

DEPARTEMENT D'ELECTRONIQUE ET ELECTROTECHNIQUE

PROJET DE FIN D'ETUDES

DIPLOME D'INGENIORAT.

المدرسة الوطنية للعلوم الهندسية
— المكتبة —

ETUDE ET REALISATION
D'UNE UNITE DE DIALOGUE

Proposé par : H. TEDJINI Dr. Ingénieur

المدرسة الوطنية للعلوم الهندسية

المكتبة

ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHEQUE

Etudié par :

BENAZZOUZ Chawky

MANSOURI Smail

A MES PARENTS

MES FRERES

MES SOEURS

MES AMIS

CHAWKY B.

A MES PARENTS

MES FRERES

MES SOEURS

MES AMIS

SMAIL M.

DEPARTEMENT D'ELECTRONIQUE ET ELECTROTECHNIQUE

PROJET DE FIN D'ETUDES



ETUDE ET REALISATION
D'UNE UNITE DE DIALOGUE

Proposé par : H. TEDJINI Dr. Ingénieur

Etudié par :

BENAZZOUZ Chawky

MANSOURI Smail

JANVIER 1980

REMERCIEMENTS

Ce travail a été effectué à la division V du Centre des Sciences et de la Technologie Nucléaires.

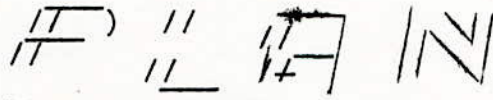
Nous formulons l'expression de notre profonde reconnaissance à:

M^r BOUALEM SANSAL , Chef de la division V " Controle et Simulation", pour avoir bien voulu nous accueillir dans sa division.

M^r HACENE TEDJINI, D^r Ingénieur au Centre des Sciences et de la Technologie Nucleaire, pour toute la confiance qu'il nous a faite en nous proposant ce sujet et pour nous avoir guidé et orienté dans notre travail.

Nous remercions Messieurs ACHOUR BOURKEB, RACHID OUGINI , SLIM et toute l'équipe de la division V. Pour tout intérêt qu'ils ont porté à notre projet et pour leur précieux conseils.

Nous voulons que nos parents, nos frères, nos soeurs, nos amis, nos maitres, nos professeurs trouvent dans ce modeste ouvrage l'expression de notre gratitude.



CHAPITRE I

=====

LES CIRCUITS D'ENTREE/SORTIE OU COUPLEURS Page 1

CHAPITRE II

=====

LE CIRCUIT D'INTERFACE SERIE ACIA " VI

CHAPITRE III

LE CIRCUIT D'INTERFACE PARALLELE PIA " XVII

CHAPITRE IV

=====

LE PERIPHERIQUE:
LE LECTEUR OPTIQUE MODELE 601 " XX

CHAPITRE V

=====

LE PERIPHERIQUE:
TERMINAL TEKTRONIX 4023 " XXVII

CHAPITRE VI

=====

"HARDWARE"
REALISATION PRATIQUE " XXXII

CHAPITRE VII

=====

"SOFTWARE"
PROGRAMMATION " XXXXVIII

INTROCUCTION

La conception d'un système à logique câblée et à microprocesseur présente d'importantes différences. Dans le cas d'une logique câblée, lorsque les composants voulus ont été fonctionnellement interconnectés, le travail est terminé. Avec un microprocesseur, la définition du "HARDWARE" n'est qu'une phase préliminaire et le véritable travail commence avec la conception et la réalisation du programme qui transformera le micro-ordinateur "NON INTELLIGENT" en un système fonctionnel spécialisé pour une tâche précise.

Un microprocesseur, tout seul, ne sait rien faire. Pour qu'il puisse travailler, il faut lui adjoindre des circuits de mémoires et d'entrée/sortie. Associé à ces circuits le microprocesseur forme un micro-ordinateur.

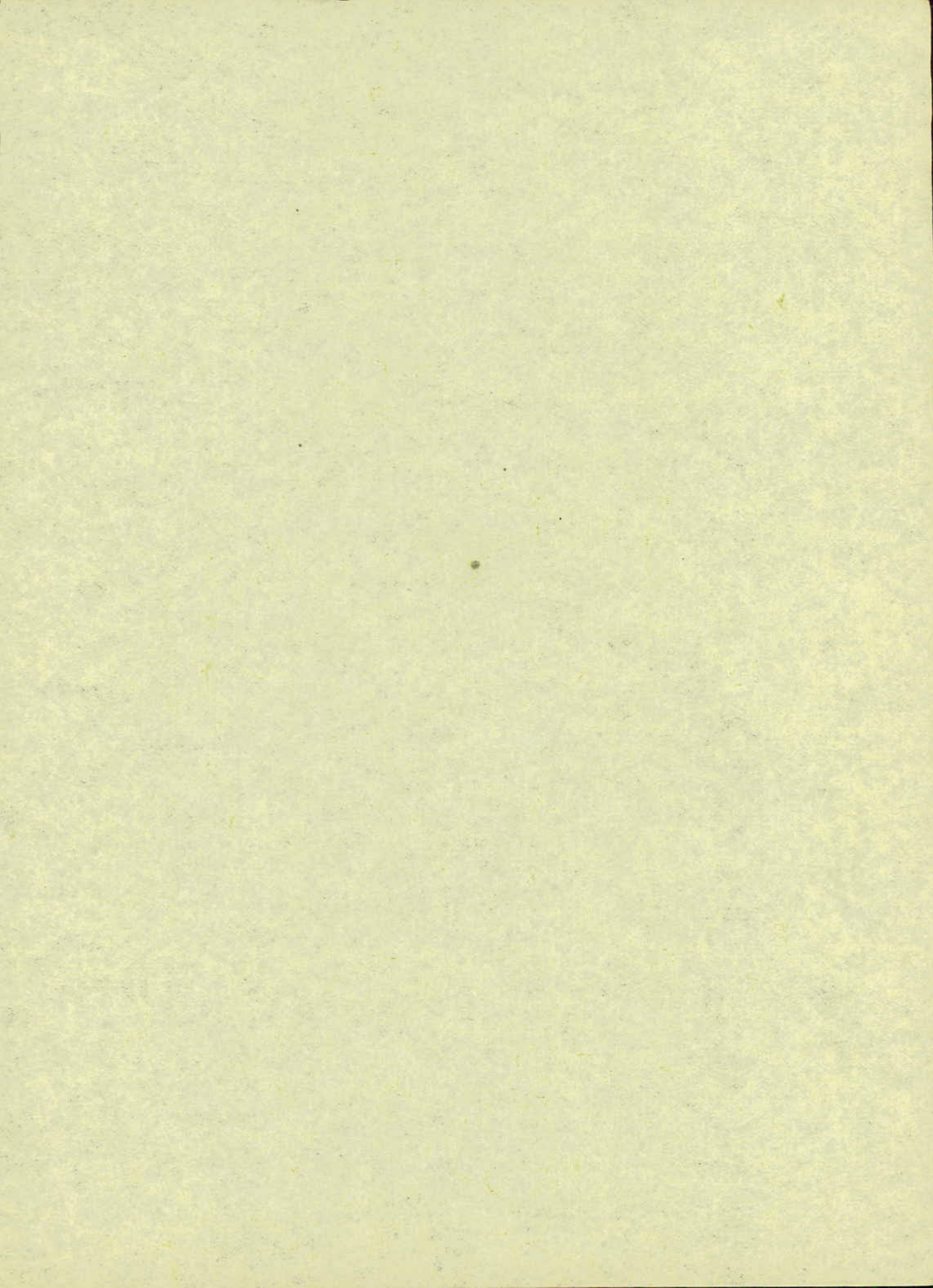
L'ordinateur est constitué de deux sous-ensembles:

+L'unité centrale.

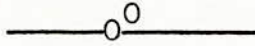
+Les périphériques (Unités de dialogue).

Ces deux unités diffèrent surtout par leurs vitesses de fonctionnement et les niveaux des signaux de sortie et d'entrée.

Dans le but de lever ces problèmes, nous nous proposons à étudier et réaliser l'interfaçage d'une unité de dialogue. Ce projet de fin d'études entre dans le cadre du grand projet "SIMULATION ET CONTROLE D'UN REACTEUR NUCLEAIRE" de la division V du centre des sciences et de la technologie nucléaire (C-S-T-N).

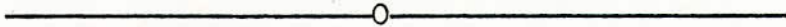


CHAPITRE I



LES CIRCUITS D'ENTREE/SORTIE OU COUPLEURS

- I INTRODUCTION
- II ORGANISATION INTERNE D'UN CIRCUIT D'ENTREE/SORTIE
- III ORGANISATION EXTERNE D'UN CIRCUIT D'ENTREE/SORTIE
- IV LES SIGNAUX D'INTERFACE ENTRE LE MICROPROCESSEUR
ET LE CIRCUIT D'ENTREE/SORTIE
- V LES SIGNAUX D'INTERFACE ENTRE LA PERIPHERIE ET LE
CIRCUIT D'ENTREE/SORTIE
- VI LES DIFFERENTS MODES D'ADRESSAGE DU CIRCUIT
D'ENTREE/SORTIE.
- VII LES DIFFERENTS METHODES DE DIALOGUE ENTRE LE MICROPROCESSEUR
ET LA PERIPHERIE
- VIII PROGRAMMATION D'UN CIRCUIT D'ENTREE/SORTIE



I INTRODUCTION:--

Un microprocesseur n'est pas fait en général pour dialoguer uniquement avec sa mémoire de programme et de données. Il doit pouvoir communiquer avec la périphérie ce qui veut dire recevoir ou transmettre des informations de ou vers le milieu extérieur, ce qui établit le dialogue entre l'homme et la machine et fait accroître considérablement le champ d'application du microprocesseur.

Le milieu extérieur peut être un clavier, un lecteur de bande perforée, une unité de disque souple; une imprimante ou tout simplement un ensemble de signaux électriques provenant de capteurs qui seront lus par le microprocesseur. Ces signaux sont mémorisés, traités et ensuite envoyés vers la périphérie pour être visualisés sur un écran ou enregistrer sur disque ou bande ou bien agir sur un système industriel (Commande numérique de machine, contrôle d'un processus...)

Un microprocesseur du type MC6800 de MOTOROLA PAR par exemple dialogue avec la périphérie à travers des circuits d'entrée/sortie programmables (la fonction est définie par programme) donc adaptables à des environnements divers.

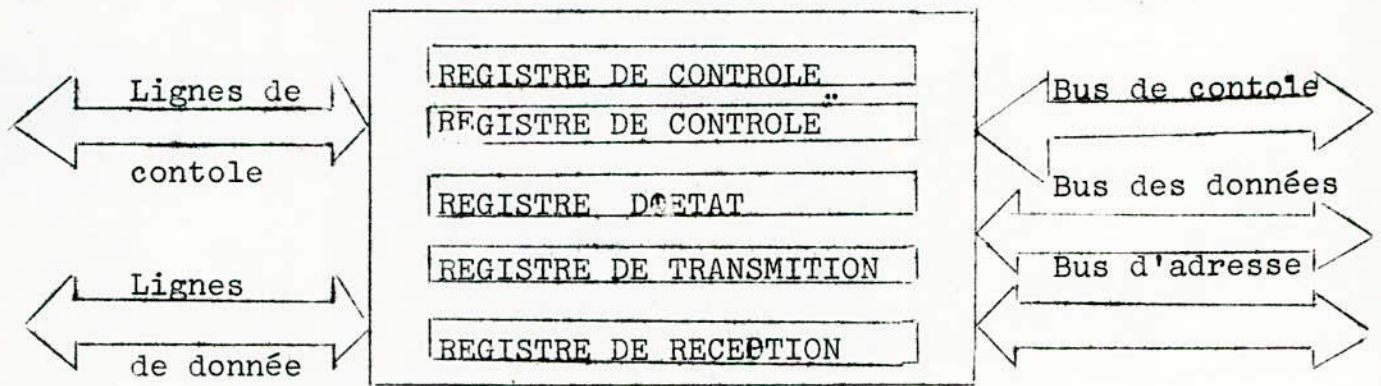
Chaque circuit d'interface d'entrée/sortie ne recouvre qu'un domaine d'application bien spécifique et que c'est à l'intérieur de ce domaine qu'il présente suffisamment de souplesse d'utilisation;

Comme exemple de circuit d'entrée/sortie qui font partie de la famille MOTOROLA on peut citer :

- ACIA (Adaptateur d'interface de communication asynchrone.)
- PIA (Adaptateur d'interface périphérique).
- SSDA (Adaptateur d'interface de données séries asynchrone).

L'ACIA et le SSDA sont utilisés pour la transmission série alors que le PIA est utilisé généralement pour la transmission parallèle.

II ORGANISATION INTERNE D'UN CIRCUIT D'ENTREE/SORTIE :



(Fig 1)

-Chaque circuit d'interface possède en générale , trois types de registres accessibles par l'utilisateur (voir) Fig1)

-Les registres de controle, qui définissent la configuration fonctionnelle du circuit. Il sont en général à écriture simplement.

-Le registre d'état contient un certain nombre de bascules qui se positionnent au niveau logique 1 ou 0 suivant l'état du circuit et celui de la communication entre la périphérie et le microprocesseur:

- demande d'interruption

- détection d'une erreur (Parité, surcharge...)

Ce registre est à lecture seulement.

-Le registre de transmission, il assure le transfert de l'information du microprocesseur vers la périphérie; il est à écriture seulement.

-Le registre de réception, il permet la transmission des données de la périphérie vers le microprocesseur, il est à lecture seulement.

REMARQUE

Dans certains circuits d'interfaces, le registre de controle et le registre d'état ne forment qu'un seul registre.

ce registre peut être évidemment lu ou écrit.

Il arrive que le registre de transmission et de réception ne forme qu'un seul, il est donc bidirectionnel (possibilité d'écriture et de lecture).

D'autre part certains registres peuvent jouer, suivant le mode de programmation, différents rôles.

III ORGANISATION EXTERNE :

Le circuit d'interface d'entrées/sorties communique à la fois avec le microprocesseur et la périphérie.

les communications se font par: (voir Fig 1)

- les signaux d'interface entre le microprocesseur et le circuit d'entrée/sortie.

- les signaux d'interface entre le circuit d'entrée/sortie et la périphérie.

IV LES SIGNAUX D'INTERFACE ENTRE LE MICROPROCESSEUR ET LE CIRCUIT D'ENTREE/SORTIE:

Le microprocesseur communique avec un circuit d'interface d'entrée/sortie par trois lignes omnibus:

Bus de données

Bus d'adresse

Bus de controle

Les registres des circuits d'entrées/sortie peuvent être considérés comme des positions mémoires dans lesquels le microprocesseur peut lire ou écrire . Ces positions mémoires peuvent être dissociés ou associées à la mémoire du microprocesseur.

Lorsque ces positions mémoires sont isolées, le microprocesseur dispose d'instructions spéciales d'entrée/sortie (en 8080 INTEL) le deuxième cas consiste à assimiler les registres du circuit d'interfacage comme des positions mémoires et dans ce cas là l'instruction d'entrée/sortie est inutile (ex le MC 6800)

V LES SIGNAUX D'INTERFACE ENTRE CIRCUIT D'ENTREE/SORTIE ET LA PERIPHERIE.

Les lignes de données par lesquelles sont véhiculées les données provenant de ou allant vers la périphérie. Leur nombre dépend du type d'application. Dans le cas de la transmission asynchrone (ACIA), on a que deux lignes:

Ligne de transmission

Ligne de réception

Pour les circuits types PIA ce nombre est de 16 permettant des transmissions sur 8 ou 16 bits, ce nombre est choisi suivant les commandes. Le sens de transmission est programmable.

Les lignes de contrôle qui permettent à la périphérie d'interrrompre le microprocesseur et de permettre à ce dernier de contrôler la périphérie. Ces lignes sont définies une fois pour toute par programme.

VI LES DIFFERENTS MODES D'ADRESSAGE DU CIRCUIT D'ENTREE/SORTIE.

Le microprocesseur peut s'adresser aux circuits d'interface d'entrée/sortie de deux manières:

Par le bus de données (séparation des opérations d'entrée/sortie et de mémoire. Fig 2)

Par le bus d'adresse (dans ce cas il y a concentration des opérations d'entrée/sortie et de mémoire Fig 3)

MEMOIRE 64K

ENTREE/SORTIE
256 MOTS

(Fig2)

MEMOIRE 32K ENTREE/SORTIE

$A_{15}=1$ $A_{15}=0$

(Fig3)

VII LES DIFFERENTS METHODES DE DIALOGUE ENTRE LE MICROPROCESSEUR ET LA PERIPHERIE.

Un micro-ordinateur doit pouvoir travailler avec la périphérie sans dégrader la performance de traitement du microprocesseur, le dialogue peut se faire de deux manières:

-Le microprocesseur prend en charge le dialogue avec la périphérie dans ce cas il teste le registre d'état de tous les circuits d'entrée/sortie pour savoir si l'un d'eux est prêt à effectuer une opération d'entrée/sortie donc le microprocesseur doit balayer périodiquement tous les registres d'états des circuits d'entrée/sortie. Cette méthode est coûteuse en temps et réduit les performances du système.

- En général, on préfère à cette méthode l'interruption du microprocesseur à chaque fois qu'une périphérie souhaite échanger des informations. Ce mode de fonctionnement nécessite alors une ligne de contrôle qui vient changer l'état du registre d'état. Le microprocesseur exécute normalement son programme et ne s'arrête que si une demande d'interruption lui est communiquée et qu'en plus elle ne soit pas masquée.

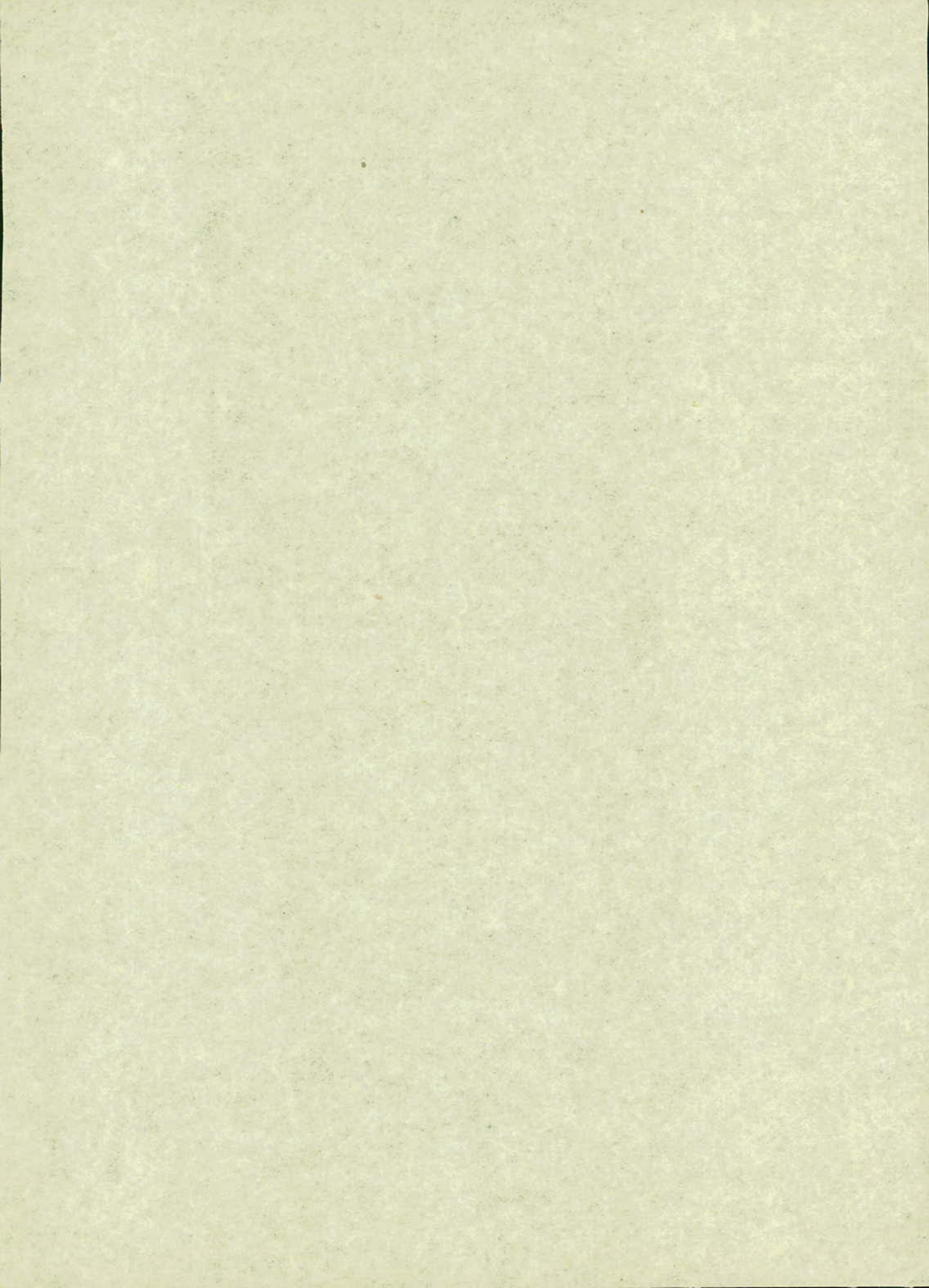
VIII PROGRAMMATION D'UN CIRCUIT D'ENTREE/SORTIE.

La fonction que doit jouer un circuit d'entrée/sortie est déterminée par programme:

-La définition de la configuration fonctionnelle se fait en écrivant dans les registres de contrôle une information indiquant la fonction que doit jouer le circuit d'entrée/sortie.

Une fois programmé il peut assurer la liaison entre le microprocesseur et la périphérie.

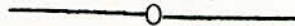
Le transfert de données entre le microprocesseur et la périphérie influe sur le registre d'état du circuit d'interface; donc la lecture du registre d'état permet de savoir quel périphérique a demandé une interruption, de savoir s'il y a erreur de parité, de format etc...



CHAPITRE II

LE CIRCUIT D'INTERFACE SERIE "ACIA"

- I INTRODUCTION
- II LIAISON SERIE ASYNCHRONE
- III LIAISON SERIE SYNCHRONE
- IV CODE ASCII
- V SYNOPTIQUE INTERNE DE L'ACIA (MC 6850)
- VI LES DIFFERENTS SIGNAUX D'INTERFACE ENTRE L'ACIA ET LE MICROPROCESSEUR
- VII LES DIFFERENTS SIGNAUX ENTRE L'ACIA ET LA PERIPHERIE
- VIII PROGRAMMATION DE L'ACIA
- IX TRASMITION ASYNCHRONE
- X RECEPTION ASYNCHRONE



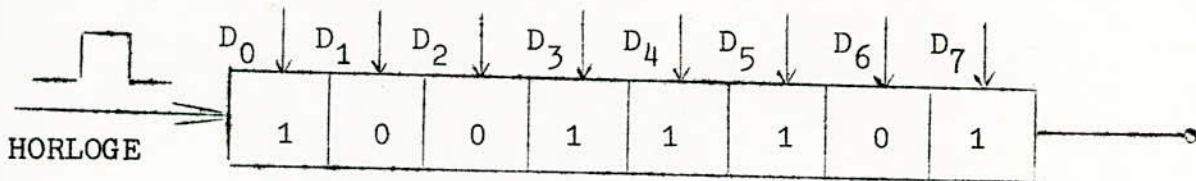
I INTRODUCTION:

Dans certaines applications ,lespériphériques transmettent les données sur une ligne et les reçoivent sur une autre. Chaque caractère (MOT) transmis par ou reçu par la périphérie est constitué généralement par 6,7 ou 8bits;unbit supplémentaire peut être ajouter pour détecter éventuellement si le caractère est erroné. Ce caractère est véhiculé sur une ligne en mode série,c'est à dire que les bits ne sont pas tranmis simultanément mais l'un derrière l'autre.

Comme on le voit un circuit "SERIALISEUR" ,"DESERIALISEUR" est nécessaire pour remplir une telle fonction.ce circuit porte un nom et s'appelle ACIA (Asynchrone Communication Interface Adapter) qui fait partie de la famille MOTOROLA 6800.

II LIAISON SERIE ASYNCHRONE:

L'idée est simple: les 8 bits parallèles passent dans un circuit approprié (un registre à décalage) pour sortir sur un seul fil sous forme serie.Les bits sortent les uns derrières les autres à une certaine vitesse (vitesse de l'horloge du registre à décalage).



Malheureusement si l'on transmet ainsi plusieurs mots de 8 bits à la queue leu leu, on se rend compte que la panique la plus complète règne sur la ligne de transmission puis qu'on ne sait plus où commence et où finit un mot donnée.

Un moyen simple consiste à précéder le mot à envoyer, d'un bit indiquant le commencement du mot et d'un bit indiquant la fin du mot transmis. Dès lors l'explication du mot ASYNCHRONE est possible; en effet les circuits d'émission et de réception des données n'ont pas à être synchronisés, car le circuit receveur saura grâce au bit de départ qu'il va recevoir un mot et grâce au bit de fin, il saura qu'il a reçu le mot.

REMARQUE:

Une standardisation à l'échelle internationale existe au niveau de ce mode de transmission. Soit une ligne de transmission série niveau TTL.

- Au repos la ligne est au niveau haut "1"
- Un niveau "1" s'appelle "MARK"
- Un niveau "0" s'appelle "SPACE"
- Le bit de début s'appelle "START" qui est toujours au niveau logique "0"
- Le bit de fin s'appelle "STOP" qui est au niveau logique "1".
- La transmission s'effectue selon un code

iii LIAISON SERIE SYNCHRONE:

Les données forment un flot continu de bits sans qu'il soit possible d'identifier les limites des caractères.

Il est donc nécessaire d'effectuer la synchronisation des caractères de début du bloc de données lors de la réception, ceci nécessite la détection d'un seul ou de deux caractères successifs de synchronisation.

Les communications à grandes distances se font par MODEM.

IV CODE ASCII (Américan Standard Code For Information Interchange):

Ce Code qu'asi universellement utilisé à été développé aux USA.

En particulier pour les télétypes; c'est à dire que toutes lettres et tous les signes existant sur un clavier d'une machine à écrire sont représentés par un code constitué de deux chiffres décimaux; le chiffre de gauche ne pouvant pas dépasser la valeur 7, alors chaque symbole est traduit par 7 bits seulement.

(ex: A est codée en ASCII par 4.1 c.à.d 1000 0001)

Pour réduire l'erreur de transmission chaque caractère de 7 bits est associé à un bit de parité.

Ce bit se traduit de la façon suivante:

-Parité impaire: Le nombre totale de "1" par caractère de 7 bits plus la parité impaire doit être un nombre impair ; le bit de parité est donc placé par consequence à "1".

ex: A=4.1 1 1000 0001

(1+1)+ parité impaire 1= nombre impair

-Parité paire: Le nombre totale d "1" par caractère de 7 bits plus parité paire doit être paire donc le bit de parité, par conséquence, doit être égale à 0

ex: A=4.1 0 1000 0001

(1+1)+ parité paire 0 = nombre paire

V SYNOPTIQUE INTERNE DE L'ACIA:

Comme tous les circuits de la famille MC 6800, L'ACIA est traité par le microprocesseur comme position mémoire. La Fig 1 montre un synoptique du 6850; nous pouvons voir qu'il comprend 4 registres internes(TDR,SR,CR,RDR).

					b ₇	0	0	0	0	1	1	1	1
					b ₆	0	0	1	1	0	0	1	1
					b ₅	0	1	0	1	0	1	0	1
b ₄	b ₃	b ₂	b ₁	Col		0	1	2	3	4	5	6	7
				15		0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	⊙	P	~	P	
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q	
0	0	1	0	2	STX	DC2	"	2	B	R	b	r	
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s	
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t	
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u	
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v	
0	1	1	1	7	BEL	ETB	/	7	G	W	g	w	
1	0	0	0	8	BS	CAN	(8	H	X	h	x	
1	0	0	1	9	HT	EM)	9	I	Y	i	y	
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z	
1	0	0	1	11	VT	ESC	+	;	K	[k	{	
1	1	0	0	12	FF	FS	,	<	L		l	!	
1	1	0	1	13	CR	GS	-	=	M]	m	}	
1	1	1	0	14	SO	RS	;	>	N	^	n	~	
1	1	1	1	15	SI	US	/	?	O	_	o	DEL	

CODE ASCII

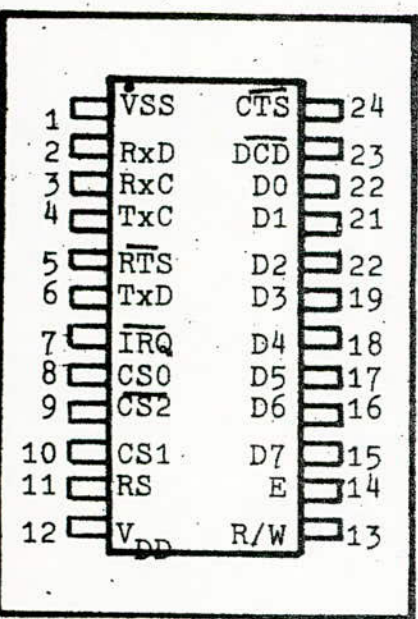
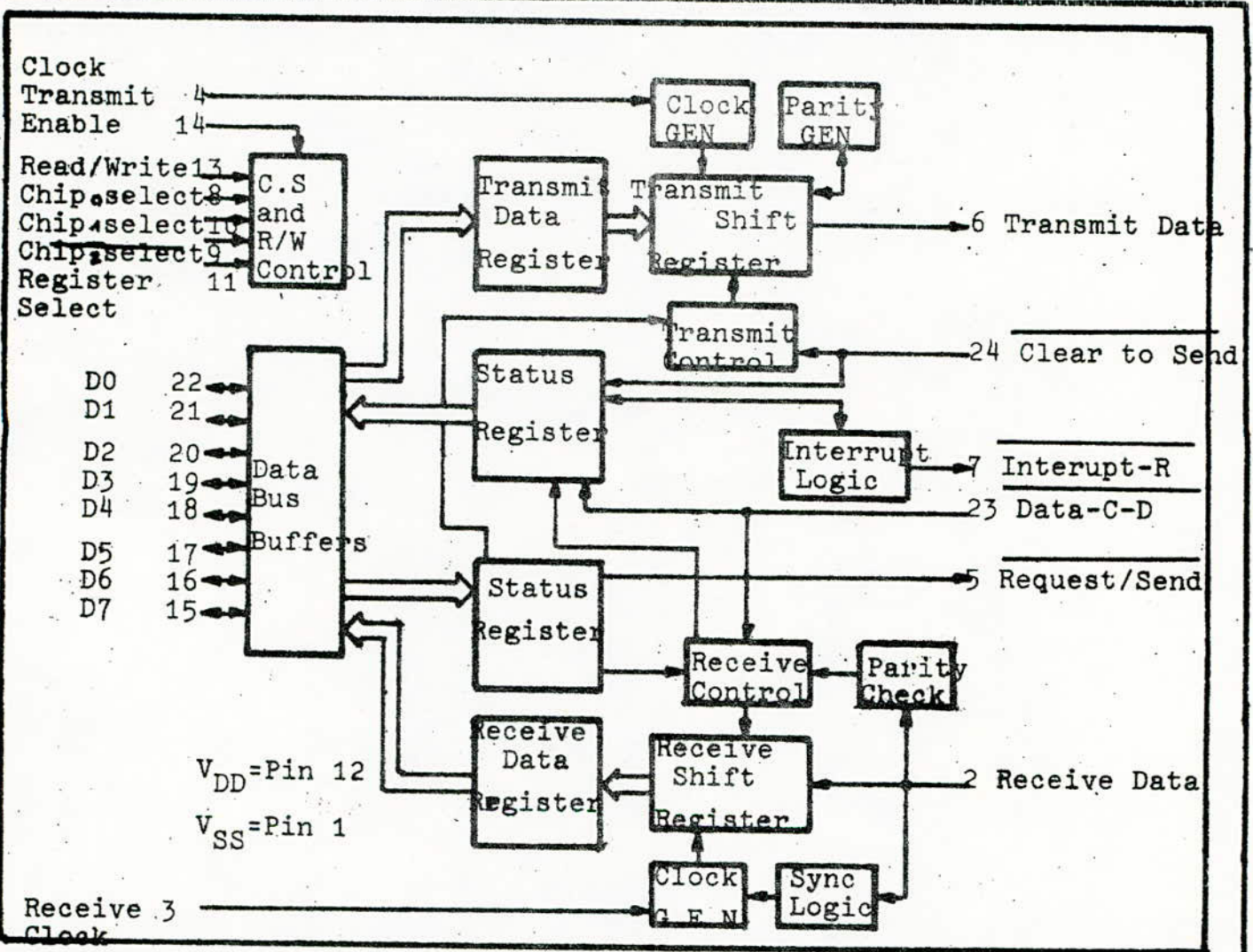


DIAGRAMME
INTERNE DE
L'ACIA

- 2 registres à lecture seulement
 - RDR (Régistre de réception des données)
 - RS (Régistre d'état).
- 2 registres à écriture seulement
 - TDR (Régistre de transmission de données)
 - CR (Régistre de contrôle)

On a donc 2 types de position mémoire qui peuvent être adressables comme 2 positions de mémoire. La sélection entre les 4 registres s'effectuant automatiquement à l'intérieur du circuit d'entrée/sortie selon que le microprocesseur lit ou écrit dans l'ACIA

OPERATIONS DE BASE DE L'ACIA

CS2	CS1	CS0	RS	R/W	Type d'opération
0	1	1	0	0	Bus de données ----> CR
0	1	1	0	1	SR ----> Bus de données
0	1	1	1	0	Bus de données ----> TDR
0	1	1	1	1	RDR ----> Bus de données

VI LES DIFFERENTS SIGNAUX D'INTERFACE ENTRE L'ACIA ET LE MICROPROCESSEUR -(Fig 2)

L'ACIA communique avec le microprocesseur par:

- Le bus de données bidirectionnel.
- Le bus d'adresse par l'intermédiaire des broches de sélection du boîtier CS0,CS1,CS2 et par l'entrée de sélection de registre RS.(L'un des chips select CS0,CS1,CS2 est généralement relié au signal de validation d'adresse VMA pour le synchronisme).La sélection de l'ACIA se fait quand CS0 et CS1 sont à l'état haut et CS2 à l'état bas.Le transfert des données de ou vers l'ACIA est alors exécuter sous le contrôle des signaux E,R/W et RS qui permet la sélection d'un des registres de l'ACIA.

-Le bus de controle par l'intermédiaire de:

-E, entrée d'activation reliée généralement à \emptyset_2 puisque E active le boitier si un "1" est appliqué; or les échanges de données sur le bus ayant lieu pendant l'état haut de \emptyset_2 , nous avons ici un moyen de synchronisation idéal.

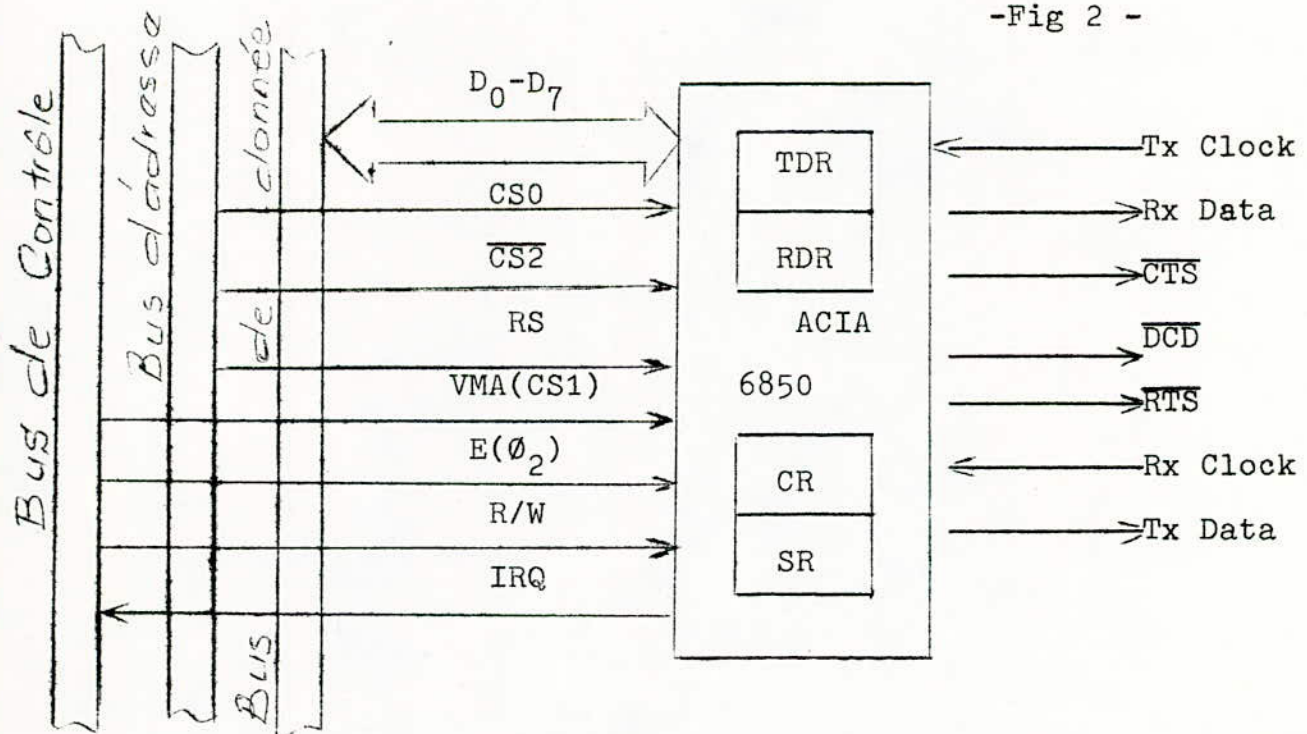
-L'entrée lecture/écriture=R/W=0 écriture

R/W=1 lecture

-La ligne de demande d'interruption $\overline{\text{IRQ}}$:

Cette broche passe à l'état bas et adresse sous certaines conditions une demande d'interruption au microprocesseur, si le registre de transmission est vide, si le registre de réception est plein, si le récepteur est en surcharge etc...

La détection d'une interruption par le microprocesseur implique le branchement à un sous programme d'identification de la cause de l'interruption (demande de transmission ou de réception de caractère...). Cette identification se fait par la lecture du registre d'état et par des rotations sur le contenu de ce registre.



VII LES DIFFERENTS SIGNAUX D'INTERFACE ENTRE LA PERIPHERIE ET L'ACIA:

L'ACIA Communique avec la périphérie par:(Fig 2)

-Les lignes de donnée par l'intermédiaire des broches de transmission Tx Data et de réception Rx Data. Le véhiculement de l'information se fait sous le controle des signaux de l'horloge Tx Clock et Rx Clock.

-Les lignes de controle véhiculent 3 signaux:

La sortie \overline{RTS} (Request to send) qui se met à l'état bas dès que l'ACIA demande à transmettre une donnée et provoque l'émission de la porteuse ligne par le MODEM.

L'entrée \overline{DCD} qui permet le controle automatique du recepteur de l'ACIA par le modem qui lui est associé.

L'entrée \overline{CTS} (Clear to send) assure le controle du modem sur la transmission. Un niveau logique bas est une autorisation de transmission, un niveau logique haut est une inhibition.

VIII PROGRAMMATION DE L'ACIA

D'un point de vue programmation, l'ACIA est extremement souple et puissant les organigrammes de la figure 3 nous aident à comprendre ce qu'il convient de faire. Il est à noter que l'ACIA ne dispose pas de broche de remise à l'état initial.

A la mise sous tension , une logique interne de mise à l'état initial détecte la transmission de la ligne d'alimentation et maintient ce circuit dans un état de mise à zéro.

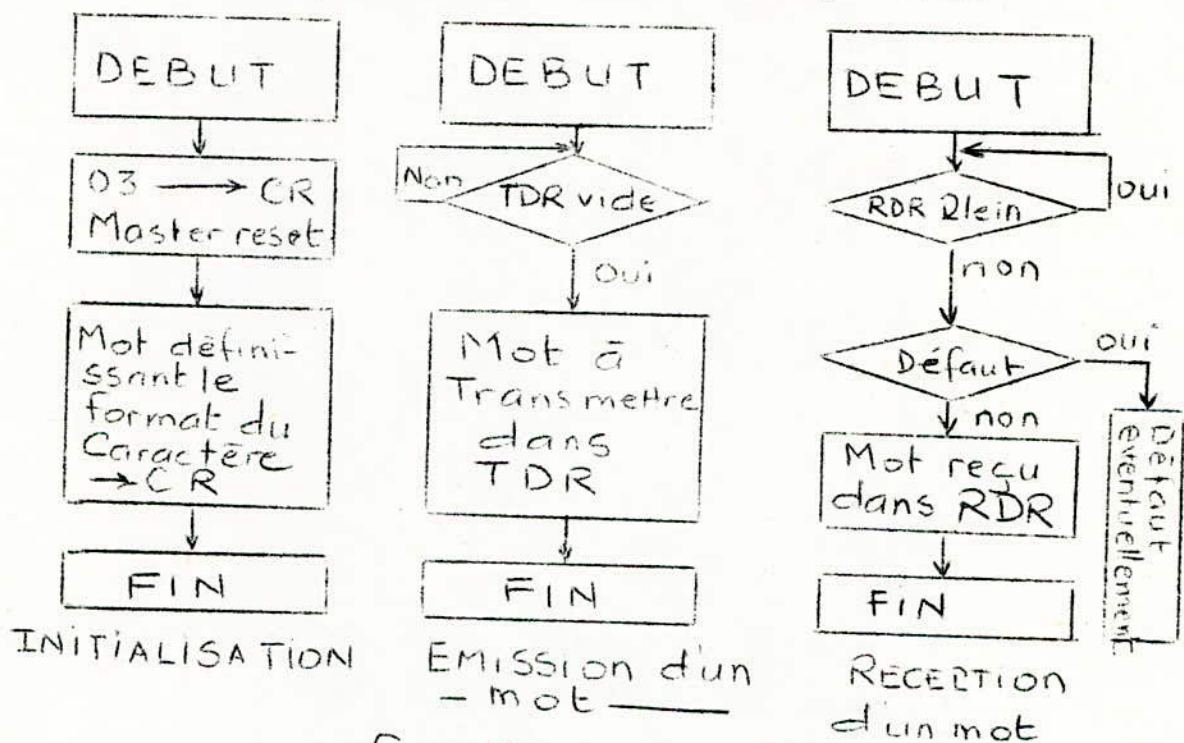
Cette logique dépend de la netteté des transitions de la ligne d'alimentation.

L'ACIA sort de la mise à zéro par la programmation des bits 0 et 1 à l'état logique "1" du registre de contrôle "MASTER RESET" qui doit être effectué avant d'utiliser l'ACIA. Après avoir effectué la mise à l'état initial, on écrit dans le registre de contrôle un mot définissant la configuration fonctionnelle de l'ACIA:

Longueur du caractère

Parité du caractère

Rapport de division de l'horloge etc...



-fig 3-

La figure 4 donne la programmation de l'ACIA (signification de chaque bit du registre de contrôle CR.)

La figure 5 donne la signification de chaque bit du SR.



REGISTRE DE CONTROLE

AUTORISATION DES INTERRUPTIONS

PAR LE RECEPTEUR -----

1 AUTORISATION

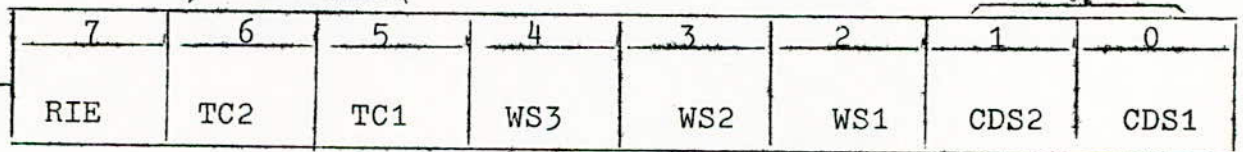
0 INHIBITION

CONTROLE DU TRANSMETTEUR

CR ₆	CR ₅	FONCTION
0	0	$\overline{\text{RTS}}$ bas inhibition des interrup.par le trans.
0	1	$\overline{\text{RTS}}$ bas autorisation des interru. par le transme.
1	0	$\overline{\text{RTS}}$ haut inhibition des interr. Par le transmet.
1	1	$\overline{\text{RTS}}$ bas envoie d'un caractère (Break) inhibition des interruptions.

SELECTION DU RAPPORT DE DIVISION DE L'HORLOGE ET REMISE A L'ETAT INITIAL

CR ₁	CR ₀	FONCTION
0	0	Division par 1
0	1	Division par 16
1	0	Division par 64
1	1	Mise à l'état initial programmé.



FORMAT DU CARACTERE

CR ₄	CR ₃	CR ₂	FONCTION
0	0	0	7 bits + Parité Paire + 2 bits Arrêt
0	0	1	7 bits + Parité Impaire + 2 bits Arrêt
0	1	0	7 bits + Parité Paire + 1 bit Arrêt
0	1	1	7 bits + Parité Impaire + 1 Bit Arrêt
1	0	0	8 bits + 2 bits Arrêt
1	0	1	8 bits + 1 bit Arrêt
1	1	0	8 bits + Parité Paire + 1 bit Arrêt
1	1	1	8 bits + Parité Impaire + bit Arrêt



REGISTRE D'ETAT

INHIBITION DE L'EMETTEUR \overline{CTS}

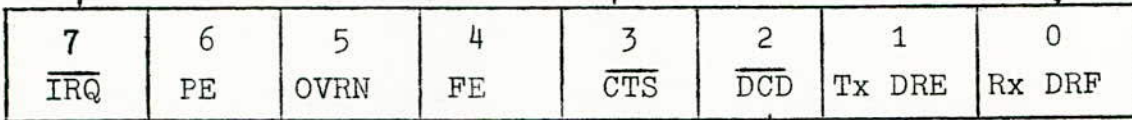
$E_3=0$ Le modem associé à l'ACIA est prêt à émettre.
 $E_3=1$ Inhibition de la transmission.

DEMANDE D'INTERUPTION IRQ

Demande d'interruption causée par E_0, E_1, E_2 , ou E_5 ; remis à 0 par la remise à l'état initial programmé.

REGISTRE DE RECEPTION PLEIN

$E_0=0$ Registre de reception vide.
 $E_0=1$ Registre de reception plein (génère une interruption)
 E_0 Lorsqu'il est à 1 est remis à 0 par la lecture du registre de reception ou par la mise à l'état haut de la broche DCD.



PRETE DE LA PORTEUSE DCD

$E_2=0$ Presence de la porteuse.
 $E_2=1$ Perte de la porteuse.
 E_2 , lorsqu'il est à 1 est remis à 0 par la lecture du registre d'état et de reception ou par la mise à l'état initial programmé.

REGISTRE DE TRANSMISSION VIDE

$E_1=0$ Registre de transmission plein.
 $E_1=1$ Registre de trans. vide.
 E_1 , lorsqu'il est à 1 est remis à 0 par le transfert d'une donnée dans le registre de de transmission ou par la mise de la broche CTS à l'état haut.

ERREURS DE RECEPTION

$E_4=1$ Erreur de format (absence du 1^{er} bit d'arrêt).
 $E_5=1$ Surchage du récepteur met E_0 à 1 et génère une interruption.
 $E_6=1$ Erreur de parité.
 E_4 est remis à 0 à la prochaine réception du caractere si le format a été rectifié.
 E_5 et E_0 sont remis à 0 par la lecture du registre de reception.
 E_6 est remis à 0 lors de la tansmission de donnéé dans l'ACIA.

IX TRANSMISSION ASYNCHRONE:

Elle est initialisée par une demande d'interruption du microprocesseur produite par les signaux émis sur la broche $\overline{\text{IRQ}}$ de l'ACIA; ensuite le microprocesseur lit systématiquement le bit Tx DRE du registre d'état pour savoir si le TDR est vide.

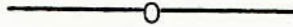
Une fois que la phase de préparation est terminée, un caractère est transmit dans le TDR puis est transféré dans le registre à décalage et ensuite envoyé en série sur une ligne de transmission de donnée précédé d'un bit de départ "START" et suivie d'un ou de deux bits d'arrêt "STOP" et d'un bit de parité éventuellement.

X RECEPTION ASYNCHRONE:

Les données de la périphérie sont reçues par l'ACIA sur la broche d'entrée(Rx DATA). Une synchronisation entre l'horloge et les données permet une réception série à la fréquence de l'horloge ou à des fréquences 16 fois, 64 fois moindres que celle de l'horloge.

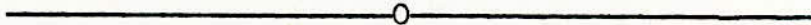
Lorsque le mot est transmis dans le RDR, l'ACIA génère des signaux d'interruption du microprocesseur sur la broche $\overline{\text{IRQ}}$. Le microprocesseur lit le registre d'étatsystématiquement pour savoir si le registre de réception est plein (bit RDRF) et éventuellement pour détecter les erreurs de parité, format... avant la lecture du registre de réception RDR.

CHAPITRE III



LE CIRCUIT D'INTERFACE PARALLELE LE "PIA"

- I INTRODUCTION
- II SYNOPTIQUE INTERNE DU PIA
- III LES DIFFERENTS SIGNAUX D'INTERFACE ENTRE LE PIA ET LE MICROPROCESSEUR
- IV LES DIFFERENTS SIGNAUX D'INTERFACE ENTRE LE PIA ET LA PERIPHERIE
- V PROGRAMMATION DU PIA



I INTRODUCTION:

Le circuit d'interface PIA (Péripheral Interface Adapter) a été conçu pour des applications générales, couvrant des domaines aussi variés que la conversion analogique/numérique, l'interface clavier afficheur numérique. Il assure l'interfaçage entre le microprocesseur et les périphériques qui ne sont pas synchronisés par l'horloge de ce micro-ordinateur. C'est un circuit programmable, destiné pour des transmissions et réception parallèles.

II SYNOPTIQUE INTERNE DU PIA :

Le PIA se compose de:

-De deux registres chacun ayant 8 bits, l'ORA et l'ORB sont les registres de données de sortie.

-De quatre registres de contrôle; deux de ces registres sont affectés au canal A et les deux autres au canal B.

* Les registres de sens de transfert de donnée le DDRA et le DDRB, associés aux canaux A et B, permettent de définir le sens de transfert des données.

A chaque bit de ces registres est associée une ligne de donnée. Un "1" correspond à une sortie c'est à dire le transfert s'effectue du PIA vers le périphérique. Par contre un "0" logique est considéré comme une entrée.

* Les registres de contrôle et d'état CRA et CRB ils permettent au microprocesseur de commander par programme les quatre lignes de contrôle CA1, CA2, CB1, CB2; et permettent aussi d'autoriser les interruptions sur IRQA, IRQB et de tester sur les bits 6 et 7 l'état des indicateurs d'interruption.

Les bits 0 à 5 peuvent être lus ou écrits par le microprocesseur par contre les bits 6 et 7 ne sont pas que lus et sont modifiés par les interruptions externes sur les lignes de contrôle CA1, CA2, CB1, CB2.

III LES DIFFERENTS SIGNAUX D'INTERFACE ENTRE LE PIA
ET LE MICROPROCESSEUR:

Le PIA dialogue avec le microprocesseur par:

-Le bus de donnée bidirectionnel (D_0-D_7)

-Le bus d'adresse comprenant:

. Les signaux de sélection du boîtier ($CS_0, CS_1, \overline{CS_2}$) généralement l'un des chips select est synchronisé avec le signal de validation, d'adresse VMA envoyé par le microprocesseur pour empêcher des lectures intempestives des registres du PIA lorsque le bus d'adresse est à haute impédance et évite le masquage des interruptions.

. De deux signaux de sélection d'un registre du PIA, RS_0, RS_1 .

-Un signal de synchronisation, appliqué sur la broche E, ce signal est généralement \emptyset_2 .

-Un signal de lecture/écriture qui permet de distinguer une opération d'entrée de donnée d'une opération de sortie. Le tableau de la figure 2 donne les opérations de base du PIA.

-Un signal RESET de remise à zéro des registres du PIA.

CS_2	CS_1	CS_0	RS_1	RS_0	B_2	R/W	TYPE D'OPERATION
0	1	1	0	0	0	0	SORTIE:BUS DONNEE → REG. DDRA
0	1	1	0	0	1	0	SORTIE:BUS DONNEE → CANAL A
0	1	1	0	0	1	1	ENTREE:CANAL A → BUS DONNEE
0	1	1	0	1	X	0	SORTIE:BUS DONNEE → REG. CRA
0	1	1	0	1	X	1	ENTREE:REG. CRA → BUS DONNEE
0	1	1	1	0	0	0	SORTIE:BUS DONNEE → REG. DDRB
0	1	1	1	0	1	0	SORTIE:BUS DONNEE → CANAL B
0	1	1	1	0	1	1	ENTREE:CANAL B → BUS DONNEE
0	1	1	1	1	X	0	SORTIE:BUS DONNEE → REG CRB
0	1	1	1	1	X	1	ENTREE:REG CRB → BUS DONNEE

(Fig 2) TABLEAU DES OPERATIONS DE BASE DU PIA.

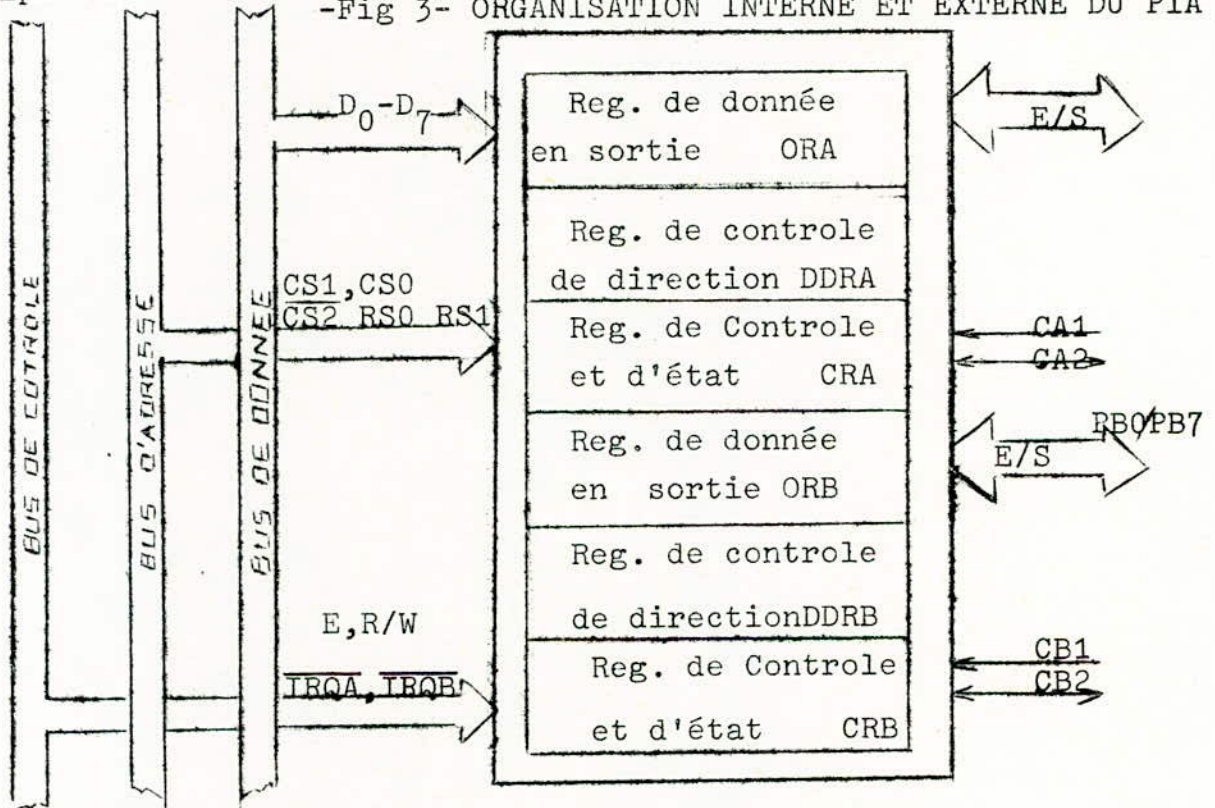
IV LES DIFFERENTS SIGNAUX D'INTERFACE ENTRE LE PIA ET LE MICROPROCESSEUR:

Le PIA communique avec la périphérie par 20 Lignes (Fig 3). Ces lignes comprennent:

- 16 lignes de données individuellement programmables.
- 4 lignes de contrôle, qui permettent d'envoyer des demandes d'interruption à partir de la périphérie.

Deux de ces lignes peuvent être programmées en sortie de commande de la périphérie.

-Fig 3- ORGANISATION INTERNE ET EXTERNE DU PIA



V PROGRAMMATION DU PIA:

Le PIA est un circuit dont la programmation est assez complexe puisqu'il possède quatre registres de controle, donc il lui faut plusieurs mots pour définir sa fonction. Par contre il possède une très grande souplesse d'utilisation puisque ces lignes de données sont programmables individuellement.

La figure 4 représente la signification de la programmation de chaque bit des registres de controle.

CHOIX DE LA TRANSITION ACTIVE DE CA1

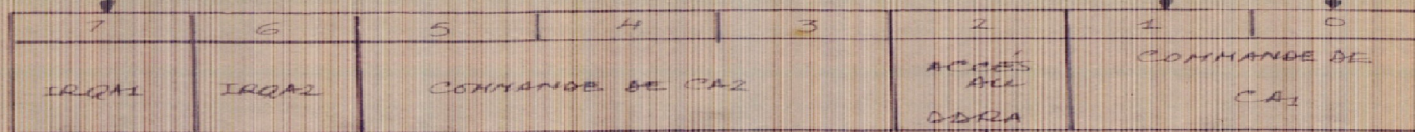
$CRA_7 = 0$, CRA_6 mis à 1 par une transition négative de CA_1
 $CRA_1 = 1$, CRA_7 mis à 1 par une transition positive de CA_1

INDICATEUR D'INTERRUPTION CRA_7

Mis à 1 par une transition active sur la ligne CA_1
 Remis à 0 par une lecture du registre de donnée correspondant et par RESET.

AUTORISATION DU SIGNAL DE DEMANDE D'INTERRUPTION

$CRA = 0$ masque les demandes d'interruption sur CA_1
 $CRA = 1$ autorise les demandes d'interruption sur CA_1



INDICATEUR D'INTERRUPTION CRA_6

Si CA_2 est une entrée d'interruption ces indicateurs sont mis à 1 par une transition active de CA_2
 Remis à 0 par une lecture du registre de donnée correspondant et par RESET.

SELECTION DU REGISTRE DE DONNÉE OU DU REGISTRE DE DIRECTION

$CRA_2 = 0$ Sélection du registre de direction
 $CRA_2 = 1$ Sélection du registre de donnée écrite
 CA_2 est obtenu par la combinaison de R_0 et R_1 .

CRA_5 CRA_4 CRA_3

1	0
---	---

CA_2 envoie une impulsion vers le microprocesseur indiquant l'activation d'une opération de lecture.
 $CRA_5 = 0$, CA_2 passe à l'état bas sur la première transition négative de CA_2 qui suit une lecture d'une donnée provenant de la périphérie A.
 CA_2 passe à l'état haut quand CRA_5 est mis à 1 par une transition positive de CA_2 . CA_2 passe à l'état bas sur la transition négative de CA_2 qui suit une impulsion de CA_2 qui suit une lecture provenant de la périphérie A.
 CA_2 passe à l'état haut à la prochaine transition négative de CA_2 .
 CA_2 envoie une impulsion vers le bus interne indiquant une écriture d'une donnée dans le registre CRB .
 CRA_4

CA_2 passe à l'état bas sur la transition positive de la 1^{ère} impulsion de CA_2 qui suit une écriture du registre de donnée CRB .

CA_2 passe à l'état haut quand l'indicateur d'interruption CRA_5 est mis à 1 par une transition active du signal CA_1 .

$CRA_5 = 1$
 CA_2 passe à l'état bas par la transition positive de la 1^{ère} impulsion de CA_2 qui suit une écriture du registre de donnée CRB .

CA_2 passe à l'état haut à la prochaine transition positive de CA_2 .

CRA_5 CRA_4 CRA_3

1	1
---	---

Generation de l'impulsion CA_2
 CA_2 passe à l'état bas quand le MRD est 0 dans CRA_3 et passe à l'état haut inversement.

CRA_5 CRA_4 CRA_3

0		
---	--	--

Autorisation du signal de demande d'interruption
 $CRA_3 = 0$ masque l'interruption (CA_2)
 $CRA_3 = 1$ autorise les demandes d'interruptions sur CA_2 .

CHOIX DE LA TRANSITION ACTIVE DE CA_2

$CRA_4 = 0$, CRA_6 mis à 1 par une transition négative de CA_2 .

$CRA_4 = 1$, CRA_6 mis à 1 par une transition positive sur CA_2 .

REGISTRE DE CONTROLE CRA

ou $R1A$

Remarque: Ces explications sont valable pour le CRB

CHAPITRE IV

0
0

LE PERIPHERIQUE: LE LECTEUR OPTIQUE MODELE 601

I _ INTRODUCTION

II _ DESCRIPTION DU LECTEUR OPTIQUE MODELE 601

III _ LES SIGNAUX DE COMMANDE DU LECTEUR OPTIQUE

IV _ QUELQUES CARACTERISTIQUES DU LECTEUR OPTIQUE

V _ ETUDE DU RUBAN A HUIT (8) PERFORATIONS

— 0 —

I INTRODUCTION

L' exécution d'un programme entraîne un échange d'informations entre le microprocesseur et sa mémoire de données et d'instructions. Cette mémoire constitue la mémoire centrale. Cette dernière a une capacité très limitée. Elle est constituée de deux parties:

R.O.M (READ ONLY MEMORY) mémoire non volatile.

R.A.M (RANDOM ACCES MEMORY) mémoire volatile.

Afin d'y remédier à cet inconvénient on associe au processeur des mémoires auxiliaires, dont le but est de stocker l'information et de la conserver, telles que la bande magnétique, le disque souple, la cassette, la bande perforée etc... Chacune des mémoires auxiliaires nécessite un circuit de lecture ou d'écriture qu'on appelle la périphérie.

Le périphérique est couplé en général au microprocesseur par un circuit d'interface d'entrée/sortie car la vitesse d'échange d'information avec les mémoires auxiliaires est différente de celle de l'unité centrale.

Dans le cadre de ce chapitre, nous étudierons le lecteur Optique Modèle 601. Mais brièvement; car l'étude approfondie de ce lecteur nous entrainerait à étudier en détail le moteur pas à pas type ADDMASTER et toute sa partie de commande qui ne font pas objet de notre projet.

Nous limiterons l'explication à certains points que nous jugerons utiles et importants pour la compréhension et la réalisation du circuit d'interface d'entrée/sortie entre le lecteur optique et le microprocesseur.

II DESCRIPTION DU LECTEUR OPTIQUE MODELE 601 C :

Le moteur d'entrainement utilisé est un moteur pas à pas type ADMASTER. Il est très rapide et peut travailler à des températures très élevées allant jusqu'à 155° C. Son circuit de commande utilise des bascules Flip-Flop, donc le moteur peut tourner dans les deux sens.

Le ruban entraîné par une roue dentée est illuminé par une rangée de diodes "L.E.D" émettrices d'un rayon infrarouge donc non visible à l'oeil nu.

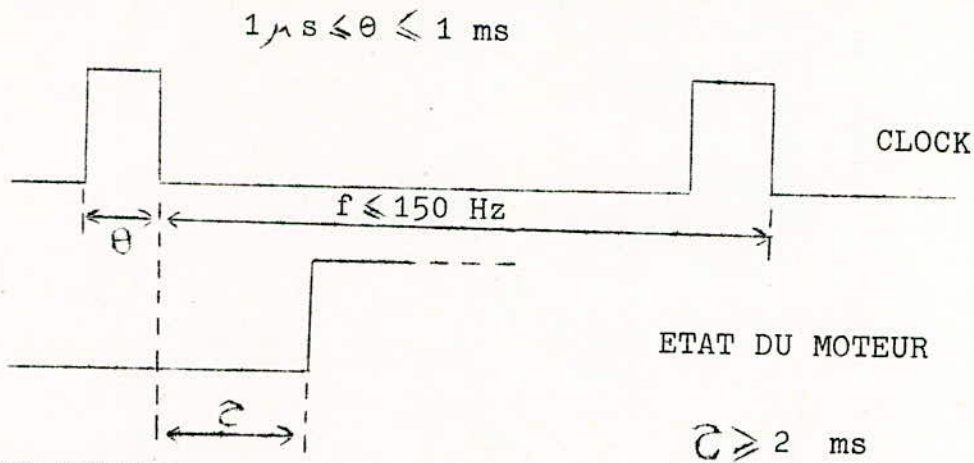
Un circuit de détection des perforations du ruban est constitué par une rangée de photo-transistors. Pour éviter que les rayons émis par les diodes divergent et perturbent les photo-transistors voisins on a placé une plaque comportant des ouvertures entre le ruban et la rangée de photo-transistor afin de canaliser les rayons infra-rouge. Le courant naissant dans les photo-transistor est amplifié puis transformé en un signal TTL.

III LES SIGNAUX DE COMMANDE DU LECTEUR OPTIQUE:

Dans ce qui suit nous allons donner la signification et le rôle de chaque signal de commande.

CLOCK:

Ce signal est une entrée pour le lecteur optique. Quand on génère une impulsion sur cette ligne le moteur avance d'un pas. Sa fréquence ne doit pas dépasser 150 Hz. La durée de l'impulsion à l'état haut doit être comprise entre 1 μ s et 1 ms. C'est la transition négative du signal clock qui fait changer l'état du moteur; mais il faut noter que le moteur possède une inertie et ne peut donc répondre qu'après 2 ms.



CLOCK ENABLE:

Ce signal est aussi une entrée pour le lecteur. Cette ligne peut ne pas être utilisée, dans ce cas là elle reste flottante.

Si ce signal est mis bas (état logique "0") le moteur s'arrête.

REMOTE START:

Cette ligne permet le contrôle du signal reader ready comme on le verra un peu plus loin. Elle est utilisée comme une entrée.

REMOTE OFF :

Ce signal contrôle aussi la ligne Reader Ready et en plus il supprime le courant dans le moteur et les diodes "L.E.D" quand il passe à l'état bas. (Etat logique "0")

READER READY:

Ce signal est une sortie pour le lecteur optique.

Reader ready est mis à l'état "1" si:

- l'interrupteur est à l'état " START "
- ou bien le signal remote start devient momentanément bas (état logique "0") et que l'interrupteur n'est pas dans la position off.

- Reader ready est mis à l'état logique "0" si:

- l'interrupteur est à l'état "off"
- le signal remote off passe à l'état logique "0" momentanément
- détection de la fin de bande.
- la bande s'est accrochée.

SPROCKET:

Ce signal lit le bit de validation.

Il indique au lecteur qu'un caractère est positionné.

Ce signal est une sortie pour le lecteur.

FORWRD/REVERSE:

Cette ligne commande le sens de rotation du moteur. Elle peut ne pas être utilisée, dans ce cas là elle reste flottante. L'état de cette ligne ne doit pas changer durant la période.

DATA BITS

Les bits de données lisent continuellement le ruban perforé; mais la lecture ne doit se faire qu'après 2ms (temps d'inertie du moteur) de la transition négative du signal clock.

La présence d'un trou est indiquée par un niveau logique "1"

IV - QUELQUES CARACTERISTIQUES DU LECTEUR OPTIQUE.

-Le moteur et les diodes "LED" peuvent être alimentés par une tension allant de 12V à 48V, mais la valeur standard est de 24V.

-Le courant dans les diodes doit être de 29 ± 3 mA

-Les signaux d'entrée et de sortie de commande du lecteur sont tous compatibles TTL:

. Un niveau logique 0 correspond à une tension inférieure à 0,5V.

. Un niveau logique 1 correspond à une tension supérieure à 2,4V.

-Le courant dans le moteur est de 0,6 A.

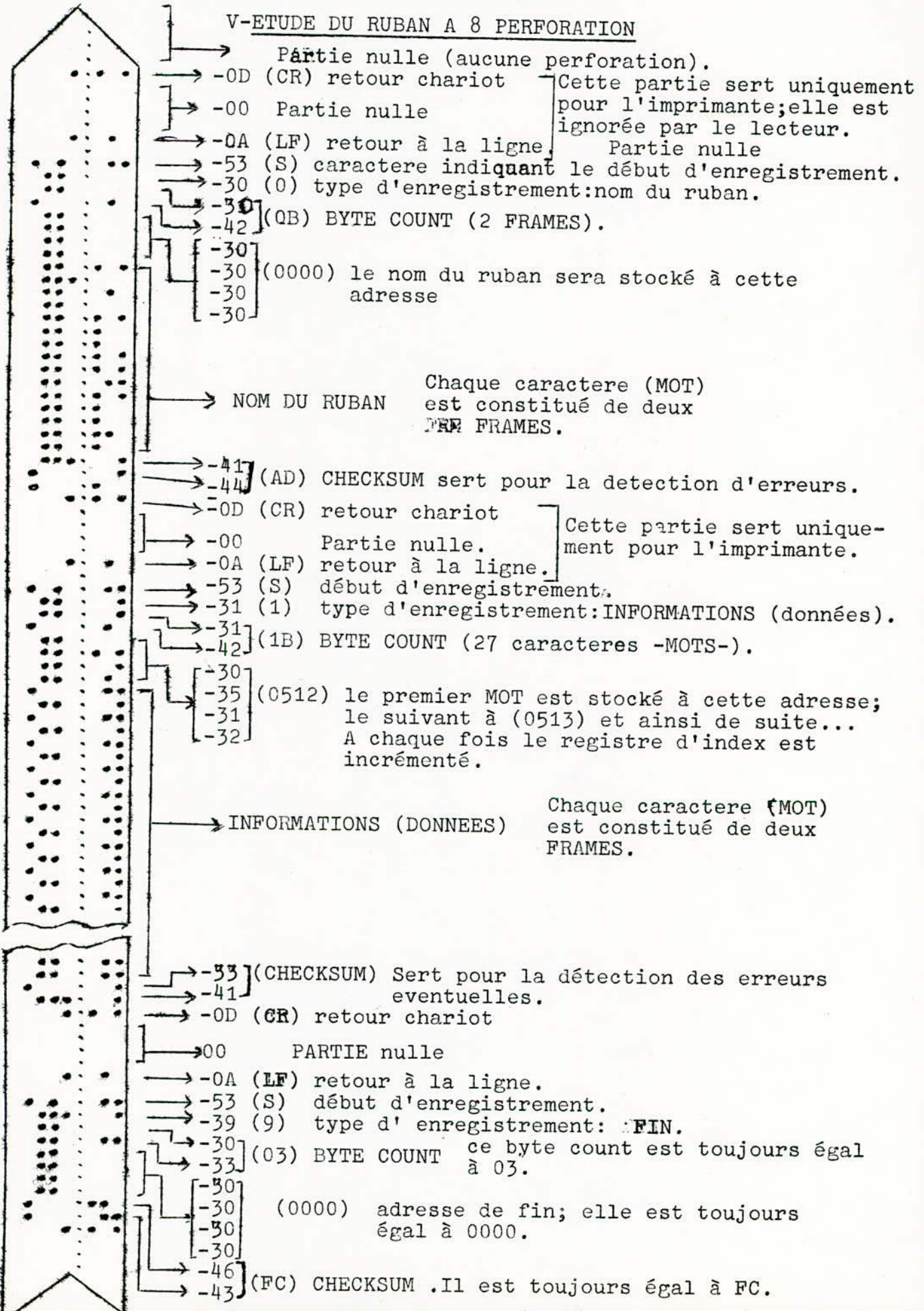
-Le lecteur optique Modèle 601 utilise plusieurs formats de ruban à 5,6,7 ou 8 perforations.

Un adaptateur est prévu pour chaque format.

V ETUDE DU RUBAN A 8 PERFORATIONS:

Une explication plus détaillée est donnée à la page suivante.

V-ETUDE DU RUBAN A 8 PERFORATION



Partie nulle (aucune perforation).

-0D (CR) retour chariot

-00 Partie nulle

-0A (LF) retour à la ligne

-53 (S) caractere indiquant le debut d'enregistrement.

-30 (0) type d'enregistrement: nom du ruban.

-30] (0B) BYTE COUNT (2 FRAMES).

-42]
-30] (0000) le nom du ruban sera stocké à cette
-30]
-30]
-30] adresse

Cette partie sert uniquement pour l'imprimante; elle est ignorée par le lecteur.

Partie nulle

NOM DU RUBAN

Chaque caractere (MOT) est constitué de deux FRAMES.

-41]

-44] (AD) CHECKSUM sert pour la detection d'erreurs.

-0D (CR) retour chariot

-00 Partie nulle.

-0A (LF) retour à la ligne.

-53 (S) debut d'enregistrement.

-31 (1) type d'enregistrement: INFORMATIONS (données).

-31] (1B) BYTE COUNT (27 caracteres -MOTS-).

-42]

-30] (0512) le premier MOT est stocké à cette adresse;
-35] le suivant à (0513) et ainsi de suite...
-31] A chaque fois le registre d'index est
-32] incrémenté.

Chaque caractere (MOT) est constitué de deux FRAMES.

INFORMATIONS (DONNEES)

-53] (CHECKSUM) Sert pour la detection des erreurs
-41] eventuelles.

-0D (CR) retour chariot

00 PARTIE nulle

-0A (LF) retour à la ligne.

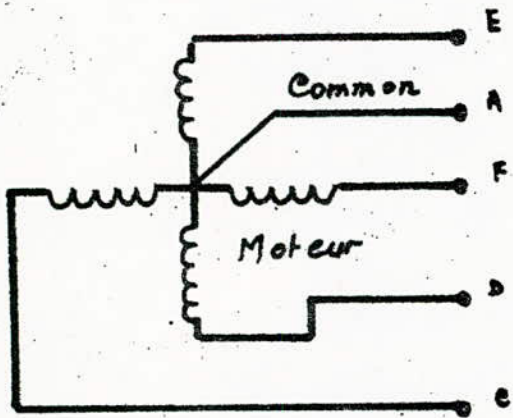
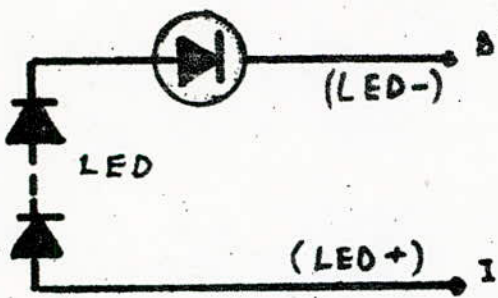
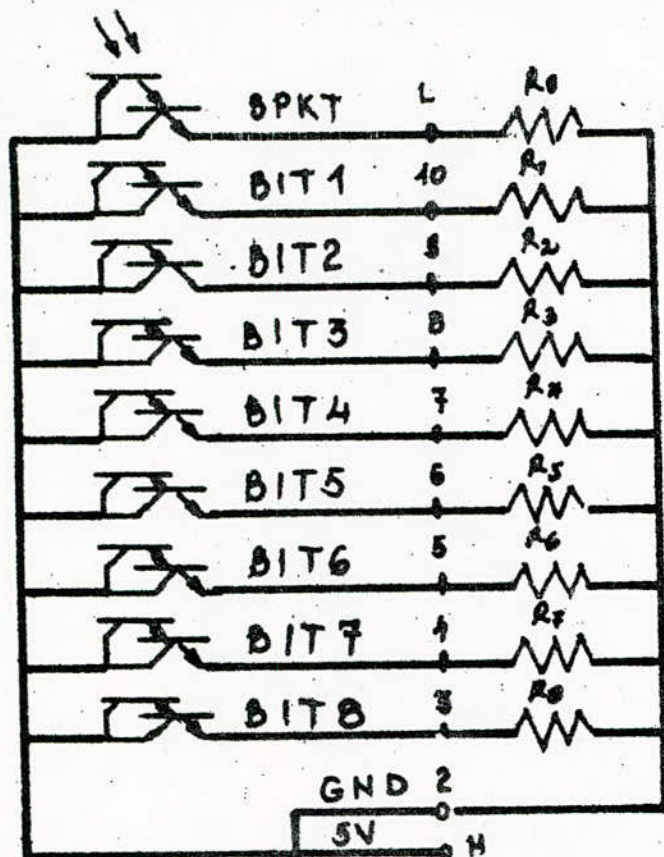
-53 (S) debut d'enregistrement.

-39 (9) type d'enregistrement: FIN.

-30] (03) BYTE COUNT ce byte count est toujours égal
-33] à 03.

-30] (0000) adresse de fin; elle est toujours
-30] égal à 0000.
-30]

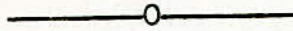
-46] (FC) CHECKSUM .Il est toujours égal à FC.
-43]



JONCTION — SHEMATIC —

— ENTREES DU CONNECTEUR —

CHAPITRE V



LE TERMINAL TEKTRONIX 4023 " UNITE DE VISUALISATION"

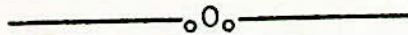
I INTRODUCTION

II SCHEMA SYNOPTIQUE

III LES LIAISONS EXTERNES DU TERMINAL
TEKTRONIX 4023

IV LES SIGNAUX DE SORTIE DE LA TEKTRONIX 4023

V QUELQUES CARACTERISTIQUES DE LA TEKTRONIX
4023.



I INTRODUCTION:

Les terminaux à écran cathodique sont un excellent moyen de communication entre l'homme et la machine.

- Il est possible d'envoyer une information à l'ordinateur et la visualiser sur un écran cathodique du type télévision.

- De visualiser des informations stockées dans la mémoire centrale; d'où un dialogue possible entre l'opérateur et l'ordinateur.

UN TEL TERMINAL COMPREND:

- Une voie de réception qui traduit sur l'écran les informations données par ce dernier, sont codées en numérique.

Aussi elles seront traduites par un convertisseur numérique-analogique.

- Une voie d'émission qui permet à l'opérateur de:

* Fournir à tout moment une information à l'ordinateur (à l'aide d'un clavier par exemple)

* Modifier une information.

* Poser une question en demandant par exemple le contenu de tel ou tel fichier etc...

Toute information ainsi transmise par l'ordinateur est aussi visualiser sur l'écran ce qui assure son contrôle.

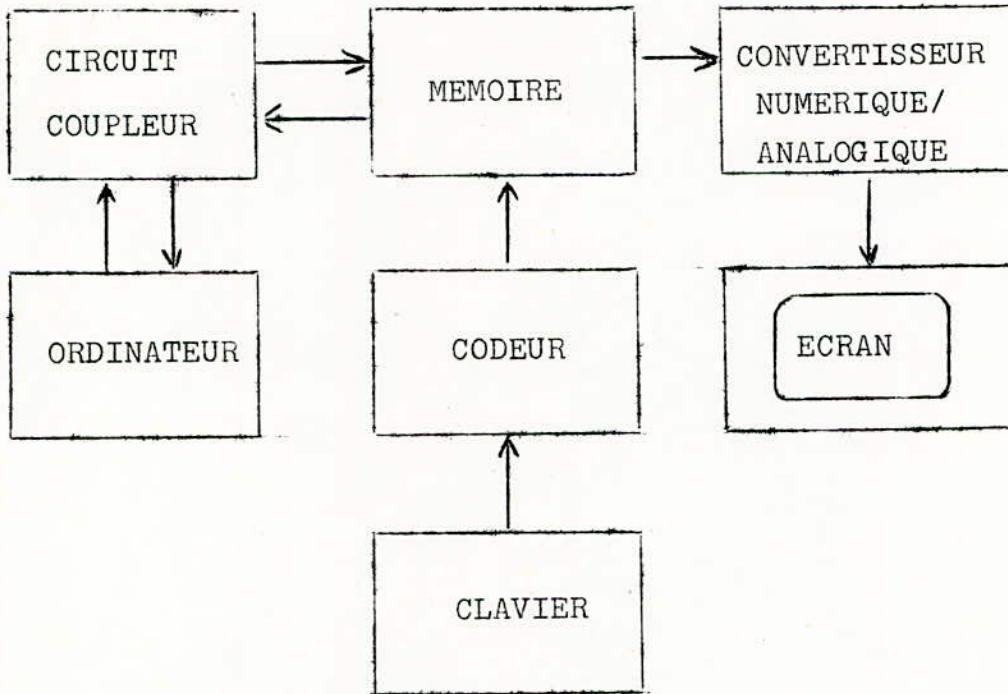
LES INFORMATIONS A VISUALISER PEUVENT ETRE:

. Purement ALPHANUMERIQUES (chiffres et lettres)
dans ce cas on a un dispositif de visualisation tabulaire.

. Graphiques (dessins de tous genres).

Dans notre projet, nous avons à établir l'interfaçage entre le système microprocesseur MC 6800 et l'unité de visualisation "TEKTRONIX 4023". Nous ne ferons pas l'étude d'un tel terminal qui est assez complexe, mais nous expliquerons certaines parties utiles pour la conception et la réalisation du circuit coupleur.

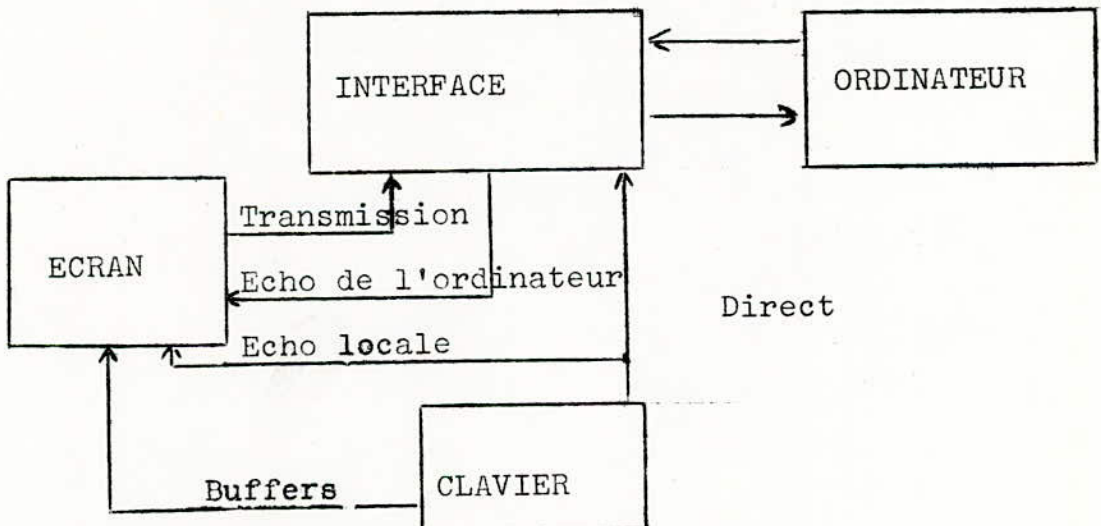
II SCHEMA SYNOPTIQUE :



III LES LIAISONS EXTERNES SE LA TEKTRONIX 4023

L'unité de visualisation TEKTRONIX 4023 peut être reliée à un ordinateur et à d'autres périphériques.

.Liaison ordinateur Terminal.



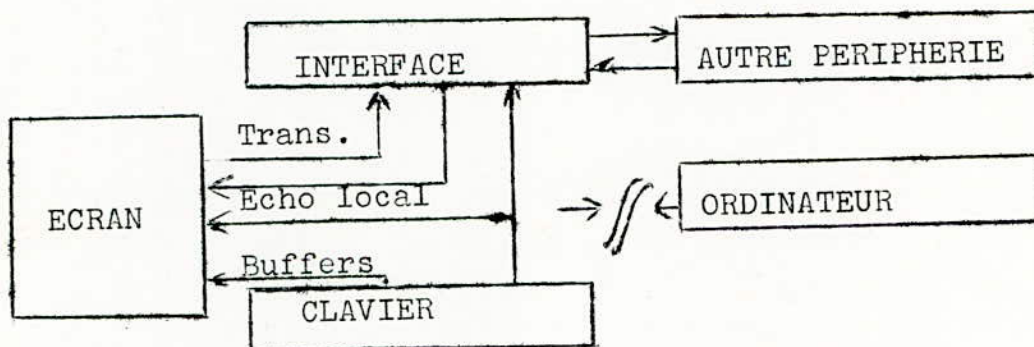
Deux types de transmission sont possibles:

. DIRECT; Les informations sont envoyées directement à l'ordinateur et en même temps visualisées sur l'écran cathodique.

. BUFFERS: Les informations sont stockées en mémoires RAM de la TEKTRONIX, l'opérateur peut faire des corrections par la suite, et ce n'est qu'à la fin qu'ils sont envoyées à l'ordinateur. Il y a toujours le phénomène d'écho qui se produit pour le contrôle de l'information.

Liaison terminal périphérique auxiliaire.

En opération LOCALE, la tektronix 4023 est isolée de l'ordinateur. Dans ce cas là on peut lui associer un périphérique.



LA TEKTRONIX 4023 est un périphérique assez puissant comme on le voit. Ce terminal possède un clavier comprenant certaines touches pouvant réaliser plusieurs fonctions.

- EFFACEMENT D'UN CARCTERE
- EFFACEMENT D'UNE LIGNE
- ECRITURE DES CARACTERES EN BLANC SUR NOIR (ou inversement)
- DEPLACEMENT DU CURSEUR ETC...

IV LES SIGNAUX DE SORTIE:

(Voir Fig 1)

Comme le montre le schéma de la Fig 1, les lignes accessibles par l'utiliasteur sont:

- . Deux lignes de transmission de données (trans. ou réception)
- . D'une ligne de retour (commun)
- . D'une masse de protection.
- . 4 Lignes de controle.

V QUELQUES CARACTERISTIQUES DE LA TEKTRONIX 4023 :

* Capacité de l'écran

-24 LIGNES

-80 CARACTERES PAR LIGNE.

* Le clavier utilise le code ASCII.

* Quand on presse une touche pendant un temps supérieur à 0,5 s il y a auto-reproduction du caractère.

* Entrée/sortie

Les lignes d'entrée/sortie et les lignes de contrôle sont compatibles avec les circuits d'interface:

- RS 232 C

- FULL DUPLEX

- Liaison série asynchrone

- START/STOP

- LSB FIRST

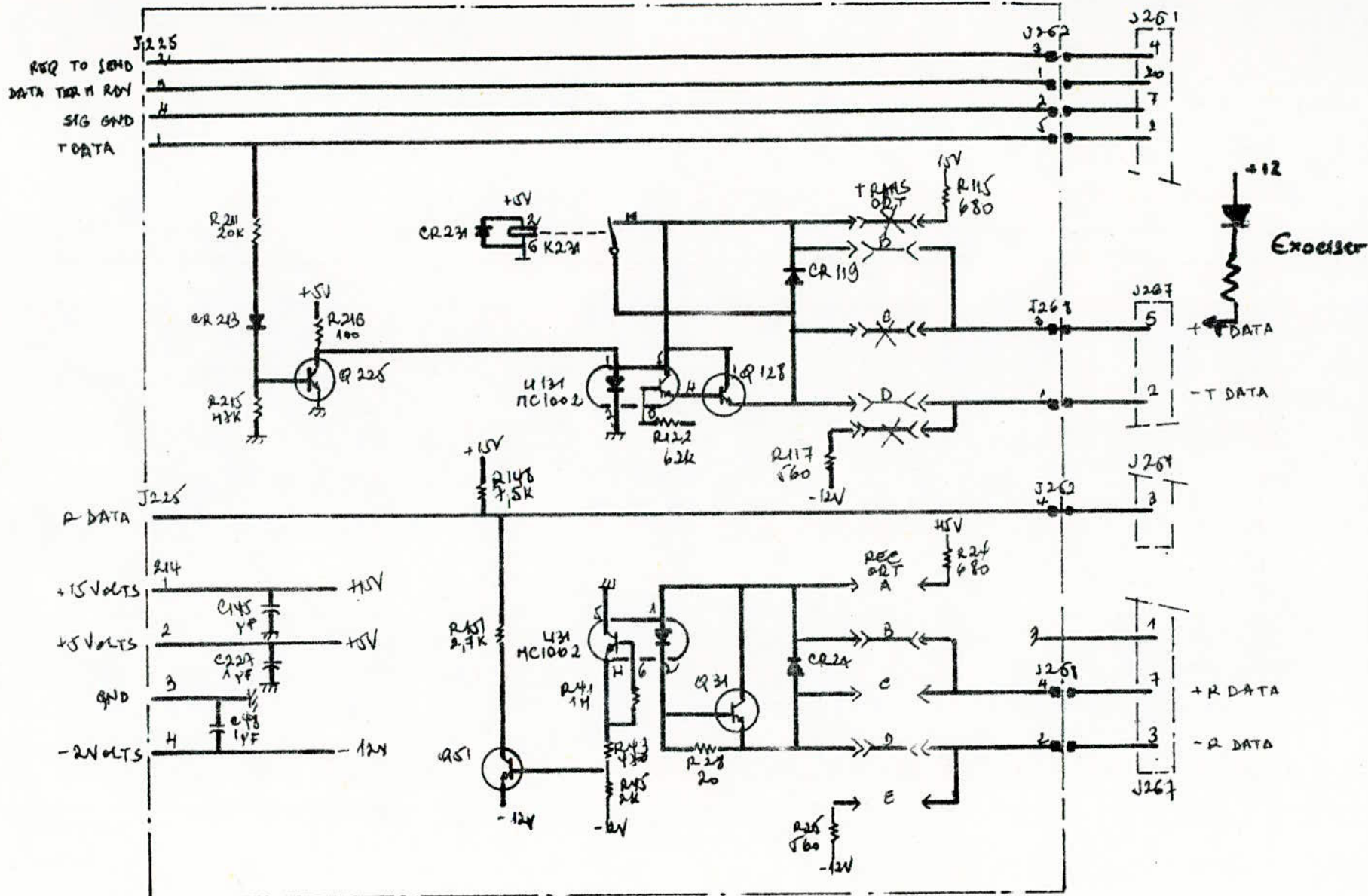
* La vitesse de transmission et de réception est donnée par le BAUD-RATE

110-160-300-600-1200-2400-4800-9600.

REMARQUE

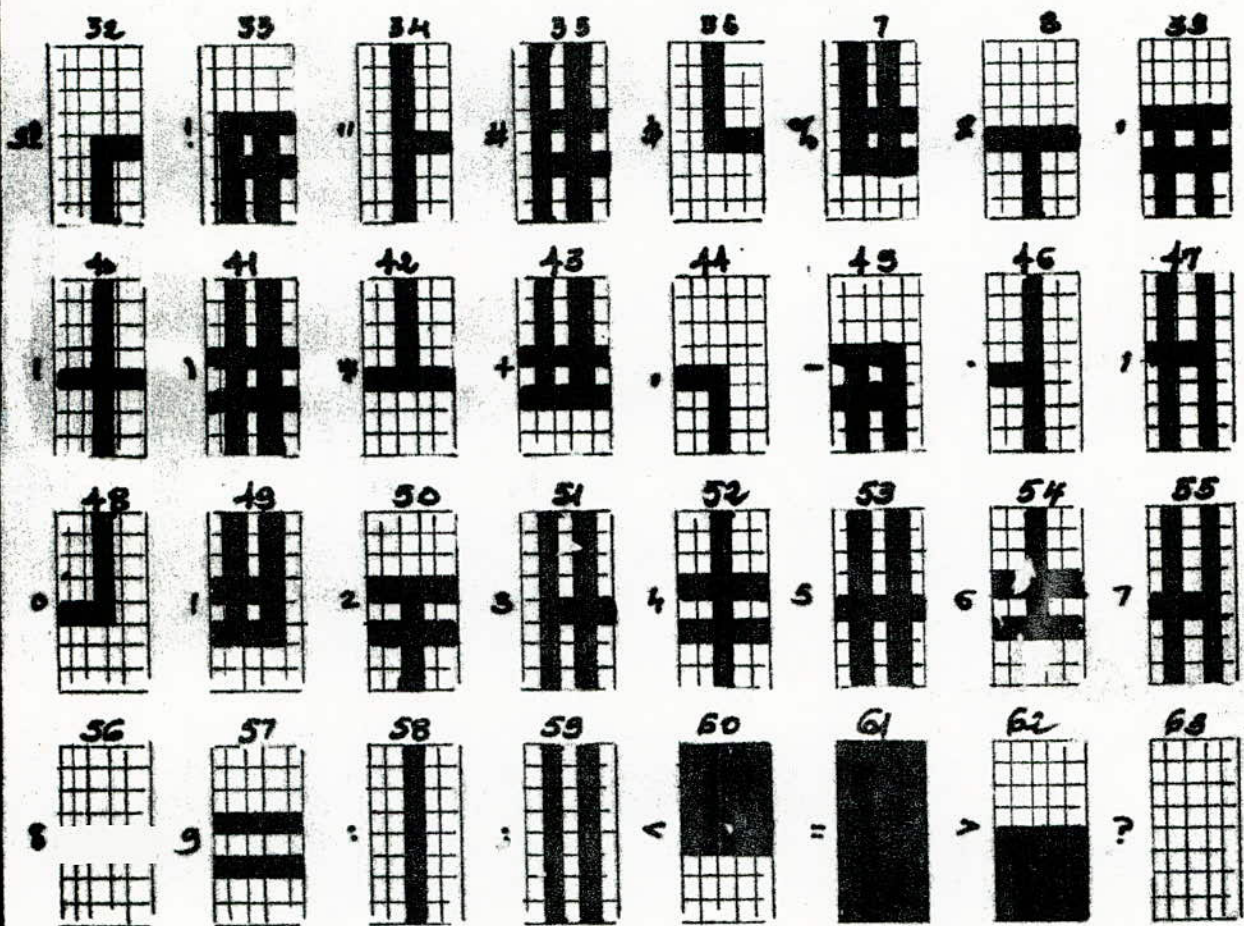
La vitesse est donnée en bauds.

UN BAUD VAUT 16 Hz.



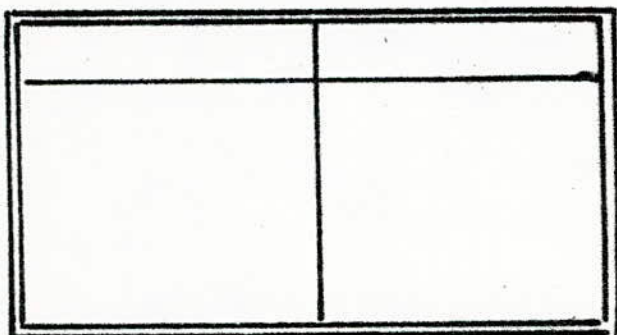
20 mA Current Loop

- * T DATA → SERIAL DATA IN
- * R DATA → SERIAL DATA OUT
- * -R DATA & -T DATA → Serial Data Common (-12V)



Elektronix 4023

Exemple d'une figure sur
l'écran.



Pour pouvoir représenter ces symboles
sur l'écran, appuyer d'abord sur
CTRL et N en même temps.

"HARDWARE" REALISATION PRATIQUE

I INTRODUCTION

II SYNOPTIQUE GENERAL DU SCHEMA.

III ETUDE DETAILLEE DE CHAQUE BLOC.

1) LIAISON MICROPROCESSEUR - COUPLEURS

*DECODEUR D'ADRESSES.

*BUFFERS.

*LOGIQUE DE COMMANDE DES BUFFERS

*RELEVÉ DES CHRONOGRAMMES DES SIGNAUX.

2) LIAISON CIRCUITS COUPLEURS - PERIPHERIQUES.

*LIAISON ACIA - UNITE DE VISUALISATION

*LIAISON PIA - LECTEUR OPTIQUE

IV ETUDE DE L'ALIMENTATION DU LECTEUR OPTIQUE

V SCHEMA GENERAL DE LA CARTE.

I- INTRODUCTION

Avant de commencer à rédiger ce chapitre nous avons longuement réfléchi et hésité sur la façon de le présenter.

En définitif, nous avons opté pour le plan suivant:

+Définition et spécification du système. C'est une partie importante qui permet de tracer les grandes lignes du projet qu'on a à réaliser. Elle permet de définir la fonction globale que doit remplir le système.

+Etudier chaque bloc constituant le système en donnant le rôle qu'il doit jouer, son utilité, ses performances, et ses limites.

+Faire des tests préliminaires d'une manière simple afin de localiser et de cerner les erreurs éventuelles qui peuvent apparaître. En effet pour la réalisation d'un circuit programmable deux problèmes peuvent apparaître:

-Erreur de programmation.

-Erreur d'un point de vue "HARDWARE"

Alors pour les tests préliminaires on évitera les longs programmes, ceci est le seul moyen de s'en sortir, Sinon on se noyera dans certaines difficultés qu'il devient pratiquement impossible de ~~pouvoir~~ détecter les erreurs.

+Faire à chaque fois une interprétation des résultats obtenus en les comparant à ceux de la théorie. Ceci permettra de savoir s'il y a conformité, sinon on réfléchira aux transformations ou améliorations que l'on doit faire.

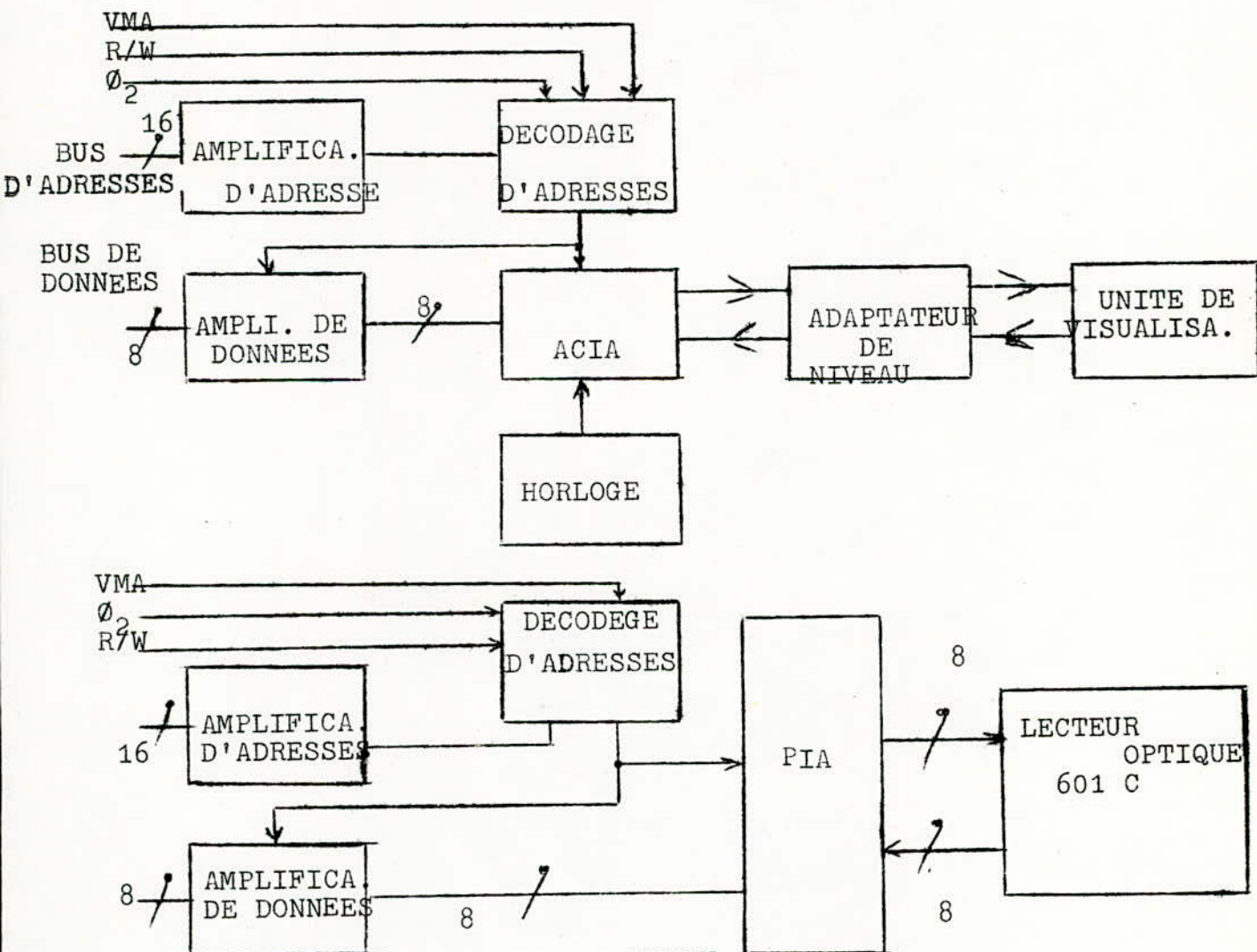
+Tester, en dernier lieu, le système dans son environnement en lui faisant exécuter la tâche qu'il doit remplir.

II SCHEMA SYNOPTIQUE GENERAL:

Avant de se lancer directement dans le schéma synoptique général, il est bon de rappeler que notre projet comporte deux parties distinctes:

- L'interfaçage entre le microprocesseur et une unité de visualisation.
- L'interfaçage entre le microprocesseur et un lecteur optique.

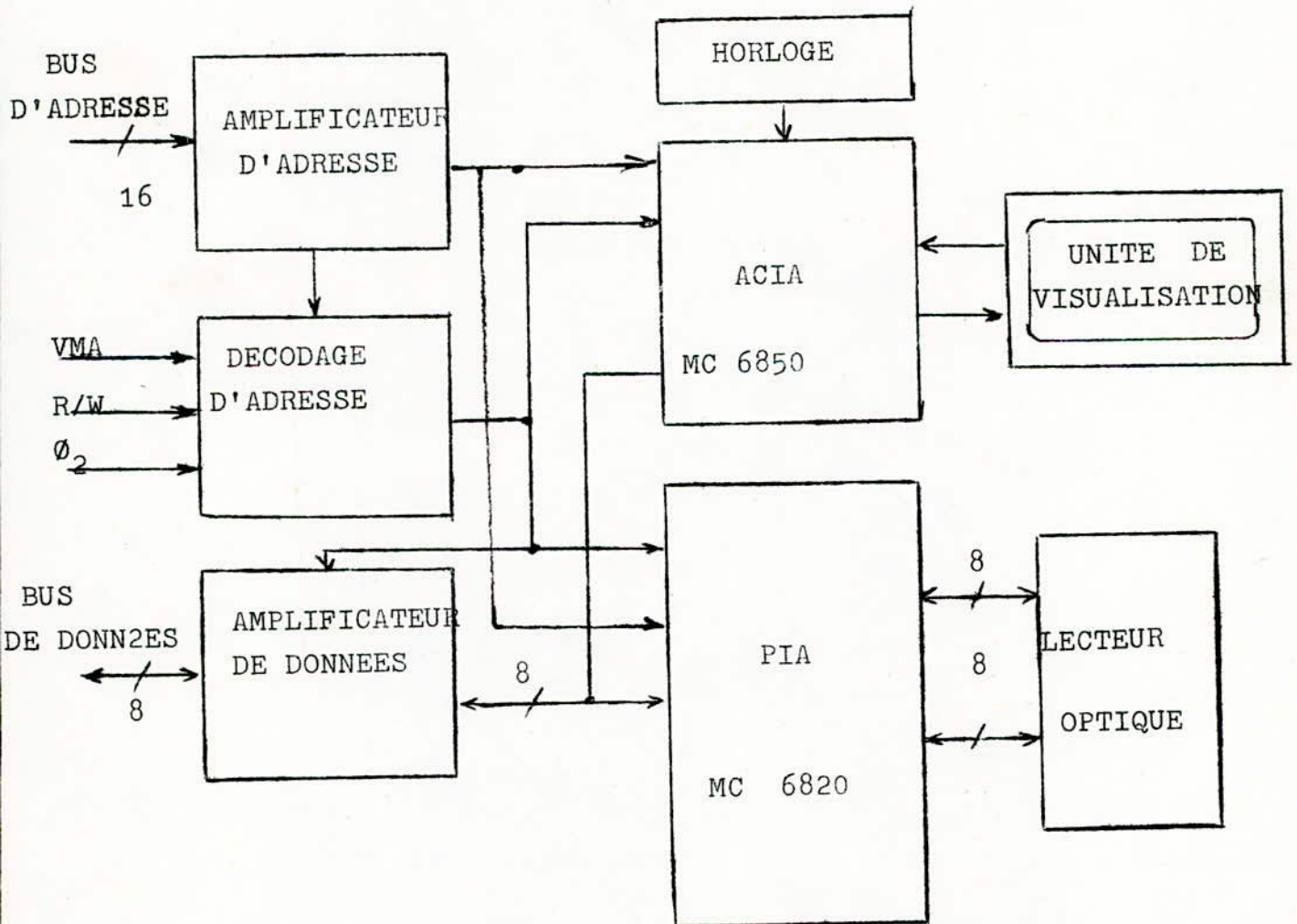
A la lueur des chapitres précédents on peut proposer les deux schemas synoptiques suivant.



Une analyse rapide des deux schémas synoptiques précédent; nous montre qu'ils sont constitués de certaines parties identiques tels que:

- L' Ampli d'adresse
- L'Ampli de données
- Décodeur d'adresse

Ces parties qui remplissent les mêmes fonctions pour chacun des deux circuits d'interface, nous ont amené à réfléchir sur la possibilité d'utiliser le même circuit pour les deux interfaces; cela nous a conduit à adopter le schéma synoptique suivant:



III ETUDE DETAILLEE DE CHAQUE BLOC (FONCTION)

1°) LIAISON MICROPROCESSEUR - CIRCUITS COUPLEURS

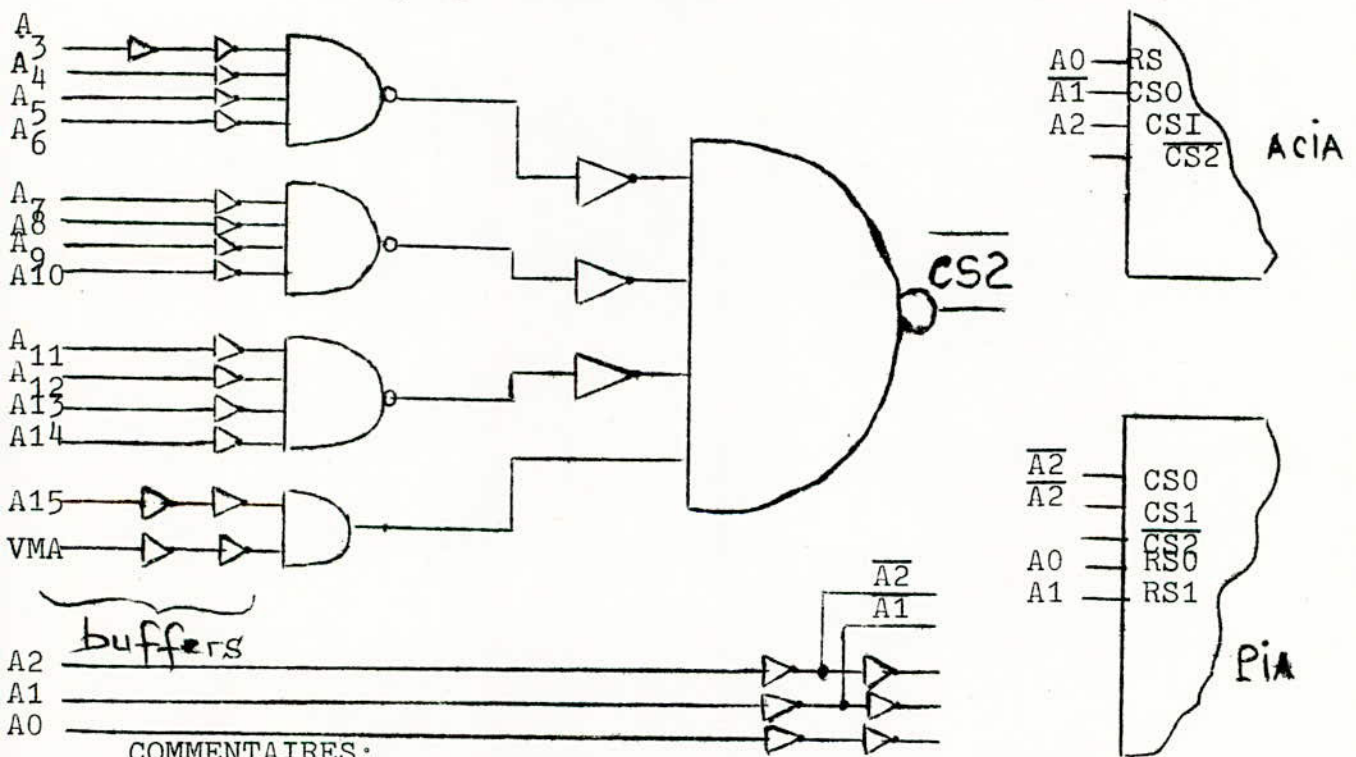
DECODAGE D'ADRESSE:

Le microprocesseur MC 6800 de motorola considère les circuits d'entrée/sortie comme des positions mémoires. Il les adresse à travers 16 lignes " BUS D'ADRESSE".

Nous rappelons que le circuit d'entrée/sortie ACIA est considéré comme deux positions mémoires et le PIA comme quatre positions mémoires.

Comme nous avons opté à réaliser un seul circuit décodeur; celui ci doit pouvoir adresser six positions memoires.

nous proposons le schéma du décodeur suivant:



COMMENTAIRES:

Une étude simple de cette logique montre:

* Le chip select $\overline{CS2}$ est égale à "0" uniquement pour la combinaison: $A_{15}=A_3=1$

$$A_{14}=A_{13}=A_{12}=A_{11}=A_{10}=A_9=A_8=A_7=A_6=A_5=A_4=0$$

Donc on a les adresses de 8008 à 800F qui seront validées par le décodeur .

* Les lignes A_0, A_1, A_2 permettent de sélectionner une seule adresse parmi les huit citées ci-dessus.

comme l'on a besoin uniquement de six adresses parmi les huit, il faut donc éliminer deux. Dans le paragraphe réservé à l'étude des buffers nous montrerons comment on y arrive.

Le PIA répond aux adresses 8008 à 800B.

L'ACIA répond à deux adresses 800C et 800D.

- La ligne VMA (Valid Mémoire Adresse) évite que les lectures ou écritures intempéstives des circuits coupleurs.

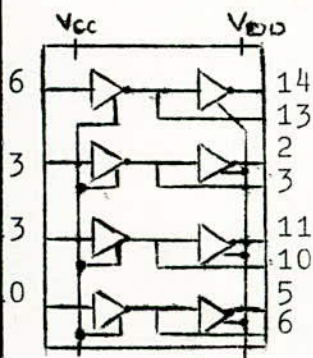
LES CIRCUITS BUFFERS

Les buffers sont des circuits amplificateurs utilisés généralement pour éviter que les circuits coupleurs viennent "POMPER" du courant sur le système microprocesseur.

On distingue deux types de buffers:

- Les buffers unidirectionnels, utilisés pour le bus d'adresses et les lignes de controle. On a utilisé comme circuit buffers les 8 T 26 qui sont des inverseurs afin de pouvoir changer facilement les adresses (ces buffers remplissent aussi la fonction des STRAPS).

- Les buffers bidirectionnels à 3 états, utilisée pour le bus de données. Le microprocesseur MC 6800 possède un bus de données inversé. Ceci nous oblige à porter notre choix sur les buffers inverseurs tels que les 8 T 26 de MOTOROLA qui sont bien étudié pour cette fonction.



N CIRCUIT "8T26"

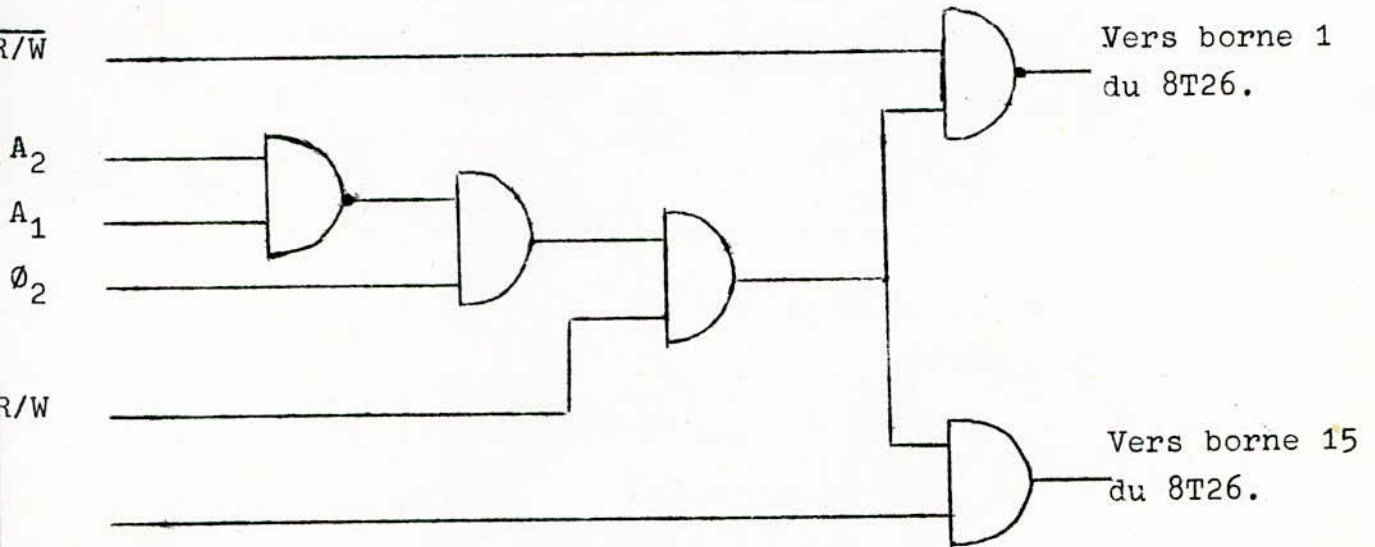
OPERATION	Borne 1 du 8 T 26	Borne 15 du 8 T 26
LECTURE	0	0
ECRITURE	1	1
HAUTE IMPED.	1	0
IMPOSSIBLE	0	1

REMARQUE: La combinaison **0 - 1** ne doit pas exister (état impossible) Cette combinaison met les deux inverseurs d'une même ligne à l'état de conduction ce qui veut dire que le microprocesseur fait une lecture et en meme temps écriture ce qui est absurde.

LOGIQUE DE COMMANDE DES BUFFERS.

Le circuit de commande des buffers doit être fait de telle sorte qu'il satisfait au tableau précédent.

Soit la logique câblée suivante:



COMMENTAIRES:

La ligne CS₂ est introduite dans le circuit de commande des buffers pour le bon verrouillage entre l'adressage et la lecture ou écriture des données. Ceci constitue un excellent synchronisme.

R/W est nécessaire pour différencier le sens de transfert des données (lecture ou écriture).

Les lignes A₁ et A₂ ont été introduites dans cette logique pour supprimer les deux adresses 800E et 800F. Quand l'une des deux adresses se présente sur le bus d'adresses, l'état de la logique de commande sera à un état tel que les buffers seront à l'état HAUTE IMPEDANCE.

TABLE DE VERITE DE LA LOGIQUE DE COMMANDE.

CS ₂	R/W	Ø ₂	Borne 1	Borne 15	Etat des Buffers.
1	0	1	0	0	ECRITURE
1	1	1	1	1	LECTURE
1	0	0	1	0	HAUTE IMPEDANCE
1	1	0	1	0	HAUTE IMPEDANCE

REMAQUE:

Quand CS₂ = 0 les buffers sont à l'état HAUTE IMPEDANCE quelque soit l'état de R/W et de Ø₂

RELEVÉ DES CHRONOGRAMMES DES SIGNAUX

Dans cette partie, nous parlerons de quelques tests que nous avons fait pour la vérification de la liaison microprocesseur coupleurs:

- Vérification de l'adresse du PIA et de L'ACIA.
- Synchronisme entre l'adressage et l'arrivée des données.

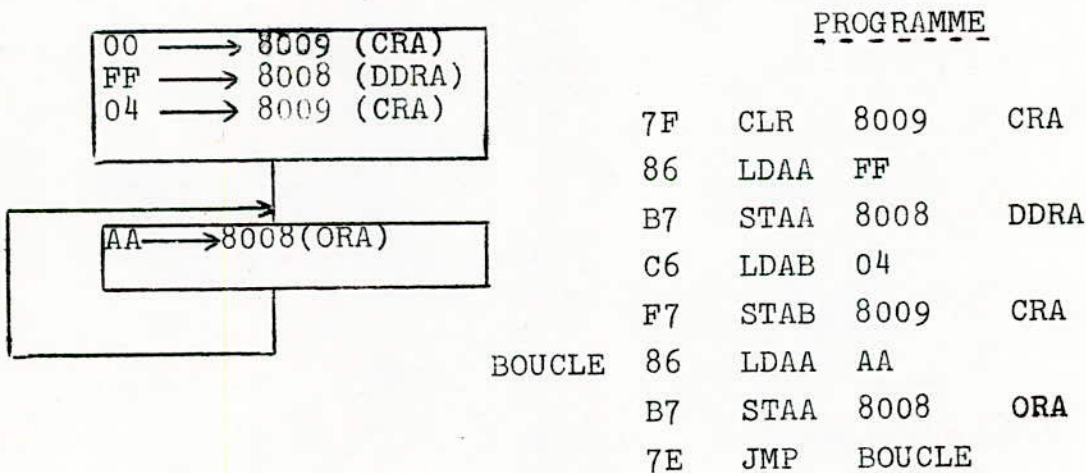
a) LIAISON MICROPROCESSEUR - PIA

Le test que nous allons faire nécessite un programme qui est constitué de deux parties.

. INITIALISATION du PIA, partie qui définit la configuration fonctionnelle de chaque registre et chaque ligne.

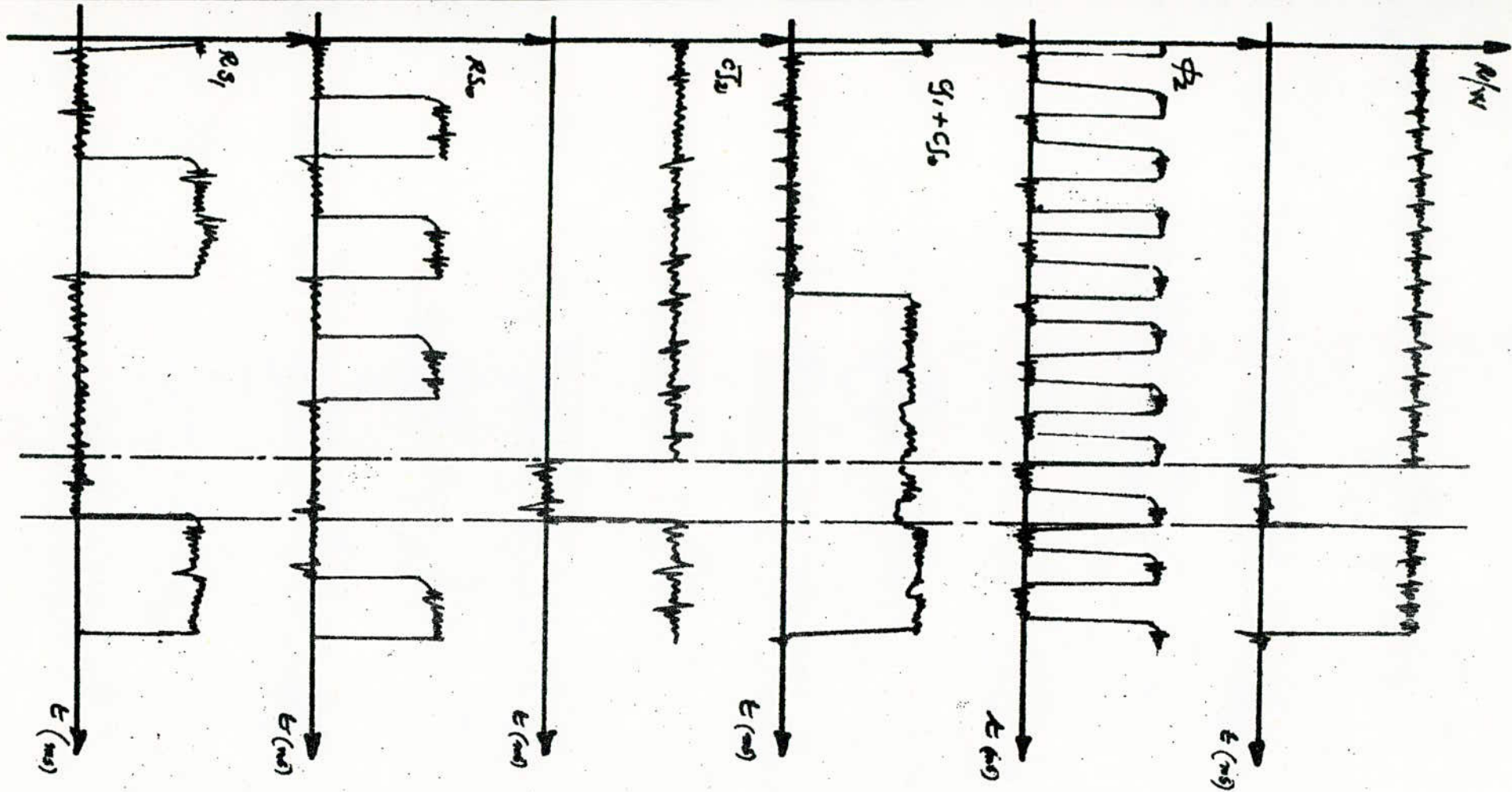
. Ecriture d'une configuration dans le registre de sortie A OU B.

Soit l'organigramme et son programme suivant:



L'opération consiste à boucler le programme ci dessus afin que le microprocesseur génère les signaux R/W, \emptyset_2 , VMA, $\overline{CS2}$, etc...

A l'aide de l'oscilloscope on a relevé les signaux qui sont dressées sur la page suivante.



Remarque : Quand $R/W = 0$ on trouve $RS_1 = RS_0 = 0$

$CS_1 = CS_0 = 1$ et $CS_1 + CS_0 = 0$; Ils constituent bien l'Adresse 8008.

— CARACTERISTIQUES —
 ADRESSE DU PIA CHOISIE 8008

1 ns
 2V

INTERPETATION DES SIGNAUX RELEVES

L'ETUDE DES CHRONOGRAMMES RELEVES NOUS MONTRE

. Que le signal $R/W = 0$ et $\phi_2 = 1$ on a :

$CS0=1$
 $CS1=1$
 $\overline{CS2}=0$

LE PIA EST SELECTIONNE.

$RS0=0$
 $RS1=0$

LE registre de sortie est atteint.

à l'aide de l'oscilloscope on vérifie les lignes PA0-PA7, on trouve bien la valeur AA c.a.d 1010 1010

. Pour la logique de commande

la borne 1 = 0

la borne 15 = 0

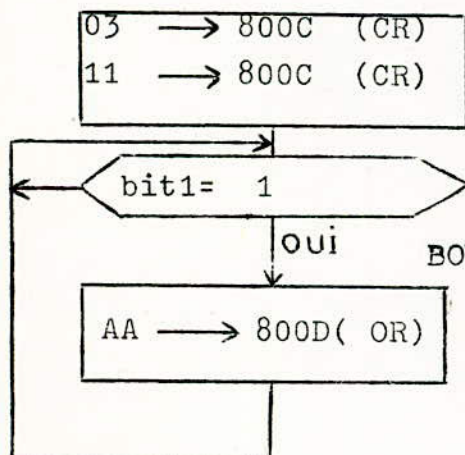
Donc le sens de transfert des données se fait du microprocesseur vers le PIA.

En conclusion , l'écriture de la donnée dans le registre de sortie

DRA se fait au même moment que l'adressage du PIA.

b) LIAISON MICROPROCESSEUR - ACIA

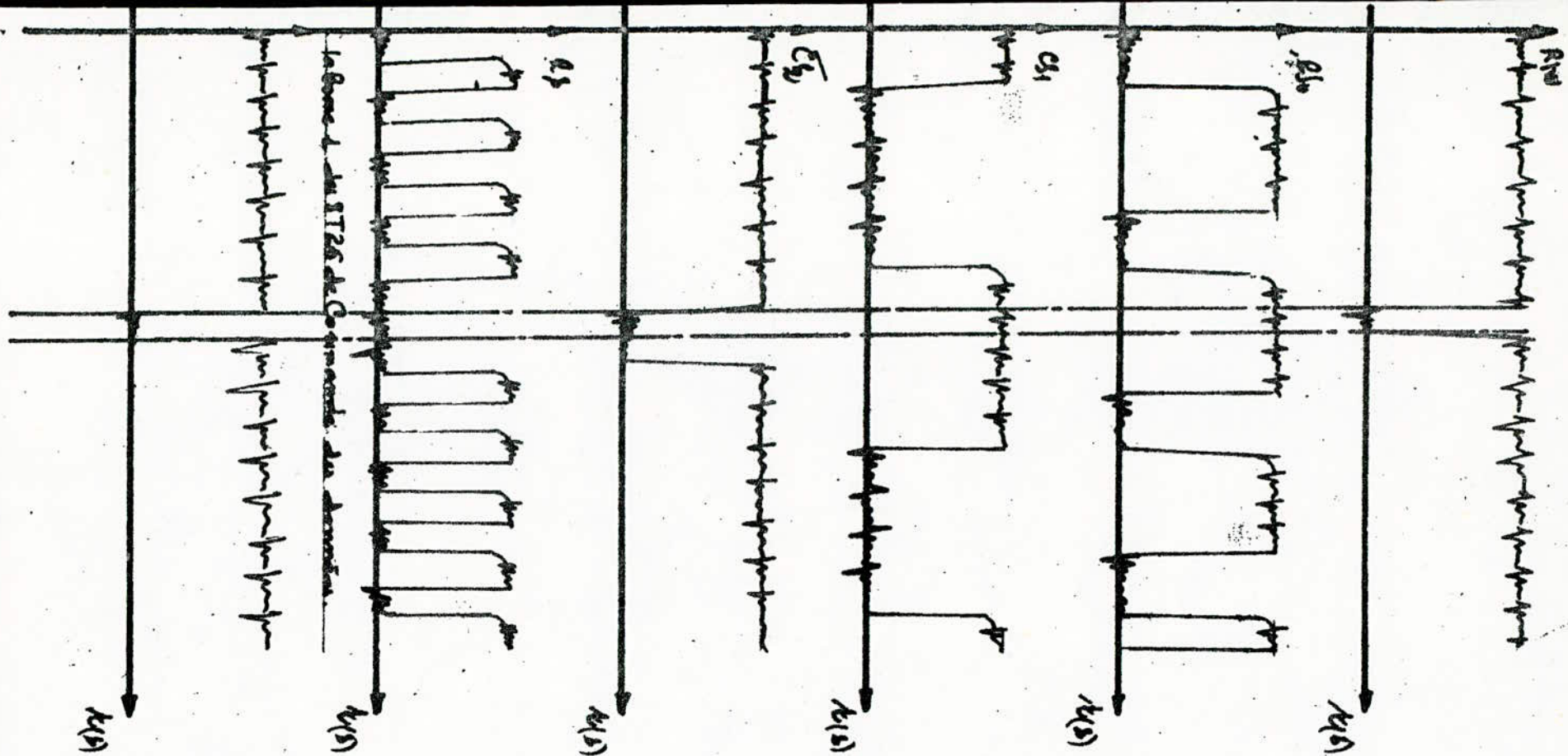
Comme le cas précédent on a établie l'organigramme avec son programme :



PROGRAMME

86	LDAA	03	
B7	STAA	800C	CR
86	LDAA	11	
B7	STAA	800C	CR
86	LDAA	AA	
F6	LDAB	800C	SR
57	ASRB		
57	ASRB		
24	BCC	F9	
D7	STAA	800D	OR
7E	JMP	BOUCLE	

-XXXX-



Remarque: Quand $R/W = 0$ on trouve $RS_1 = 0$
 $CS_1 = CS_0 = 1$ et $CS_2 = 0$
 on a bien l'Adresse 800C

5ns

==== CARACTERISTIQUES ====
 A DRESSE DE L'ACIA CHOISIE 800C

Cette partie étant un peu plus complexe car la transmission des données ne peut se faire que si l'ACIA est relié à un autre circuit (charge). Ceci nous a conduit à connecter à l'ACIA un circuit adaptateur de niveau dont l'étude sera faite un peu plus loin.

Les chronogrammes relevée sont représentés sur la page suivante:

INTERPRETATION DES SIGNAUX RELEVES

Une étude des chronogrammes nous montre:

. Quand le signal $R/W = 0$ et $\emptyset_2 = 1$ on a:

CS0 = 1	}	L'ACIA EST SELECTIONNER
CS1 = 1		
$\overline{CS2} = 0$		

RS = 0

Le registre de sortie est atteint.

Pour la logique de commande

La borne 1 = 0

La borne 15 = 0 Donc on est bien en opération de lecture

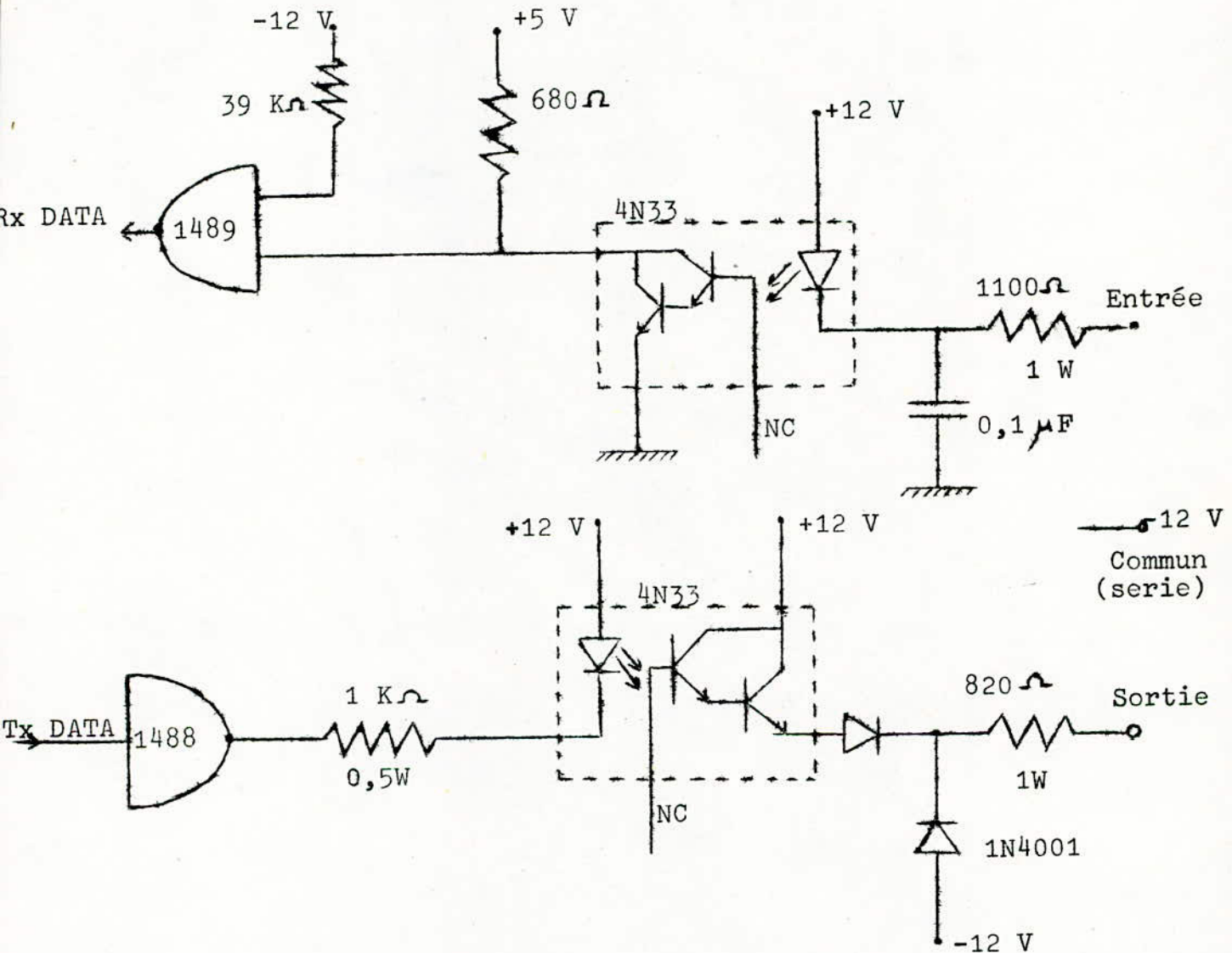
On conclusion, il y a synchronisme entre l'adresse et le transfert de dl donnée.

2. LIAISON CIRCUITS COUPLEURS-PERIPHERIQUES :

+Liaison ACIA (MC6850)-Unité de visualisation TEKTRONIX 4023.

La peripherie TEKTRONIX 4023 travaille en boucle de courant de 20 mA en entrée et en sortie. Cette caractéristique nous oblige à interconnecter entre l'ACIA et le terminal un circuit adaptateur de niveau pouvant remplir cette fonction.

Soit le circuit suivant:



COMMENTAIRES:

• Circuit d'entrée:

+Un intrégrateur a été placé juste à l'entrée pour supprimer les variations brusques (PICS) accidentelles du signal d'entrée qui peuvent détériorer le photo-coupleur.

+Un photo coupleur qui isole l'ACIA de la boucle de courant de l'unité de visualisation.

+La boucle de courant de 20 mA est obtenue grace au circuit formé par la diode du photo-coupleur et de la résistance 1100 Ω qui sont alimentées entre ± 12 V donc $24V:1200 = 20$ mA (environ)

+Le transmetteur MC1489 amplifie le signal sortant du photo-coupleur 4N33 avant d'attaquer l'ACIA(MC6850)

Circuit de sortie:

+Le circuit MC1488 est aussi un transmetteur. Il attaque le circuit 4N33 à travers une résistance dont la valeur est calculée de telle sorte qu'un courant suffisant doit circuler dans la diode du photo-coupleur pour que celle-ci puisse émettre des photons qui attaqueront la base du transistor.

+La boucle de courant (20 mA) est obtenue de la même manière que pour le circuit d'entrée.

+Les deux diodes 1N4001 permettent la mise en forme du signal sortant du 4N33.

REMARQUE:

L'alimentation de ce circuit adaptateur de niveau nécessite deux sources d'alimentation l'une délivrant +5V l'autre ± 12 V symétrique. Ces deux alimentations sont disponibles sur le bus de l'EXOCISER mais ont des masses différentes non reliées. Par conséquent il faut faire attention; lors du câblage, de ne pas "mélanger" les masses.

* Liaison PIA (MC 6821) Lecteur optique.

Comme le lecteur optique délivre ou reçoit des signaux TTL, il est possible de lui relier directement le PIA (MC6821) sans ajouter d'autres circuits auxiliaires.

La liaison se fait comme suit:

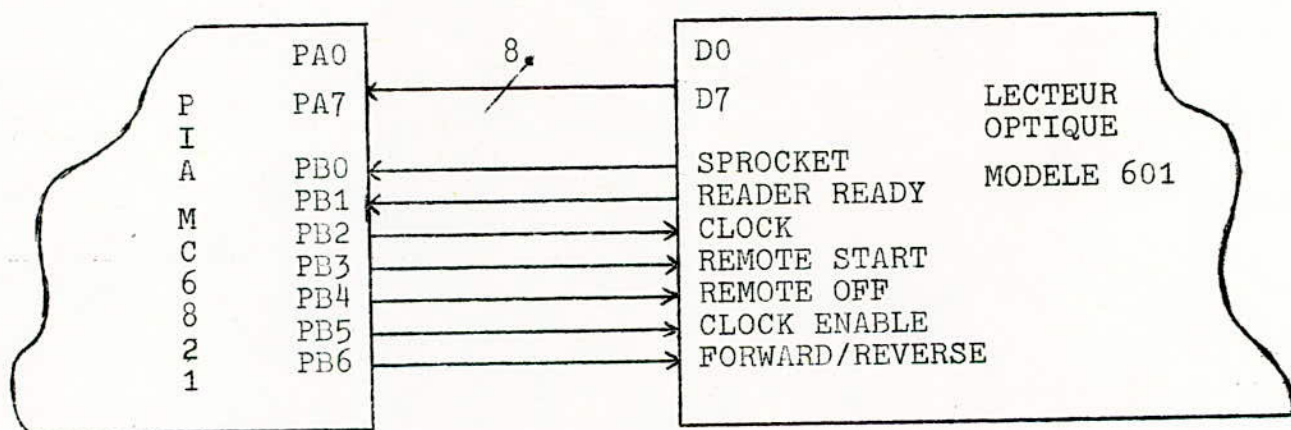
+Les huit lignes de données sont reliées directement au port A du PIA, qui seront programmées en entrées puisque l'on fait uniquement une lecture du ruban.

+Le port B du PIA est utilisé pour la commande du lecteur optique:

.PB0 et PB1 sont reliées respectivement au SPROCKET et READER READY qui sont deux lignes de sorties du lecteur optique donc PB0 et PB1 devront être programmées en entrées.

.PB2 à PB6 doivent être reliées au CLOCK, CLOCK ENABLE, REMOTE START, REMOTE OFF, respectivement et doivent être programmées en sorties.

Soit le schéma suivant:

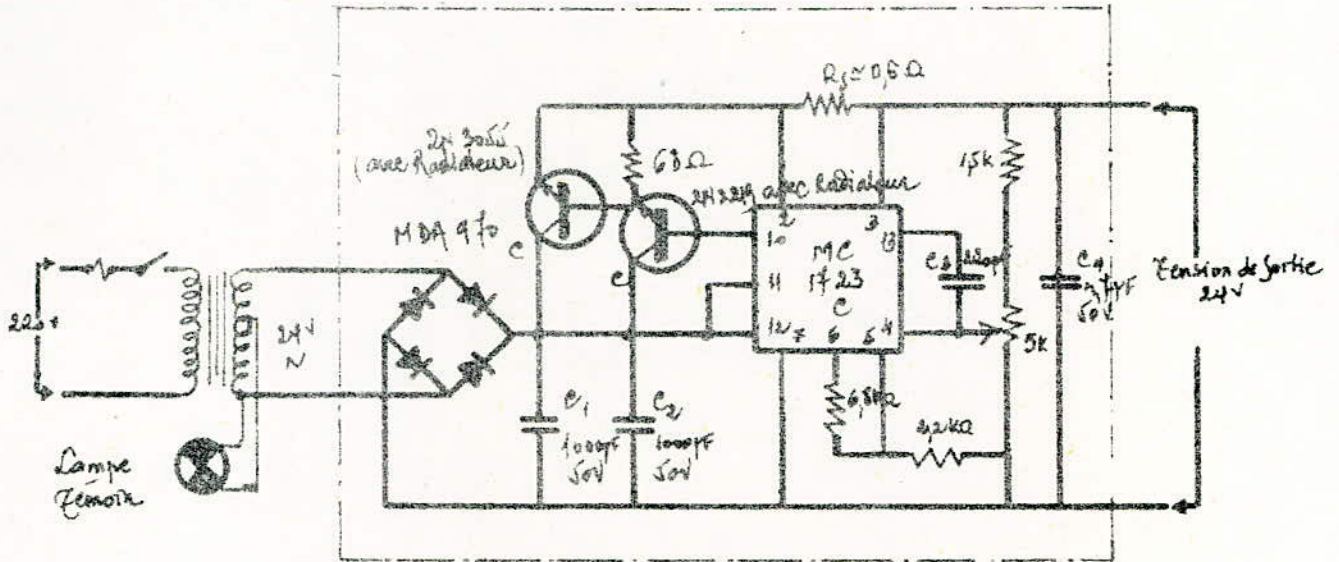


IV-ETUDE DE L'ALIMENTATION DU LECTEUR OPTIQUE -MODELE 601- :

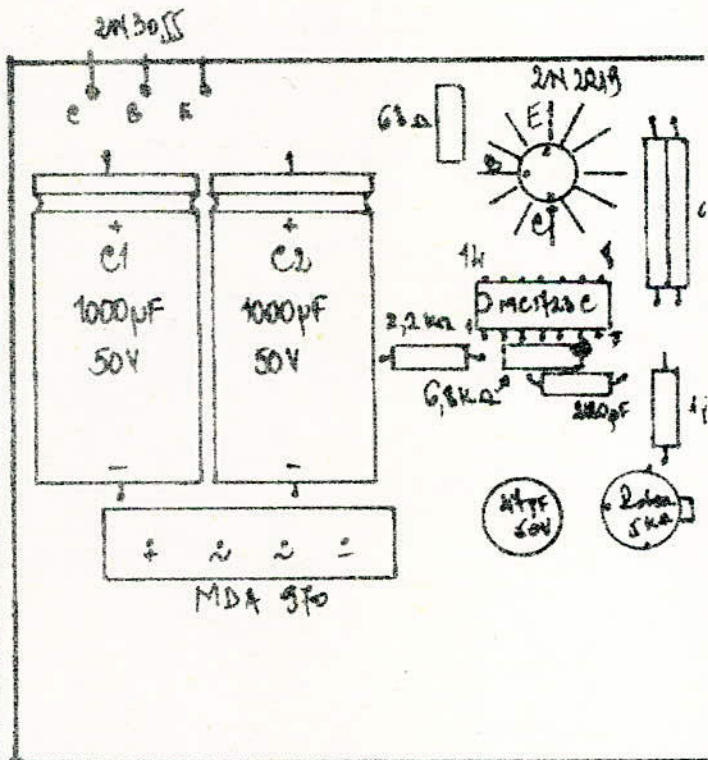
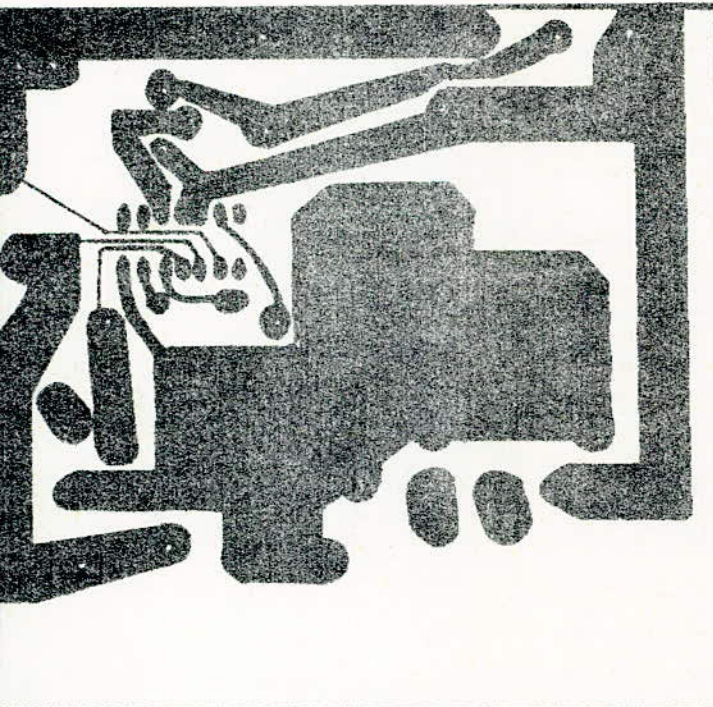
Le lecteur optique MODELE 601 a besoin d'une source pour alimenter son moteur et ses diodes "LED". Cette alimentation doit fournir une tension de + 24v \pm 5% avec un débit de 0,6 A. Cette alimentation n'existe pas sur le bus de l'EXORCISER, donc on a été obligé de réaliser une alimentation auxiliaire.

Le schéma électrique ainsi que celui du circuit imprimé sont donnés en page suivante.

SCHÉMA DE PRÎNCÎPE

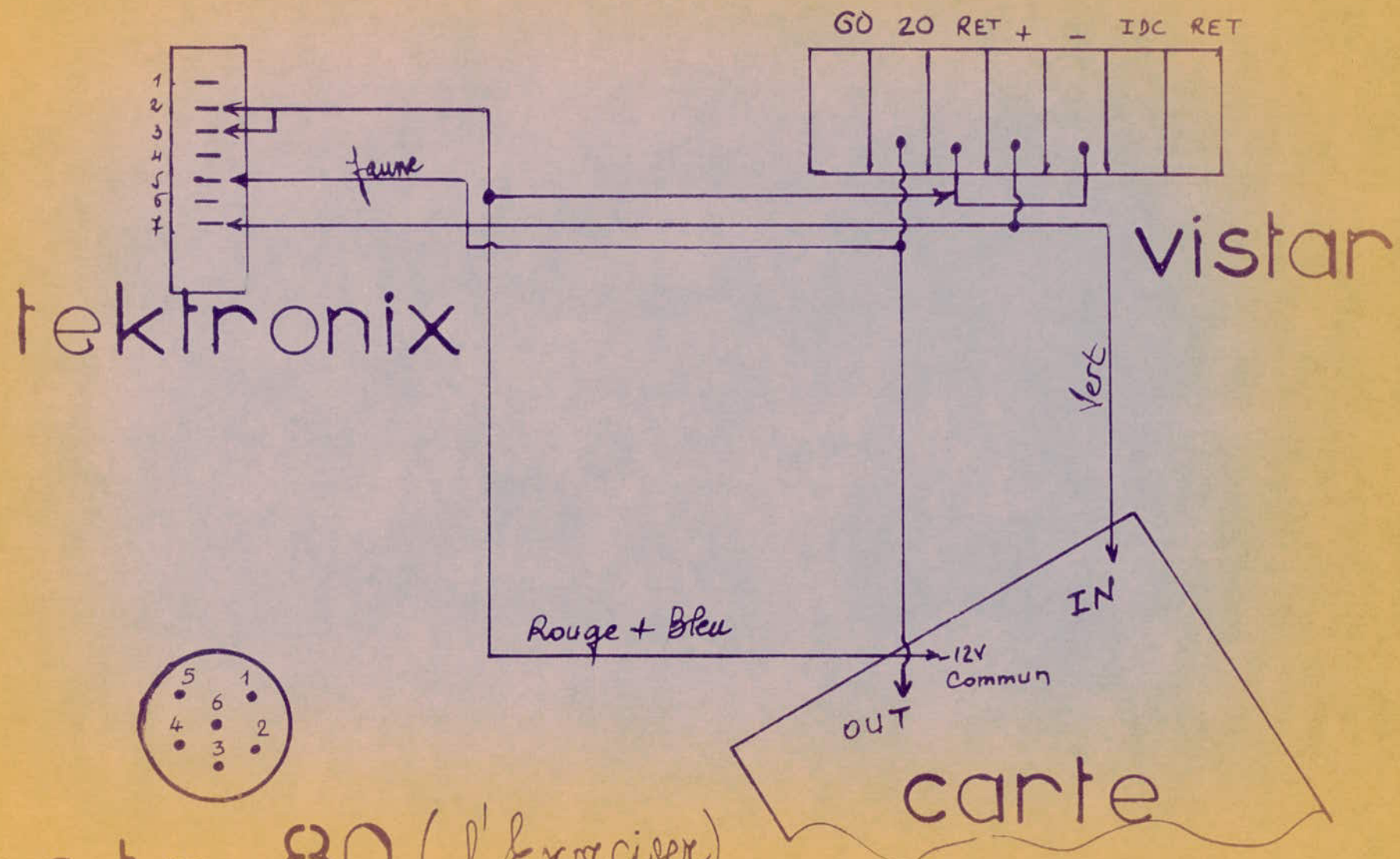


CÎRCUIT ÎMPRÎME




ALIMENTATION (24V, 0,6A)

liaisons carte - périphéries



c.s.t.n 80 (l'Exerciser)

CHAPITRE VII

"SOFTWARE"  PROGRAMMATION

I INTRODUCTION

II ORGANIGRAMMES ET PROGRAMMES ASSOCIES A L'UNITE
DE VISUALISATION TEKTRONIX 4023.

III ORGANIGRAMMES ET PROGRAMMES ASSOCIES AU LECTEUR OPTIQUE
MODELE 601 C.

IV PROGRAMME GENERALE

V AMELIORATIONS POSSIBLES



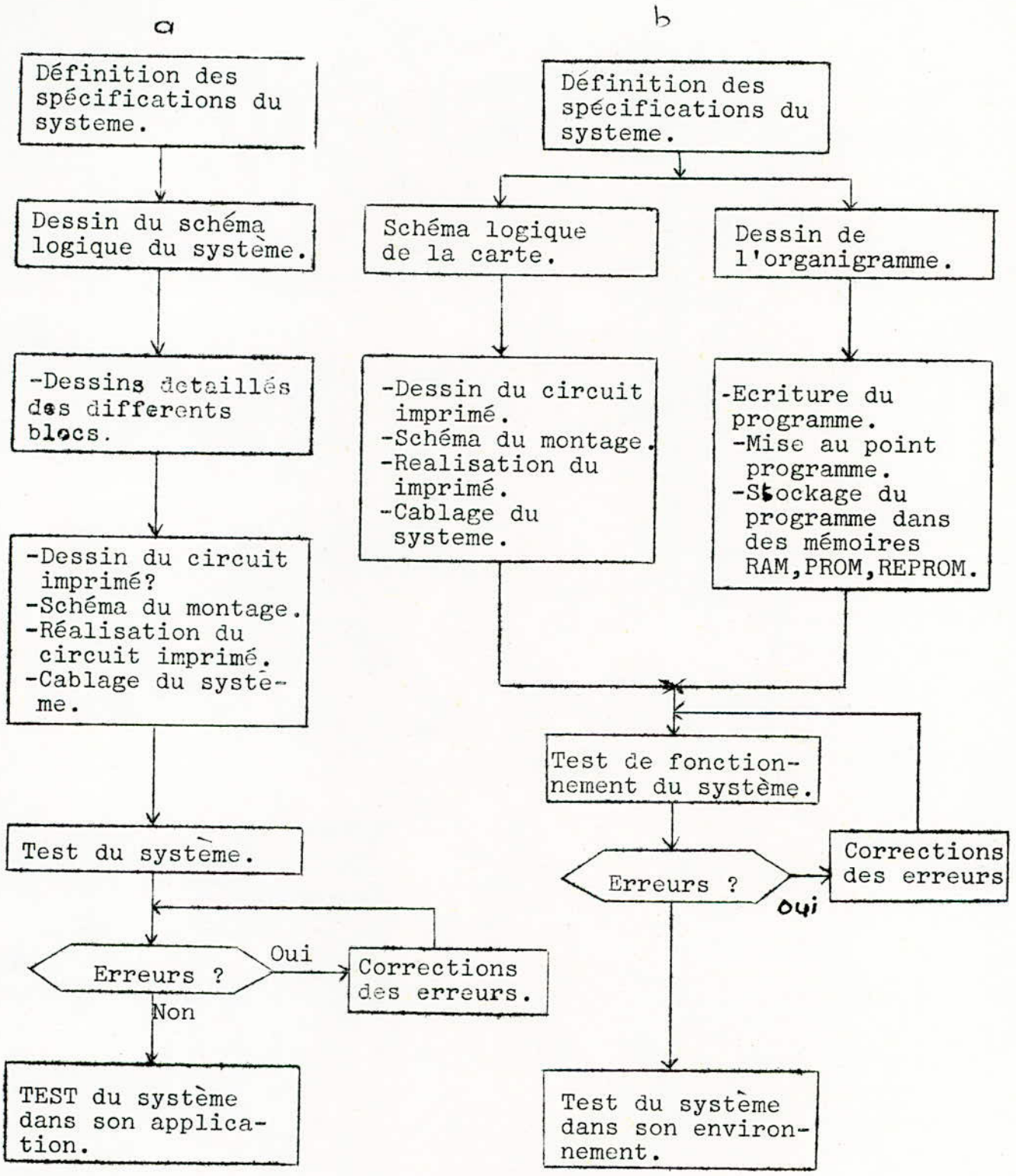
I INTRODUCTION:

Le "SOFTWARE" est la partie la plus difficile et la plus longue à élaborer lors de la réalisation d'un circuit d'interface d'entrée/sortie. Ce dernier fonctionne sous le contrôle du microprocesseur qui lui même est contrôlé par un programme, qui est une suite de séquence d'instructions, qu'il doit exécuter:

-Lors de l'élaboration d'un projet de logique câblée ou d'un système analogique, le réalisateur trace d'abord le schéma synoptique global. Puis il détaille chaque bloc du système en définissant le rôle et l'utilité de chaque élément qui entre dans la constitution de ce bloc(Portes, Bascules, ou les montages de base à transistors dans le cas analogique...)

-La conception d'un "SOFTWARE" est semblable à celle de la logique câblée: Le réalisateur commence par donner les fonctions que doit remplir son système. Ces fonctions seront alors traduites en organigrammes qui montrent graphiquement les différentes tâches que doit effectuer le microprocesseur. Chaque tâche est développée en une suite de séquence d'instructions en code machine.

Les deux organigrammes de la figure 1 permettent de schématiser la marche à suivre lors de la réalisation d'un système soit en logique câblée ou en logique programmée.



a) Développement d'un système en logique câblée.

b) Développement d'un système en logique programmable.

II ORGANIGRAMMES ET PROGRAMMES ASSOCIES A L'UNITE DE VISUALISATION
TEKTRONIX 4023

Après la réalisation du circuit d' interface d'entrée/sortie entre le microprocesseur et l'unité de visualisation, nous allons aborder dans ce paragraphe la partie "SOFTWARE".

Notre travail est d'élaborer et d'associer au circuit d'interface un programme qui comprend les deux parties suivantes:

* Stockage d'une page d'information à partir du clavier du terminal; c'est à dire mémoriser une page de données dans les RAM associées au microprocesseur.

* La sortie d'une page d'informations:
Visualiser sur l'écran le contenu des RAM.

REMARQUE:

Une page d'informations est constituée de 24 lignes de 80 Caractères (MOTS) chacune. C'est la capacité maximale d'informations qu'on peut visualiser sur toute la surface de l'écran de la TEKTRONIX 4023.

Dans ce qui suit nous donnerons uniquement les divers organigrammes. Les programmes seront données dans le paragraphe IV qui englobe toute la programmation. Les organigrammes étant clairs et explicites ne necessitent pas de commentaires.

Les organigrammes sont établis comme suit:

- * Organigramme d'initialisation
- * Organigrammes d'entrée et de sortie d'un caractère.
- * Organigrammes d'entrée et de sortie d'une ligne.
- * Organigrammes d'entrée et de sortie d'une page.

III- ORGANIGRAMMES ET PROGRAMMES ASSOCIES AU LECTEUR OPTIQUE

MODELE 601

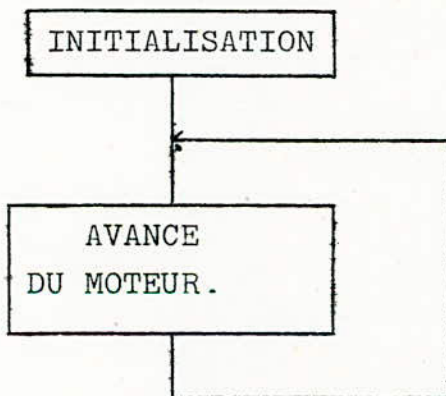
Avant de se lancer dans l'élaboration de l'organigramme et le programme général gérant le lecteur optique, nous avons jugé utile de procéder par étape.

Nous avons établis certains test préliminaires qui nous ont permis de comprendre, de maitriser et de verifier le fonctionnement du lecteur optique.

1° TESTS PRELIMINAIRES

-Avance continue du moteur.

Ce test consiste à faire tourner uniquement le moteur Pour cela on a établi l'organigramme d'avance du moteur (Voir organigramme "subroutine de commande du lecteur optique")



REMARQUE:

Les signaux de commande restent inchangés pendant toute la periode du signale CLOCK.

CLOCK ENABLE = 1

FORWARD/REVERSE=0

REMOTE START = 1

REMOTE OFF = 1

Ce test est répété plusieurs fois afin de s'assurer de la bonne fiabilité du moteur.

-Mouvement en sens inverse du moteur.

Pour ce test il suffit de changer uniquement l'état de la ligne FORWARD/REVERSE = 1.

L'état des autres lignes reste inchangé.

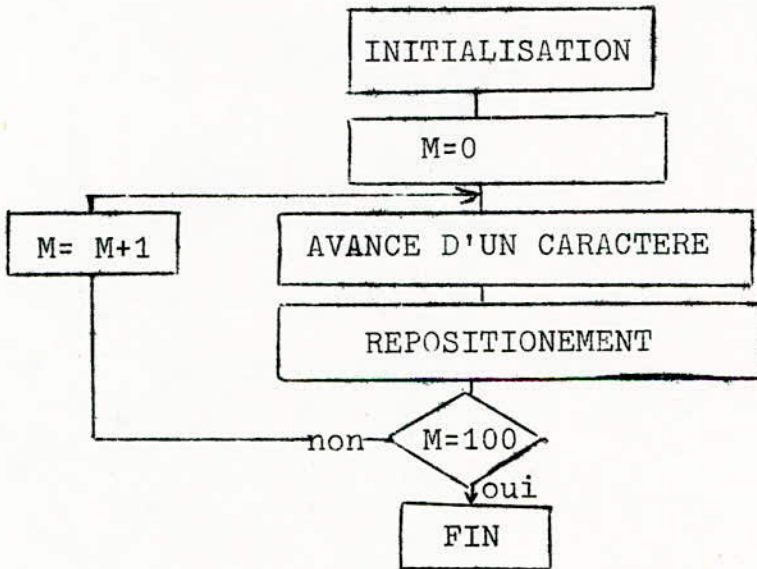
TEST AVANCE DU MOTEUR PAS A PAS

Une fois qu'on est assuré que le moteur tourne dans les deux sens, on passe à une autre phase de test, qui consiste à faire tourner le moteur pas à pas.

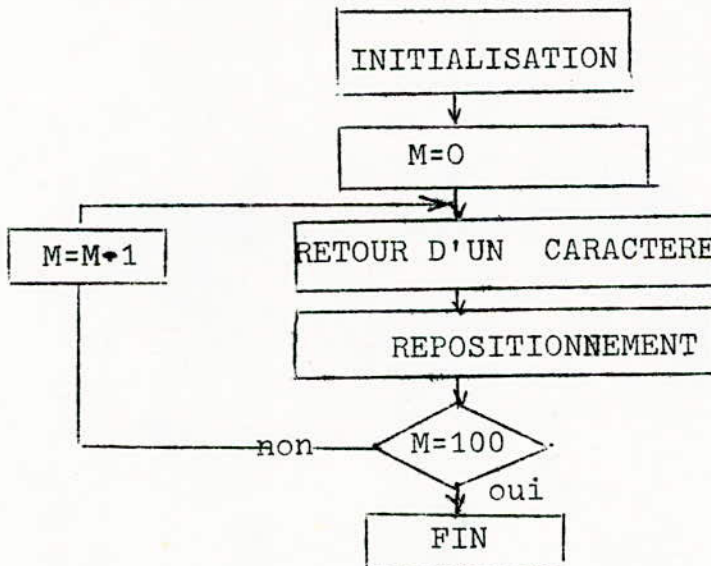
Donc il faut changer l'état logique de Clock Enable durant une avance d'un pas, afin de permettre au moteur d'établir un temps d'arrêt "repositionnement" pour chaque pas.

On limite l'avance du moteur à 100 pas seulement.

Pour cela l'organigramme sera le suivant.

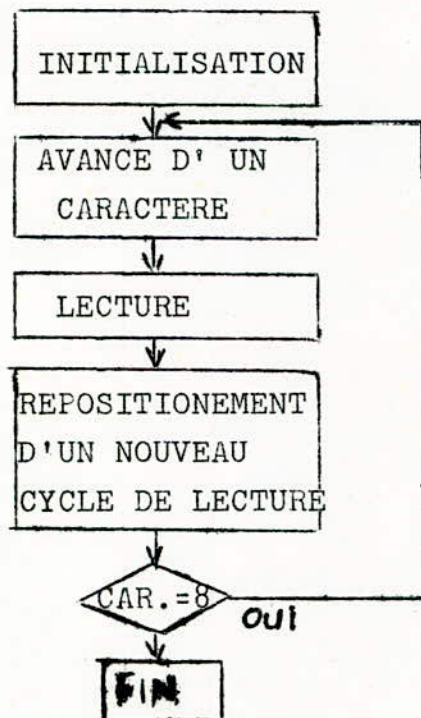


De même le test de retour du moteur pas à pas. Seulement on change l'état logique de F/W.



TEST DE LECTURE :

Ce test consiste à détecter un chiffre "A" parmi toute une série de chiffres "B".
C'est un test intéressant et efficace qui permet de savoir si le lecteur lit les informations.



Pour ce test on perfore plusieurs chiffres "8" sur un ruban et on intercale un chiffre "7" parmi les "8".

Quand le lecteur optique détecte le chiffre "7" doit s'arrêter.

AVEC CE TEST ON S'ASSURE QUE LE LECTEUR LIT BIEN.

NOTE: Comme la fréquence du signal clock du moteur ne doit pas dépasser 150 Hz. Il est nécessaire d'établir un programme d'attente (Programme de repositionnement d'un nouveau cycle de lecture).

COMMENTAIRE :

Ce test nous a permis de relever les courbes des différents signaux de commande du lecteur.

Aussi nous a permis d'ajuster les temps de ces signaux.

A la page suivante nous donnons le relevé.

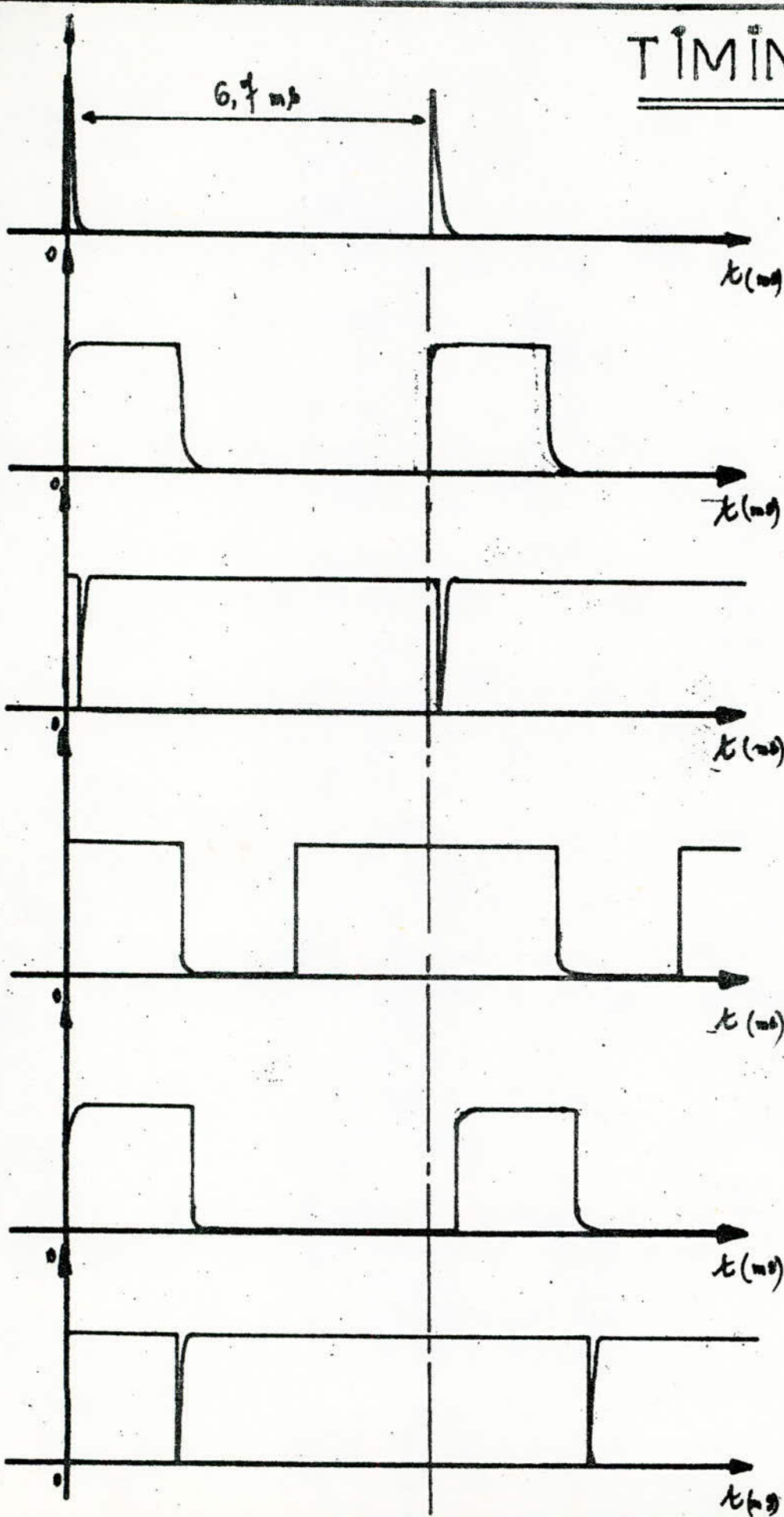
TIMING

— 2x

— 1ms

- Clock "10 μ s"

6.7 μ s



- Clock / Enable

Remote Start
"10 μ s"

Sprocket

Reader

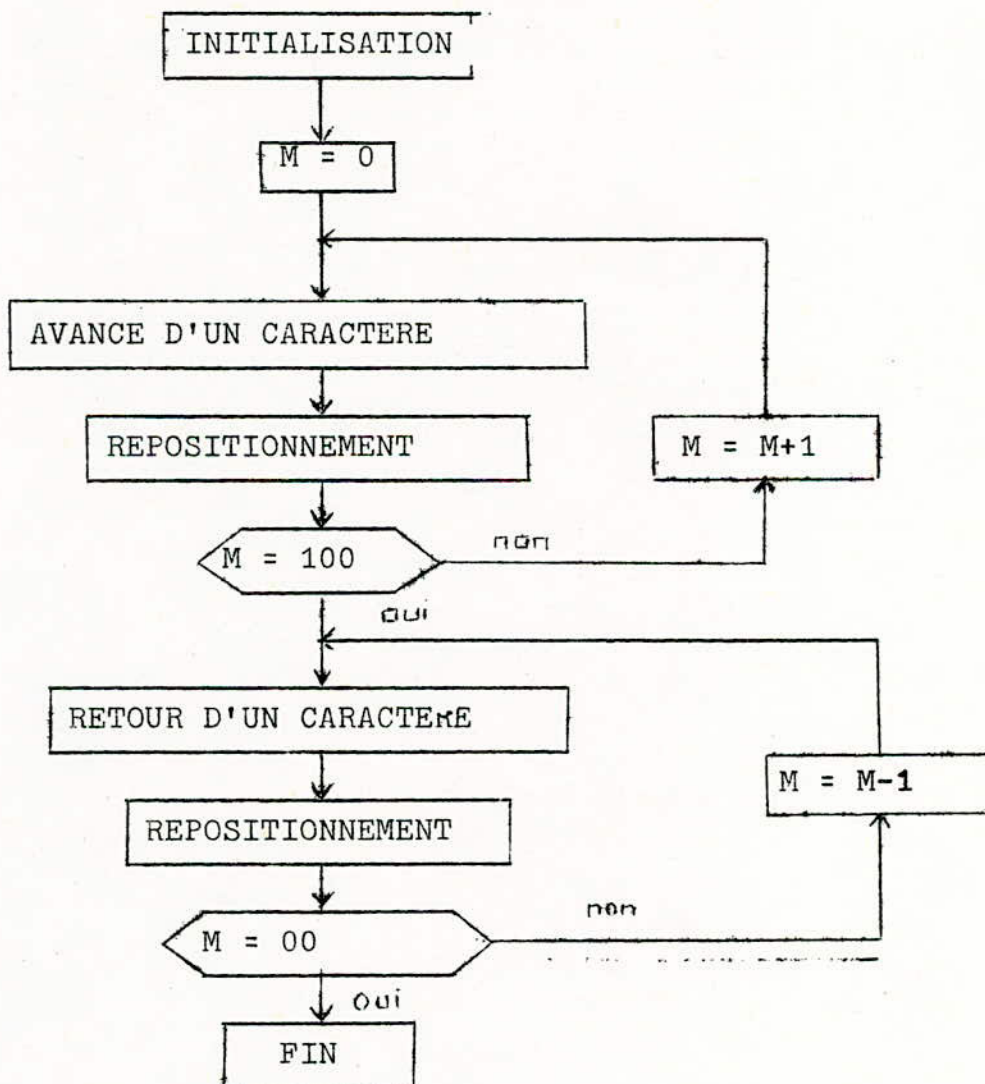
Remote Off
"10 μ s"

TEST DE DETECTION D'ERREUR:

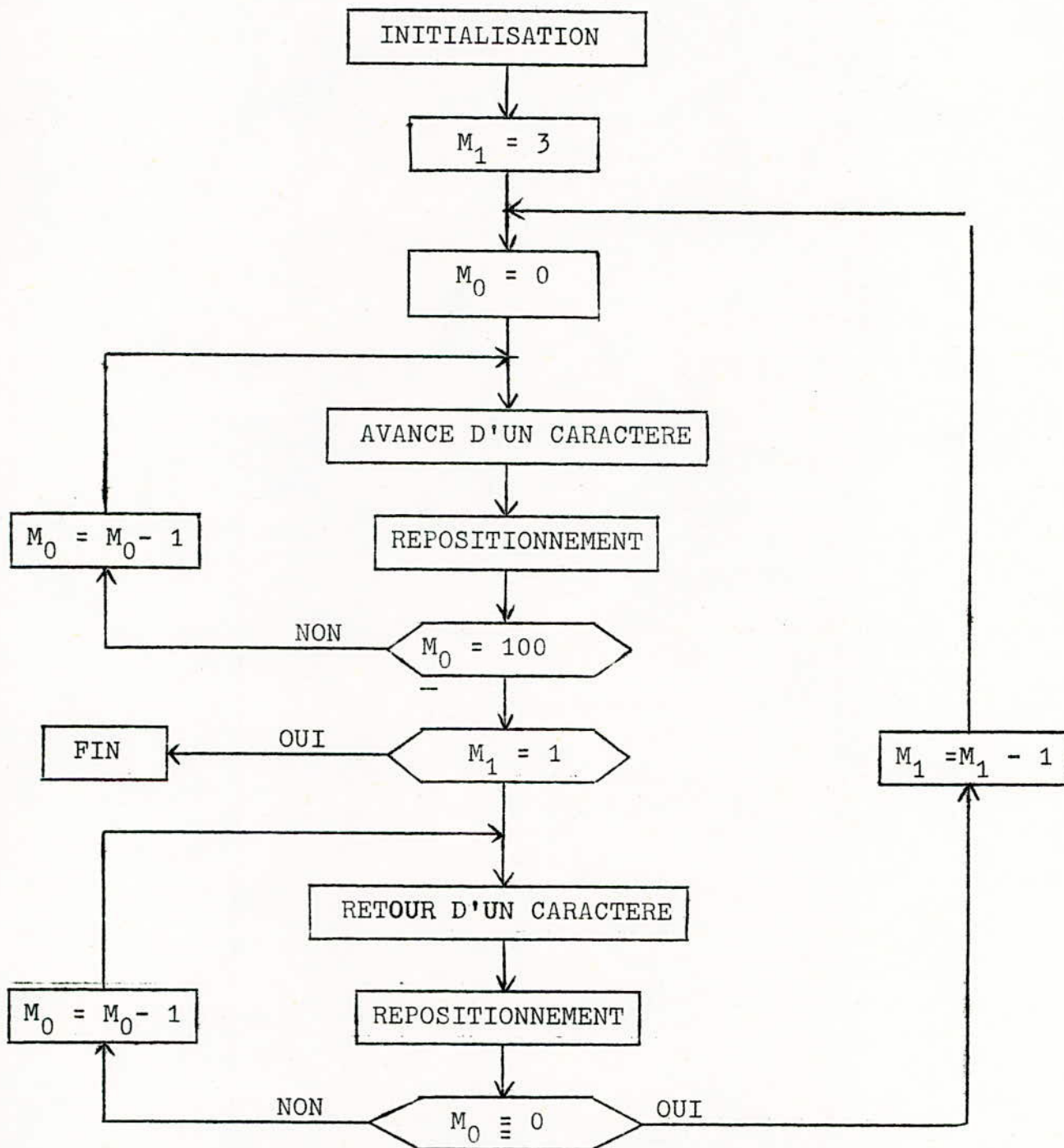
Comme nous avons à établir un sous programme de detection d'erreur "CHECKSUM" nous avons jugé utile de faire deux tests:

oLe premier test consiste à faire avancer le moteur de 100pas, puis de changer immédiatement son sens de rotation en effectuant aussi 100 pas vers l'arrière afin de repositionner le ruban à sa position de départ.

Pour cela on a élaborer un programme qui corespond à l'organigramme suivant:



Le deuxième test consiste à faire la même opération que précédemment mais trois fois de suite. Pour cela nous avons établis le programme qui correspond à l'organigramme suivant:



Commentaires: Quand le lecteur detecte une erreur CKSM il doit relire le ruban pour bien s'assurer qu'elle existe Il doit lire le ruban trois (3) fois, et si à la troisième lecture l'erreur existe toujours il doit s'arreter. C'est dans ce but que le test ci dessus a été établi.

EXEMPLE D'UN CONTENU D'UN RUBAN

SC 0B 00 00 58 20 49 4E 50 41 47 45 C8
S1 05 01 00 20 30 AF
S9 03 00 00 FC

POUR ELABORER L'ORGANIGRAMME DU LECTEUR OPTIQUE NOUS RAPPELONS QU'UN RUBAN EST CONSTITUE DES PARTIES SUIVANTES:

La première ligne donne le nom du programme.
La deuxième ligne donne les données et les adresses où sont stockées.
La troisième ligne la fin du programme.

S: Caractère de début d'enregistrement.

0;1;9 Type d'enregistrement.

0 début d'enregistrement

1 enregistrement des données.

9 Fin d'enregistrement

L'octet juste après appelé le byte count il donne le nombre d'octets se trouvant sur la ligne.

Ex: dans la première ligne on lit 0B ce qui signifie qu'il y a 11 octets.

Les deux octets suivant représentent l'adresse de la première donnée.

Ex: dans la deuxième ligne on lit 01 00 20 ... cela veut dire que la donnée 20 est stockée à l'adresse 0100 est bien entendu la donnée suivante 30 est stockée à l'adresse 0101.

Pour la première ligne on lit après l'adresse 0000

58 20 49 4E 50 41 47 45 sont stockées respectivement aux adresses suivants 0000,0001,...,0007.

Donc toujours le nom du programme est stocké aux adresses 0000;...,0007 est toujours.

Enfin le dernier octet de la première ligne représente le checksum. sa valeur est déterminé de la manière suivante.

C'est la somme des octets constituant la ligne (le byte count et l'adresse sont compris) puis cette somme complétée à 1.

Ex: Prenant le cas le plus simple, la troisième ligne
On lit (page suivante). -XXV-

On lit FC qui représente le checksum pour cette ligne.

Ce FC est obtenu en faisant la somme des octets de la troisième ligne puis le tout complémenter à 1

$$\text{Donc FC} = \overline{03 + 00 + 00}$$

$$\begin{array}{r} \phantom{\text{compl. à 1}} \phantom{\text{FC}} \phantom{\text{"CHEKSUM"}} \\ \phantom{\text{compl. à 1}} \phantom{\text{FC}} \phantom{\text{"CHEKSUM"}} \\ \phantom{\text{compl. à 1}} \phantom{\text{FC}} \phantom{\text{"CHEKSUM"}} \\ + \phantom{\text{compl. à 1}} \phantom{\text{FC}} \phantom{\text{"CHEKSUM"}} \\ \phantom{\text{compl. à 1}} \phantom{\text{FC}} \phantom{\text{"CHEKSUM"}} \\ \phantom{\text{compl. à 1}} \phantom{\text{FC}} \phantom{\text{"CHEKSUM"}} \\ \hline = \phantom{\text{compl. à 1}} \phantom{\text{FC}} \phantom{\text{"CHEKSUM"}} \\ \text{compl. à 1} \phantom{\text{FC}} \phantom{\text{"CHEKSUM"}} \end{array}$$

Cette brève explication reste valable pour toutes les lignes.

REMARQUE : Une ligne ne peut comporter plus de 24 données.

À partir des explications précédentes, il est bon d'établir l'organigramme mais d'une manière plus schématique, la figure 2 permet de gérer le lecteur optique.

Comme on le voit elle représente les fonctions suivantes:

Détection du type d'enregistrement.

Formation d'un byte (somme de deux frames)

Lecture et sauvegarde du byte count

Formation d'adresse

Lecture et stockage des données à leur adresse.

Test de detection d'erreurs.

Emission des messages:

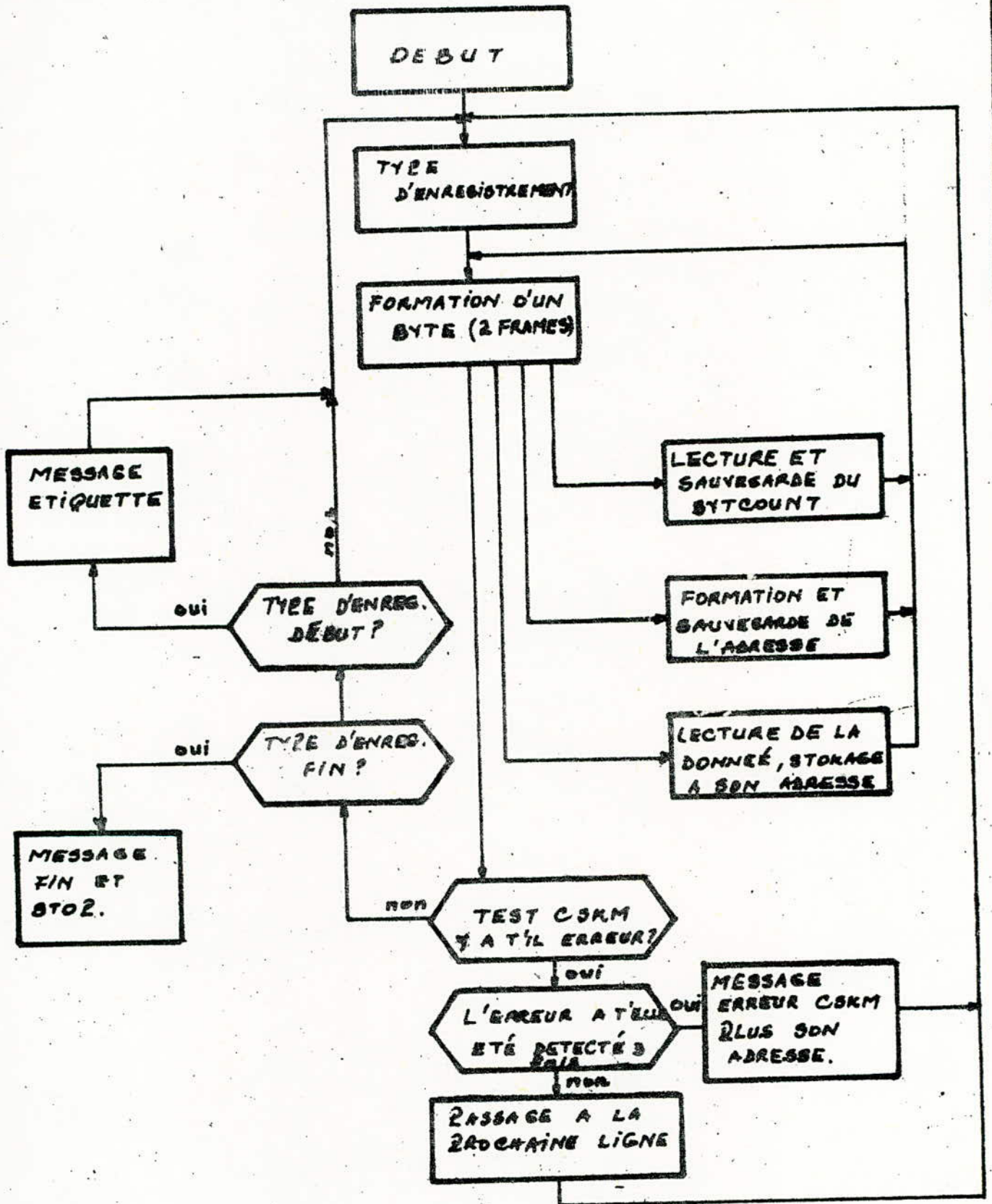
ETIQUETTE (nom du ruban)

CHARGEMENT TERMINE (fin du ruban)

ERREUR CSKM + ADRESSE. (s'il y a erreur)

LES DIFFERENTS FONCTIONS PRECEDENTES NCUS ONT PERMIS DE TRACER L'ORGANIGRAMME GENERAL D'UNE MANIERE PLUS DETAILLE.

(VOIR ORGANIGRAMME GENERALR)

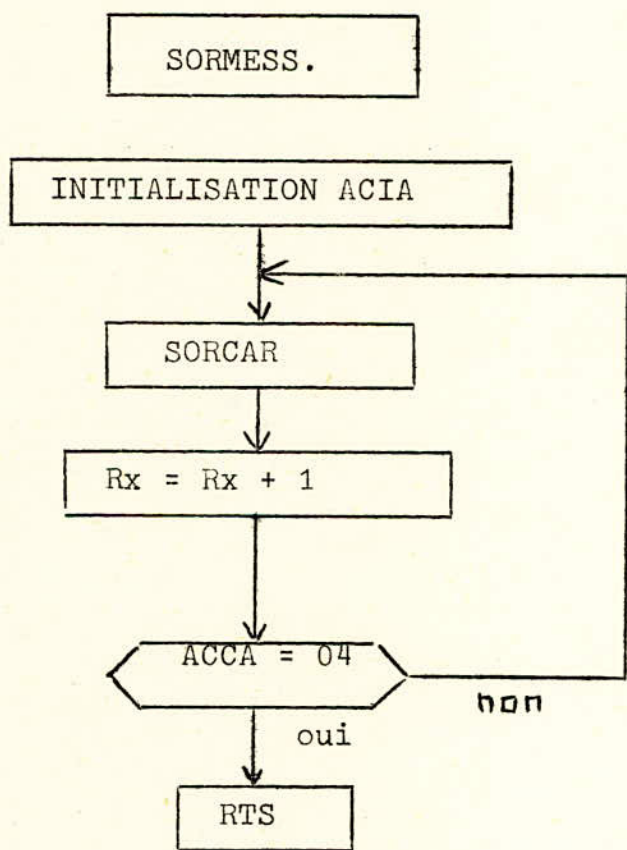
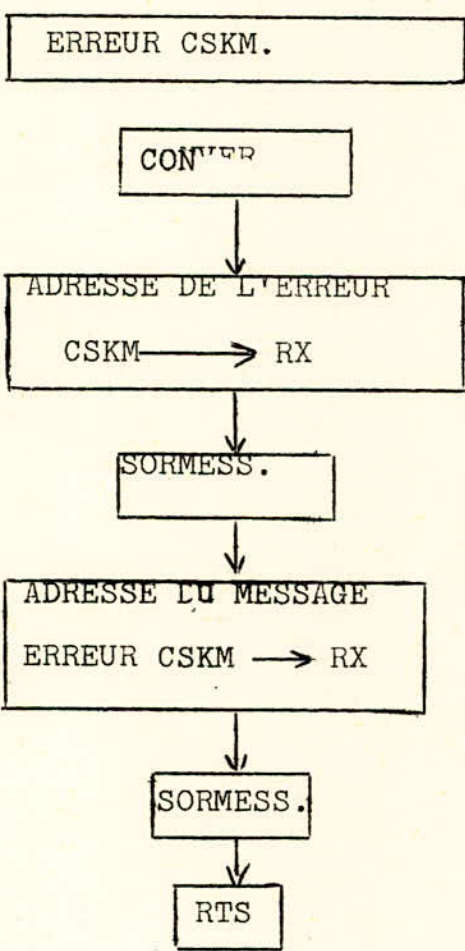
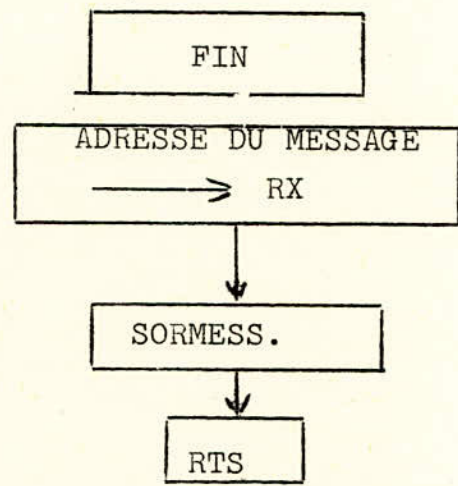
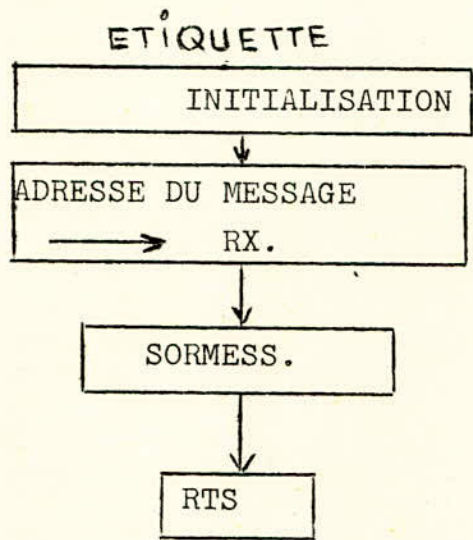


- fig 2 -

Une fois cette étape étant faite, il faut dresser les différentes "SUBROUTINES" (sous programmes); mais avant de se lancer dans la programmation; il est plus commode de commencer par dessiner les organigrammes suivants:

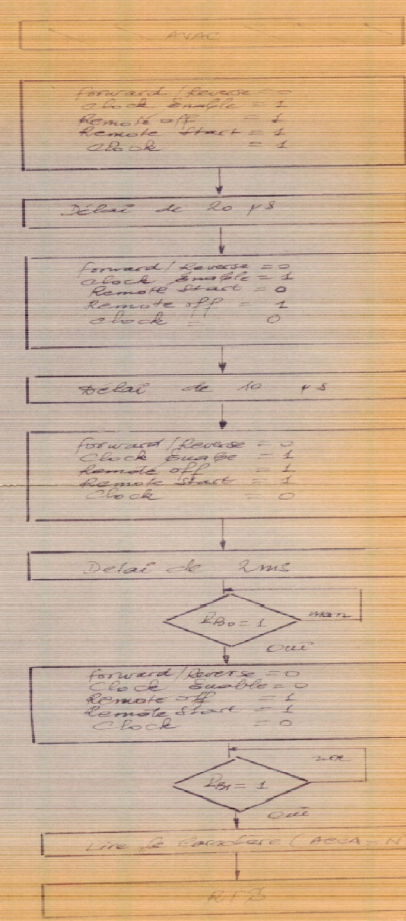
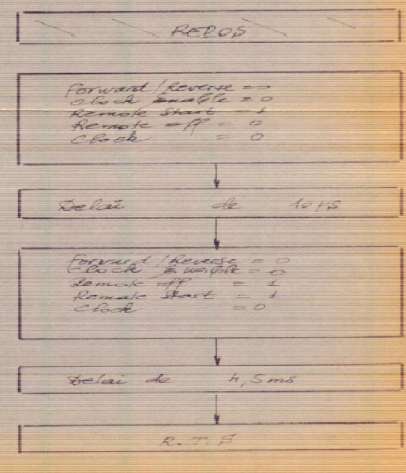
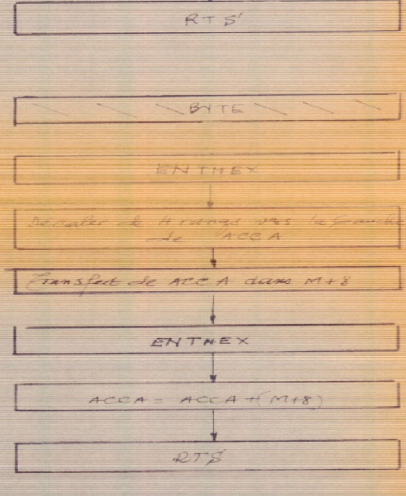
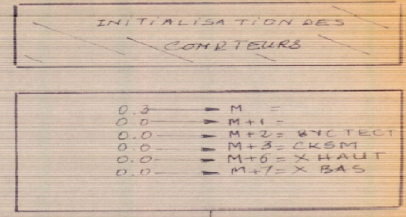
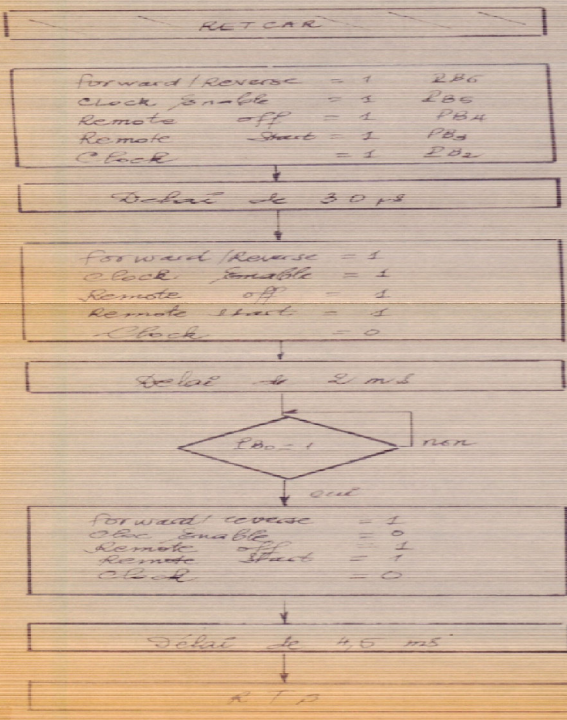
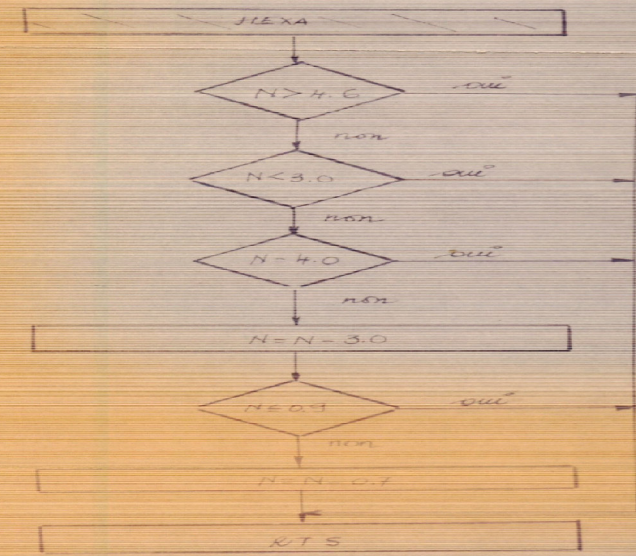
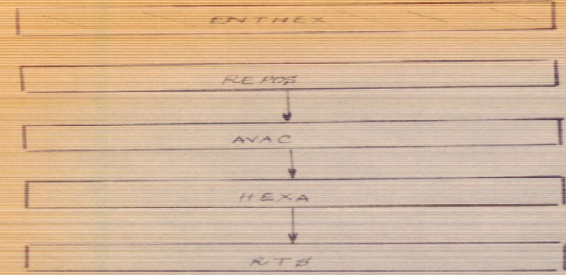
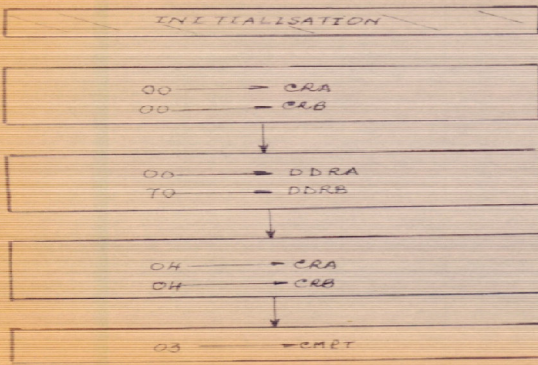
- +AVAC avance d' un caractère: subroutine d'avance d'un pas du moteur et lecture d'une frame.
- +REPOS repositionnement d'un nouveau cycle de lecture.
- +HEXA conversion ASCII-HEXADECIMAL d'un chiffre.
- +ENTHEX entrée d'un caractère hexadécimal.
- +RETCAR retour de caractères: utile dans le cas de la détection d'une erreur CKSM pour une nouvelle lecture de la ligne.
- +INTCO initialisation des divers compteurs.
- +ETIQUETTE subroutine permettant la visualisation du nom du ruban.
- +FIN subroutine indiquant la fin de la lecture du ruban (CHARGEMENT TERMINE).
- +ERREUR CKSM permet l'emission du message erreur CKSM avec l'adresse à quel niveau se situe t'elle.
- +SORMESS sortie d'un message::
 - ERREUR CKSM
 - CHARGEMENT TERMINE
 - ETIQUETTE (NOM DU RUBAN)
- +CONVER subroutine de conversion de l'adresse de l'erreur CKSM du code HEXADECIMAL au code ASCII pour permettre sa visualisation.

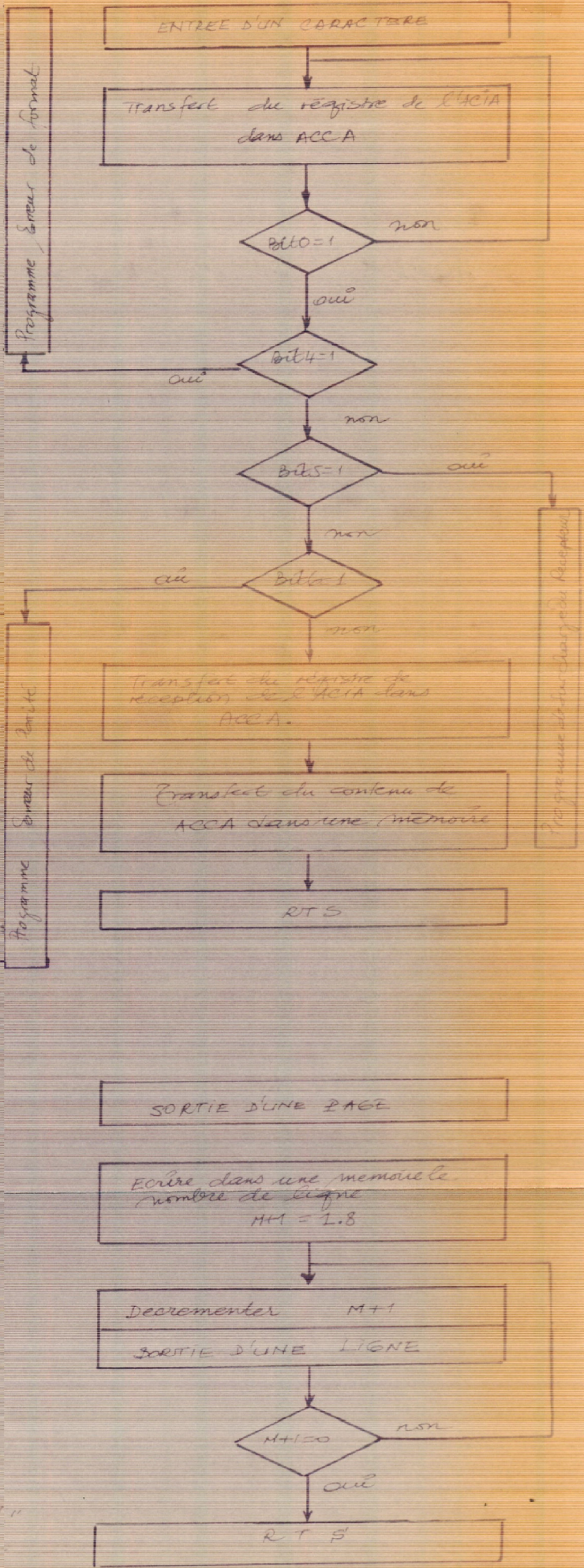
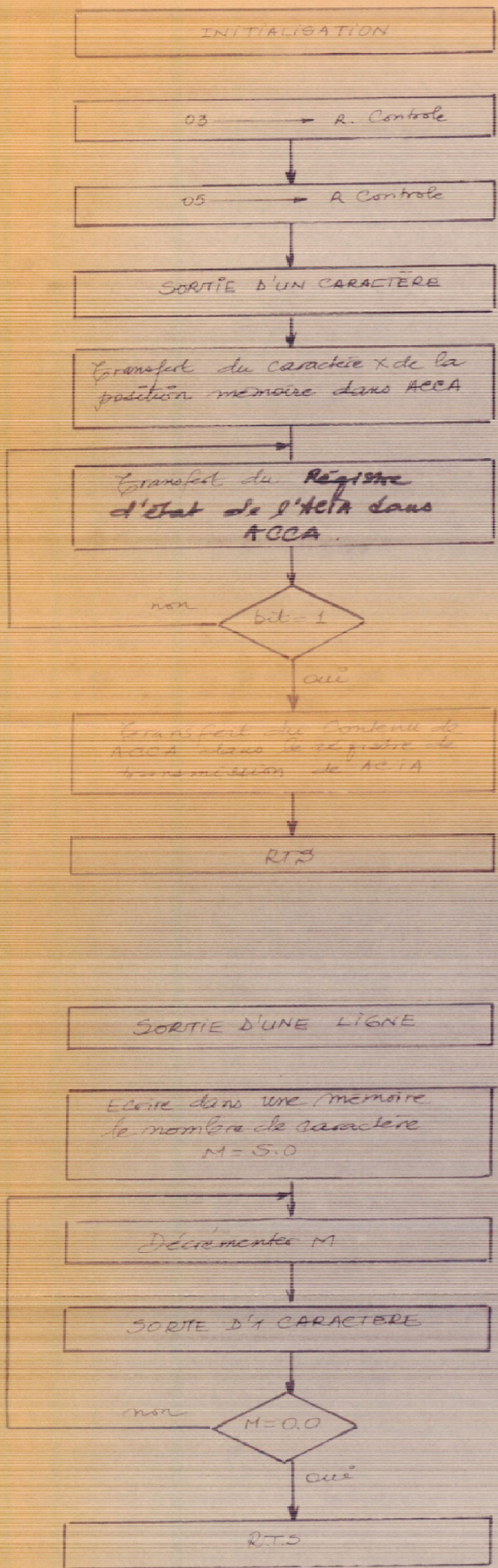
Aux pages suivantes nous donnons quelques organigrammes des messages utilisés.



SUBROUTINES "FIN" , "ERREUR CSKM " , " ETIQUETTE" , " SORMESS"

SUBROUTINES DE COMMANDE DU LECTEUR OPTIQUE

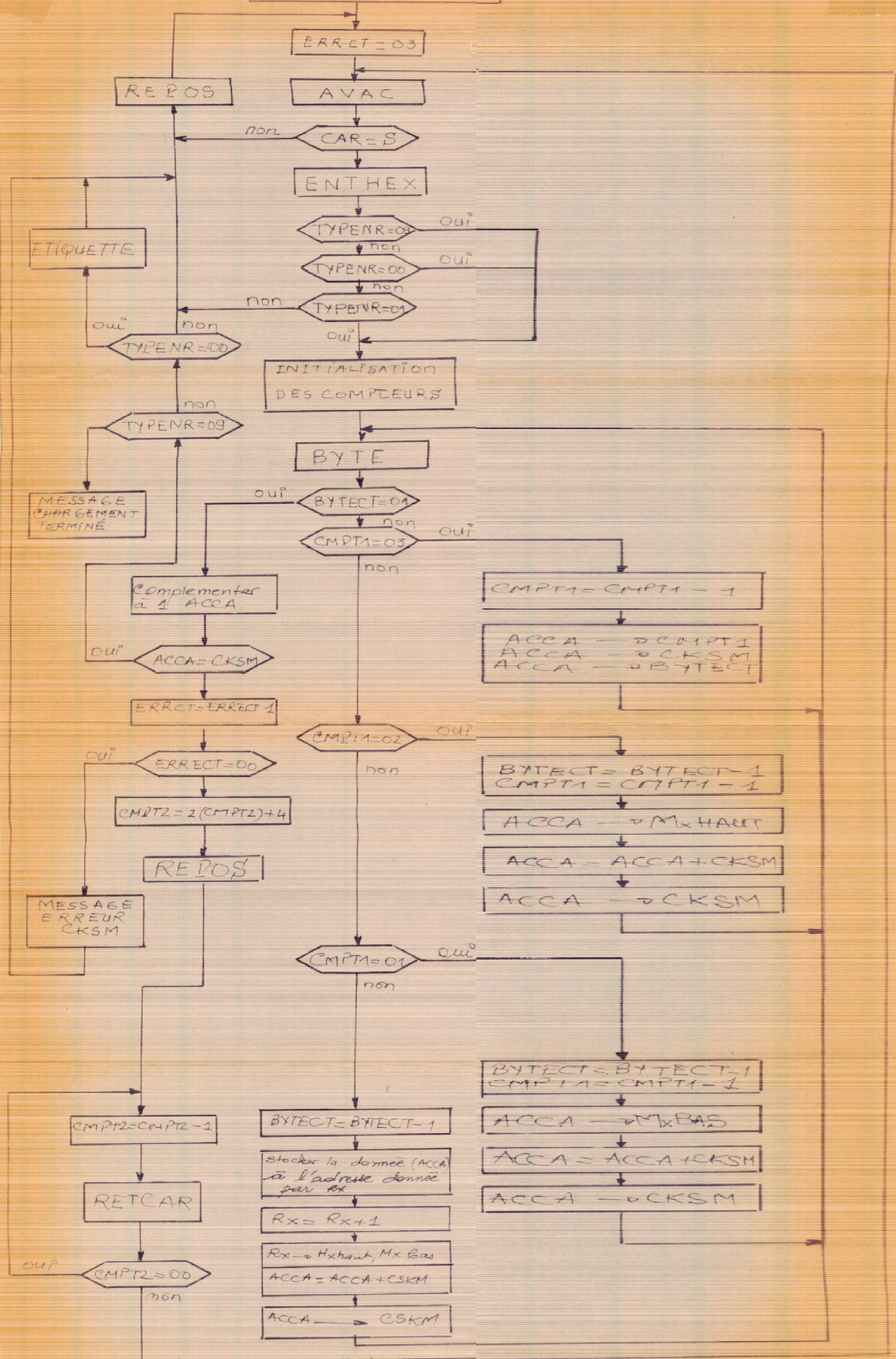


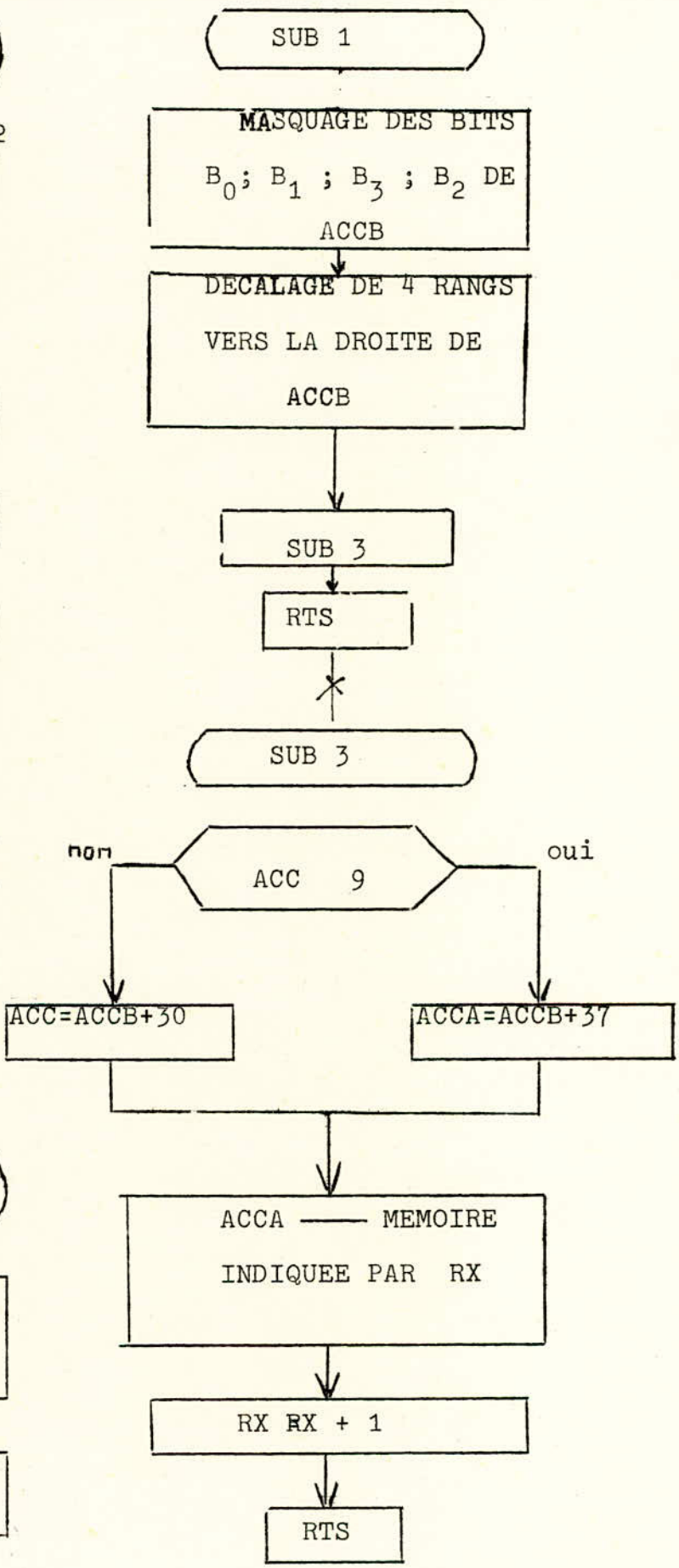
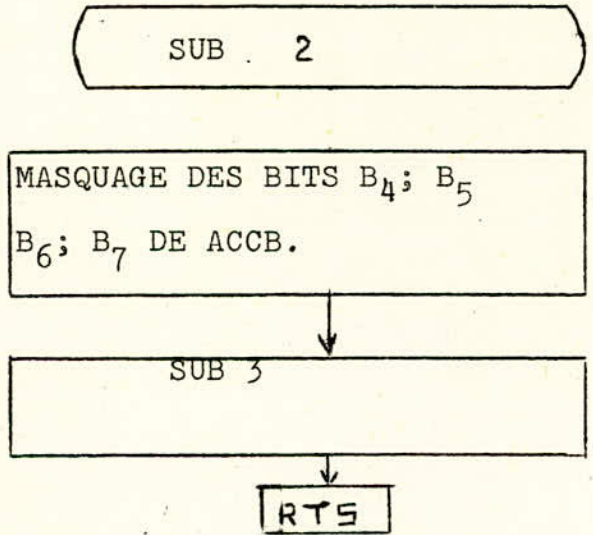
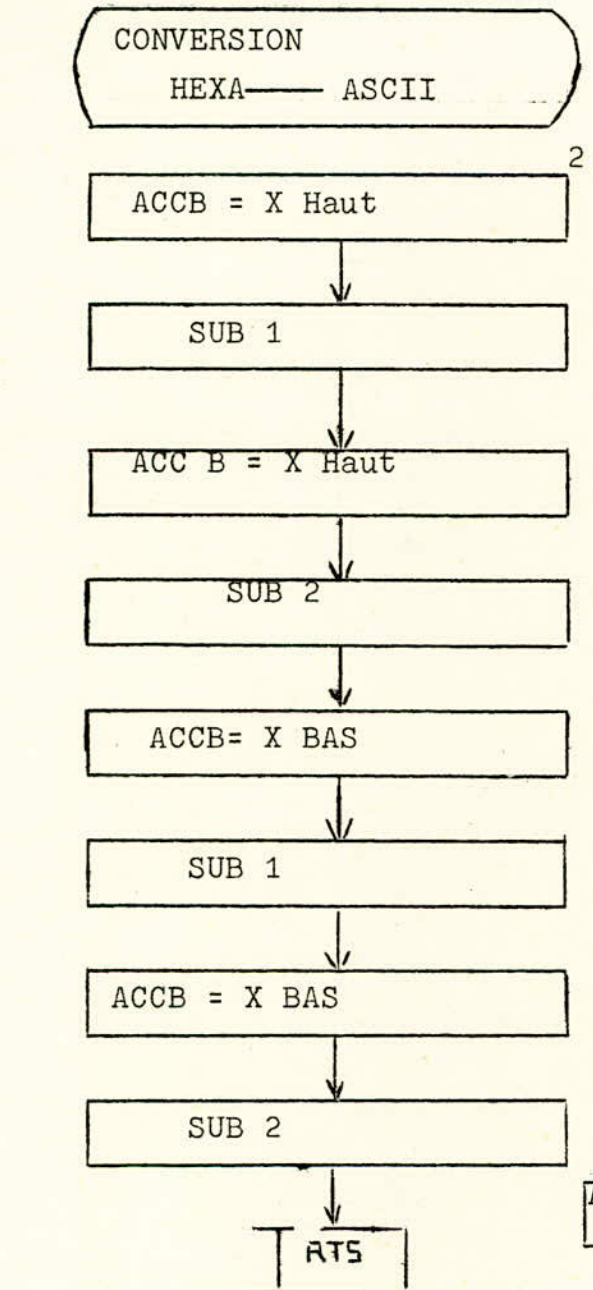


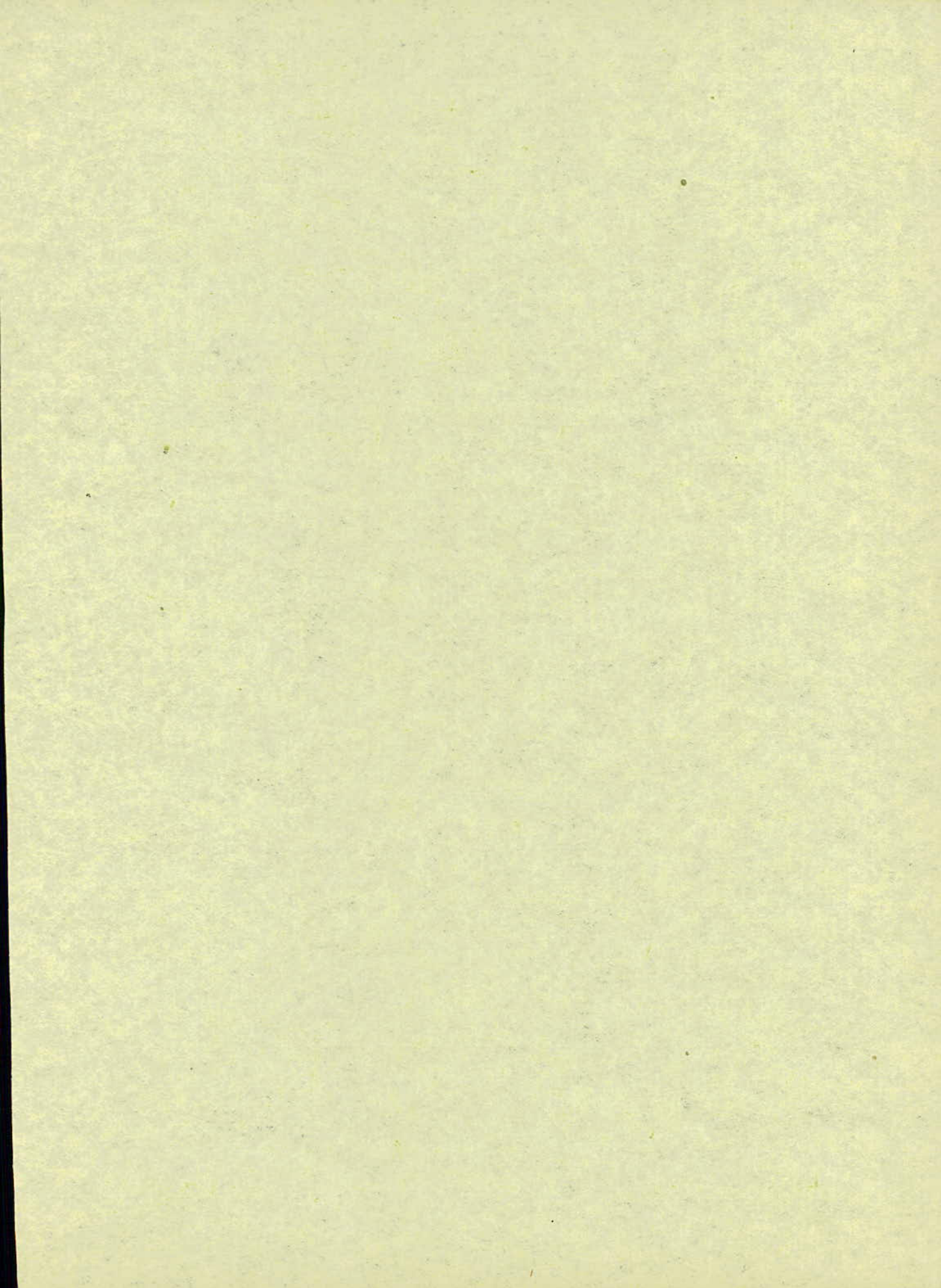
REMARQUE: l'organigramme "entrée d'une ligne" est identique à l'organigramme ci-dessus; et faut remplacer "Sortie d'un caractère" par "entrée d'un caractère". Cette remarque est valable pour les deux organigrammes.

RTS = (RETOURNER À LA SOUBROUTINE)

INITIALISATION DU PAA







IV PROGRAMME GENERAL GERANT LE LECTEUR OPTIQUE
ET L'UNITE DE VISUALISATION.

INITIALISATION DES COMPTEURS

02F3	B7	058A	INTCO	STAA	TYPENR
02F6	7F	0589		CLR	CMPT2
02F9	7F	058C		CLR	BYTECT
02FC	7F	058D		CLR	CKSM
02FF	86	03		LDAA	
0301	B7	0588		STAA	CMPT1
0304	39			RTS	

AVANCE D'UN CARACTERE

0305	C6	3C	AVAC	LDAB		CLOCK = 1
0307	F7	800A		STAB	ORB	
030A	86	02		LDAA		DELAI DE 20 MICROSECONDES
030C	4A			DECA		
030D	26	FD		BNE		
030F	C6	30		LDAB		CLOCK = 0
0311	F7	800A		STAB	ORB	REMOTE START = 0
0314	86	02		LDAA		DELAI DE 20 MICROSECONDES
0316	4A			DECA		
0317	26	FD		BNE		
0319	C6	38		LDAB		REMOTE START = 1
031B	F7	800A		STAB	ORB	
031E	CE	0100		LDX		DELAI DE 2 MILLISECONDES
0321	09			DEX		
0322	26	FD		BNE		
0324	B6	800A		LDAA	ORB	
0327	44			LSRA		
0328	24	FA		BCC		SPROCKET == 1 ?
032A	C6	18		LDAB		
032C	F7	800A		STAB	ORB	CLOCL ENABLE = 0
032F	B6	800A		LDAA	ORB	
0332	44			LSRA		
0333	44			LSRA		
0334	24	F9		BCC		READER READY = 1 ?
0336	B6	8008		LDAA	ORA	LECTURE DU RUBAN.
0339	84	7F		ANDA		MASQUAGE DU BIT DE PARITE.
033B	39			RTS		

ENTREE D'UN CARACTERE HEXADECIMAL

033C	8D	18	ENTHEX	BSR	REPOS
033E	8D	C8		BSR	AVAC
0340	8D	2A		BSR	HEXA
0342	39			RTS	

CONSTRUCTION D'UN BYTE (DEUX FRAMES,

0343	8D	F7	BYTE	BSR	ENTHEX
0345	48			ASLA	
0346	48			ASLA	
0347	48			ASLA	
0348	48			ASLA	
0349	B7	0590		STAA	FRAME1
034C	8D	EE		BSR	ENTHEX
034E	BB	0590		ADDA	FRAME1
0351	39			RTS	

MEMOIRES RESERVEES POUR SAUTS

0352	20	9F	INTCO1	BRA	INTCO
0354	20	AF	AVAC1	BRA	AVAC

REPOSITIONNEMENT D'UN NOUVEAU CYCLE
DE LECTURE

0356	C6	08	REPOS	LDAB	REMOTE OFF = 1
0358	F7	800A		STAB	
035B	C6	02		LDAB	DELAI DE 20 MICROSECONDES
035D	5A			DECB	
035E	26	FD		BNE	
0360	C6	18		LDAB	REMOTE OFF = 1
0362	F7	800A		STAB	
0365	CE	0230		LDX	
0368	09			DEC	DELAI DE 4,5 MILLISECONDES
0369	26	FD		BNE	
036B	39			RTS	

TEST HEXADECIMAL

036C	81	46	HEXA	CMPA	
036E	2E	10		BGT	NONHEX
0370	81	30		CMPA	
0372	2D	0C		BLT	NONHEX
0374	81	40		CMPA	
0376	27	08		BEQ	NONHEX
0378	80	30		SUBA	
037A	81	09		CMPA	
037C	2F	02		BLE	
037E	80	07		SUBA	
0380	39		NONHEX	RTS	

MEMOIRES RESERVEES POUR SAUTS

0381	20	C0	BYTE1	BRA	BYTE
0383	20	D1	REPOS1	BRA	REPOS

INITIALISATION DU PIA

0385	7F	8009	INTPIA	CLR	CRA
0388	7F	800B		CLR	CRB
038B	7F	8008		CLR	DDRA
à38E	86	7C		LDAA	
0390	B7	800A		STAA	DDRB
0393	86	04		LDAA	
0395	B7	8009		STAA	CRA
0398	B7	800B		STAA	CRB

SAUVEGARDE DU POINTEUR DE PILE

039B	8E	0020		LDS	PILE
------	----	------	--	-----	------

INITIALISATION DU COMPTEUR D'ERREUR

039E	86	03	INERCT	LDAA	
03A0	B7	058B		STAA	ERCT
03A3	8D	AF	ALPHA	BSR	AVAC1
03A5	81	53		CMPA	S
03A7	27	04		BEQ	ENTHEX1
03A9	8D	AB		BSR	REPOS
03AB	20	F1		BRA	INERCT
03AD	8D	8D	ENTHEX1	BSR	ENTHEX
03AF	81	09		CMPA	
03B1	27	0C		BEQ	INTCO2
03B3	81	00		CMPA	
03B5	27	08		BEQ	INTCO2
03B7	81	01		CMPA	
03B9	27	04		BEQ	INTCO2
03BB	8D	99		BSR	REPOS2
03BD	2C	DF	INERCT1	BSR	INERCT
03BF	8D	91	INTCO2	BSR	INTCO
03C1	8D	BE	BOUCLE	BSR	BYTE1
03C3	F6	058C		LDAB	BYTECT
03C6	C1	01		CMPB	
03C8	27	57		BEQ	TSTCKSM
03CA	F6	0588		LDAB	CMPT1
03CD	C1	03		CMPB	
03CF	27	42		BEQ	INBYTE
03D1	C1	02		CMPB	
03D3	27	2D		BEQ	ADHAUT
03D5	C1	01		CMPB	
03D7	27	18		BEQ	ADBAS

STOCKAGE DE LA DONNEE A SON ADRESSE

03D9	7A	058C	STOCK	DEC	BYTECT
03DC	FE	058E		LDX	Mx HAUT, Mx BAS
03DF	A7	00		STAA	X
03C1	08			INX	
03C2	FF	058E		STX	Mx HAUT, Mx BAS
03C5	BB	058D		ADDA	CKSM
03C8	B7	058D		STAA	CKSM
03CB	20	D4		BRA	BOUCLE

MEMOIRES RESERVEES POUR SAUTS

03CE	20	94	REPOS2	BRA	REPOS1
03D0	20	B2	ALPHA1	BRA	ALPHA

CONSTRUCTION D'ADRESSE X BAS

03D2	B7	058F	ADBAS	STAA	Mx BAS
03D5	7A	0588		DEC	CMPT
03D8	7A	058C		DEC	BYTECT
03DB	BB	058D		ADDA	CSKM
03DE	B7	058D		STAA	CSKM
03F1	20	BF		BRA	BOUCLE

CONSTRUCTION DE L'ADRESSE X HAUT

03F3	B7	058E	ADHAUT	STAA	Mx HAUT
03F6	7A	0588		DEC	CMPT
03FA	7A	058C		DEC	BYTECT
03FD	BB	058D		ADDA	CKSM
0400	B7	058D		STAA	CSKM
0403	20	AE		BRA	BOUCLE

ENTREE D'UN BYTE (DEUX FRAMES)

0405	7A	0588		DEC	CMPT1
0408	B7	0589		STAA	CMPT2
040B	B7	058C		STAA	BYTECT
040F	B7	058D		STAA	CKSM
0412	20	A0		BRA	BOUCLE

TEST CHECKSUM

0414	43		TSTCKSM	COMA	
0415	B1	058D		CMPA	CKSM
0418	26	11		BNE	RETLINE
041A	F6	058C		LDAB	TYPENR
041D	C1	09		CMPB	
041F	27	62		BEQ	FIN
0421	C1	00		CMPB	
0423	26	02		BNE	REPOS3
0425	8D	4E	REPOS3	BSR	ETIQUETTE
0427	8D	B7		BSR	REPOS2
0429	20	85		BRA	INERCT

RETOUR A LA LIGNE POUR UNE NOUVELLE LECTURE

042B	7A	058B	RETLINE	DEC	ERCT
042E	F6	058B		LDAB	ERCT
0431	26	02		BNE	REPOS4
0433	20	54		BRA	CKSM
0435	8D	A9	REPOS4	BSR	REPOS2
0437	86	04		LDAA	
0439	78	0589		ASLA	CMPT2
043C	BB	0589		ADDA	CMPT2
043F	B7	0589		STAA	"
0442	7A	0589	RETOUR	DEC	"
0445	8D	07		BSR	RETCAR
0447	F6	0589		LDAB	CMPT2
044A	26	F6		BNE	RETOUR
044C	20	94		BRA	ALPHA1

RETOUR DE CARACTERES

044E	C6	7C	RETCAR	LDAB	
0450	F7	800A		STAB	ORB
0453	86	03		LDAA	
0455	4A			DECA	
0456	26	FD		BNE	
0458	C6	78		LDAB	
045A	F7	800A		STAB	ORB
045D	CE	0100		LDX	
0460	09			DEX	DELAI DE 2 MILLISECONDES
0461	26	FD		BNE	
0463	B6	800A		LDAA	ORB
0466	44			LSRA	
0467	24	FA		BCC	
0469	C6	58		LDAB	
046B	F7	800A		STAB	ORB
046E	CE	0240		LDX	
0471	09			DEX	DELAI DE 4,5 MILLISECONDES
0472	26	FD		BNE	
0474	39			RTS	

VISUALISATION DU NON DU RUBAN

0475	86	04	ETIQUETTE	LDAA	
0477	B7	0008		STAA	
047A	B7	000D		STAA	
047D	CE	0000		LDX	ADRESSE DU NOM
0480	8D	4A		BSR SORMESS	SORTIE DU MESSAGE
0482	39			RTS	

MESSAGE CHARGEMENT TERMINE

0483	CE	055F	FIN	LDX	ADRESSE DU MESSAGE FIN
0486	8D	44		BSR SORMESS	SORTIE DU MESSAGE
0488	39			RTS	

MESSAGE ERREUR CKSM ADRESSE DE L'ERREUR

0489	CE	0009	ERRCKSM	LDX	
048C	8D	0C		BSR CONVER	
048E	CE	0577		LDX	ADRESSE DU MESSAGE
0491	8D	39		BSR SORMESS	
0493	CE	0009		LDX	ADRESSE DE L'ADRESSE
0496	8D	34		BSR SORMESS	
0498	20	8D		BRA REPOS3	

CONVERSION DE L'ADRESSE HEXA—— ASCII

049A	F6	058E	CONVER	LDAB Mx HAUT	
049D	8D	10		BSR SOUB1	
049F	F6	058E		LDAB Mx HAUT	
04A2	8D	14		BSR SOUB2	
04A4	F6	048A		LDAB Mx BAS	
04A7	8D	06		BSR SOUB1	
04A9	F6	058F		LDAB Mx BAS	
04AC	8D	0A		BSR SOUB2	
04AE	39			RTS	

SOUBROUTINE 1

04AF	C4	F0	SOUB1	ANDB	
04B1	54			LSRB	
04B2	54			LSRB	
04B3	54			LSRB	
04B4	54			LSRB	
04B5	8D	03		BSR SOUB 3	
04B7	39			RTS	

SOUBROUTINE 2

04B8	C4	0F	SOUB2	ANDB	
------	----	----	-------	------	--

SUBROUTINE 3

04BA	C1	09	SUB3	CMPB	
04BC	2E	07		BGT	SUB4
04BE	86	30		LDAA	
04C0	1B			ABA	
04C1	A7	00		STAA	X
04C3	08			INX	
04C4	39			RTS	

SUBROUTINE 4

04C5	86	37	SUB4	LDAA	
04C7	1B			ABA	
04C8	A7	00		STAA	X
04CA	08			INX	
04CB	39			RTS	

SORTIE DE MESAAGE

04CC	86	03	SORMESS	LDAA	
04CE	B7	800C		STAA	RC ACIA
04D1	86	11		LDAA	
04D3	B7	800C		STAA	RC ACIA
04D6	8D	06	BETA1	BSR	SORCAR
04D8	08			INX	
04D9	81	04		CMPA	CARACTERE DE FIN DE MESSAGE
04DB	26	F9		BNE	BETA1
04DD	39			RTS	

SORTIE D'UN CARACTERE

04DE	A6	00	SORCAR	LDAA	X
04E0	F6	800C	ATTENTE	LDAB	SR ACIA
04E3	57			ASRB	
04E4	57			ASRB	
04E5	24	F9		BCC	ATTENTE
04E7	B7	800D		STAA	TDR ACIA
04EA	39			RTS	

SORTIE D'UNE LIGNE

04EB	C6	50	SORLIGNE	LDAB	
04ED	F7	0591		STAB	COCAR
04F0	8D	EC	BETA2	BSR	SORCAR
04F2	08			INX	
04F3	7A	0591		DEC	COCAR
04F6	F6	"		LDAB	COCAR
04F9	26	F5		BNE	BETA2
04FB	39			RTS	

INITIALISATION DE L'ACIA

04FC	86	03	INTACIA	LDAA	
04FE	B7	800C		STAA	CR ACIA
0501	86	11		LDAA	
0503	B7	800C		STAA	CR ACIA
0507	FE	0593		LDX	ADRESSE DE LA PAGE
0509	39			RTS	

SORTIE D'UNE PAGE

050A	8E	0020	SORPAGELDS	PILE	
050D	8D	ED		BSR	INTACIA
050F	C6	18		LDAB	NOMBRE DE LIGNE = 24
0511	F7	0592		STAB	COLIGNE
0514	8D	D5	BETA3	BSR	SORLIGNE
0516	7A	0592		DEC	COLIGNE
0519	26	FD		BNE	BETA3
051E	39			RTS	

ENTREE D'UN CARACTERE

051F	F6	800C	ENCAR	LDAB	SR ACIA
0522	57			ASRB	
0523	24	FA		BCC	ENCAR
0525	57			ASRB	
0526	57			ASRB	
0527	57			ASRB	
0528	57			ASRB	
0529	24	F4		BCC	ENCAR
052B	57			ASRB	
052C	24	F1		BCC	ENCAR
052E	57			ASRB	
052F	24	EE		BCC	ENCAR
0531	B6	800D		LDAA	RDR ACIA
0534	A7	00		STAA	X
0536	39			RTS	

ENTREE D'UNE LIGNE

0537	C6	500	ENLIGNE	LDAB	NOMBRE DE CARACTERES = 80
0539	F7	0591		STAB	COCAR
053C	8D	E1	BETA	BSR	ENCAR
053E	8D	9E		BSR	SORCAR
0540	08			INX	
0541	7A	0581		DEC	COCAR
0544	F6	0591		LDAB	COCAR
0547	26	F3		BNE	BETA4
0549	39			RTS	

ENTREE D'UNE PAGE

054A	8E	0020	ENPAGE	LDS	PILE	
054D	8D	AD		BSR	INTACIA	
054F	C6	18		LDAB		
0551	F7	0592	BETA5	STAB	COLIGNE	NOMBRE DE LIGNE = 24
0554	8D	E1		BSR	ENLIGNE	
0556	7A	0592		DEC	COLIGNE	
0559	B6	0592		LDAB	COLIGNE	
055C	26	F6		BNE	BETA5	
055E	39			RTS		

MESSAGE CHARGEMANT TERMINE

055F	0A		SAUT A LA LIGNE SUIVANTE
0560	0D		RETOUR CHARIOT
0561	20		LAISSER UN ESPACE
0562	43	C	
0563	48	H	
0564	41	A	
0565	52	R	
0566	47	G	
0567	45	E	
0568	4D	M	
0569	45	E	
056A	4E	N	
056B	54	T	
056C	20		LAISSER UN ESPACE
056D	54	T	
056E	45	E	
056F	52	R	
0570	4D	M	
0571	49	I	
0572	4E	N	
0573	45	E	
0574	0A		SAUT A LA LIGNE SUIVANTE
0575	0D		RETOUR CHARIOT
0576	04		CARACTERE INDIQUANT LA FIN DU MESSAGE

MESSAGE ERREUR CKSM

0577	0A	SAUT A LA LIGNE SUIVANTE
0578	0D	RETOUR CHARIOT
0579	20	LAISSER UN ESPACE
057A	45	E
057B	52	R
057C	52	R
057D	45	E
057E	55	U
057F	52	R
0580	20	LAISSER UN ESPACE
0581	43	C
0582	4B	K
0583	53	S
0584	4D	M
0585	20	LAISSER UN ESPACE
0586	20	LAISSER UN ESPACE
0587	04	CARACTERE INDIQUANT LA FIN DU MESSAGE

ADRESSE DE L'ERREUR CKSM

0009	XX	
000A	XX	
000B	XX	
000C	XX	
000D		CARACTERE INDIQUANT LA FIN DU MESSAGE

MESSAGE DE L'ETIQUETTE

0000	XX	
0001	XX	
0002	XX	
0003	XX	
0004	XX	
0005	XX	
0006	XX	
0007	XX	
0008	04	CARACTERE INDIQUANT LA FIN DU MESSAGE

REMARQUE:

XX: CARACTERE VARIANT SUIVANT LE NOM DU RUBAN
ET DE L'ADRESSE DE L'ERREUR CKSM.

MEMOIRES (RAM) RESERVEES

0588	CMPT1	COMPTEUR PERMETTANT DE DISTINGUER LE TYPE DE LECTURE / BYTECT , PROGRAMME OU ADRESSE
0589	CMPT2	COMPTEUR SERVANT AU COMPTAGE DU NOMBRE DE PAS DE RETOUR DU PROGRAMME DANS LE CAS OU LE LECTEUR DETECTE UNE ERREUR CKSM.
058A	TYPENR	SERT A DETECTER LE TYPE D'ENREGISTREMENT POUR L'EMISSION DU MESSAGE "CHARGEMENT TERMINE" OU DE L'ETIQUETTE.
058B	ERCT	COMPTEUR D'ERREUR
058C	BYTECT	COMPTEUR DE BYTE
058D	CKSM	CHECKSUM
058E	MxHAUT	MEMOIRE POUR LA SAUVEGARDE DE L'ADRESSE HAUT.
058F	MxBAS	MEMOIRE POUR LA SAUVEGARDE DE L'ADRESSE BAS.
0590	FRAME1	MEMOIRE SERVANT A LA SAUVGARDE DE LA PREMIERE FRAME LUE.
0591	COCAR	COMPTEUR DE CARACTERES POUR UNE LIGNE.
0592	COLIGNE	COMPTEUR DE LIGNES.

V-CERTAINES AMELIORATIONS POSSIBLES:

Vu le temps très limité que nous avons eu pour étudier et réaliser ce projet de fin d'études nous nous proposons à donner certaines améliorations possibles pouvant être faites pour donner plus de souplesse et un champ d'application très vaste à notre système:

+Le programme général peut être introduit dans le programme de l'EXBUG de l'EXORCISER.

+Associé a notre programme du lecteur optique un sous-programme tel-que:

-Le lecteur optique peut lire plusieurs programmes incorporés dans un même ruban.

-Le lecteur optique lit un seul programme parmi toute un serie **enregistrée** sur un même ruban en détectant son nom "ETIQUETTE".

+Associe au programme de l'unité de visualisation d'autres programmes qui permettent un meilleur dialogue entre l'opérateur et la machine.

CONCLUSION

— 00 —

TOUTE CREATION TECHNIQUE PORTE EN SOI SA
PROPRE LIMITATION.

FRANCHIR CES LIMITES JUSQU'A EN RENCONTRER
DE NOUVELLES, ET AINSI DE SUITE, CONSTITUE LA
TACHE DU CHERCHEUR ET DU TECHNICIEN.

LES ETAPES DE CETTE PROGRESSION SONT LES
MAILLONS DE LA CHAINE DU PROGRES

----- LE PROGRES -----

— 0 —