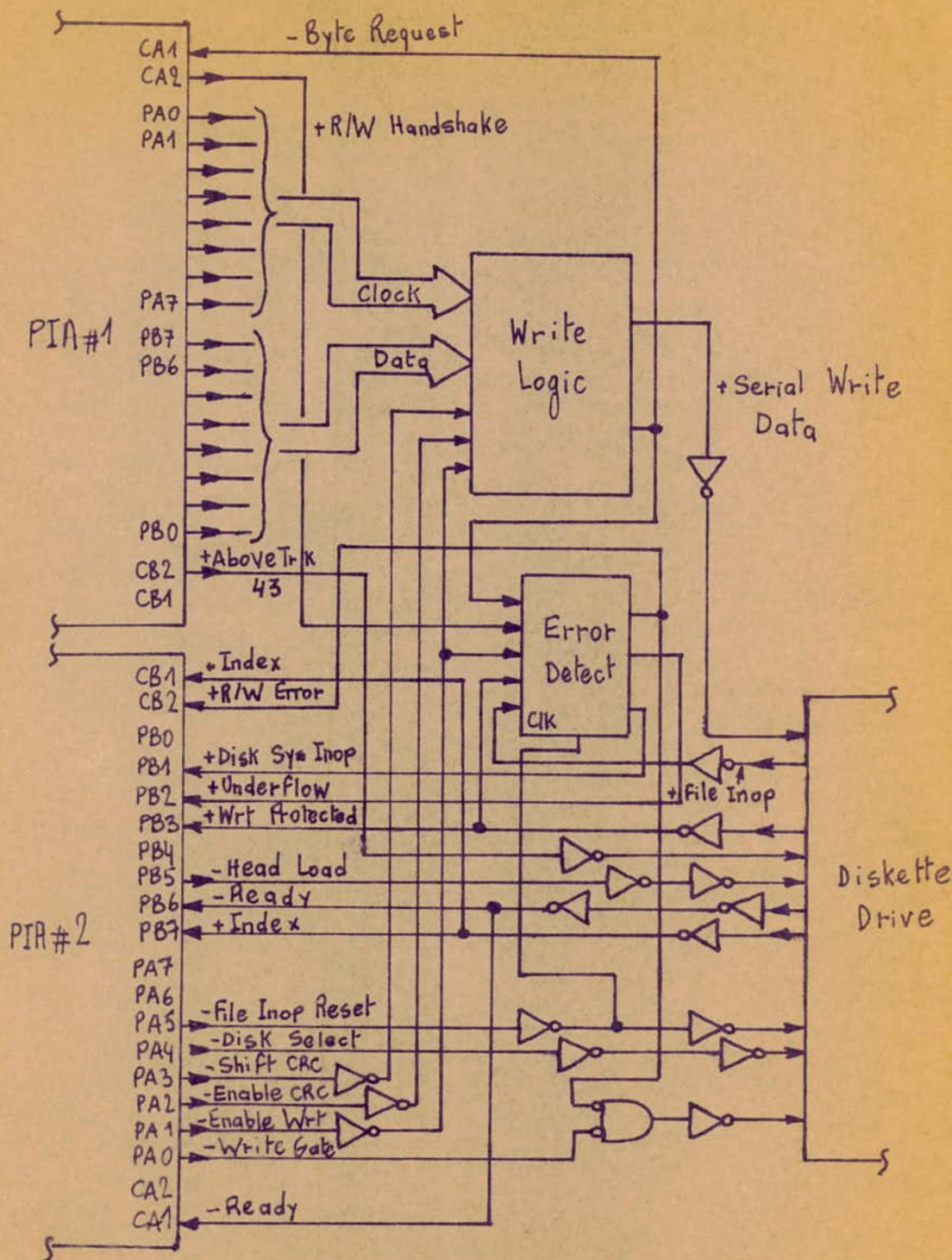
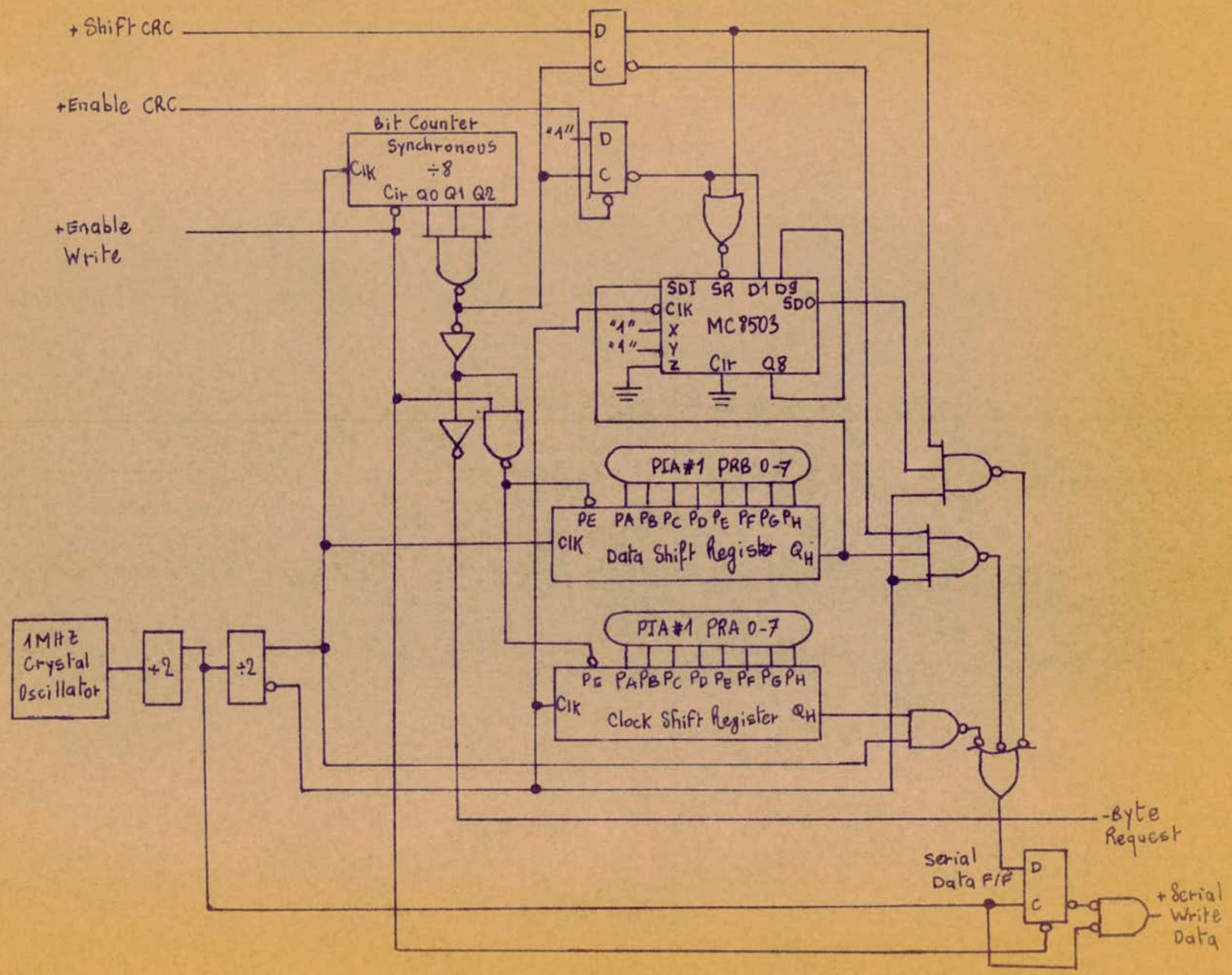


Figure IV-7. Write Operation Interface

Figure IV-8 - Floppy Disk Write Interface



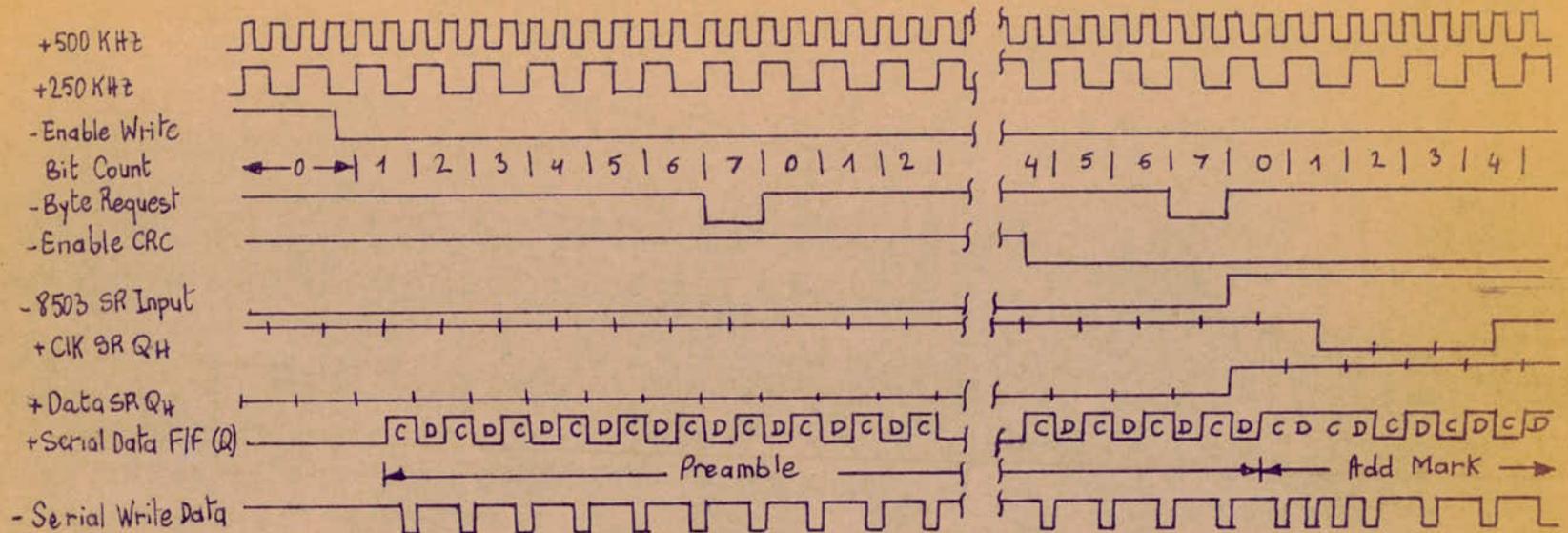


FIGURE 9. Beginning Write Data Timing

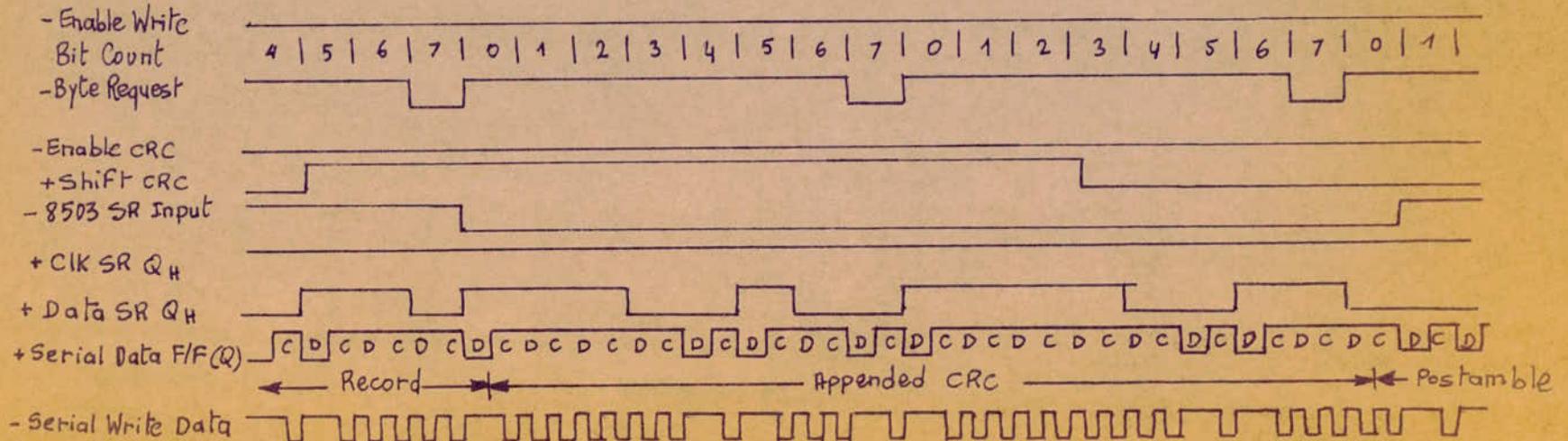


FIGURE 10. Append CRC Timing

Dans les chapitres II et III, nous avons présenté et étudié deux méthodes qui permettaient l'exploitation des unités d'entraînement du disque souple.

- La première méthode (chapitre II, paragraphe II) et dont nous avons donné à titre d'illustration une opération d'écriture (chapitre IV, et paragraphe V) consistait à relier directement les deux P I A à l'unité d'entraînement de disque et à un ensemble de circuits (R/W logic, ERROR detect logic....) qui est abusivement appelé FORMATTER.

- La deuxième méthode consistait à l'utilisation d'un FORMATTER (CA1COMP 1140) entre le MPU et les unités d'entraînement du disque et dont les interfaces sont totalement différentes de celles de la première méthode.

Notre étude pratique a pour objet de relier le FORMATTER au MPU donc de proposer une interface MPU/FORMATTER, puis une interface FORMATTER/DRIVES et enfin de donner les organismes de quelques commandes du FORMATTER.

I) INTERFACE MICROPROCESSEUR/FORMATTER.

1°) Constitution de l'interface MPU/FORMATTER.

Les lignes d'interface du système MPU/FORMATTER sont constituées de:

- 16 lignes bidirectionnelles de données
- 10 lignes du Formatter au MPU
- 5 lignes du MPU au Formatter

Ces lignes ont été expliquées au chapitre III.

2°) Interface proposée (voir fig.V.1)

L'utilisation et le rôle des différentes lignes du PIA ont été présentés dans le chapitre deux(II). En nous basant sur cette présentation, nous expliquons le choix de certaines lignes pour les signaux utilisés dans l'interface MPU/FORMATTER.

CA1(DATA IN): C'est une ligne de demande de service.

Une transition active de la ligne CA1 met à "1" le bit b7 du registre de contrôle A. Après que le microprocesseur ait reconnu que l'indicateur (flag) a été mis à "1", une donnée à lire ou à écrire est transférée du PIA ou vers le PIA.

.../....

CA2(DATA OUT) :

Cette ligne est la réponse du MPU au système dedisque indiquant que la demande de service à CA1 à été acceptée.

DATA OUT est la réponse du MPU au signal DATA IN venant du Formatter.

CB1(T/R) :

C'est une ligne qui indique la direction de transfert des données.

Quand (T/R=1), elle indique une transmission de données du Formatter, donc les lignes de données seront programmées en entrées.

Quand (T/R =0), cette ligne prépare le FORMATTER à recevoir des données du micro-processeur.

Mises à part les lignes de données (DATA BUS), le sens de toutes les autres lignes est fixe. Ce qui fait que les lignes PA0-PA7 et PBO-PB7 du PIA +1 doivent etre repositionnées suivant que le MPU envoie un mot de commande et un mot de données ou qu'il envoie un mot de commande et recoit un mot de données.

Les lignes PA0-PA7 et PBO-PB7 du PIA +1 sont utilisées pour le transfert des données.

Les lignes PA0-PA7 et PBO-PB3 du PIA + 2 constituent les lignes de controle.

II/ INTERFACE FORMATTER/UNITE DE DISQUES (voir fig.V.2)

Le Formatter est utilisé pour "INTEFACE" jusqu'à quatre unités de disques reliées en Daisy-chain, mais les lignes SELECT et READY sont envoyées séparément du Formatter pour chaque unité de disque.

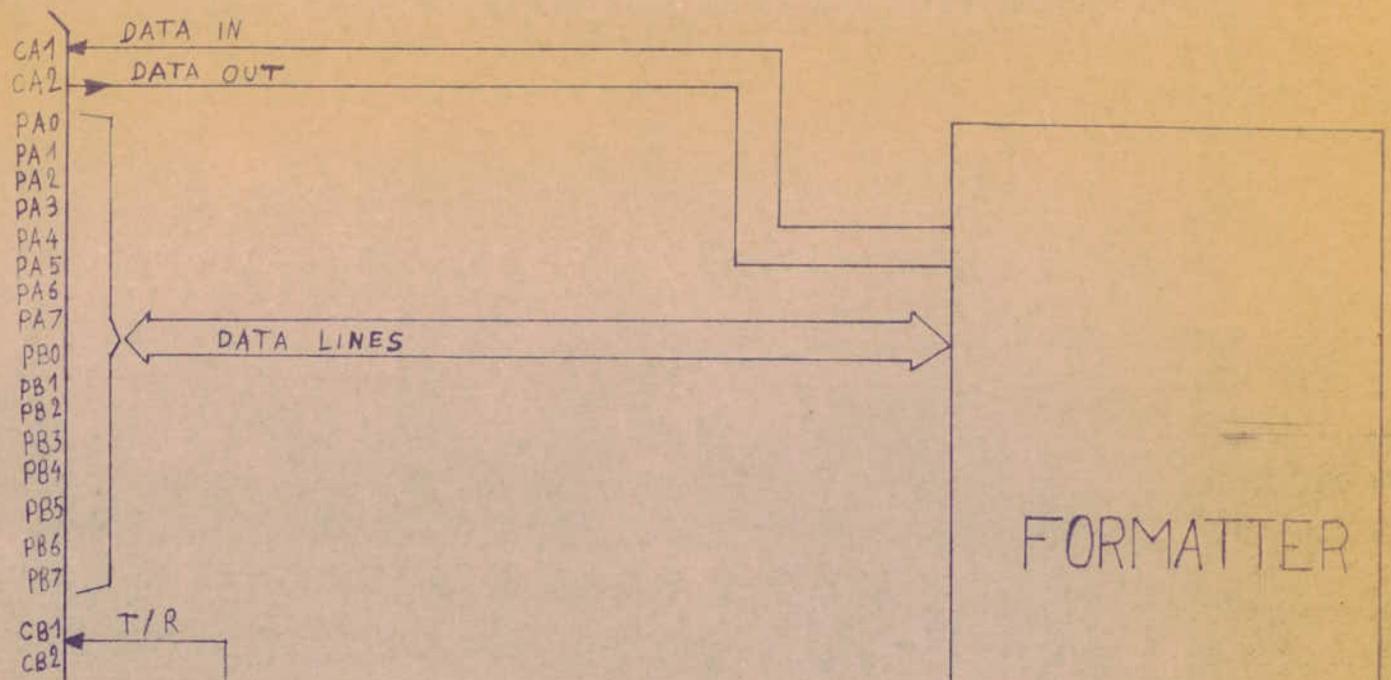
Les signaux d'interruptions sont : INDEX et READY

Les signaux d'états sont: TRACKOO, READY et INDEX.

Les signaux de controle sont:

- WRITE ENABLE
- WRITE ENABLED
- READ CLOCK
- STEP
- DIRECTION
- ABOVE TRACK 43
- SELECT
- WRITE DATA
- HEAD LOAD.

PIA#1



PIA#2

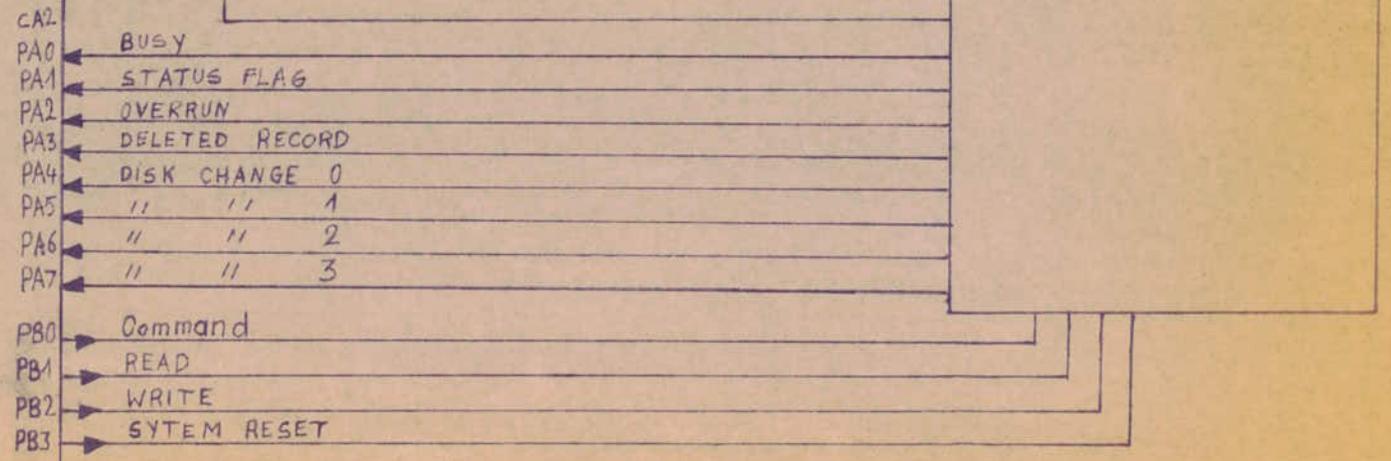


Fig V.1 Interface MPU/Formatter

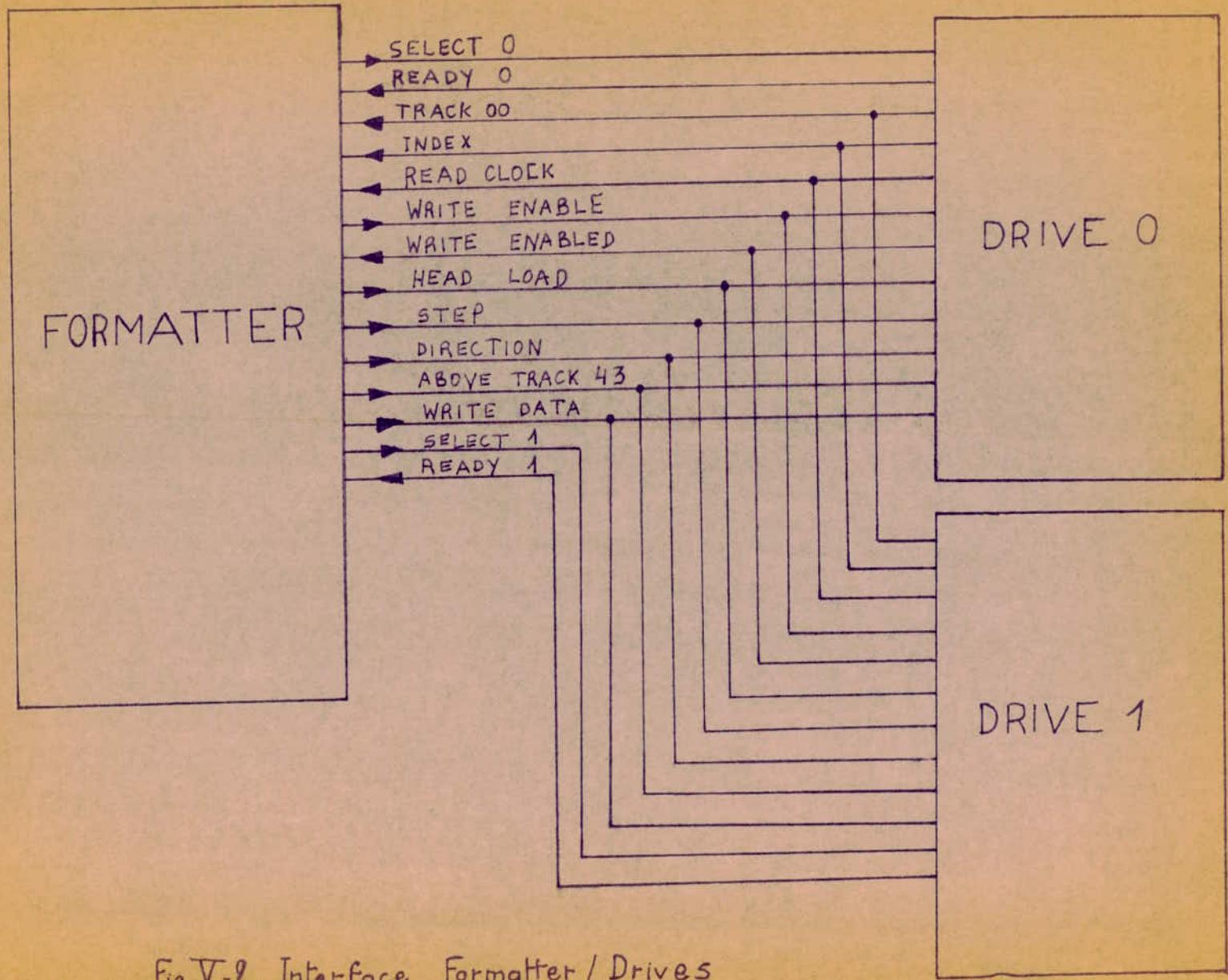


Fig V-2. Interface Formatter / Drives

CHAPITRE VI. ELABORATION DES ORGANIGRAMMES

L'organigramme d'une commande du Formatter peut être fait à partir du timing correspondant et que nous avons déjà traité au chapitre III.

Nous présenterons à titre d'exemple, les organigrammes des commandes SELECT, READ STATUS, READ, WRITE et nous commenterons celui de READ.

1°) SELECT : (voir fig.VI.1)

2°) READ STATUS: (fig VI.2)

3°) READ :

a) organigramme : (fig. VI.3)

b) Commentaire:

- Positionnement des PIA # 1, 2 (a) : Les lignes de données du PIA # 1 doivent être sortantes pour l'envoi du mot contenant la commande. Celles du PIA # 2, ont un sens fixe.

Le MPU doit mettre READ à zéro, envoyer la commande, puis de mettre la ligne COMMAND à zéro pour que le Formatter puisse s'occuper du traitement de la commande. Il répondra par BUSY qui doit être testé par le MPU pour repositionner les lignes de données du PIA # 1 dès que BUSY change d'état.

Les lignes T/R et DATA IN doivent être testées pour que le MPU sache si le Formatter est disponible pour la transmission ou la réception de données. Si les premiers tests de DATA sont négatifs, c'est à dire "DATA IN =1", le MPU doit prévoir un test pour OVERRUN qui le renseignera sur l'absence de données duquel cas le MPU ordonnera la fin de lecture. Par contre si "OVERRUN=1" le MPU testera encore DATA IN jusqu'à ce qu'il y ait fin de lecture ou bien passage à la suite du programme (DATA IN =0), c'est à dire lecture de données. Après ceci, le MPU mettra DATA OUT à 0 puis testera DATA IN.

Quand DATA IN =1, ceci signifie que le FORMATTER s'est acquitté du mot données; le MPU mettra alors DATA OUT à 1 signifiant qu'il a accompli la lecture du premier mot,....

Après la lecture du dernier mot de données, le MPU mettra DATA OUT à 0, READ à 1 puis testera DATA IN. Quand DATA IN =1, il mettra DATA OUT à 1 puis testera BUSY jusqu'à ce que BUSY =1, alors c'est la fin de lecture.

..../....

.../...

3°) CONCLUSION:

Compte tenu de la vitesse de transfert (250 K bits/s) soit environ 32500 mots/seconde, il convient d'étudier le temps nécessaire dans nos organigrammes pour le transfert d'un mot.

- Pour un ordinateur lent (M6800;....), il faut envisager un dispositif d'accès mémoire rapide.

- Pour un ordinateur rapide, notre solution est envisageable.

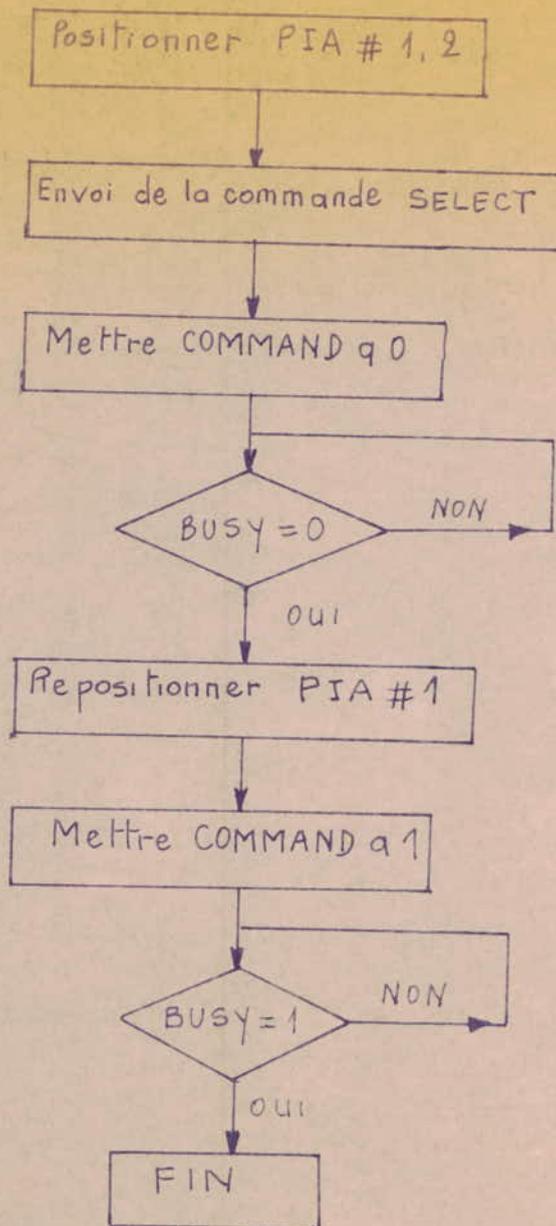
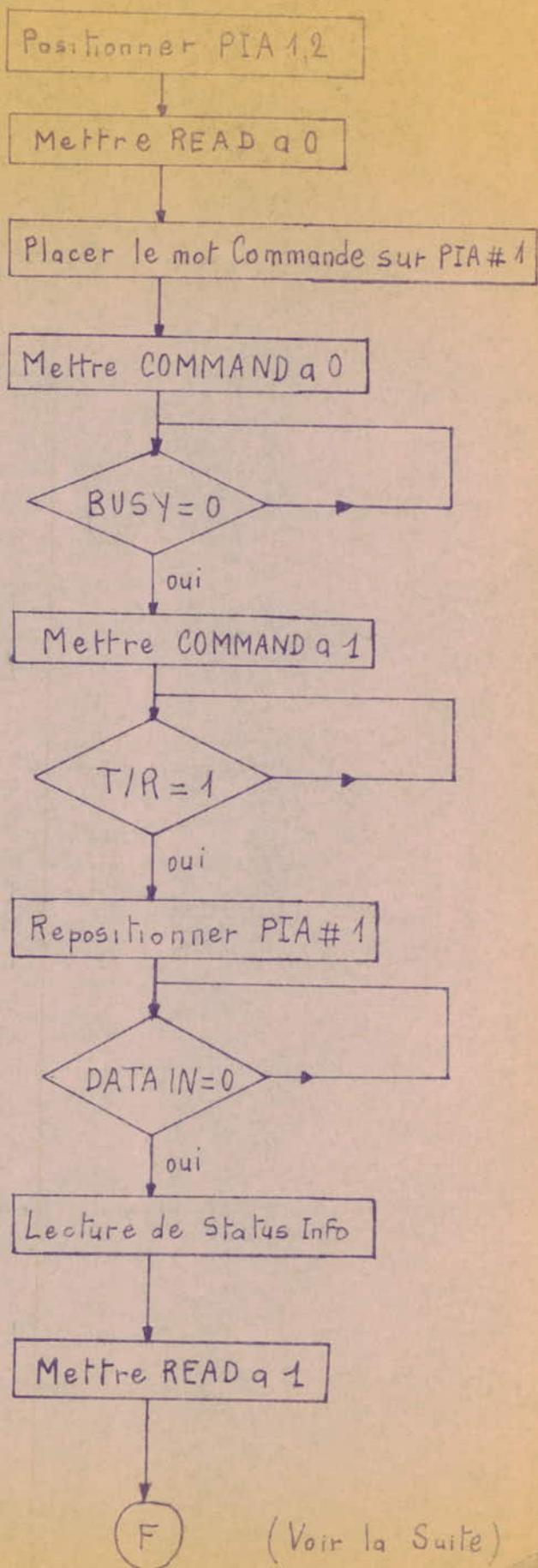


fig VI. 1. Organigramme de la Commande SELECT

Pour les Commandes : SEEK, RESTORE, et SECTOR LENGTH l'organigramme est le même, excepté, la commande qui change.



(Voir la Suite)

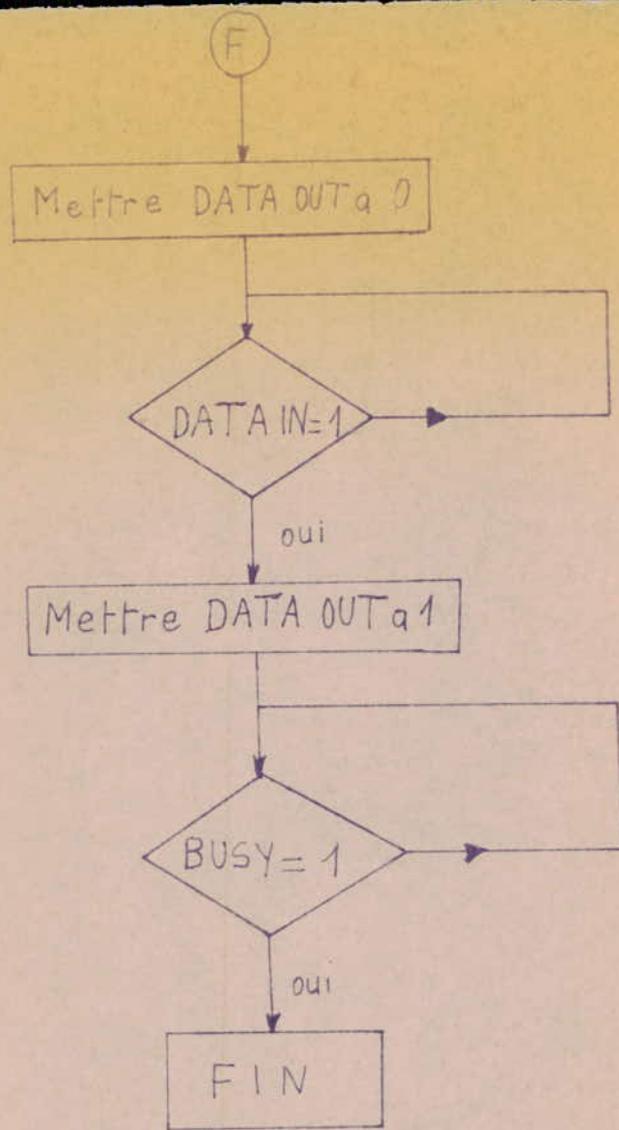
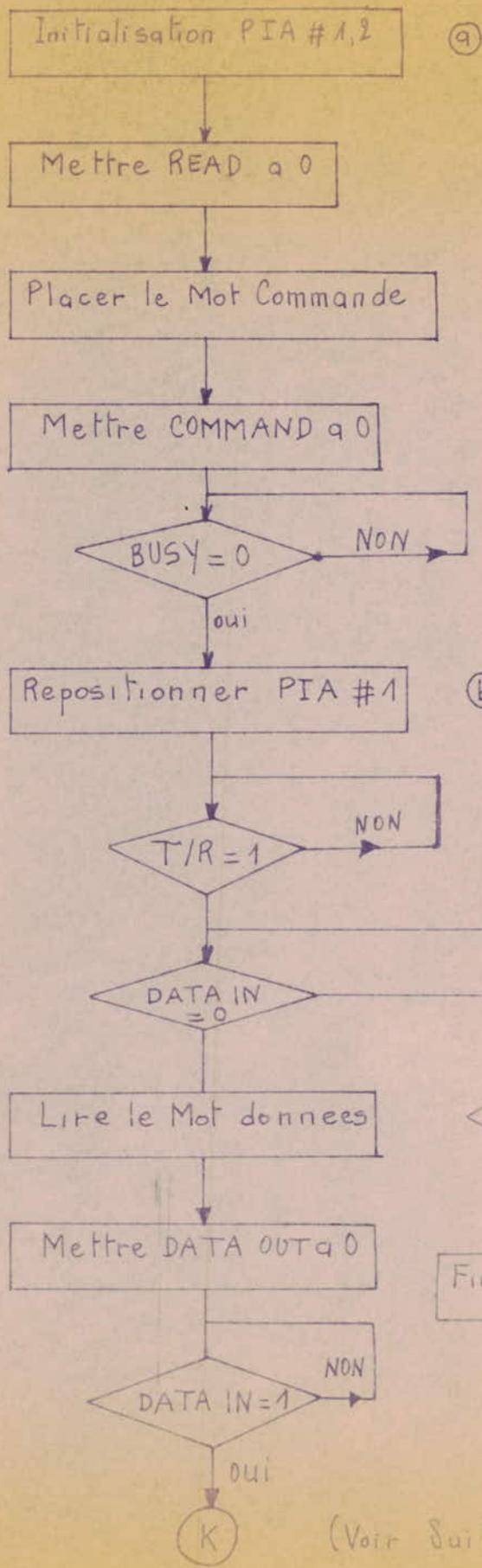


Fig VI-2 Organigramme de READ STATUS



(Voir Suite)

Dernier Mot de
donnees a Lire

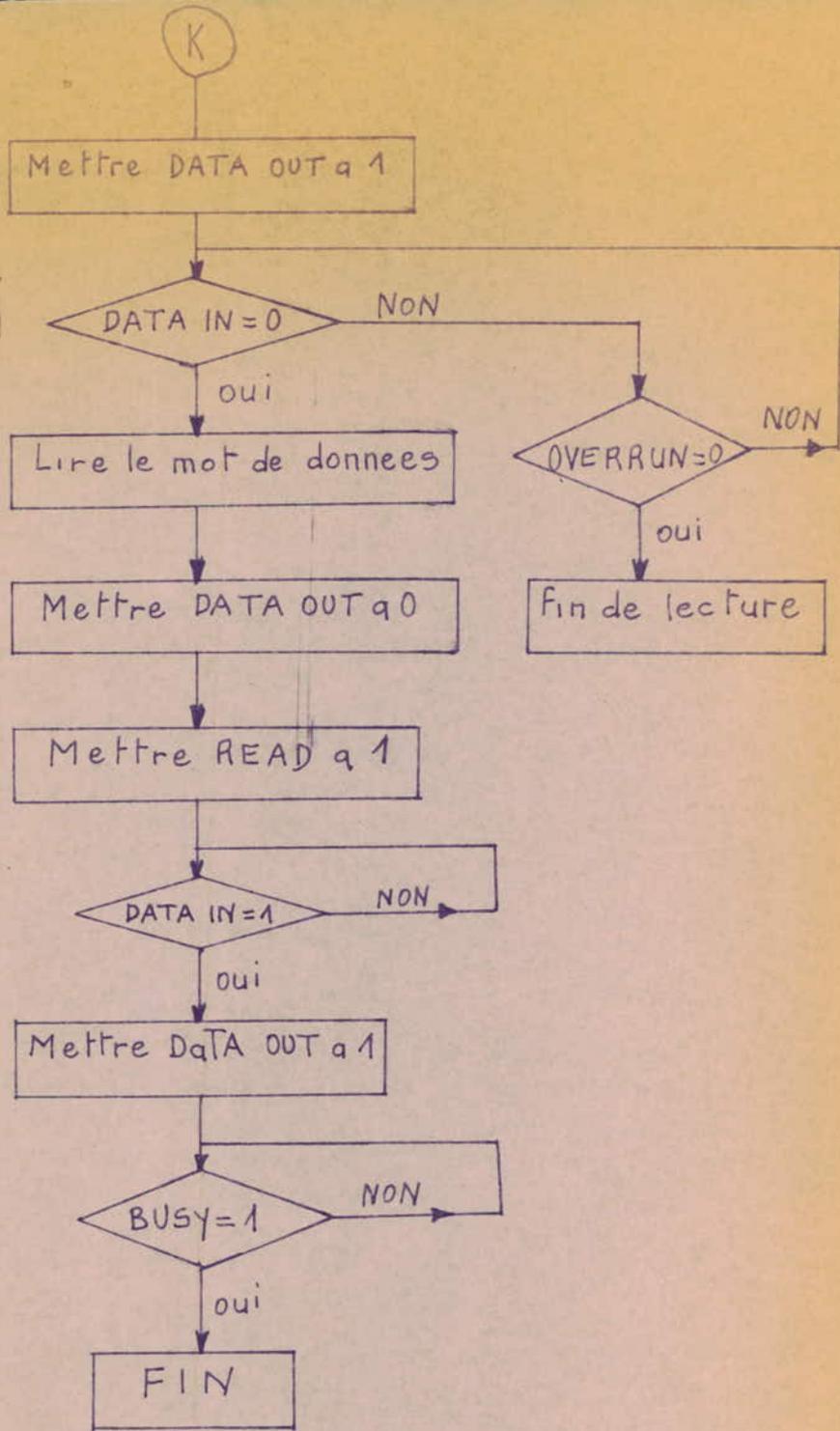
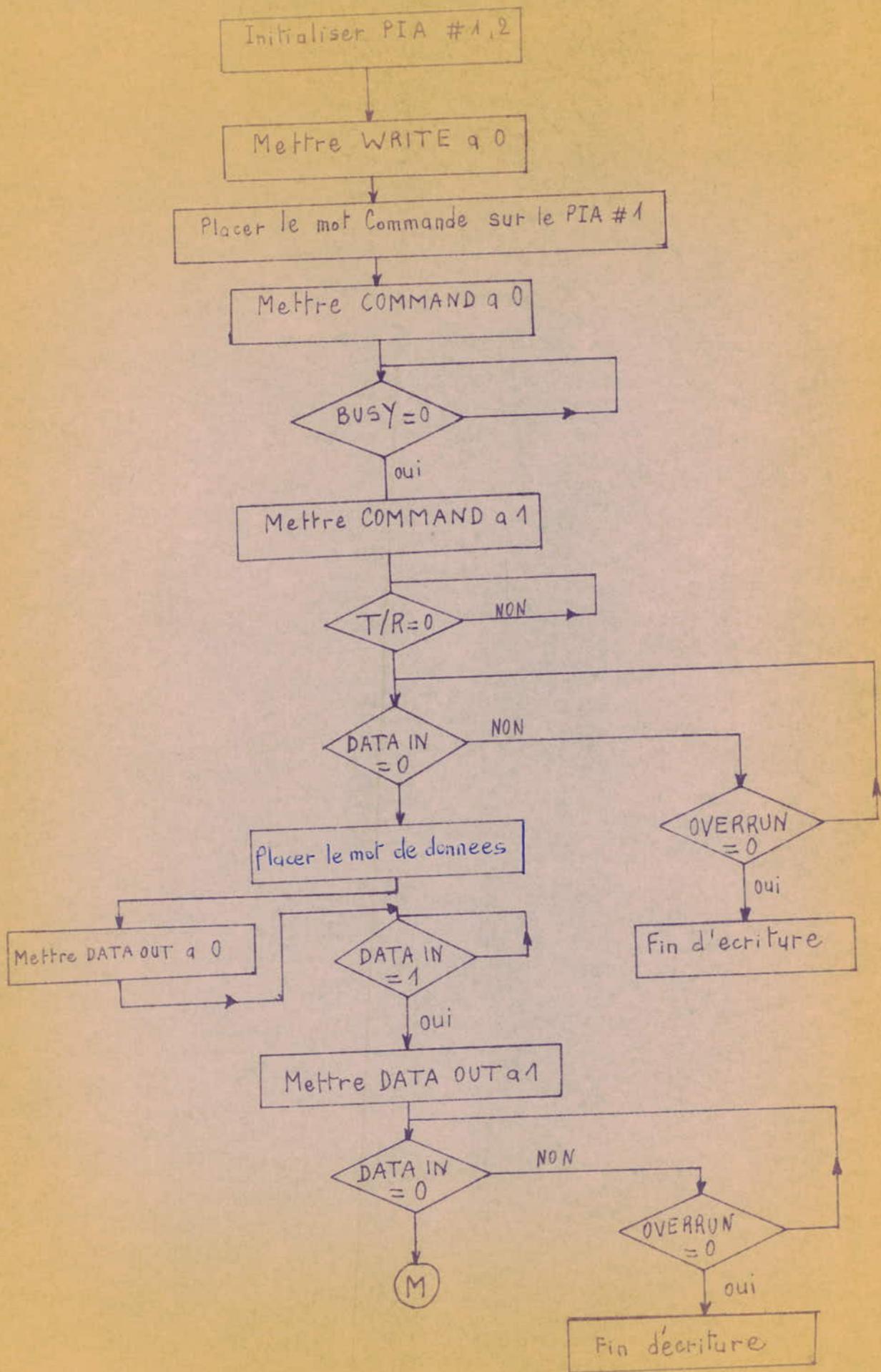


Fig VI.3. Organigramme de la Commande READ



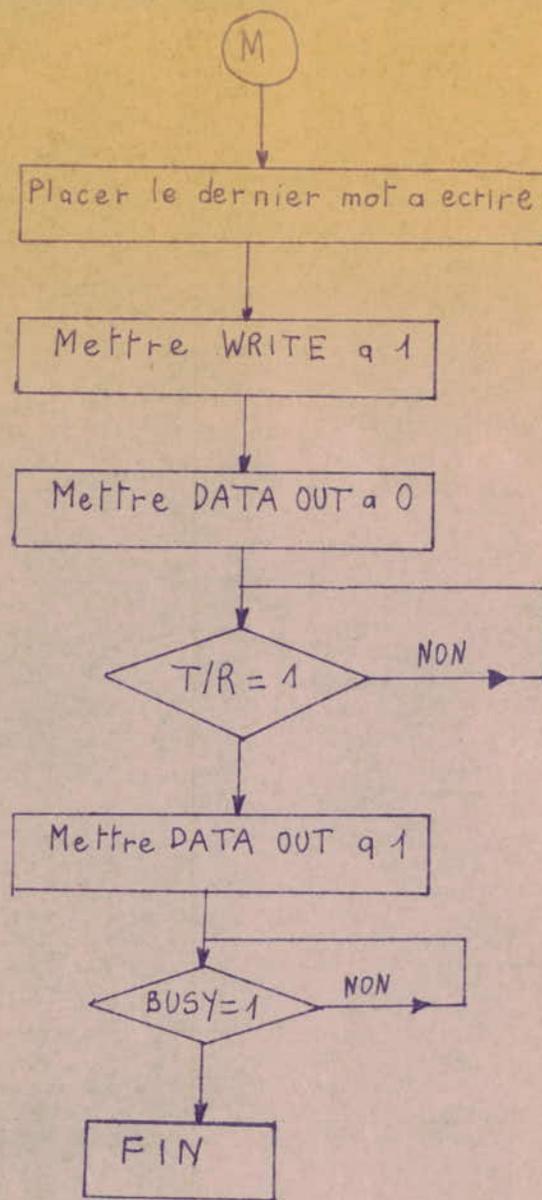


Fig VI.4 - Organigramme de la Commande WRITE

-- O N C L U S I O N --

Après près de 800 micro-instructions programmées par FIRMWARE dans le R O M du microprogramme, le formater est devenu un élément indispensable non seulement pour organiser la gestion des unités d'entraînement du disque souple, mais encore pour exécuter des vérifications, des recherches d'erreurs et des corrections ainsi que pour procéder à des tests divers.

Ainsi, son utilisation minimise beaucoup plus le SOFTWARE que pour la gestion directe des unités de disque à partir du microprocesseur. Et, grâce aux micro-processeurs, tout autre format à la demande peut être réalisé très aisément.

Le disque souple - ou « FLOPPY DISK » - semble tout destiné aux micro-ordinateurs : faibles volume et prix, excellent temps d'accès et compatibilité quasi-totale des dispositifs commercialisés.

Le disque souple accepte une densité d'information de 3200 bits par pouce, soit quelque 1260 bits par centimètre. De ce fait, il fournit en volume global de plus de trois millions de bits d'information, et cela, pour un prix modeste, ce qui a contribué à l'intérêt de ce type de mémoire.

Désormais, l'unité à disque souple rejoint donc, en volume, celles des cassettes magnétiques. Ses applications relèvent du traitement de texte, de l'édition elle servira de mémoire de programme pour les minis et les micros, et de périphérique pour les systèmes de frappe de texte, les calculatrices et tous les mini-systèmes « intelligents ».

Nous espérons que cette étude, contribuera à la mise en ligne des unités d'entraînements du disque souple avec les mini-ordinateurs et les micro-ordinateurs.

LEXIQUE ET ABREVIATIONS

Above : au dessus

AM : marquage d'adresse

Append(to) : Annexer, joindre

Available : disponible

Auto Status : etat automatique

Bit : chiffre binaire

Bit counter : compteur binaire

Buffer : mémoire tampon

Bus adress : bus d'adresse

Bus control : bus de contrôle

Busy : occupé

Byte : serie de huit bits, octet

Byte requête : demande de byte

Carry : retenue , report

Change(to) : changer, modifier

check(to) : vérifier , contrôler

check sum : vérification

clock : horloge

Commande ward: mot de commande ou mot contenant une commande

CR : registre de contrôle

CRC : contrôle cyclique

Data : données

Data bus : lignes de données

Data mark : marque de données

Detect (to) : detecter

Delete (to) : effacer, eliminer

Drive(to) : conduire, gérer(Ici) unité d'entraînement du disque

DDR : registre des directions de données

Enable(to) : valider, mettre en service

Peripheral : peripherique
P.I.A : interface d'adaptation du peripherique au M.P.U
Read (to) : lire
READY : pret , disponible
Read Operation : operation de lecture
Record (to) : enregistrer
Recover (to) : recuperer
Request^R(to) : demander
Reset(to) : remettre à zero , initialiser
Restore(to) : remise à zero (remise à la piste 80 par exemple)
R/W : lecture ;écriture

Sector : secteur
Seek(to) : rechercher
Select (to) : choisir ,selectionner
Shift(to) : decaler ,deplacer
Software :logiciel ,programme
Stack pointer :pointeur de pile
Status :etat

Track : piste
Transmit (to) : transmettre

Underflow : depassement de capacité
Valid(to) :valider
Write (to): ecrire

1148 (CALCOMP) :FORMATTER
~~XXXX~~ 140(CALCOMP) :unité d'entrainement du disque souple
...../..

Efficiency : efficacité

Error : erreur

File : fichier, unité de disque (abusivement employé)

Find(to) : trouver

Flag : indicateur , drapeau

Floppy disk : disque souple

Format : longueur et dimension

FORMATTER : formateur

Gate : porte

Halt(to) : arrêter

Handshake(to) : établir une liaison

Hardware : matériel , électronique utilisée

Head Load : charge de la tête lecture/écriture

ID : identifi cat(pour identification)

Index register : registre index

Inoperable: indisponible

Interrupt (to) : interrompre

Interrupt mask : masque d'interruption

Interrupt request : demande d'interruption

L

Latch : bascule à verrouillage

Length : longueur

Link (to) : relier, unir enchaîner

Load (to) : charger

Masquable : masquable

Next : prochain, suivant

OE : registre d'entrée/sortie

Overrun, overflow : dépassement de capacité

B I B L I O G R A P H I E

- CALCOMP 1140 : FORMATTER MANUEL
CALCOMP 140/142 : FLOPPY DISK DRIVE MANUEL
MOTOROLA : MICROPROCESSOR APPLICATIONS MANUEL
H. LILEN : INTRODUCTION DE LA MICROINFORMATIQUE DU
MICROPROCESSEUR AU MICROORDINATEUR.

