

UNIVERSITE D'ALGER

19/79

1 ex

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT ELECTRICITE

ELECTRONIQUE



THESE DE FIN D'ETUDES

ETUDE DE LA LIAISON
MICROPROCESSEUR-ECRANTY
A L'AIDE DU CRTC MC 6845



Propasé par :

Mme G. MONDON

Etudié par :

MESSAD Cherif

NACER ALI

Promotion Janvier 1979

UNIVERSITE D'ALGER

ECOLE NATIONALE POLYTECHNIQUE
DEPARTEMENT ELECTRICITE

THESE DE FIN D'ETUDES

ETUDE DE LA LIAISON
MICROPROCESSEUR-ECRANTY
A L'AIDE DU CRTC MC 6845

Propasé par :

Mme G. MONDON

Etudié par :

MESSAD Cherif

NACER ALI

Promotion Janvier 1979

A mon père , à ma mère.

A mes frères et soeurs .

A mes amis .

M. Chérif .

_____ 0 _____

A mon père.

A mes frères et soeur.

A mes amis .

Nacer Ali .

Que M^{me} G. MONDON, Docteur Ingénieur, daigne trouver ici, l'expression de tous nos remerciements et toute notre gratitude pour nous avoir guidé dans l'élaboration de ce travail.

Nos remerciements vont également à Monsieur TEDJINI , Docteur Ingénieur au CSTN , pour nous avoir accepté dans son service et pour les précieux conseils qu'il nous a prodigués.

Introduction

Chapitre I :

- Le microprocesseur MC 6800 .

I.1. Organisation interne du MC 6800 .

I.2. Differentes lignes du MC 6800 .

Chapitre II:

Support du MC 6800 :

II .1. La mémoire vive RAM .

II . 2. La mémoire morte ROM (PROM, E PROM)

II . 3. Les interfaces d'entrées/Sorties.

II.3.1 Le PIA .

II.3.2 L'ACIA .

Chapitre III:

Modes d'adressage :

III.1. Inhérent .

III.2. Immédiat.

III.3. Etendu et direct.

III.4. Relatif.

III.5. Indexé.

Chapitre IV :

Système de Visualisation:

IV.1. Rappels sur la T.V. (Standard 625 lignes).

IV.2. Interface de Visualisation (Etude du MC 6845).

IV.3. Description du Système de visualisation:

IV.3.1. Determination de la fréquence de l'oscillateur.

IV.3.2. Fonctionnement du Système.

IV.3.3. Mémoire de rafraichissement .

IV.3.4. Mémoire de caractères.

Chapitre V :

Programmation :

V.1. Initialisation.

V.2. Adressage du générateur de caractères et visualisation.

ANNEXE

Determination des circuits partenaires :

1. La mémoire de rafraichissement.
2. Le registre à décalage.
3. La Sortie Vidéo.
4. Les "BUFFERS à 3 états."
5. Le multiplexeur.
6. Le circuit tampon "LATCH".
7. Le clavier alphanumérique .

- CONCLUSION -

00.

Bibliographie .

Introduction .

L'objet de notre travail est l'étude de la liaison

Processeur - écran TV (Standard 625 lignes) .

Topologiquement il s'agit d'une chaîne de traitement dont une extrémité est constituée par le processeur et l'autre par une unité d'affichage.

Le processeur constitué par le microprocesseur MC 6800 et des mémoires qui stockent le programme de travail et les données, constitue l'unité centrale de traitement.

Quant aux fonctions complexes d'interfaces entre le processeur et l'écran TV, elles sont assurées par un circuit spécialisé appelé contrôleur d'écran Cathodique. (CRTC) . Dans notre application on a utilisé le MC 6845 de Motorola .

Ce circuit est capable de gérer les signaux nécessaires à l'affichage du contenu d'une mémoire sous forme de caractères sur l'écran TV .

Comme tel , il est chargé des tâches suivantes :

gestion des transferts, gestion de la mémoire de rafraichissement, mise en oeuvre des générateurs de caractères, gestion de l'écran et du curseur.

Le boîtier du " CRTC " doit s'attacher correctement à quelques boîtiers intégrés partenaires : La mémoire de rafraichissement de l'écran (qui est en fait une mémoire vive ordinaire) , le générateur de caractères et ses prolongements dans la vidéo (- registre à décalage, convertisseur Digital/ analogique , et modulateur VHF), et les compteurs aux bus de programmation tant du CRTC que du texte en mémoire.

(Multiplexeur, Buffers a 3 états, latch) .

L'unité d'affichage est en fait un téléviseur (Standard 625 lignes) dont la partie de synchronisation et balayage du texte sur l'écran est gérée par le CRTC.

Pour être utilisable ce circuit doit être commandé par un programme.

D'abord le MPU charge les registres internes du CRTC à partir d'une table firmware, pour l'adapter aux conditions de balayage et de synchronisation du **moniteur**

Ensuite il générera des adresses pour adresser le générateur de caractères lequel va délivrer à sa sortie une information affichable.

On va ainsi aboutir à une topologie plus complexe d'une unité de visualisation conversationnelle.

Chapitre I

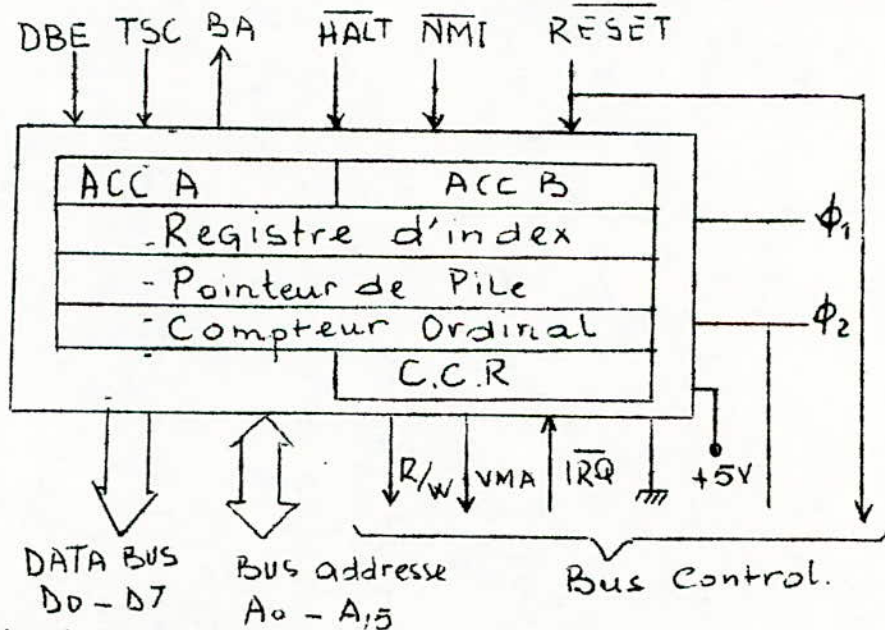
M . P . U .

Le microprocesseur MC 6800 constitue l'unité centrale de traitement commandé par un programme et remplissant les fonctions de micro - ordinateur.

Pour être utilisable, le microprocesseur doit être relié à des mémoires qui stockent le programme de travail et les données. Il a des circuits d'entrées -Sorties qui le branchent sur le monde externe que l'on regroupe sous le nom de périphériques: les unités de dialogue avec l'utilisateur.

Dans ce chapitre nous donnerons un aperçu sur le contenu du MPU accessible au programmeur.

Ainsi que les différentes lignes d'entrées/sorties et les différents signaux de contrôle. Finalement les différents modes d'adressage utilisé par le MC 6800.



1. 1. Organisation interne du MPU :

Le MPU est constitué de :

1) - Une unité arithmétique et logique . (ALU).

C'est elle qui exécute les opérations arithmétiques et logiques.

2°) Deux accumulateurs (A et B) de 8 bits chacun qui ont pour rôle principal de maintenir les operandes et les resultats de l'ALU .

3°) Un registre d'index de 16 bits utilisé pour le stockage de données et surtout pour specifier une adresse precise.

4°) Un pointeur de pile ou "Stack Pointer" qui est également positionné de 16 bits. La pile est faite de façon à apparaître comme une mémoire dans laquelle les octets peuvent être empilés les uns après les autres. Elle est utilisée pour la sauvegarde des données contenues dans les registres lors des interruptions.

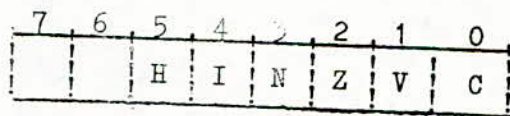
Le pointeur de pile retient l'adresse de la prochaine position libre dans la pile.

5°) Un compteur de programme ou "Programme Counter (PC)", est un registre de 16 bits qui retient l'adresse de la prochaine instruction du programme qui doit être exécutée. Lorsque le MPU a trouvé l'instruction adressée par le contenu du PC, ce dernier est incrémenté à l'adresse de la prochaine instruction.

6°) Un registre de code de condition (C C R) :

- Il est composé de 8 bits, qui après chaque opération effectuée par le MPU prennent des états pouvant nous renseigner sur le résultat.

Les bits 6 et 7 sont sans attribution, ils sont constamment à " 1 " :



a/ " Half carry " H

H = 1 veut dire qu'il existe une retenue par suite d'une addition entre les 4 premiers bits des accumulateurs A et B .

b/ Interrupt mask : I

L'indicateur de masquage intervient lorsqu'une interruption vient d'être prise en compte et que l'on veut interdire l'accès à toute autre interruption.

Il est remis à zéro lorsqu'on autorise de nouveau les interruptions à se manifester.

Pour I = 1 la demande d'interruption (\overline{IRQ}) est rejetée.

c) " Negatif " N :

Si le résultat de l'opération qui vient d'être exécutée a mené à un résultat négatif, ce bit se positionne à 1.

Il sert lors de branchements conditionnels qui seront définis par son test.

d) " Zero " Z :

Il signale que le résultat d'une opération est nul en se positionnant à 1. Si ce résultat n'est pas nul, il reste à Zéro.

L'une de ses fonctions est suivante: lors d'un décomptage, on détecte le passage à zéro en interrogeant uniquement cet indicateur.

Ce dernier interviendra par conséquent dans les ordres de branchements conditionnels.

e) Overflow V :

Ce bit nous renseigne sur le dépassement de capacité des registres A et B .

f) Carry C :

Il indique s'il y a une retenue (report) provenant du bit 7 à la suite d'une opération entre accumulateurs.

I.2 Differentes lignes du MPU:

Les entrées/sorties du MC 6800 peuvent être divisées en 3 catégories de fonctions:

- Fonction Data - Data bus
- Fonction Adresse - Bus adress
- Fonction Control - Bus control.

I.2.1 Data Bus (Do - D₇) :

Il est composé de 8 lignes bidirectionnelles permettant le transfert de données entre le MC 6800 et les différents éléments de la famille qui lui sont reliés.

I.2.2 Bus adress (Ao - A 15):

Ces 16 lignes sont sortantes du MPU et possèdent 3 états (0, 1, off);
l'état off correspond à une haute impédance (ligne déconnectée),
l'état off est utilisé pour le mode D.M.A (Direct Memory access).

I.2.3 Bus Control:

Constitué par les lignes suivantes:

a) - R/W Read ~~write~~ :

Il indique la direction de transfert des données

R/W = 1: le MPU effectue une opération de lecture

R/W = 0: le MPU effectue une opération d'écriture.

b) - VMA Valid Memory adress:

Valid l'adresse présente sur le bus adress.

c) $\overline{\text{Reset}}$:

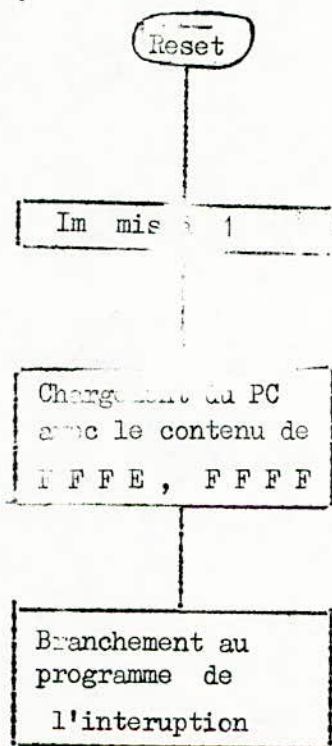
On peut considérer le " reset " comme une interruption.

Il vise à initialiser le système, à la mise sous tension, par exemple.

De ce fait, il mettra souvent en service un programme dit d'initialisation, qui fournira aux divers registres les informations utiles de démarrage: Valeur initiale du PC, du SP, etc.

Il interviendra également chaque fois qu'on voudra revenir à la situation initiale en cours de service.

Par conséquent, il n'y a plus lieu de sauvegarder des contenus de registres, non plus que de tester la ou de le porter à 1"; l'organigramme suivant résume la fonction du "Reset " :



d) Horloge $\overline{Q_2}$:

C'est une phase de l'horloge en opposition de phase avec $\overline{Q_1}$.

Elle sert à mettre en liaison l'élément sélectionné du système avec le MPU via le bus data et ceci seulement lorsque le bus adresse et le VMA sont stables.

e) Interrupt request $\overline{I R Q}$:

$\overline{I R Q}$: Ce signal est généré par le PIA, l'ACiA .

Il indique la présence ou l'absence de demande d'interruption de la part de ces dernières.

Tout d'abord, le MPU termine l'instruction en cours; puis, il teste le bit de masquage, noté Im, du CCR.

Si un masque est établi (Im = 1) le MPU poursuit l'exécution de son programme; sinon (Im=0), le MPU range le contenu de ses registre PC,CCR,ACCA,ACCB dans la pile de sauvegarde, dont le niveau de remplissage est indiqué par un pointeur de pile. (" Stack pointer").

Après quoi, le MPU établit le masque en portant I_m à 1, ce qui permettra éventuellement au programme de déterminer dans quel ordre les interruptions seront traitées.

Si d'autres interruptions prioritaires sont autorisées à intervenir, I_m sera aussitôt remis à zéro.

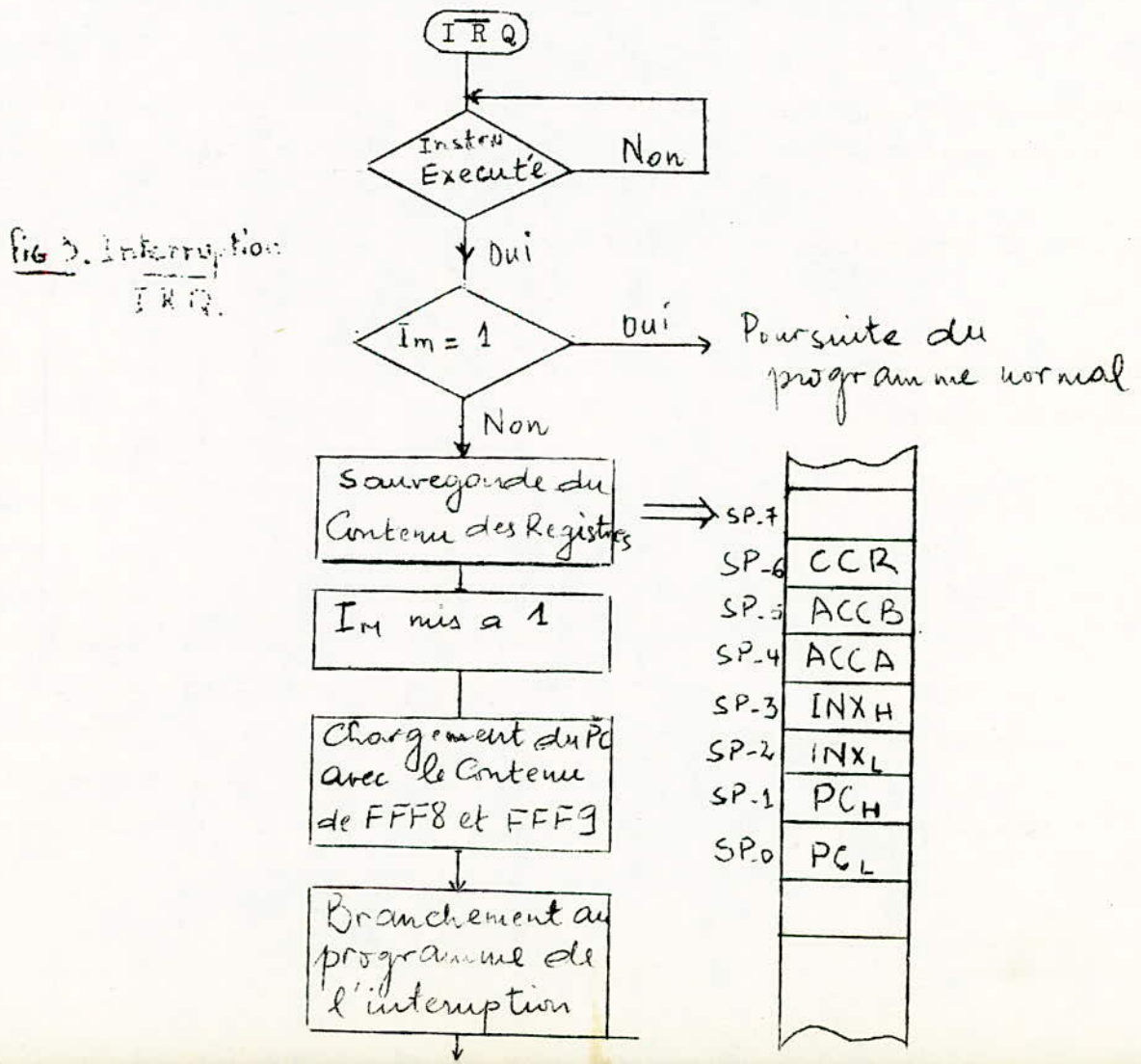
Le MPU se branche alors sur l'adresse du Vecteur \overline{IRQ} et va lire dans les cellules mémoires qui sont ici obligatoirement F F F 8 et F F F 9 l'adresse du sous-programme spécifique de l'interruption.

L'adresse contenue dans les cellules F F F 8 et F F F 9 sur 16 bits, est chargée dans le compteur ordinal.

Désormais commencera le traitement de l'interruption proprement dite, la procédure de son acquisition étant terminée.

A l'issue du traitement, une instruction de fin d'interruption et par conséquent de retour au programme principal (R T I), termine le programme propre à l'interruption.

Nous illustrons cette procédure par l'organigramme de la figure suivante:



Les pins VMA et BA sont forcés dans leurs niveau bas .

TSC n'a aucun effet sur le data bus .

TSC est utilisé par exemple pour mettre l'accès direct aux mémoires :

(DMA)

Notant que le MPU est un système dynamique, il ne peut rester dans cet état un temps dépassant (415 μ s) sinon les données qui sont à l'intérieur du MPU seront perdues.

i) Halt :

Quand ce signal est à un niveau bas, toute activité dans la machine est arrêtée après l'exécution de l'instruction en cours.

Tous les signaux caractérisant les 3 états des lignes seront à l'état haute impédance; le VMA est à zéro et BA au niveau haut .

k) Bus Available:

Ce signal est généralement à l'état inactif (état " 0 "); le MPU fonctionne alors normalement.

Il est porté à l'état " 1 " actif lorsque survient une halte ou un état d'attente (Wait)

La sortie Bus available doit être utilisée pour signaler à un élément extérieur que le MPU ne possède plus le contrôle des bus, ce qui est intéressant pour un multiprocessing ou un accès direct aux mémoires.

f) ϕ_1 : Une des 2 phases d'horloges prévue pour activer le MPU : $\phi_1 = \overline{\phi_2}$

g) NMI Non Maskable Interrupt:

Le MPU dispose d'une entrée d'interruption non masquable à laquelle est attribué le niveau prioritaire.

Par exemple, une baisse de tension d'alimentation annonçant sa coupure se traduira par une commande d'interruption de toute première urgence.

$\overline{\text{NMI}}$ est similaire à $\overline{\text{IRQ}}$ seulement, puisque celle-ci dispose de la priorité absolue ; par conséquent, la situation du bit d'état I_m , au départ ne joue aucun rôle. Le vecteur se trouve dans les cellules F F F C et F F F D car cette interruption est également vectorisée.

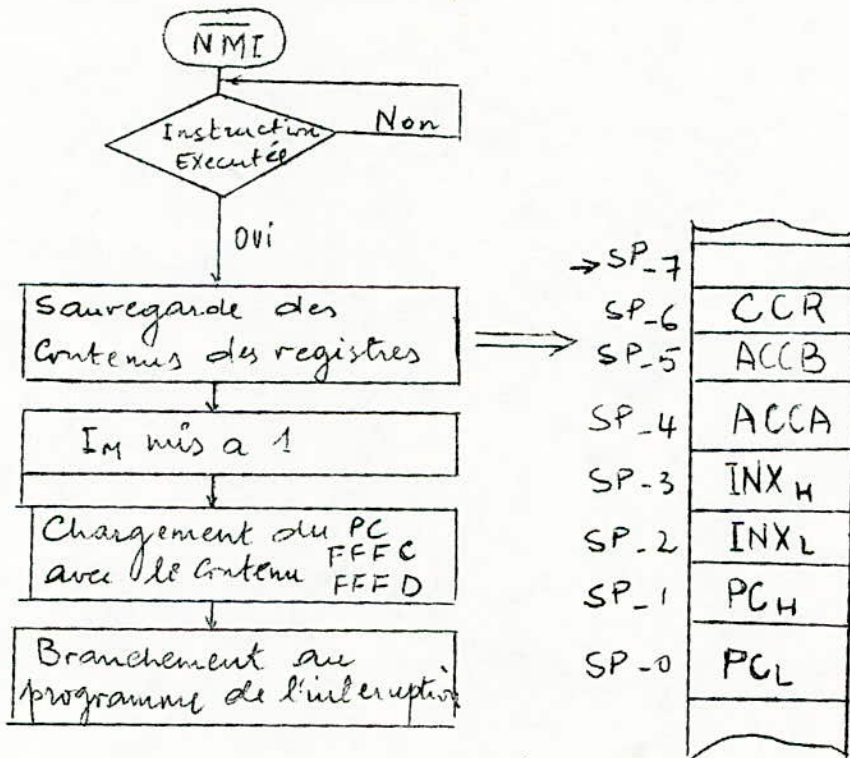


FIG 4. NMI

h) DBE Data Bus Enable:

C'est un signal qui caractérise les 3 états du bus data :

- l'état fermé sens rentrant,
- l'état fermé sens sortant,
- l'état ouvert.

DBE = 1, le bus data est dans l'état " 0 ou 1 "

DBE = 0, le bus data est dans l'état off

Ce signal dérive généralement de ϕ_2

i) TSC Three State Control:

Ce signal place les lignes d'adresse et R/W dans l'état off (haute Impedance).

Avec le MC 6800 comme organe central, les autres éléments de la famille M6800 lui sont reliés directement par le réseau des bus.

II.1 La Mémoire RAM:

La mémoire vive (RAM : Random Access Memory) est utilisée pour stocker des informations variables dans un système microprocesseur.

Ainsi l'unité centrale, sous contrôle du programme, peut à volonté lire ou changer le contenu d'une position mémoire .

Les mémoires vives peuvent être de deux sortes: statiques ou dynamiques .

Dans ces dernières, l'information est stockée sous forme électrique dans la capacité porte (Gate) d'un transistor MOS. Ces mémoires nécessitent un rafraîchissement périodique.

Les RAM statiques n'ont pas besoin d'être rafraîchies, car les cellules mémoires sont à 2 états et de conception similaire au flip-flop traditionnel.

La plupart des mémoires vives à semi conducteur sont volatiles; cela veut dire que l'information stockée est perdue lorsque l'on supprime l'alimentation.

Exemple la RAM MCM 6810 :

La figure ci-dessous illustre cette RAM est organisée en mots de 8 bits. Sept entrées d'adresse ($A_0 - A_6$) permettent d'adresser n'importe laquelle des 128 positions ou octet.

Elle possède quatre entrées " chip select " sensibles à un niveau bas et deux sensibles à un niveau haut.

Lorsque la mémoire est constituée par plusieurs boîtiers,

- 1er) Il faut sélectionner le boîtier,
- 2em) Il faut adresser la position donnée au sein de ce boîtier.

Le premier de ces niveaux d'adresse est appelé "chip select"(sélection du boîtier).

II.2 . Mémoire Morte (ROM).

A l'opposé de la RAM, la ROM est non volatile.

Elle permet de stocker des informations figées (programme des instructions et des données constantes) et de les restituer sur demande.

Les ROM peuvent être réparties en trois groupes:

Le 1er est composé des ROM programmées par masque: la programmation est exécutée par le fabricant selon les directives de l'utilisateur. Une fois ces ROM fabriquées, leur contenu ne peut plus être modifié.

Le 2^{eme} groupe est celui des ROM programmables ou PROM qui sont programmés par l'utilisateur.

Il existe un autre type de PROM dans laquelle l'information peut être effacée (EPROM) en exposant le substrat à un flux de radiation Ultra Violet.

Le dernier groupe de ROM se compose des EAROM(programmables electriquement) . Ces produit sont initialement destinés a êtres utilisés en tant que ROM .

II. 3 Les interfaces d'entrée/ Sortie:

LES interfaces d'entrée/Sortie permettent de relier le microprocesseur avec le milieu exterieur.

II. 3. 1 Interface Parallele (PIA) :

Dans la famille M 6800, le PIA (MC 6820) permet de relier un ou plusieurs peripheriques à transfert parallèles au MPU .

(Generalment sans aucun circuit supplémentaire .)

Le PIA Contient deux blocs de circuits d'E/S dont chacun est capable de gérer independamment un bus peripherique à 8 bits.

Une particularité du PIA, réside dans ses registres internes qui non seulement le rendent programmable, mais permettant également un stockage temporaire des données afin de faciliter leur transfert.

Associés à chaque partie, il y a 8 signaux de données (PA₀ - PA₇) pour la partie A et (PB₀ - PB₇) pour la partie B .

Chaque partie possède 2 signaux de controle (CA₁ , CA₂) partie A , (CB₁ ,CB₂) partie B.

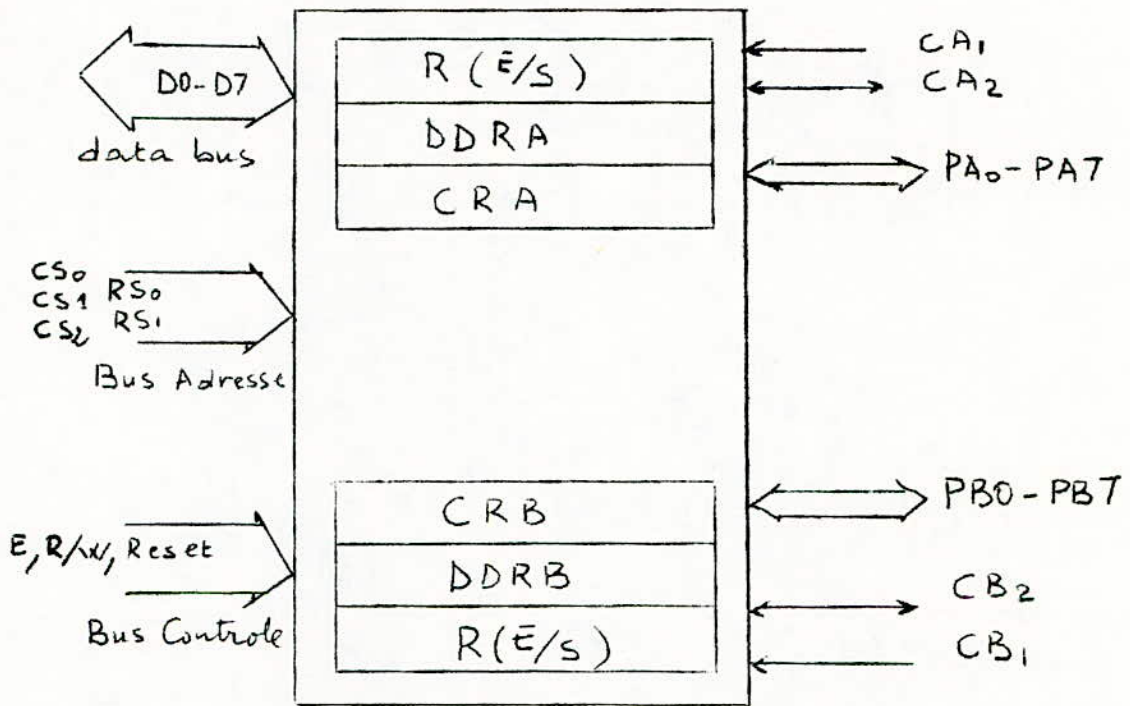
Ces lignes de données et ces signaux de controle, du coté peripherique, peuvent être programmées comme entrée ou sortie (grace à DDRA , DDRB).

L'interface du PIA avec le MPU comprend :

- 8 bits pour les données (D₀ - D₇),
- 5 bits adresses parmi les 16 bits du Bus Adress (CS₀, CS₁, CS₂ ,RS₀,RS₁),
- 5 lignes de controles. (E, R/W , Resst).

Il est à noter que le fonctionnement du PIA est entièrement programmé par le MPU .

FIG. PIA Registers



DDRA : Registre de direction de données partie A.

CRA : Registre de controle

R(E/S): Registre d'entrée sortie

II.3.2. Interface Serie Asynchrone (ACIA):

Cette interface convient pour les transmissions de données en mode serie, alors que le bus du systeme M6800 ne fonctionne qu'en mode parallèle.

L'interface MPU / ACIA necessite :

- 8 bits pour les données ,
- 4 bits adresse parmi les 16 du bus adress ,
- 3 lignes de controle.

Modes d'adressage

Par définition une instruction signifiera une expression complète, en programme Source, Spécifiant une action déterminée et une donnée ou l'adresse d'une donnée représentant ce sur quoi l'action doit être faite.

Malheureusement le microprocesseur ne comprend pas le langage mnémotechnique. Par conséquent, il va falloir lui traduire tout cela en binaire.

Un programme est une suite d'instructions placées en mémoire. Le MPU exécute ce programme en allant chercher une par une dans l'ordre, les instructions dans le mémoire puis les exécuter.

Généralement l'instruction est composée, d'un **byte** portant le code opération. Ce byte est suffisant pour définir le mode d'adressage.

Le second byte (éventuellement le troisième) contiennent soit un **opérande**, soit une adresse ou un déplacement.

La lecture du 1er byte engendre une séquence de micro-instructions.

Avec le MC 6800 cinq modes d'adressage sont utilisés.

- Adressage inherent
- Adressage immédiat
- Adressage Indexé
- Adressage étendu et direct
- Adressage relatif

III. 1er Adressage Inherent.

C'est l'instruction elle-même, plus exactement son code opération, qui contient l'adresse où se trouve la donnée sur laquelle va porter l'opération.

Exemple :

byte 1	byte 2
ADD A	3C2F H

signifie additionner au contenu de l'accumulateur A la valeur 3C2F (en hexa). Si l'on avait voulu additionner cette même valeur au contenu de l'accumulateur B on aurait simplement modifié le code opération l'instruction devenant **alors :**

ADD B 3C2F H

Le principe de cet adressage:

le MPU est relié à une RAM qui n'interviendra pas, et à une ROM. Le P.C contient l'adresse de l'instruction, celle-ci est lue, revient dans le registre d'instruction puis est exécutée.

III . 2 . Adressage Immédiat :

Dans ce mode d'adressage, l'instruction ne contient pas d'adresse du tout, à la place de l'adresse figure l'opérande sur lequel va porter l'opération .

(Il n'est donc pas besoin de cycles mémoire) .

Deux cas sont à envisager, Suivant que l'on s'adresse à des registres de 8 bits (ACCA et ACCB) ou à des registres de 16 bits (R INX, SP.)

Pour le premier l'instruction nécessite 2 byte :

EX : LDAA 10 (en code machine 86 10)

Cette instruction demande le chargement immédiat de l'accumulateur par le nombre 10 .

L'opérande peut prendre au maximum la valeur FF il y a donc 255 possibilités.

Mais une instructions chargeant un registre de 16 bits nécessitera un 2^{em} octet de donnée soit 3 octets au total.

EX : LDX Ao oI en code machine CE Ao oI le registre d'indexe est chargé immédiatement par Ao. oI.

III . 3 . Adressage Direct et Étendu :

Les modes d'adressage étendu et Direct, ne diffèrent que par le nombre de bytes formant l'opérande.

Le mode direct ne permet d'adresser que des positions de mémoire comprise entre 0 et 255 .

Exemple : Le microprocesseur après avoir lu l'instruction STAB à la position mémoire 5004 trouve l'adresse (100) de la position de mémoires vers laquelle la MPU va envoyer le mot 35 contenu dans l'accumulateur B : **STAB 100**

L'adressage direct se fait sur 2 bytes et permet par rapport à l'adressage étendu de réduire la capacité de mémoire de programme de 1/3 et gagne un cycle mémoire lors de la lecture de l'adresse.

L'adressage étendu utilise 3 bytes et permet d'adresser des positions de mémoires de 0 à 65336.

III . 4. Adressage relatif :

Ce mode n'est pas utilisé pour définir la donnée de l'instruction, mais pour permettre de modifier le contenu du PC , et par ce fait de changer le déroulement du programme.

Dans le cas du MC 6800, il n'est utilisé que pour les instructions de branchement.

Il comprend 2 bytes de l'instruction "branch" est un nombre donnant le déplacement que doit effectuer le PC pour que le MPU puisse continuer son programme.

Après que le PC ait rencontré un "branch", lui le déplacement qu'il doit effectuer il s'incrémente automatiquement. Cette nouvelle position du PC est prise comme référence.

III . 5 Adressage indexé :

Dans l'adressage indexé on ajoute à l'adresse offerte par l'instruction le contenu du registre d'index, afin d'obtenir l'adresse effective.

Cependant, et c'est la différence avec l'adressage relatif, le contenu de l'index peut être incrémenté (ou décrémente) lors de chaque operation, fournissant ainsi des adresses effectives successives. De ce fait, l'adresse fournie par l'instruction peut servir pour des operations multiples ou répétitives. Ainsi, l'adresse effective est obtenue en additionnant :

- 1^e) l'adresse fournie par l'instruction :
- 2^e) mot d'index contenu dans le registre d'index.

Pour préciser qu'il s'agit d'un adressage indexé, le mot d'instruction peut contenir un bit spécial, dit bit d'indexation.

L'adresse effective pourra être, elle, stockée temporairement dans un registre d'adresses.

On distingue deux façons d'indexer l'adresse : la pré-indexation: c'est la méthode de base qu'on vient de décrire, et la post-indexation qui est un adressage indirect.

TABLE 3 - ACCUMULATOR AND MEMORY INSTRUCTIONS

OPERATIONS	MEMONIC	ADDRESSING MODES					SOUL: AN ARITHMETIC OPERATION	COND. CODE REG.
		IMMED	DIRECT	INDEX	EXTEND	IMPLIED		
		DP	DP	DP	DP	DP		
Add	ADDA	88 2 2	9E 3 2	A0 5 2	B0 1 3		A ← M + A	
Add Register	ADDB	8F 2 2	9D 3 2	9F 5 2	FB 4 3		B ← M + B	
Add with Carry	ADCA	89 2 2	99 3 2	A9 5 2	B9 4 1	18 2 1	A ← M + C + A	
And	ANDA	80 2 2	D9 3 2	E9 5 2	F9 4 1		A ← M & A	
And Register	ANDB	81 2 2	D1 3 2	E1 5 2	F1 4 1		B ← M & B	
Bit Test	BTTA	84 2 2	94 3 2	F4 5 2	F4 4 3		S ← M	R
Clear	CLRA	8E 2 2	9E 3 2	AE 5 2	BE 4 3		A ← 0	R
	CLR B	85 2 2	95 3 2	BE 5 2	FE 4 3		B ← 0	R
	CLRA			CF 7 2	FF 6 3		CF ← 0	R
	CLRB					1F 2 1	CF ← 1	R
	CLRD					5F 2 1	CF ← 1	R
Compare	CPXA	81 2 2	91 3 2	A1 5 2	B1 4 3		A ← M - A	R
Compare Register	CPXB	82 2 2	92 3 2	E1 5 2	F1 4 3		B ← M - B	R
Compare Address	CPA	83 2 2	93 3 2	F3 5 2	F3 4 3	11 2 1	A ← M	R
Compare Register	CPB			03 7 2	13 6 3		B ← M	R
Compare Address	CPM					4D 2 1	A ← A	R
Compare Register	CPM					5D 2 1	B ← B	R
Complement	COMA						A ← ~A	R
Complement Register	COMB						B ← ~B	R
Complement Address	COM			6D 7 2	7D 6 3		CF ← ~M	R
Complement Register	COM					9D 2 1	CF ← ~A	R
Complement Address	COM					0D 2 1	CF ← ~B	R
Complement Register	COM					1D 2 1	Convert Binary Field of OP characters into BCD format	R
Decrement	DECA			6E 7 2	7A 6 3		M ← M - 1	R
	DECB					1A 2 1	A ← A - 1	R
	DECD					5A 2 1	B ← B - 1	R
Exclusive OR	EXRA	88 2 2	98 3 2	A8 5 2	B8 4 3		A ← M ⊕ A	R
Exclusive OR Register	EXRB	89 2 2	99 3 2	E8 5 2	F8 4 3		B ← M ⊕ B	R
Increment	INCA			3C 7 2	4C 6 3		M ← M + 1	R
	INCB					4C 2 1	A ← A + 1	R
	INCD					5C 2 1	B ← B + 1	R
Load Register	LDBA	85 2 2	95 3 2	A5 5 2	B5 4 3		M ← A	R
Load Register	LDBB	86 2 2	96 3 2	E5 5 2	F5 4 3		M ← B	R
Load Register	LDBA	8A 2 2	9A 3 2	A8 5 2	B8 4 3		A ← M	R
Load Register	LDBB	8B 2 2	9B 3 2	E8 5 2	F8 4 3		B ← M	R
Push Data	PSHA			FA 5 2	FA 4 3		A ← M	R
Push Data	PSHB					3E 4 1	A ← M, SP ← SP - 1	R
Push Data	PSHA					4E 4 1	B ← M, SP ← SP - 1	R
Push Data	PSHB					5E 4 1	SP ← SP - 1, M ← A	R
Rotate Left	ROL			6F 7 2	7F 6 3		SP ← SP - 1, M ← B	R
	ROLA					0D 2 1	A ← (A << 1) C	R
	ROLB					5D 2 1	B ← (B << 1) C	R
Rotate Right	ROR			8F 7 2	9F 6 3		M ← (M >> 1) C	R
	RORA			8D 7 2	9D 6 3		A ← (A >> 1) C	R
	RORB					9D 2 1	B ← (B >> 1) C	R
Shift Left, Arithmetic	ASL			8E 7 2	9E 6 3		M ← (M << 1) C	R
	ASLA					8E 2 1	A ← (A << 1) C	R
	ASLB					9E 2 1	B ← (B << 1) C	R
Shift Right, Arithmetic	ASR			9F 7 2	AF 6 3		M ← (M >> 1) C	R
	ASRA					9F 2 1	A ← (A >> 1) C	R
	ASRB					AF 2 1	B ← (B >> 1) C	R
Shift Right, Logical	LSR			84 7 2	94 6 3		M ← (M >> 1) 0	R
	LSRA					84 2 1	A ← (A >> 1) 0	R
	LSRB					94 2 1	B ← (B >> 1) 0	R
Store Address	STAA		97 4 2	A7 6 2	B7 5 3		A ← M	R
	STAB		07 4 2	E7 6 2	F7 5 3		B ← M	R
Subtract	SUBA	80 2 2	90 3 2	A0 5 2	B0 4 3		A ← M - A	R
	SUBB	81 2 2	D0 3 2	E0 5 2	F0 4 3		B ← M - B	R
Subtract Register	SBA					18 2 1	A ← A - A	R
Subtract with Carry	SCBA	87 2 2	97 3 2	A2 5 2	B2 4 3		A ← M - C - A	R
	SCBB	88 2 2	98 3 2	E2 5 2	F2 4 3		B ← M - C - B	R
Transfer Register	TAB					16 2 1	A ← B	R
	TBA					17 2 1	B ← A	R
Test, Zero or Sign	TST			8D 7 2	9D 6 3		M ← 0	R
	TSTA					4D 2 1	A ← 0	R
	TSTB					5D 2 1	B ← 0	R

LEGEND:

- OP - Operation Code (Hexadecimal)
- DP - Number of CPU Cycles
- DP - Number of Program Bytes
- A - Arithmetic Plus
- - Arithmetic Minus
- B - Binary AND
- SP - Contents of the stack pointer to be Stack Pointer

- ⊕ - Boolean Inclusive OR
- ⊗ - Boolean Exclusive OR
- ~ - Complement of M
- ← - Transfer into
- ← - Bit-Zero
- 00 - Bit-Zero

CONDITION CODE SYMBOLS:

- H - Half carry from bit 5
- I - Interrupt mask
- N - Negative (sign bit)
- Z - Zero (byte)
- V - Overflow 2's complement
- C - Carry from bit 7
- R - Reset Allowed
- S - Set Allowed
- - Test and set if true, cleared otherwise
- - Test Affected

Note: Accumulator addressing mode instructions are included in the column for IMPLIED addressing.



TABLE 5 - JUMP AND BRANCH INSTRUCTIONS

OPERATIONS	MNEMONIC	RELATIVE		INDEX		EXTND		IMPLIED		BRANCH TEST	COND. CODE REG.						
		OP	~	#	OP	~	#	OP	~		#	5	4	3	2	1	0
		H	I	N	Z	V	C	H	I		N	Z	V	C			
Branch Always	BRA	20	4	2						None	•	•	•	•	•	•	
Branch If Carry Clear	BCC	24	4	2						C = 0	•	•	•	•	•	•	
Branch If Carry Set	BCS	25	4	2						C = 1	•	•	•	•	•	•	
Branch If = Zero	BEQ	27	4	2						Z = 1	•	•	•	•	•	•	
Branch If ≠ Zero	BNE	28	4	2						Z = 0	•	•	•	•	•	•	
Branch If > Zero	BGT	2E	4	2						N ⊕ V = 0	•	•	•	•	•	•	
Branch If Higher	BHI	22	4	2						Z + (N ⊕ V) = 0	•	•	•	•	•	•	
Branch If = Zero	CLE	21	4	2						C + Z = 0	•	•	•	•	•	•	
Branch If Lower Or Same	BLS	23	4	2						Z + (N ⊕ V) = 1	•	•	•	•	•	•	
Branch If < Zero	BLT	20	4	2						C + Z = 1	•	•	•	•	•	•	
Branch If Minus	BAN	2E	4	2						N ⊕ V = 1	•	•	•	•	•	•	
Branch If Not Equal Zero	BNE	26	1	2						N = 1	•	•	•	•	•	•	
Branch If Overflow Clear	BVC	20	4	2						Z = 0	•	•	•	•	•	•	
Branch If Overflow Set	BVS	29	4	2						V = 0	•	•	•	•	•	•	
Branch If Plus	BPL	2A	4	2						V = 1	•	•	•	•	•	•	
Branch To Subroutine	BSR	8D	8	2						N = 0	•	•	•	•	•	•	
Jump	JMP				6E	4	2	7E	3	3	See Special Operations	•	•	•	•	•	
Jump To Subroutine	JSR				AD	8	2	BD	9	3		•	•	•	•	•	
No Operation	NOZ										Advanced Prog. Only	•	•	•	•	•	
Return From Interrupt	RTI								07	2		1	•	•	•	•	•
Return From Subroutine	RTS									3B	10	1	•	•	•	•	
Software Interrupt	SWI									29	5	1	•	•	•	•	
Wait for Interrupt*	WAI									7F	17	1	•	•	•	•	
										3E	9	1	•	•	•	•	

*WAI puts Address Bus, R/W, and Data Bus in the three state mode while VMA is held low.

TABLE 4 - INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

POINTER OPERATIONS	MNEMONIC	IMMED		DIRECT		INDEX		EXTND		IMPLIED		BOOLEAN/ARITHMETIC OPERATION	COND. CODE REG.							
		OP	~	#	OP	~	#	OP	~	#	OP		~	#	5	4	3	2	1	0
		H	I	N	Z	V	C	H	I	N	Z		V	C						
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2		X _H - M, X _L - (M + 1)	•	•	•	•	•	•	•	
Decrement Index Reg	DEX										09	4	1	•	•	•	•	•	•	
Decrement Stack Ptr	DES										04	4	1	•	•	•	•	•	•	
Increment Index Reg	INX										08	4	1	•	•	•	•	•	•	
Increment Stack Ptr	INS										31	4	1	•	•	•	•	•	•	
Load Index Reg	LDX	CC	3	3	DE	4	2	EE	6	2	FE	5	3	•	•	•	•	•	•	
Load Stack Ptr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3	•	•	•	•	•	•	
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3	•	•	•	•	•	•	
Store Stack Ptr	STS				SF	5	2	AF	7	2	BF	6	3	•	•	•	•	•	•	
Index Reg ← Stack Ptr	TXS													35	4	1	•	•	•	
Stack Ptr ← Index Reg	TSX													30	4	1	•	•	•	

OPERATIONS	MNEMONIC	IMPLIED		BOOLEAN OPERATION	COND. CODE REG.											
		OP	~		#	5	4	3	2	1	0					
Clear Carry	CLC	9C	2	1	0 → C	•	•	•	•	•	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	•	•	•	•	•	•	•	•	•	R
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	•	•	•	•	•	•	R
Set Carry	SEC	6D	2	1	1 → C	•	•	•	•	•	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	•	•	•	•	•	•	•	•	•	S
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	•	•	•	•	•	•	S
Accmtr A → CCR	TAP	06	2	1	A → CCR	•	•	•	•	•	•	•	•	•	•	S
CCR → Accmtr A	TAP	07	2	1	CCR → A	•	•	•	•	•	•	•	•	•	•	S

CONDITION CODE REGISTER NOTES. (Bit set if test is true and cleared otherwise)

- 1 (Bit V) Test: Result = 10000000?
- 2 (Bit C) Test: Result = 00000000?
- 3 (Bit C) Test: (Decimal value of most significant BCD Character greater than nine? (Not cleared if previously set.)
- 4 (Bit V) Test: Operand = 10000000 prior to execution.
- 5 (Bit V) Test: Operand = 01111111 prior to execution.
- 6 (Bit V) Test: Set equal to result of NFOC after shift has occurred.
- 7 (Bit N) Test: Sign bit of most significant (MS) byte = 1?
- 8 (Bit V) Test: 2's complement overflow from subtraction of MS bytes?
- 9 (Bit N) Test: Result less than zero? (Bit 15 = 1)
- 10 (Bit) Load Condition Code Register from Stack. (See Special Operations)
- 11 (Bit) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
- 12 (Bit) Set according to the contents of Accumulator A.



Chapitre IV

IV . Etude du systeme de Visualisation .

IV.1 Rappel sur la TV (Etude du standard 625 lignes) .

IV.2 Etude de l'interface de visualisation le CRTC (MC 6845).

IV.3. Etude du bloc diagramme ,
et Mode de fonctionnement.

IV.1 Rappel sur la TV (Etude du standard 625 lignes):

Le bloc terminal de tout notre systeme etant un recepteur de television (625 lignes), il est donc necessaire de faire un rappel sur les principes d'analyse et de transmission de l'image .

IV 1.1 Schema de principe d'un recepteur TV :

La fig montre un schema synoptique de TV noir et blanc

L'attaque du poste par le signale video peut se faire de 2 façons, soit:

- Directement sur l'entrée video,
- Soit par l'intermediaire d'un modulateur,
et l'entrée se fera sur la prise d'antenne.

IV 1. 2 Principe de l'analyse TV:

L'image résulte de 2 operations simultanées :

l'analyse de la surface de l'écran par le faisceau cathodique suivant des lignes horizontales successives, de haut en bas, chaque ligne pouvant être elle même considérée comme analysée point par point, successivement de la gauche vers la droite .

Le faisceau est soumis a 2 champs orthogonaux variables suivant une loi dite en dents de scie et la trace du faisceau sur l'écran constitue une trame.

IV . 1 . 3 Balayage entrelacé .

Le balayage decrit precedement donne lieu, à la reception, à un effet de scintillement désagréable.

Pour l'éviter on fait appel au balayage " entrelacé " d'ordre 2 . Dans ce balayage on analyse toutes les lignes impaires puis les lignes paires.

De cette façon on analyse donc en $1/50^{\text{eme}}$ de seconde la trame impaire puis pendant la même durée la trame paire. On obtient alors une image en $1/25^{\text{s}}$.

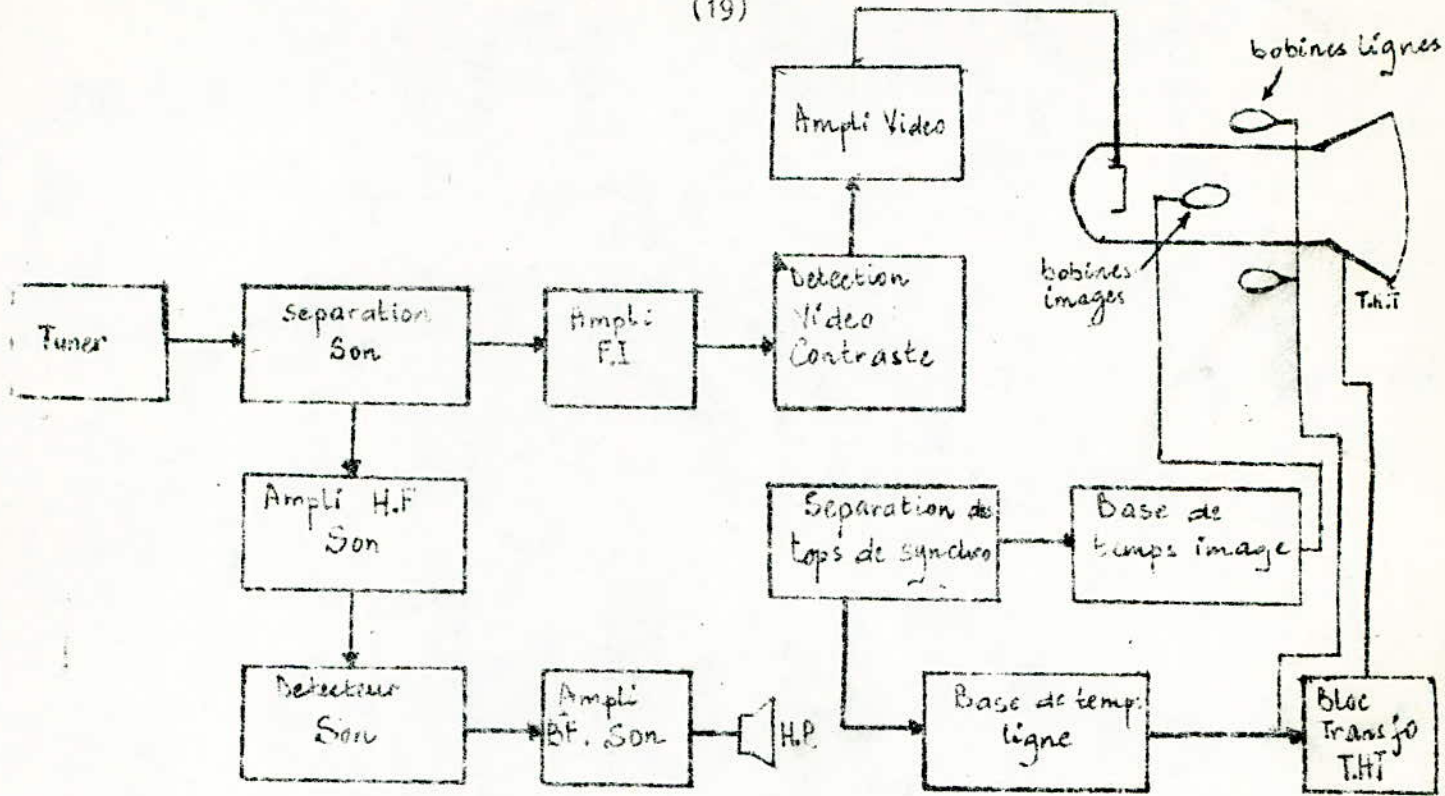
(Fig IV.1 b)

IV 1 . 4 Bande de frequence necessaire .

L'image etant rectangulaire de format $\frac{L}{H} = \frac{4}{3}$, le nombre de points sur une ligne, en negligant la durée du retour $\frac{H}{L}$, sera :

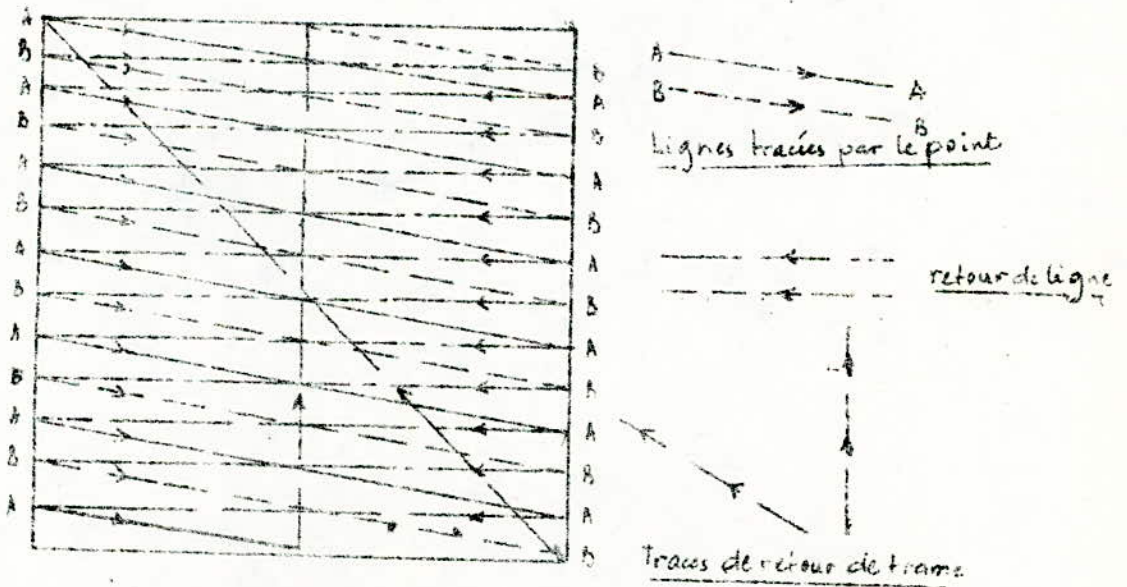
$$625 \times \frac{4}{3} = 830 \text{ points}$$

(19)



a/

Fig. 1. Schéma synoptique d'un récepteur TV (Noir et blanc)



2/ Principe de l'analyse entrelacée

(en supposant que les definitions horizontales et verticale identiques:
Cas general)

Pour toute l'image cela represente :

$$830 \times 625 = 508750 \text{ points}$$

or la frequence image etant de 25 Hz on a donc à transmettre une gamme de frequence de :

$$\frac{508750 \times 25}{2} = 6,5 \text{ MHz}$$

IV 1 . 5 Reception:

La restitution de l'image à la reception utilise un CRT à deviation magnetique.

Pour cela il faut :

- 1) Moduler le faisceau electronique de façon à ce que les variations reproduisent l'image.
- 2) Déplacer le faisceau de façon rigoureusement identique au déplacement du faisceau d'analyse, à la fois dans le temps et dans l'espace .

La 1^{ere} condition est realisée à l'aide des procedés habituels (signal-image modulant une onde porteuse qui ,recueillie et détectée, donne un signal modulant le faisceau electronique du tube).

La 2^{eme} par contre amène certaine complexité ; on utilise à la reception 2 generateurs de dents de scie fournissant des signaux semblables à ceux de l'émission.

Mais comme il est necessaire,

- 1) Que les frequences soient egales ,
- 2) Que le debut des dents de scies

ait lieu au même instant à l'émission et à la reception. Il faut prévoir un dispositif particulier permettant de synchroniser ces 2 generateurs: Ce sera la base de temps.

IV 1 . 6 Signal video . Signaux de Synchronisation :

Le signal video, est un signal électronique dont la tension est fonction de la luminance de chacun des points de l'image analysée .

Quand aux signaux de synchro, on incorpore au signal image pendant le retour de la dent de scie, un signal supplementaire (top de synchro) qui donne l'ordre au recepteur de changer de ligne (top de fin de ligne : 4,7 μ s) et un (top de fin d'image qui permet de changer d'image.

Les tops de fin de lignes sont plus courts que les tops de fin d'image. (fig 2

fig 3) IV. 2 et IV 3)

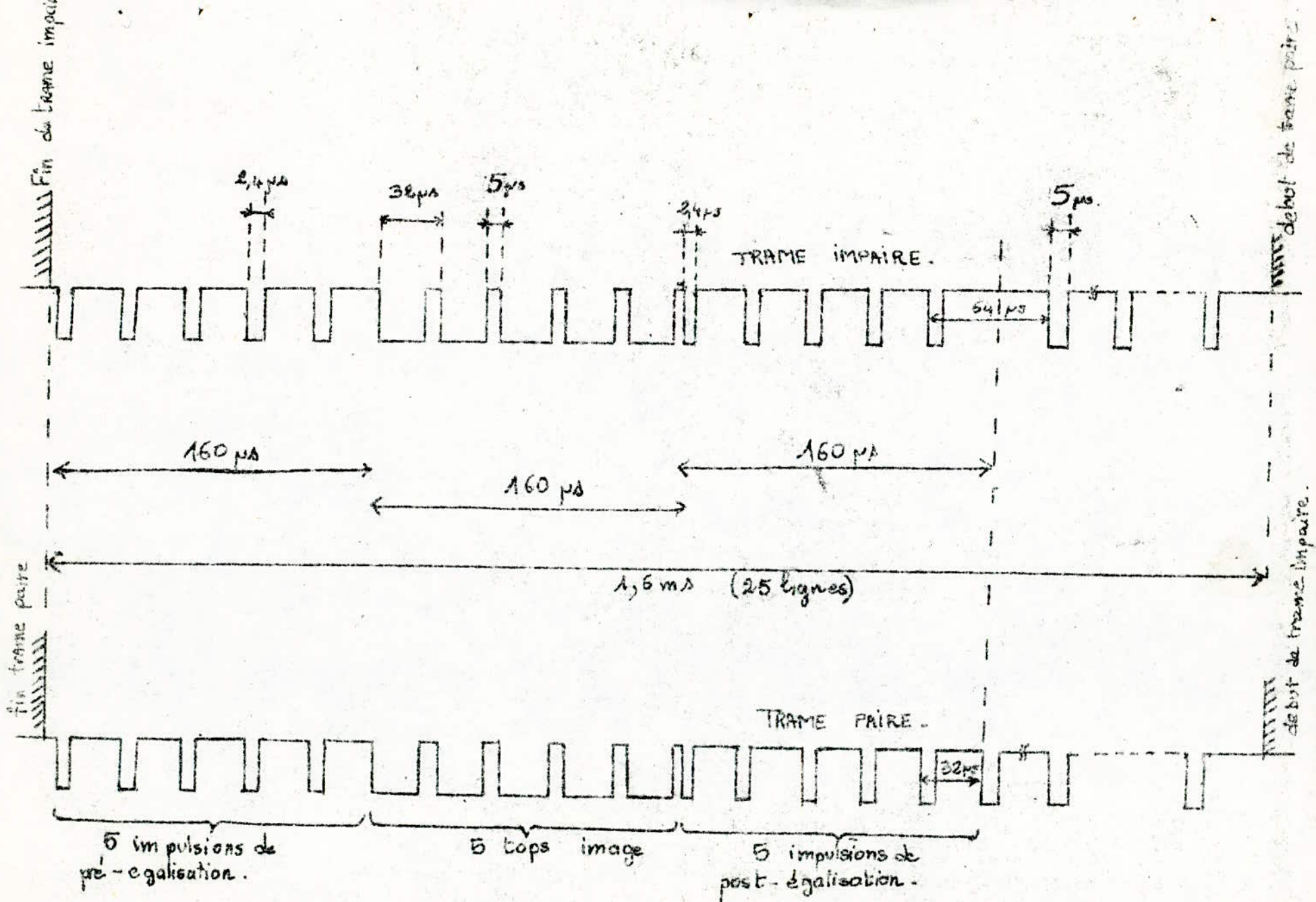


Fig 2 Signaux de synchronisation image (standard 625 lignes)
CCIR

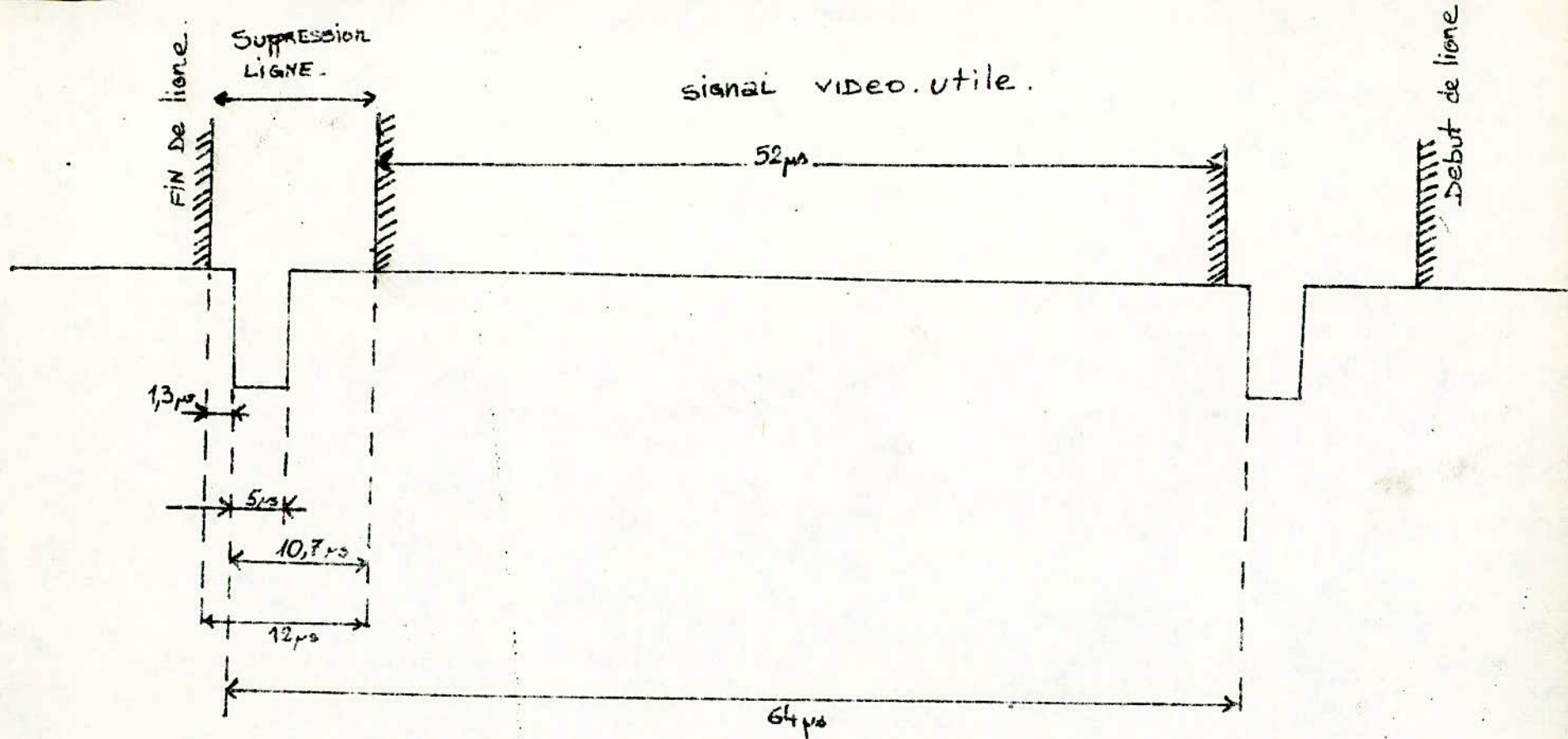


Fig IV 3 Signaux de synchronisation ligne (Standard 625 lignes)
CCIR

IV. 1. 7 Affichage sur écran TV.

On dispose de signaux, appelés signaux d'image, qui modulent l'intensité du faisceau en fonction de la brillance du point concerné.

On va donc substituer à ces signaux image les impulsions issues de la voie de luminance.

On analyse alors la 1^{ère} ligne du 1^{er} caractère puis la 1^{er} ligne du 2^{ème} caractère et ce jusqu'au dernier de la rangée, on passe ensuite à la 2^{ème} ligne du 1^{er} caractère et ainsi de suite jusqu'à la dernière ligne des caractères de la 1^{er} rangée.

En synchronisme avec le balayage on engendre les signaux correspondants aux points allumés sur la ligne considérée.

Il est à remarquer que du point de vue de la qualité, la brillance est constante.

En balayage entrelacé chaque point de la matrice (9 X 7) définissant un caractère sera en réalité constitué sur l'écran par deux points :

- Un point correspondant à la ligne paire
- Un point correspondant à la ligne impaire.

Par conséquent les 9 lignes de la matrice de caractère seront représentées sur l'écran par 9 lignes impaires et 9 lignes paires.

IV. 1. 8 Rafraichissement de l'image.

La persistance maximale de l'image sur l'écran ne dépassait pas quelques secondes. Cela nécessite donc de rafraichir l'image à une fréquence suffisante (toute les 20 ms) pour la conserver avec une bonne qualité, sans scintillement en particulier.

IV. 1. 9 Nombre de caractères par rangée :

La fréquence maximale du signal électronique pour le 625 lignes est de 6,5 MHz.

En appliquant des impulsions de luminances sur le ~~videotexte~~ à des moments précis durant le balayage, des caractères peuvent être générés.

Par conséquent la fréquence de ces points ne doit pas dépasser la fréquence maximale vidéo.

Soit N le nombre de points illuminés par ligne de balayage et si 2 caractères sont séparés par 2 points, on aura la condition :

$$N \cdot 625 \cdot 25 \leq 6,5 \text{ MHz}$$

ou encore

$$N \cdot 625 \cdot 25 \leq \frac{830 \cdot 625 \cdot 25}{2}$$

d'où

$$N \leq 415$$

Chaque caractère occupe 7 colonnes, et entre 2 caractères on laisse 2 blancs
on aura donc :

$$N = (7 + 2) M$$

cù M est le nombre de caractères affichés par rangée

d'où

$$9 M \leq 415 \text{ soit } M \leq \frac{415}{9}$$

$$\underline{M < 46}$$

IV . 1 . 10 Nombre de rangées de caractères par écran :

Si nous laissons 5 lignes de balayage non écrites entre 2 rangées de caractères, et si (X) est le nombre total de rangées de caractères nous aurons la relation suivante :

$$(9 + 5) \cdot X < \text{nombre total de lignes de balayage par trame}$$

$$14 X < Y$$

En laissant libre les 6 lignes du haut et du bas de la trame, on obtient :

$$Y = 312,5 - 6 - 6 = 300,5 \text{ soit } Y = 300 \text{ lignes}$$

$$14 X < Y, \quad 14 X < 300$$

d'où

$$X < \frac{300}{14} \quad X < \underline{\underline{21}}$$

Pour notre cas on choisira une image de 20 lignes de 40 caractères chacune
soit au total 800 caractères par écran T.V .

IV . 2 Interfaces de visualisation :

Lorsqu'on veut visualiser 16 lignes de 64 caractères ou 24 lignes de 80 caractères, l'utilisation des afficheurs n'est rentable du point de vue économique .

Dans ce cas, la meilleure solution est d'utiliser un tube cathodique. Il existe sur le marché un certain nombre de circuits LSI intégrant une grande partie de cette logique . Nous avons choisi, dans notre application le CRTC MC 6845 .

Le contrôleur de tube cathodique MC 6845 assure les fonctions complexes d'interface entre le MPU et un terminal à écran cathodique . Parmi les applications, on peut citer la visualisation de textes au format téléimprimeur, les terminaux " intelligents " , le traitement des mots et la visualisation directe des informations .

IV . 2 . 1 Differentes lignes du MC 6845 :

IV.2.1.1/ Interface avec le processeur. (fig 3 | 4)

Le CRTC interface avec le processeur sur le bus bidirectionnel de données (D₀ - D₇) utilisant CS (Chip select) , RS (Register select) , E (Enable) et R/W̄ (read/ Write) pour signaux de contrôle.

a/ Bus de données (D₀ - D₇) ;

Les lignes de données permettent le transfert de données entre la file de registres internes du CRTC et le processeur. Les sorties du bus de données possèdent des buffers à trois états, qui ne prennent l'état haute impédance seulement quand le processeur exécute une opération lecture du CRTC .

Un niveau haut sur ces lignes correspond à un " 1 " logique .

b/ Enable (E) ;

Se signal enable est une entrée compatible (TTL/ Mos) qui rend libre l'entrée et la sortie des buffers au bus de données, et compte les données entrantes et sortantes du CRTC .

Ce signal dérive de l'horloge du processeur, sa zone active est celle entre le niveau haut et le niveau bas .

c/ CS (Chip Select) : Sélection du boîtier :

Cette ligne est une entrée compatible (TTL/ Mos) .

Elle permet de sélectionner le CRTC quand on veut lire ou écrire dans la **file** de ces registres internes.

Ce signal deviendrait actif seulement quand il y a une adresse stable fournie au préalable par le processeur.

d/ RS (Selection du registre .) :

C'est une entrée compatible (TTL/ Mos) qui sélectionne un registre d'adresse (RS = 0) ou un des registres de données (RS = 1) de la **file** interne du CRTC .

c/ R/ \bar{W} (Lecture/écriture)

Cette entrée permet de définir si la **file** des registres internes est lue ou écrite .

L'écriture correspond à " 0 " .

IV . 2 . 1 . 2 / Lignes de Contrôle :

Le CRTC produit la synchronisation horizontale (HS) , la synchronisation verticale (VS) et les signaux de " Display Enable " .

a/ Synchronisation Verticale (V sync) :

Cette sortie compatible (TTL) dirige directement le moniteur, et détermine la position verticale du texte visualisé.

b/ Synchronisation horizontale (H sync) :

Cette sortie compatible (TTL) est un signal actif à un niveau haut, qui détermine la position horizontale du texte visualisé .

c/ Display Enable (Validation de la visualisation) :

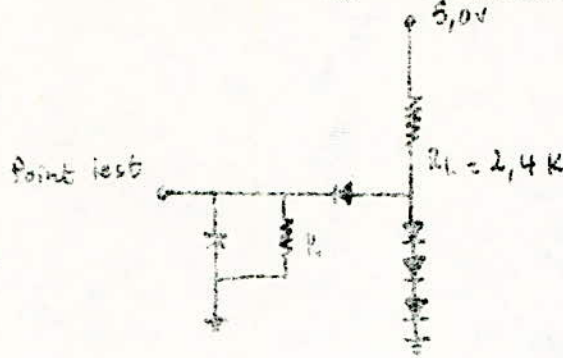
Cette sortie compatible (TTL) est un signal actif à un niveau haut qui indique que le CRTC est entrain de produire un adressage de l'aire active de visualisation .

d/ Lignes d'adresses de la mémoire de rafraichissement (MA0 - MA 13)

Ces 14 sorties sont utilisées pour produire les adresses de la liste de visualisation pour rafraichire l'écran du tube cathodique, cette liste se trouve dans une RAM de 16 K .

Un niveau haut sur ces sorties est un "1" logique.

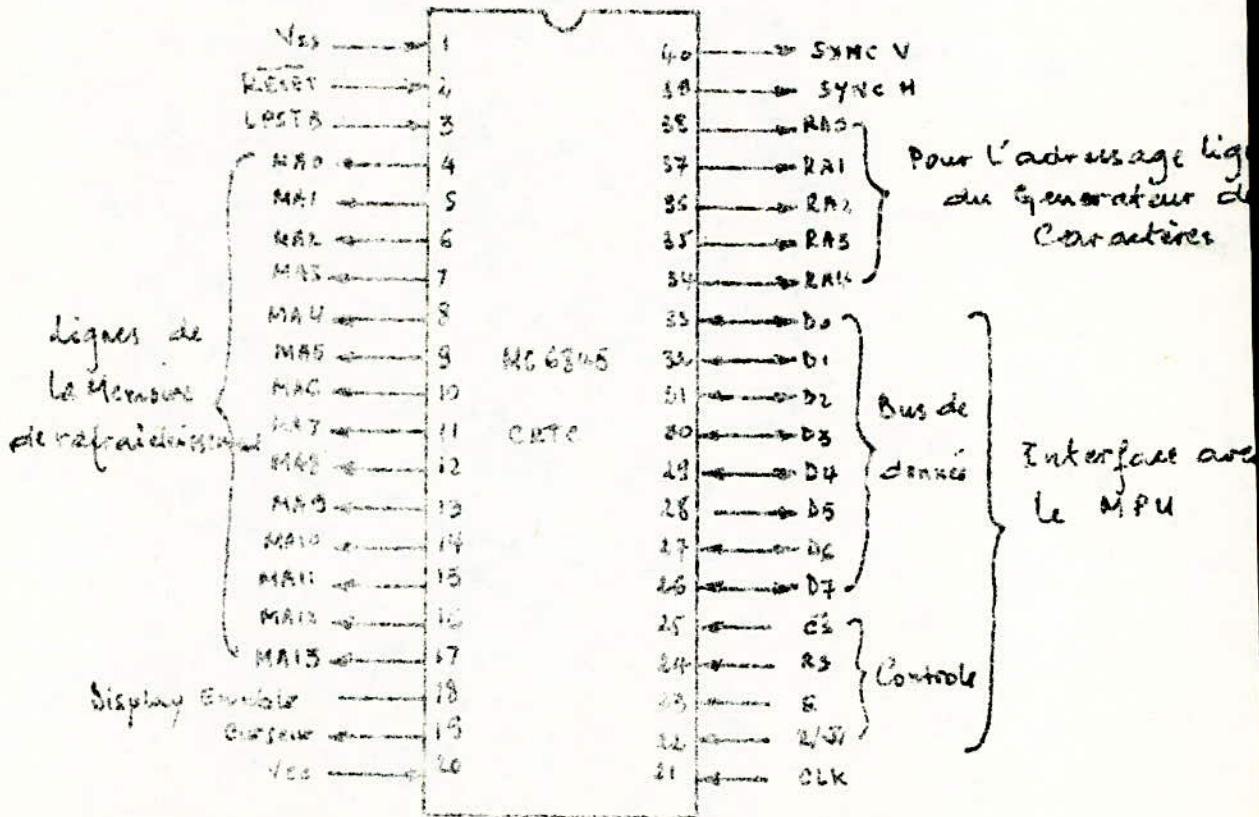
Charge de test du sequencement des bus



$C = 130 \text{ pF}$ pour D0-D7
 $= 30 \text{ pF}$ pour MA0-MA13
 RA0-RA4, bispea, Synch
 Synch V, et Curseur

$R = 11K\Omega$ pour D0-D7
 $= 24K\Omega$ pour toutes les autres sorties.

Figure 4 Affectation des broches



b/ Sorties pour la selection ligne (RA0 - RA4) :

Ces cinq sorties du compteur interne de lignes permettent d'adresser la ROM de caractères par une selection ligne. Un niveau haut sur (RA0 - RA4) est un "1" logique .

IV . 2 . 1 . 3 Autres lignes du CRTC :

a/ Curseur : utilisé pour visualiser un curseur qui determine la position du prochain caractère a visualiser.

b/ L'horloge (CLK) :

L'entrée du CLK compatible (TTL/ Mos) est utilisée pour la synchronisation de tous les signaux de controle du tube cathodique .

La zone active est la transition du haut jusqu'à bas .

c/ Le strobe du photostyle . (LPSTR) :

Cette entrée compatible (TTL/ Mos) en niveau haute impedance sert a "verrouiller" l'adresse de rafraichissement courante dans la file de registre.

d/ $\overline{\text{RES}}$ Reset:

L'entrée $\overline{\text{RES}}$ est utilisée pour arrêter le fonctionnement du CRTC . Un niveau bas sur $\overline{\text{RES}}$ force le CRTC à passer aux états suivants .

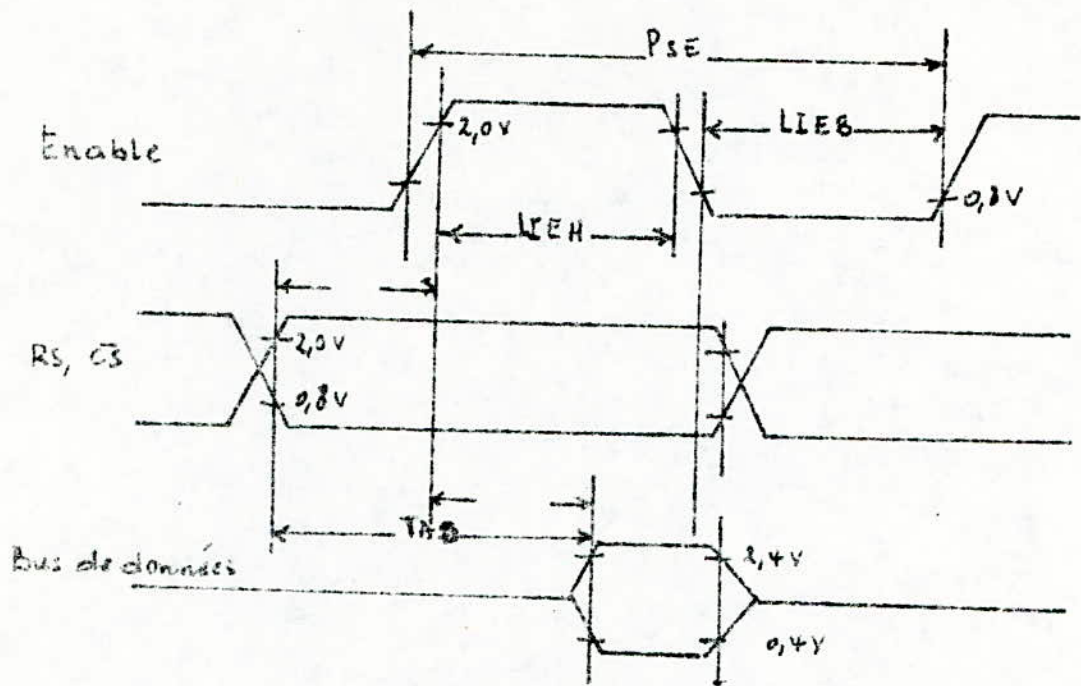
- 1) Tous les compteurs dans le CRTC sont vidés et le dispositif stoppe l'opération de visualisation .
- 2) Toutes les sorties passent à leur niveau bas .
- 3) Les registres de contrôle dans le CRTC ne sont pas affectés et demeurent inchangés.

Ce signal diffère des autres (Reset) de la famille M 6800 dans les fonctions suivantes:

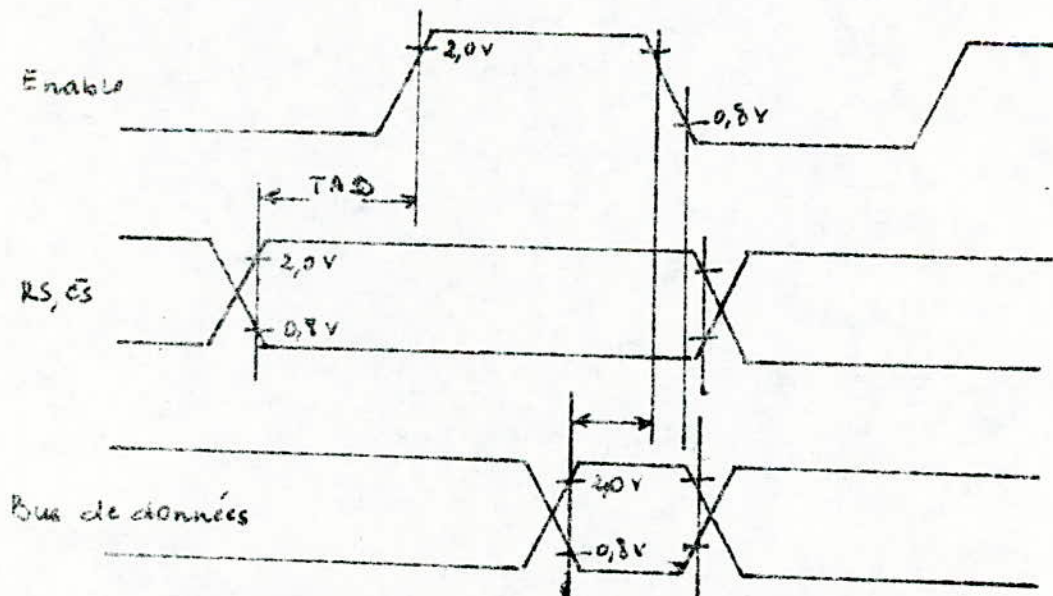
- 1) Le signal $\overline{\text{RES}}$ ne peut remplir la fonction $\overline{\text{RES}}$ que si le strobe du photostyle est à un niveau bas.
- 2) Dès que $\overline{\text{RES}}$ passe au niveau bas les sorties (MA0 - MA 13) et (RA0 - RA4) synchronisé par le niveau bas de l'horloge, passent toutes au niveau bas .
(Il faut au moins un cycle memoire pour Reset) .

Fig 16 (32) Diagramme des Sequencements des Bus

a/ Lecture d'une information dans Le CRTC



b/ Ecriture d'une information dans Le CRTC



- PSE periode du signal Enable

- LIEH : largeur de l'impulsion Enable niveau haut

- LIEB : largeur de l'impulsion Enable " bas

c) Le CRTC commence l'opération de visualisation immédiatement après la relaxation du signal $\overline{\text{RES}}$.

$\overline{\text{RES}}$	LPSTB	Mode opératoire du CRTC
0	0	Reset
0	1	Mode test
1	0	Mode normal
1	1	Mode normal

IV . 2 . 2 Organisation Interne du CRTC

19 registres dans le CRTC peuvent être accessibles par l'intermédiaire du bus de donnée.

a/ Registre d'adresse.

C'est un registre de 5 bits à écriture seulement, utilisé comme registre pointeur. Son contenu est l'adresse de l'un des 18 autres registres de la file.

* Quand $\overline{\text{RS}}$ et $\overline{\text{CS}}$ sont à un niveau bas ce registre n'adresse plus. Quand $\overline{\text{RS}}$ devient actif (niveau haut) la file de registre est accessible.

b/ Les registres de timing horizontal (R0, R1, R2, R3).

Le point de référence de ces registres est fixé comme la position du caractère la plus à gauche.

Ces registres sont programmables en unité de " temps de caractère", tout en respectant la référence.

1) Registre Horizontale total (R0)

Ce registre de 8 bits d'écriture seulement détermine la fréquence de la synchro- horizontale.

2) Registre de balayage horizontal (R1)

Ce registre de 8 bits d'écriture seulement détermine le nombre de caractère à visualiser par ligne horizontale.

3/ Registre de la position de la Synchronisation horizontale (R2):

Ce registre de 8 bits à écriture seulement détermine la position de la synchro - horizontale sur la ligne horizontale.

4/ Registre de la largeur de la Synchro - horizontale (R3):

Ce registre de 4 bits d'écriture seulement détermine la largeur de l'impulsion de synchro- horizontale .

c/ Registres de Timing vertical (R4, R5, R6, R7, R8, R9,):

Le point de référence pour ces registres est la position la plus haute du caractère à visualiser .

Ces registres sont programmés en temps d'une rangée de caractères ou en temps de ligne de balayage (.....) .

1) Registre vertical total (R4) et registre d'ajustement (R5)

La fréquence de la synchro - verticale est déterminée par ces 2 registres (R4 et R5) .

Le temps d'une ligne de caractères est généralement un entier plus une fraction .

Le nombre entier moins 1 est programmé par R4 de 7 bits d'écriture seulement la fraction par R5 de 5 bits d'écriture seulement, comme un temps de ligne de balayage .

2) Registre de balayage Vertical (R6):

Ce registre de 7 bit à écriture seulement détermine le nombre de rangées de caractères visualisés sur l'écran, et il est programmé en temps de rangée de caractères .

3) Registre de la position de la synchronisation Verticale (R7)

Ce registre de 7 bits d'écriture seulement détermine la position de la synchronisation verticale tout en respectant la référence. Il est programmé en temps de rangées de caractères .

4) Registre du mode entrelacé. (R8) :

Ce registre de 2 bit controle le mode d'analyse de l'image :

Bit	Bit	Mode
1	0	Mode
0	0	Mode de Synchronisation normale (Non entrelacé)
1	0	Mode de synchro- entrelacé
0	1	Mode de synchro- entrelacé
1	1	Mode entrelacé et vidéo

Pour notre cas c'est un mode entrelacé, imposé par le standard TV 625 lignes .

2) Registre d'adresse de départ (Haut et bas) (R12, R13) :

C'est un registre de 14 bits d'écriture seulement qui determine la première adresse pour le rafraichissement après un blanking vertical.

Il consiste en un registre de 8 bits inférieures et un registre de 6 bits supérieures .

3) Registres du photostyle. (Haut et bas) (R16 ,R17) :

Ce registre de 14 bits d'écriture seulement est utilisé pour emmagasiner le contenu du registre d'adresse (Haut et bas) quand l'entrée du LPSTB est à un niveau haut .

Ce registre consiste en un registre 8 bits inferieurs et un autre de 6 bits superieurs .

4) Registre du curseur . (Haut et bas) (R14, R15) :

Ce registre de 14 bits d'écriture et lecture emmagasine la **position** du curseur .

Il est composé de 2 registres l'un de 8 bits inferieurs et l'autre de 6 bits superieurs .

8/ Registre d'adressage du maximum de lignes de balayage (R9) :

Ce registre de 5 bit d'écriture seulement determine le nombre de lignes par rangée de caractères plus les lignes constituant l'interligne entre 2 rangées .

d) Autres registres :

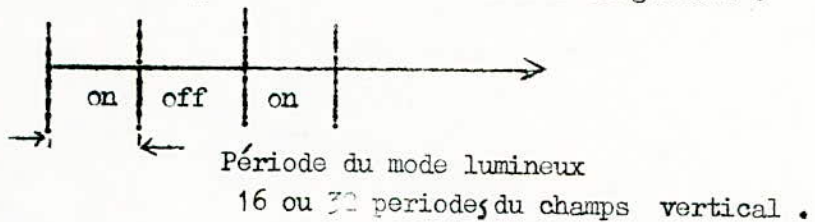
1) Registre de depart du curseur (R 10) :

Ce registre de 7 bits d'écriture seulement controle le format du curseur (voir fig. suivante)

Le bit 5 controle le timing clignotement du curseur .

Bit	Bit	Mode de visualisation du curseur
6	5	
0	0	Non lumineux
0	1	Le curseur n'est pas manifesté
1	0	Lumineux 1/16 vitesse du champ vertical
1	1	Lumineux 1/32 vitesse du champ vertical

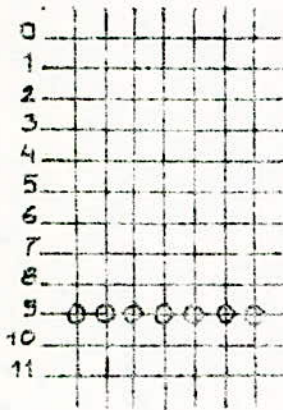
Le bit 6 utilisé pour autoriser un curseur clignotant :



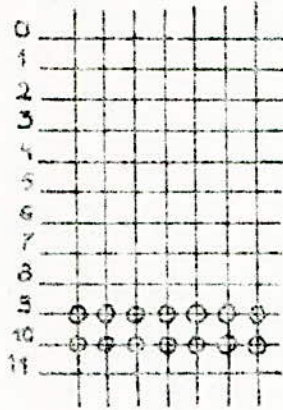
2) Registre de fin du curseur (R11) :

Ce registre de 5 bit d'écriture seulement fixe la ligne de balayage de fin du curseur .

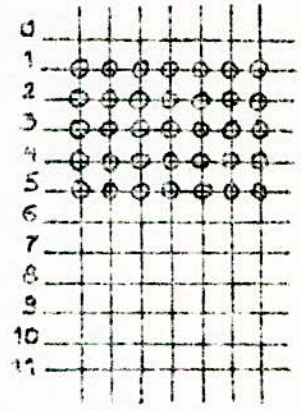
Exemples de modes de Visualisation du Curseur.



adresse de depart curs = 9
 adresse de fin du curs = 9



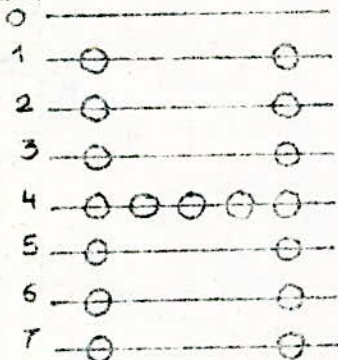
adr. dep. Curs. = 9
 adr. fin Curs. = 10



adr. dep curs. = 1
 adr. fin. curs. = 5

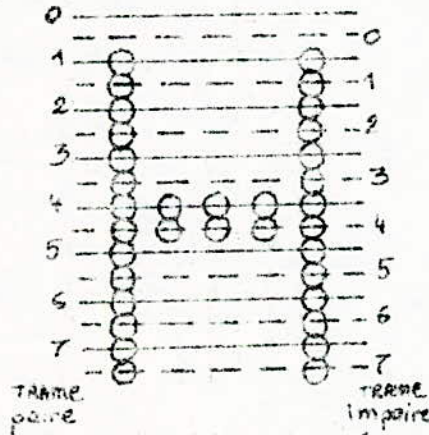
Fig. IV 7 Modes d'INTERFACE.

adresse de
 la ligne sur
 l'écran.



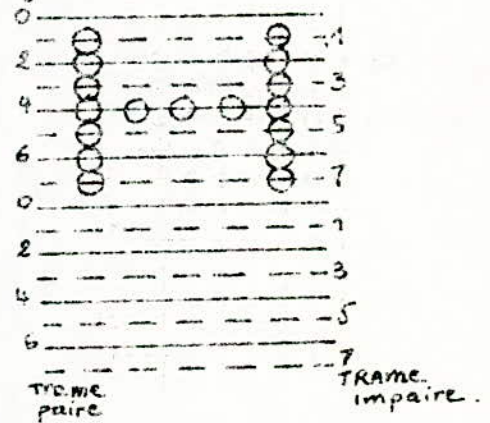
SYNCRD. NORMALE.

adr de
 la ligne
 écran.



SYNCRD. ENTRELACÉ

adr de
 la ligne écran.



SYNCRD. ENTRELACÉ VIDEO.

Atribution des registres internes du CRTC

R S	Registre d'adress					Registre	File de registres	Unité de Programme	Lecture	Ecriture	Nombre de Bits
	4	3	2	1	0	##					
X	X	X	X	X	X	X	-----	----	----	----	-----
0	X	X	X	X	X	X	Registre d'adress	+	Non	Oui	5
1	0	0	0	0	0	R0	Horizontal Total	Caractere	Non	Oui	8
1	0	0	0	0	1	R1	Balayage horizontal	"	"	"	8
1	0	0	0	1	0	R2	Position de la synchro horizontale	"	"	"	8
1	0	0	0	1	1	R3	Largeur de la synchronisation	"	"	"	4
1	0	0	1	0	0	R4	Vertical total	Ligne Carac	"	"	7
1	0	0	1	0	1	R5	Ajustement vertical total	"	"	"	5
1	0	0	1	1	0	R6	Balayage vertical	"	"	"	7
1	0	0	1	1	1	R7	Position de la synchro-vertical	"	"	"	7
1	0	1	0	0	0	R8	Mode entrelacé	"	"	"	2
1	0	1	0	0	1	R9	Maximum lignes adressé	Lignes de Balayage	"	"	5
1	0	1	0	1	0	R10	Debut du curseur	"	"	"	7
1	0	1	0	1	1	R11	Fin du curseur	"	"	"	5
1	0	1	1	0	0	R12	Adresse de départ (Haut)	"	"	"	6
1	0	1	1	0	1	R13	Adress de départ (Bas)	"	"	"	8
1	0	1	1	1	0	R14	Curseur (Haut)	"	Oui	Oui	6
1	0	1	1	1	1	R15	Curseur (Bas)	"	Oui	Oui	8
1	1	0	0	0	0	R16	Photostyle (Haut)	"	Oui	Non	6
1	1	0	0	0	1	R17	Photostyle (Bas)	"	Oui	Non	8

IV. 2. 3 Caractéristiques du CRTC :

- adresse de rafraichissement à 14 bits capable d'adresser 4 pages de 4 K caractères ou 8 pages de 2 K caractères, etc ...

Le générateur d'adresse linéaire est dirigé par l'horloge (CLK), et permet de loger la position relative d'un caractère en memoire avec sa position sur l'écran .

Utilisant le registre d'adresse de depart le déroulement vertical est possible à travers les 16 K caractères .

Le générateur d'adresse lineaire répète la même sequence pour chaque ligne de balayage de la rangée de caractères .

- Visualisation avec deroulement vertical ou en page.
- Trois modes d'interface.
- Capacité alphanumerique et graphique complète.
- Matrice de caractères programmable (nombre de points trame) .
- Nombre de caractères par ligne programmable .
- Nombre de lignes par écran programmable .
- Largeur et position de la synchronisation horizontale programmable.
- Position de la synchronisation verticale programmable.
- Registre du curseur programmable: détermine la position et la dimension du curseur sur l'écran.
- Registre du curseur intégré .
- Registre du photostyle intégré .
- Aucun " buffer " de ligne nécessaire .
- Compatibilité totale avec la logique TTL .
- Alimentation unique + 5 v .
- Compatibilité totale avec le MC 6800 .

Les registres internes du CRTC sont programmables par le MPU à travers le bus de données (D0 - D7) et les signaux de controle = R/\bar{W} , \bar{CS} , RS et E .

IV - 3 . Description du Système de Visualisation .

Comme le montre la figure 14.5, le contrôleur d'écran cathodique est la partie intelligente du terminal graphique. Il assure l'interface entre le MPU et l'écran de visualisation. Comme tel, sa fonction principale est la gestion de la mémoire de rafraichissement par un réseau de 14 lignes d'adresse (MA0 - MA13), la mise en oeuvre de la mémoire de caractères par une sélection lignes (RA0-RA4), la synchronisation du système d'affichage avec le balayage du moniteur (H sync , V sync), ainsi que la gestion de l'écran par la sortie " apte à visualiser (" Display Enable "), et celle du curseur.

Tout le séquençement - dans le CRTC dérive de l'entrée d'horloge (CLK) .

Dans les terminaux alphanumériques ce signal caractérise la période du caractère. Cette période est un multiple de la période de pilotage du système d'affichage (ou - fréquence vidéo) , quand cette dernière est supérieur à 3 MHz .

Le processeur communique avec le CRTC à travers le bus de données, par une lecture ou écriture dans la file interne de registre du CRTC.

Les lignes d'adressage de la mémoire de rafraichissement sont multiplexées entre MPU et le CRTC . Les données apparaissent sur un bus secondaire; entre ce dernier et la ROM de caractères existe un registre tampon ou de verrouillage. Ce registre est validé par le signal de l'horloge haute fréquence .

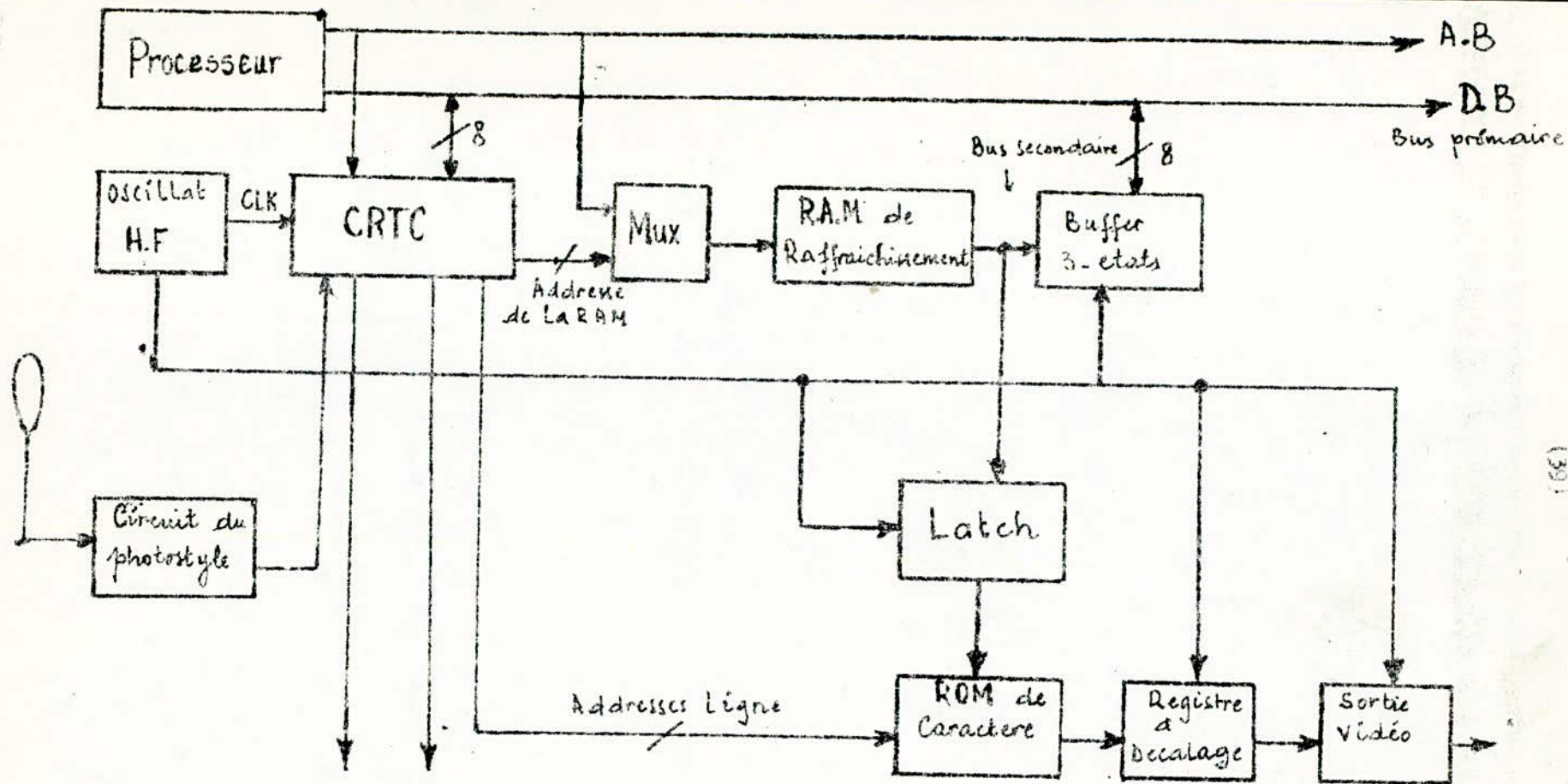
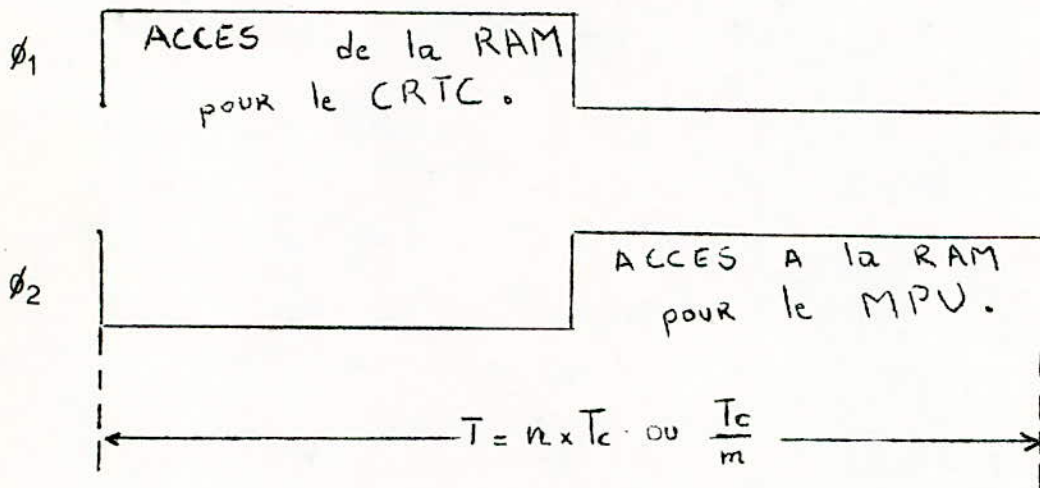


Fig IV 8 Application typique du Contrôleur de TRC

Pour résoudre le contentieux entre le MPU et le CRTC pour l'accès à la mémoire de rafraîchissement, un certain nombre d'approches est possible :

- 1er/ Le processeur a toujours la priorité .
- 2°/ Le processeur a la priorité périodiquement ; il peut être synchronisé par une interruption pour exécuter l'accès uniquement pendant le temps de relaxation horizontale et verticale ; c'est - à - dire dans la zone non active de l'écran .
- 3°/ Synchronisation du processeur par un cycles mémoire " WAIT " .
- 4°/ Synchronisation du processeur avec la période du caractère :



- Quand m , n sont des entiers, T_c est la période du caractère.

La famille M 6800 s'y prête bien à cette configuration du fait de la constance de la longueur de la période d'horloge . Cette méthode ne produit aucune charge sur le processeur , parcequ'il n'y a plus de rivalité pour l'accès à la mémoire.

La conception du bus secondaire est prévue pour l'utilisation de la RAM pour d'autres intentions.

Par exemple utilisant l'approche (4) , une RAM de 64 K byte peut remplir les fonctions de stockage de données et de rafraîchissement . Donc la mémoire d'écran est utilisée comme n'importe quelle mémoire. On dit alors que le contrôleur de CRT est transparent , vu d'un microprocesseur.

II - 3 . 1 Détermination de la fréquence de l'oscillateur :

Pour délivrer à sa sortie des points en synchronisme avec le balayage ligne , le registre à décalage doit être piloté par un oscillateur HF bien déterminé.

Nous nous proposons de calculer la fréquence de cet oscillateur , afin d'obtenir une image stable sur l'écran.

En effet un texte de (N) caractères par ligne demandera un passage de chaque ligne de caractère en $(64/N)$ μ s, temps de chargement parallèle dans le registre à décalage inclus.

En réalité la durée utile, compte tenu du palier de garde avant (1,3 μ s) , de la durée de l'impulsion ligne (5 μ s) et du palier de fin d'impulsion ligne (5,7 μ s), représentés explicitement sur la fig 7 (Chapitre ; Rappel sur la TV) se trouve être que de 52 μ s seulement .

Ses problèmes de haute fréquence liés à l'emploi de circuits MOS commenceront à se poser dès que l'on veut atteindre 64 ou 80 caractères par ligne.

Dans notre cas les rangées ne sont que de 40 caractères et les problèmes de ce genre sont pratiquement éliminés.

Si l'on désigne par :

Nrc : le nombre de rangées de caractères.

Nc/r : le nombre de caractères par rangée.

Le nombre total de caractères par écran sera :

$$N = Nc/r \times Nrc .$$

Dans notre cas le format est de 20 x 40 ,

d'où $N = 20 \times 40 = 800$ caractères par écran .

D'autres part le format du caractère est de 9 x7 points.

Donc le nombre de points analysés par ligne de caractère, en tenant compte des 2 colonnes d'espacement entre 2 caractères , sera :

$$(7 + 2) = 9 \text{ points .}$$

D'où le nombre de points analysés par ligne de balayage:

$$N_{pa/Lb} = 40 \times 9 = 360$$

Ce qui nous permet de déterminer la durée par point analysé :

$$d_{pa} = \frac{52}{360} = \underline{0,144 \text{ us}} = \underline{144 \text{ ns}}$$

Ceci nous imposera une fréquence minimale d'horloge de:

$$f_{\text{Min}} = \frac{1}{d_{pa}} = \frac{1}{144 \cdot 10^{-9}} = \underline{6,9 \text{ MHz}} .$$

On prendra une fréquence de 7 MHz d'horloge pour avoir une image stable .

IV.3.2 Fonctionnement du système de visualisations.

Nous avons vu dans le chapitre consacré au rappel sur la télévision que l'image affichée est très fugitive. Il est donc nécessaire de stabiliser cette image fugace. Pour ce faire on doit la recycler (toutes les 20 ms). Cela signifie que la liste d'affichage doit être décrite autant de fois que l'on désire réafficher son contenu.

IV.3.3 Mémoire de rafraichissement.

Comme l'indique la (figure du schéma global) la liste de visualisation est stockée dans une mémoire RAM spécialisée.

La transmission de caractères du clavier vers la visu se fait à travers le microprocesseur.

Les caractères ASCII provenant du microprocesseur sont stockés dans cette RAM, laquelle sera activée et contrôlée par le CRTC. Il est donc possible de conserver en mémoire, la liste de visualisation, qui sera appelée séquentiellement en avant ou en arrière par l'intermédiaire d'une petite logique associée et moyennant les registres d'adresse de départ, et d'adresse fin de liste .. sortie

Cette RAM délivrera les caractères ASCII qui serviront d'adresses pour le générateur de caractères, sous forme de 7 poids d'adresse (A0 - A6)

Les lignes de sorties constituent un bus de données secondaire, lequel bus est relié au bus principal par l'intermédiaire de "buffers" à 3 états. Ce bus secondaire pourra être utilisé surtout pour l'utilisation de la *RAM associée au CRTC pour le stockage de données et pour le rafraichissement.

IV.3.4 La mémoire de caractères:

Une mémoire de caractères est une simple mémoire, organisée de manière à pouvoir adresser séparément soit ses lignes et le type de caractère soit ses colonnes et ses caractères. Dans ce type de mémoire le caractère est inscrit dans une matrice de base (7x5 ou 9x7)

Le spot décrit toutes les positions de cette matrice séquentiellement point par point et ligne par ligne.

Pour notre application nous avons choisi la ROM MCM 6575. Cette mémoire présente une capacité alphanumérique de 128 caractères.

IV.3.5 Etude de la MCM 6575

La fig montre que la MCM 6575 est constituée de six blocs: (Fig IV 6)

(43)

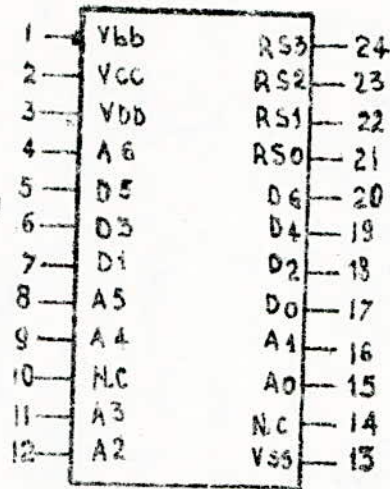


FIG AFFECTATION des Broches

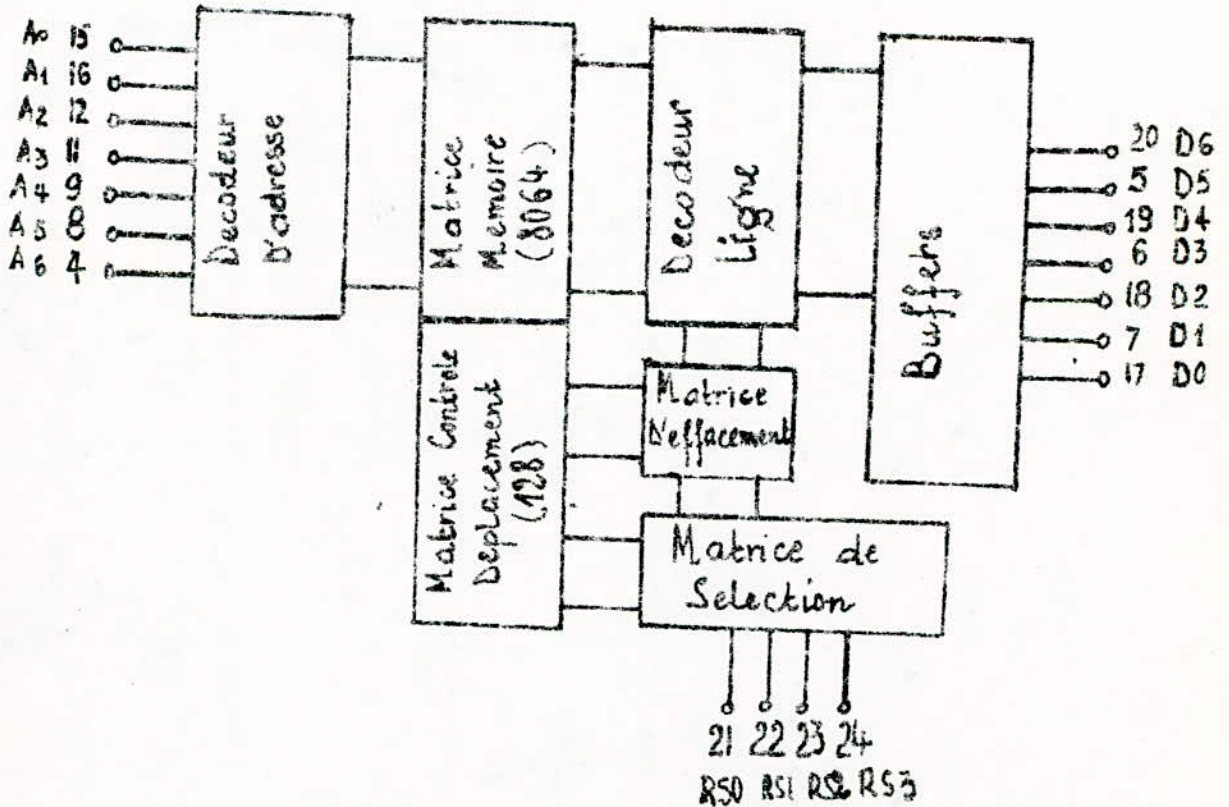


FIG 9 Schema Fonctionnel du MCM 6575

- Le décodeur d'adresses.
- D'une matrice mémoire de 8064 bits.
- D'une mémoire de contrôle de déplacement de 128 bits.
- D'un décodeur ligne.
- D'une matrice d'effacement.
- D'une matrice de sélection.
- De "buffers" de sorties.

1) Caractéristiques.

- Le temps d'accès (ou temps de lecture) maximum est de 500 ns.
- Le temps maximum d'accès d'adresse : 500 ns.
- Le temps d'accès maximum de section ligne 500 ns.
- Dissipation maximale : 800 mw.
- Capacité de déplacement des caractères.
- Compatibilité TTL/MOS.
- 3 Sources d'alimentation + 3,0v ; + 5 V ; 12 V.

2/ Mode d'adressage:

Elle reçoit 7 poids d'adresse en code (ASCII) aux entrées (A0--A6), permettant d'adresser un des 128 caractères, et suivant la valeur des bits (RS0 - RS4), les lignes du caractère adressé seront sélectionnées. Parce que le caractère est formé de 9 rangées de 7 points, l'adresse du type de caractère doit être appliquée séquentiellement 9 fois aux entrées (A0 - A6) en synchronisme avec le balayage ligne, tandis que les bits (RS0 - RS4) évoluent de 0 à 16.

Pour cette mémoire un point allumé correspond à un "1" logique en sortie alors qu'un point éteint correspondrait à un "0" logique.

La sortie du contenu du mot adressé s'effectue en parallèle sur les sorties (D0 - D6). On dispose d'un registre à décalage piloté par l'horloge HF.

Ce registre à décalage convenablement chargé par ce mot parallèle, pourra fournir à la cadence de l'horloge, les points séries.

Le chargement parallèle s'effectue pendant un bref instant entre les caractères.

La sortie série de ce registre est mixée au signal vidéo et on obtient sur l'écran une ligne complète du générateur de caractère.

Au 9^{ème} top on effectue un nouveau chargement parallèle. Ainsi aux 7 tops suivants on aura ((posé)) une deuxième ligne de la matrice du second caractère séparée de la première par 2 tops (qui peuvent se matérialiser par un noir, si la vidéo reste au noir pendant le changement parallèle).

V /

Programmation

Les registres internes du CRPC sont programmables par le processeur, par l'intermédiaire du bus de données et les signaux de contrôle R/\bar{W} , \bar{CS} , RS et E .

V . 1 Initialisation :

Les registres (R0 - R15) doivent être initialisés après la mise sous tension du système. Le processeur charge les registres du CRPC séquentiellement à partir d'une table firmware.

Dans notre cas où le moniteur est un récepteur de TV (CCIR 625 lignes) , l'initialisation du (MC 6845) consiste à l'adapter aux modes de balayages et de synchronisation du récepteur .

V . 1 . 1 Initialisation des registres de séquençement horizontal : (Fig. V 1)

- La durée d'une ligne de balayage en 625 lignes est de $64 \mu s$.

La fréquence d'horloge qui doit piloter le registre à décalage étant de 7 MHz donc la période est de $0,144 \mu s$.

Pour l'analyse d'une ligne d'un caractère il faut analyser 9 points (7 points-effectifs) et 2 de séparation).

D'où la période d'un caractère affiché :

$$T_c = 9 \times 0,144 \mu s = 1,296 \mu s .$$

- Le Nombre de caractères total qu'on peut afficher sur une rangée est programmé par R0 .

$$\frac{64}{1,296} = 49,382 \mu s . \text{ On prend } 49 \text{ pour avoir un nombre entier de périodes de}$$

caractères .

Le nombre programmable est : $49 - 1 = \underline{48} T_c$.

soit en hexadécimal : 30 .

1/ Nombre de caractères à visualiser par rangée : programmé par R1 =

On avait choisi une rangée de 40 caractères ,

d'où :

$$40 \times 1,296 = \underline{51,848} \mu s$$

Ce qui représente à peu-près la durée utile de la ligne de balayage écran qui est est de $52 \mu s$.

La valeur programmable est $\underline{40} T_c$ soit en hexa : 28 .

2/ Position de la synchro - horizontale ; (programmée par R2) :

Elle est de $52 \mu s$ durée utile pour $1,3 \mu s$ pour le palier avant

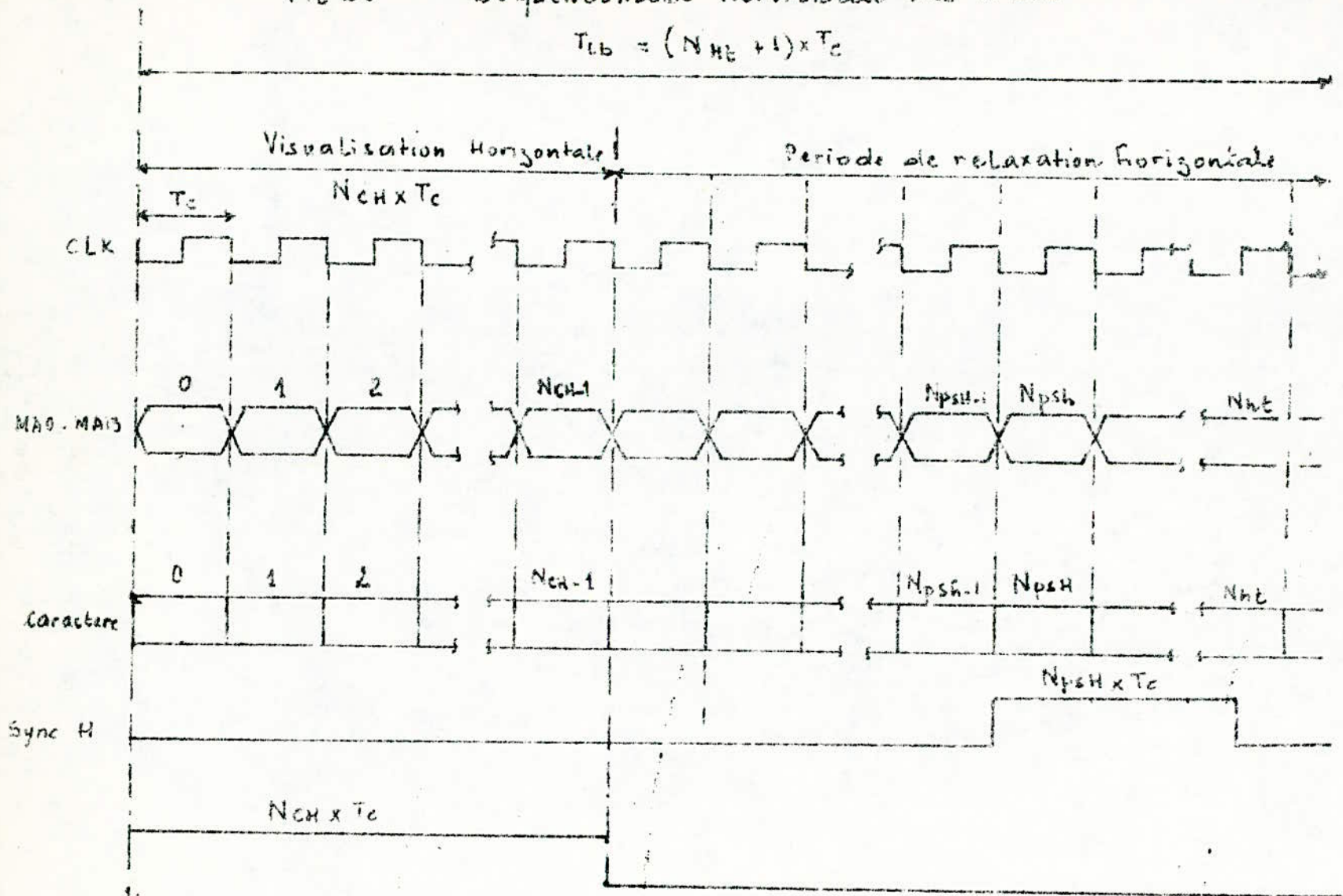
d'où :

$$\frac{53,3}{1,296} = \underline{41,126} \text{ on prendra } \underline{41} .$$

Fig 22

Sequencement horizontal du CRT

$$T_{LB} = (N_{CH} + 1) \times T_c$$



N_{psH} : nombre de cycle pour la largeur de la synchro

N_{CH} : nombre de caractere a visualiser horizontalement

- Le diagramme represente le sequencement pour la 1^{er} rangée seulement et pour $R_{12}/R_{13} = 0$

L'adresse memoire initial (MA) est determinée par le contenu de R_{12}/R_{13}

La valeur programmable est 40 soit en hexa : 31 .

3/ R3 . Largeur de l'impulsion de synchro :

En (625 lignes CCIR) cette impulsion varie

entre : $5 \pm 0,2 \mu s$.

d'où , $\frac{5}{1,296} = \underline{3,85}$ on prendra $\underline{4} T_c$

La valeur programmable est 4 .

V . 1 . 2 Initialisation des registre de séquençement vertical : (Fig V 2)

Pour l'analyse d'une rangée de caractères , il faut analyser 14 lignes (9 - lignes effectives est 5 interlignes) d'où la durées d'une rangées de caractères :

$$14 \times 64 \mu s = 896 \mu s$$

- 1/ Nombre de rangées susceptible d'être affichées :

Le balayage d'une trame dure $20 \mu s$, d'où :

$$\frac{20.000}{896} = \underline{22,321} .$$

Le nombre entier 22 sera programmé par R4 et la fraction (0,321) sera programmée par R5 .

Pour R4 la valeur programmable est $22 - 1 = 21$.

$$22 \times 896 = \underline{19,72} \text{ ms} .$$

Pour R5 la valeur programmable est :

$$\frac{0,320}{64} = \underline{5} .$$

2/ Nombre de rangées à afficher (R6) :

on a choisi 20 rangées pour notre format, soit :

$$20 \times 896 = \underline{17,92} \text{ ms}$$

3/ (R7) position de la synchro trame :

1,6 ms est nécessaire pour le retour trame d'où ,

$$20 \text{ ms} - 1,6 \text{ ms} = \underline{18,4} \text{ ms}$$

La valeur programmable est :

$$\frac{18,4}{896} = 20,53 ; \text{ on prendra } 20 :$$

$$20 \times 896 = \underline{17,92} \text{ ms}$$

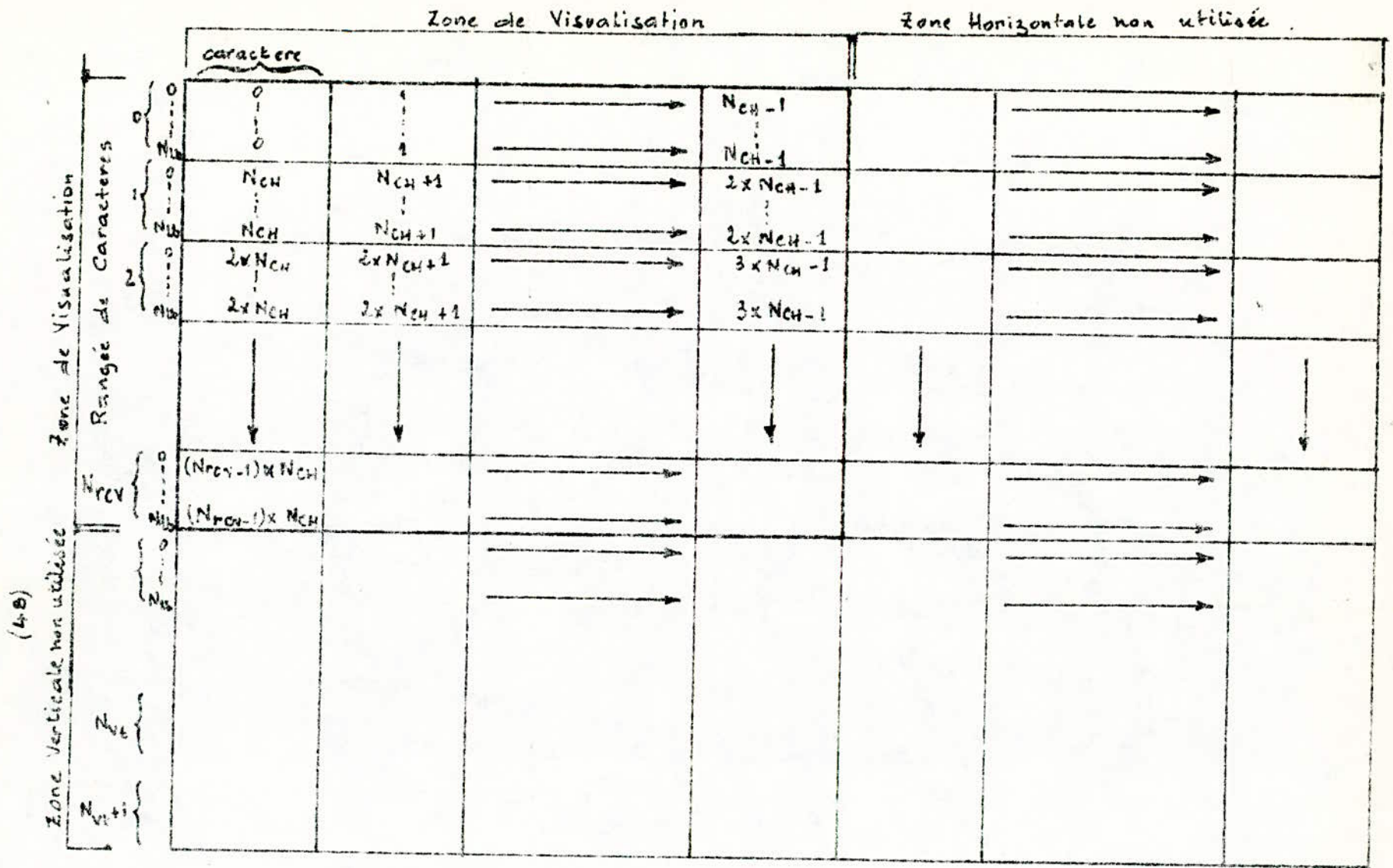
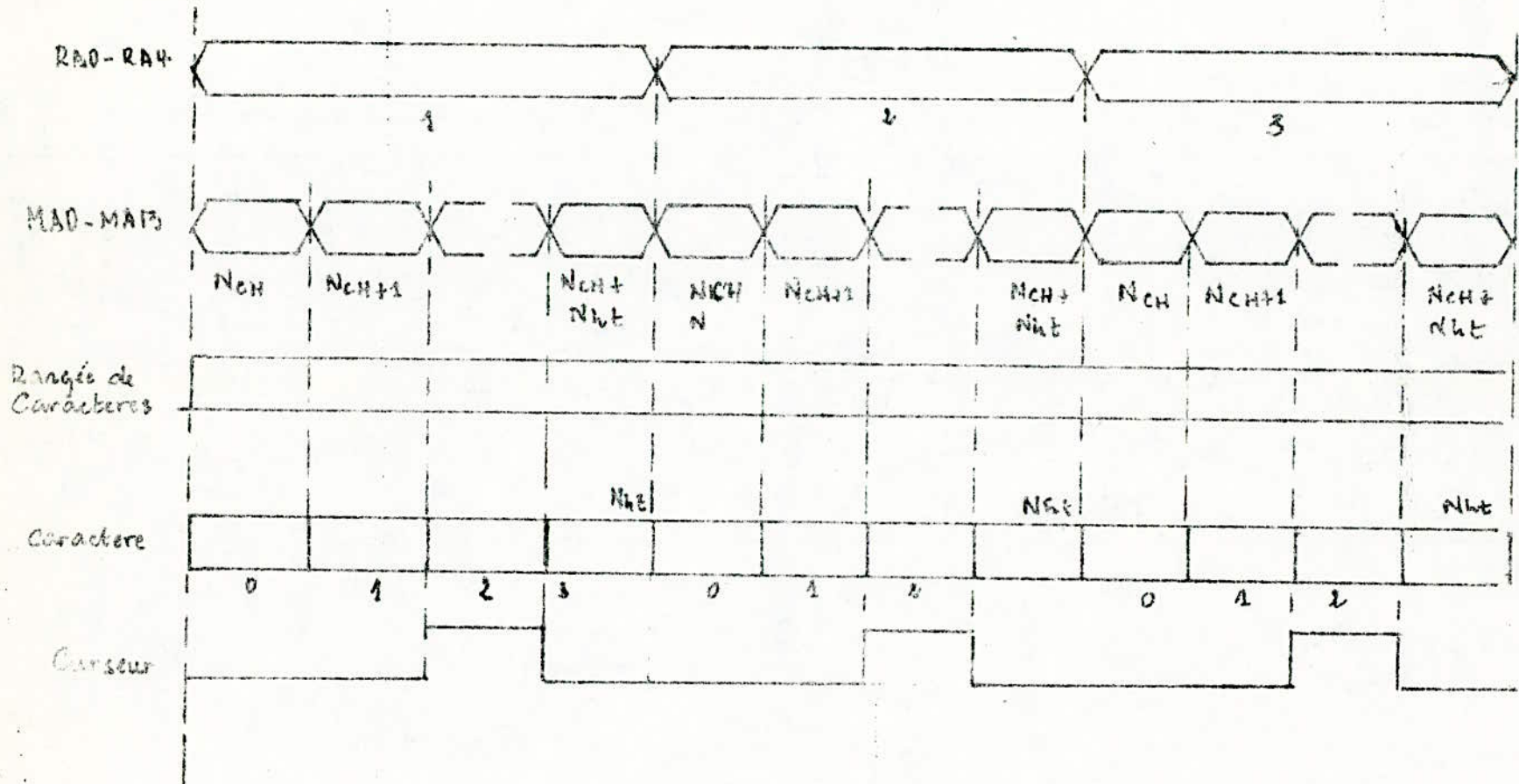


fig V2 Format de l'écran

Fig 13

Sequencement du Curseur



- NLT : Nombre de caractere total qu'on peut afficher horizontalement
 Sequencement pour les modes entrelacé et non entrelacé
 L'exemple est pour un Curseur programmé pour

- Registre du Curseur = $NCH + 2$

Les registres de {
 - debut du Curseur = 1
 - Fin du Curseur = 3

4/ (R8) Mode Entrelacé :

Dans notre cas le balayage est entrelacé d'où :

- le bit 0 du registre R8 doit être à 1 ,
- le bit 1 du registre R8 doit être à 0 .

Et la valeur programmable est " 1 ", soit en hexa 01 .

5/ (R9) fixe le nombre de lignes de balayage par rangées de caractères plus l'interlignes.

Dans notre cas ce nombre est 14 . (qui est la valeur programmable) .

6/ (R10) Début du curseur : (Fig 3)

Il se situe à la 11^{em} ligne de balayage c'est à dire en dessous du caractère de 2 interlignes .

7/ (R11) Fin du curseur :

A la 11^{em} ligne de balayage .

8/ (R12) Adresse de départ (H) : 0 (00 en hexa) .

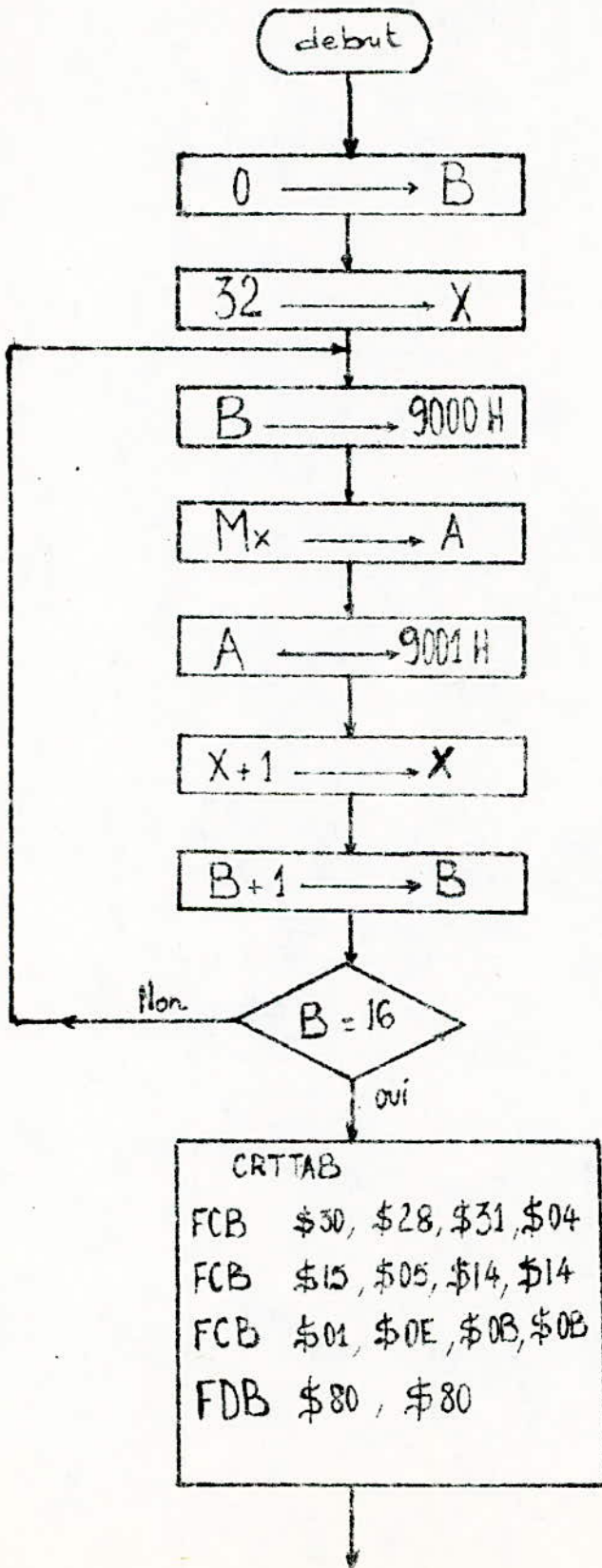
9/ (R13) Adresse de départ (L) : 128 (80 en hexa) .

Initialisation du CRTC pour le format 40x20

Pour le standard 625 lignes

Registres	Unité de Programme	Calcul	Valeurs Programmables	
			Decimal	Hexadecimal
R0	Tc	$49 \times 1,296 = 63,5 \mu s$	$49 - 1 = 48$	Nct = \$30
R1	Tc	$40 \times 1,296 = 51,84 \mu s$	40	Nch = \$28
R2	Tc	$41 \times 1,296 = 53,136 \mu s$	41	Npsh = \$31
R3	Tc	$4 \times 1,296 = 5,184 \mu s$	4	NLsh = \$04
R4	Ter	$22 \times 896 = 19,72 ms$	$22 - 1 = 21$	Nve = \$15
R5	Tsl	$5 \times 64 \mu s = 0,32 ms$	5	Nagst = \$05
R6	Ter	$20 \times 896 = 17,92 ms$	20	Nrav = \$14
R7	Ter	$20 \times 896 = 17,92 ms$	20	Npsv = \$14
R8	-			\$01
R9	Tsl		14	\$0E
R10	Tsl		11	\$0B
R11	Tsl		11	\$0B
R13	-			\$00
R13	-		128	\$80
R14	-			\$00
R15	-		128	\$80

Tsl: durée d'une ligne de balayage (64 μs).



Programme d'initialisation

PAGE	001	CRTINT			
1				NAM	CRTINT
2	00			ORG	\$0
3	00	5F		CLR B	
4	01	CE	0020	LDX	# \$ 20
5	04	F7	9000	CRTI1 STA B	\$ 9000
6	07	16	00	LDA A	0,X
7	09	B7	9001	STA A	\$ 9001
8	0C	08		INX	
9	0E	5C		INC B	
10	0E	CI	10	CMP B	# \$ 10
11	10	26	F2	BNE	CRTII
12	12	3F		SWI	
13	20			ORG	\$20
14	20	30	CRTTAB	FCB	\$30, \$28, \$31, \$04
15	24	15		FCB	\$15, \$05, \$14, \$14
16	28	01		FCB	\$01, \$0E, \$0B, \$0B
17	2C	80		FDB	\$80, \$80
18		00		END	
	CRTII	4	CRTTAB	20	

Commentaire

L'accumulateur B est utilisé comme compteur, au debut il est vide.

D'autre part l'indexe pointe la zone de stockage des donnees et il est chargé par la valeur 32 .

Le contenu de l'accumulateur B est stocké à chaque fois dans la position de memoire 9000 qui est l'adresse de la file de registres du CRTC. Donc en évoluant de zero à seize l'accumulateur B va permettre le comptage des 16 registres à initialiser .

Le MPU va chercher le contenu de l'adresse pointée par le registre d'index et le stockage dans l'accumulateur A . Ce contenu est ensuite chargé dans le registre du CRTC correspondant .

On incremente ensuite le registre d'indexe et l'accumulateur B d'une unite pour passer au registre suivant et a son contenu, et ainsi de suite jusqu'au dernier registre du C R T C .

V.1-3 Addressage - du générateur de caractères et Visualisation

Definition du Problème :

Le microprocesseur transmet dans la mémoire associée au MC 6845 un caractère ASCII sous forme de sept poids d'adresse.

Le générateur de caractères reçoit sept poids d'adresse et suivant les valeurs de (RS0 - RS3) envoie après la conversion parallèle série, sur la grille du tube cathodique, un mot de 7 bits correspondant à une rangée de la matrice du caractère à visualiser. Parce que le caractère est formé de 9 rangées de 7 points, l'adresse (A0 - A6) doit être appliquée séquentiellement 9 fois au générateur de caractères.

Pour l'affichage d'une rangée de 40 caractères, le premier caractère ASCII est envoyé sur le générateur; Le mot de 7 bits correspondant à RS0 = 0 ,RS1 = 0, RS2=0 RS3 = 0 , délivrés par le générateur de caractères est transmis vers un registre parallèle/série . Ce mot est visualisé sur la ligne de balayage 1 . Le second caractère ASCII correspondant est délivré par la RAM de donnée, sous contrôle du MC 6845 et sert d'adresse pour le générateur de caractère qui transfère vers l'écran les 7 bits de la première ligne du second caractère. Ces bits sont visualisés sur la ligne de balayage 1 décalés par rapport au premier caractère de 2 bits. Le même processus continue, sous contrôle du MC 6845, jusqu'à ce que la première ligne de balayage soit complètement parcourue.

La même séquence est répétée jusqu'à ce que la première rangée de caractères (9 lignes de balayage) ait été complètement affichée.

De la même façon on affiche une deuxième rangée, ensuite une troisième et ainsi de suite jusqu'à 20^{em}.

1) Transfert de donnée

Pour le transfert de données de la RAM du MPU dans la RAM associée au CRTIC , on utilise comme pointeur le registre d'index et le pointeur de pile.

L'index pointe vers le début de la zone de départ, dont l'adresse de début est (Beta + 1) . Le pointeur de pile vers la zone d'arrivée dont l'adresse de début est (Gamma) .

Remarquons que l'utilisation du pointeur de pile comme pointeur implique le masquage des interruptions et la sauvegarde du contenu du pointeur de pile dans les positions de mémoire SAUV et SAUV + 1 à chaque transfert .

Le programme ne comporte qu'une boucle.

Seu test de fin de zone est simple et n'est fait sur le registre d'index. Le principe de la copie est très simple:

- Transfert d'un octet par l'intermédiaire de l'accumulateur A
- du pointeur X

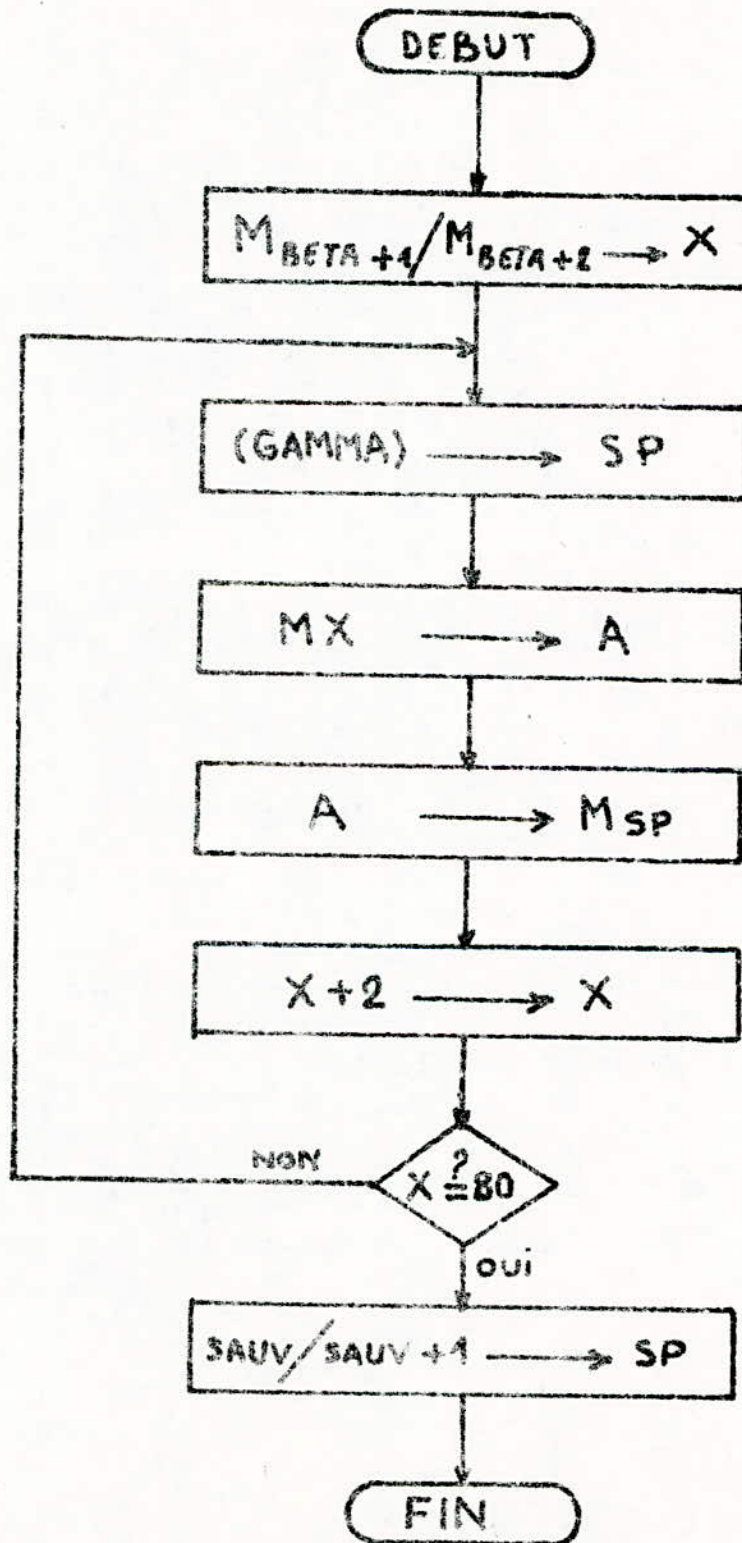
- La décrementation du contenu du pointeur de pile SP se fait automatiquement à chaque transfert
- Test de X .

2) Adressage du Générateur de caractères et Visualisation

L'horloge (CLK) amène le CRTC à adresser la mémoire associée à l'aide de (MA0 - MA13) en quête de caractères .

Ces caractères ont la forme de mots de 7 bits(ASCII) et sont envoyés à l'entrée du générateur de caractères. Celui -ci décharge ses matrices de points dont on adresse les lignes et les caractères dans un registre à décalage qui déverse son contenu dans la vidéo.

L'élaboration d'un caractère résulte du balayage de neuf lignes de 7 bits, donc de la lecture dans la ROM de caractères de 9 mots de 7 bits .



M_{SP}: Position de mémoire indiquée par le SP
 MX: Position de mémoire indiquée par l'index.

3 / Programme de transfert de donnée

	LDX	# Debut
	SEI	
	STS	
	LDS	# Gamma - 40
Boucle	LDAA	0,X
	PUSHA	
	INX	
	CPX	# Z fin
	BNE	Boucle
	LDS	
	END	

Commentaire.

- Chargement en Immédiat de l'adresse du début de zone de départ.
- Sauvegarde du pointeur de pile
- Chargement en Immédiat de l'adresse de fin de zone d'arrivée.
- Chargement de l'accumulateur avec la position de mémoire définie par X.
- Transfert du contenu de l'accumulateur vers la position de mémoire définie par le SP .
- Incrementation du contenu de X .
- Détection de fin de zone .
- Saut à l'adresse " boucle " si la copie n'est pas finie .
- Rechargement du pointeur de pile avec le contenu avant transfert.

- ANNEXE -

7.1 La mémoire de rafraichissement :

La figure montre une RAM de 16 K bytes qui servira au stockage de la liste de visualisation afin de pouvoir rafraichir l'écran toutes les 20ms.

La RAM utilisée est la MCM 6616 d'une capacité de 16 K bits. Pour avoir les 16 K bytes dont on a besoin on a regroupé 8 de ces mémoires de façon à ce que chacune d'elles serve à la mémorisation d'un bit. A la sortie on obtient des mots de 8 bits dont 7 bits serviront à l'adressage de la ROM de caractères.

Bien qu'on a besoin de mots de 7 bits seulement à la sortie, on a choisi des mots de 8 bits.

Ceci explique le fait que la mémoire de rafraichissement pourrait être utilisée pour d'autres fonctions ("stockage de données" du système MPU).

Par exemple une mémoire de 64 K bytes peut remplir les fonctions de rafraichissement et de stockage de données; en même temps.

Caractéristiques de la MCM 6616 L4 :

- Elle est organisée en 16.384 x 1 bit .
- Le temps d'accès est rapide : 250 us.
- Elle possède 2 lignes de selection utilisant des strobes ($\bar{C}\bar{A}\bar{S}$ et $\bar{R}\bar{A}\bar{S}$).
- Sa dissipation de puissance est basse: moins de 600 mw .

alimentations .

- 3 sources d'alimentation sont nécessaires :
- V_{DD} de 10,8 à 13,2 Voc
- V_{CC} de 4,5 à 5,5 V.c
- V_{BB} de - 4,5 V.c à - 5,5 V.C .

7.2 Le registre à décalage :

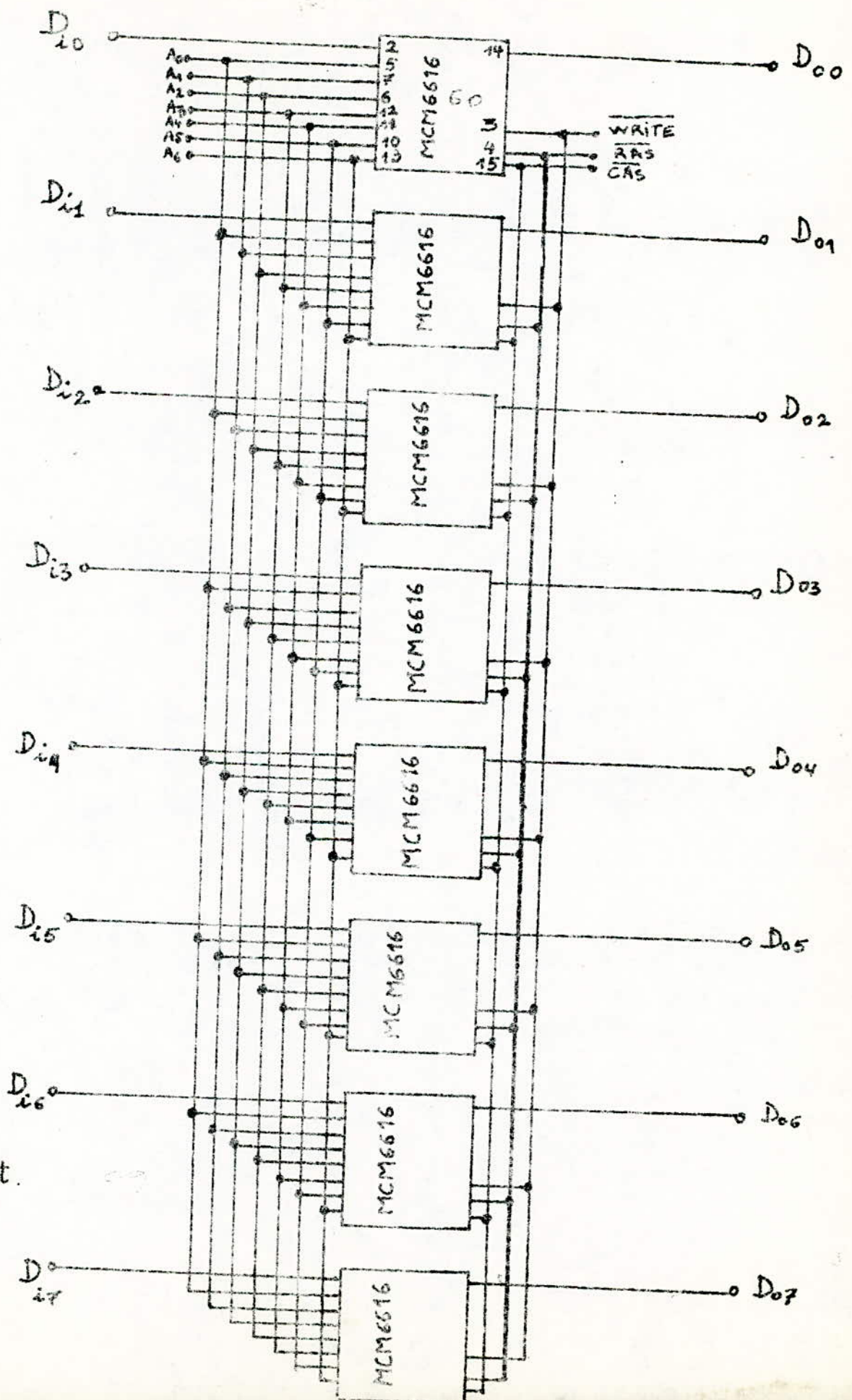
Pour la conversion de l'information affichable présente aux sorties (D0 -D6) de la ROM de caractères , de façon à ce que le spot soit excité séquentiellement par ces 7 bits , on utilise un registre à décalage. Ce registre est solidairement lié à la ROM et il est piloté par l'oscillateur HF (7 MHZ) de manière à ce qu'il y ait un synchronisme entre l'analyse et l'affichage du caractère.

Nous avons choisi le MC 14021 en technologie C -MOS qui est un registre statique à 8 bits .

Ce registre peut être utilisé soit en mode serie soit en mode parallèle.(fig)

Pour son utilisation en mode serie l'entrée P/S est constamment à "0"

D'autre par comme on a besoin que de 7 de ces bascules, notre sortie sera en Q7.



MEMOIRE RAM
rafraichissement.

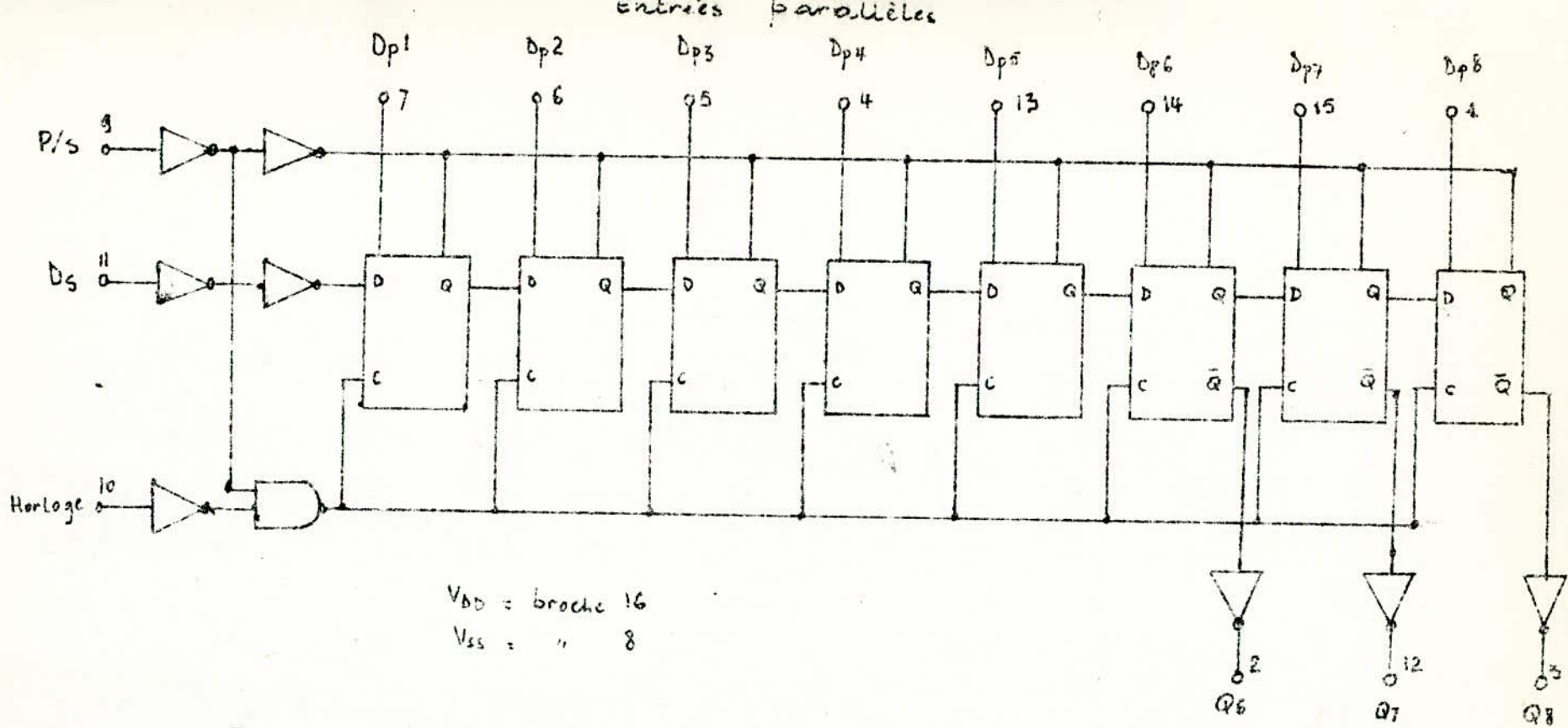


Fig Schema Logique du registre à decalage statique à 8 bits
"MC 14021 B"

Mode Serie

t	Horloge	Ds	P/s	Q6 t = n+6	Q7 t = n+7	Q8 t = n+8
n		0	0	0	?	?
n+1		1	0	1	0	?
n+2		0	0	0	1	0
n+3		1	0	1	0	1
		X	0	Q6	Q7	Q8

Mode parallele.

Horloge	Ds	P/s	Dm	* Qm
X	X	1	0	0
X	X	1	1	1

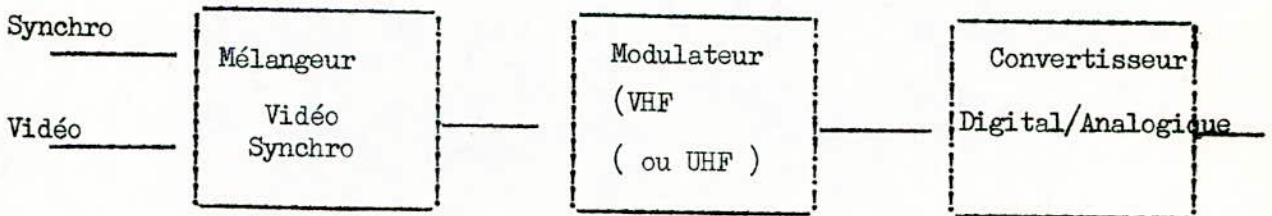
* Q6, Q7 et Q8 sont disponibles
 X sans attribution 0 ou 1

V / .3 La sortie vidéo :

La conversion du parallèle en serie ainsi effectuée, pour être visualisées ces informations doivent subir au préalable un traitement.

D'abord pour l'obtention du signal composite (vidéo ~~à~~ **synchro**), il faut prévoir prévoir un étage mélangeur vidéo synchro. D'autres part pour pouvoir entrer sur la prise d'antenne il faut que ce signal soit modulé. Dans ce cas on utilisera un modulateur (UHF ou VHF) . (Il existe des modulateur digitaux)

Finalement le signal obtenu est digital pour l'utiliser il faut un troisième étage qui fera la conversion du digital à l'analogique .



V / .4 Les " buffers " à 3 etats "

Ce sont des portes de puissances dont la sortie peut prendre 3 états: un état au n niveau logique "1" , un état au niveau logique "0" et un troisième état de haute impédance dans lequel la sortie est équivalente à un circuit ouvert. Ce troisième état avantage énormément la mise en parallèle de plusieurs circuits dont un seul est en service à la fois.

Pour nos applications nous avons choisi le MCM 14503 en technologie C - MOS .

Ce circuit comporte 6 portes et 2 entrées de validation " Disable " (voirfig)

V.5. Le Multiplexeur :

C'est un interrupteur sélectif à N entrées et une sortie. On sélectionne l'entrée qui sera envoyée à la sortie à l'aide d'entrées de sélection ,

Pour faire ce choix le multiplexeur recoit une adresse de n bits telle que :
 $N = 2^h$.

Nous avons choisi le MCM 14052 en technologie C - MOS , Pour multiplexer les entrées d'adresse de la mémoire de rafraichissement entre le bus adresse (A0 - A15) du MPU et Les 14 lignes (MA0 - MA13) du CRTIC . (voirfig)

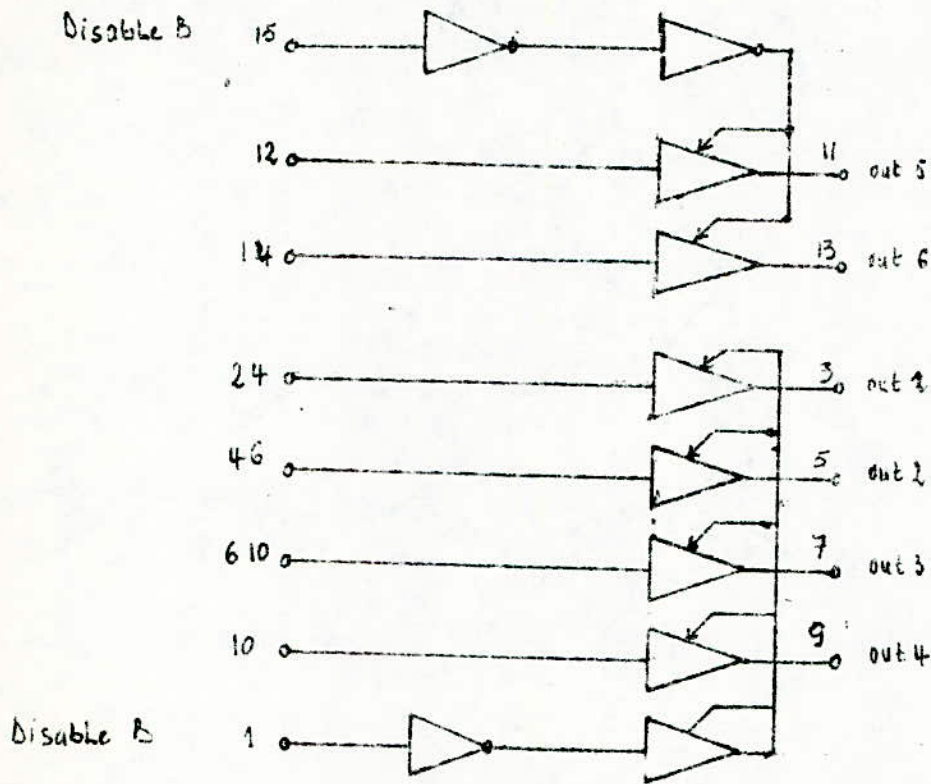


Fig Schema du MC 14503B "Buffers à 3-états"

Table de vérité

In_n	entrée Disable	Out n
0	0	0
1	0	1
X	1	Haute Impédance

X = 0 ou 1

V / .6 Le circuit tampon " Latch " :

Entre la RAM de rafraichissement et la ROM de caractères se trouve un circuit tampon .

Ce circuit délivrera donc à sa sortie l'information présente sur les sorties (D0 - D7) de la MCM 6616, ~~des~~ qu'il reçoit un strobe (horloge) et que ses entrées (D0 - D3) sont à un niveau "0" ou à un niveau "1" .

Il est verouillé sur les fronts arrière ou avant des impulsions des entrées (D0 - D4) .

L'emploi de ces mémoires tampons à été rendu nécessaire pour maintenir l'information permanente en sortie de la RAM, tandis que l'adressage (A0 - A6) de ces RAM est entrain de changer .

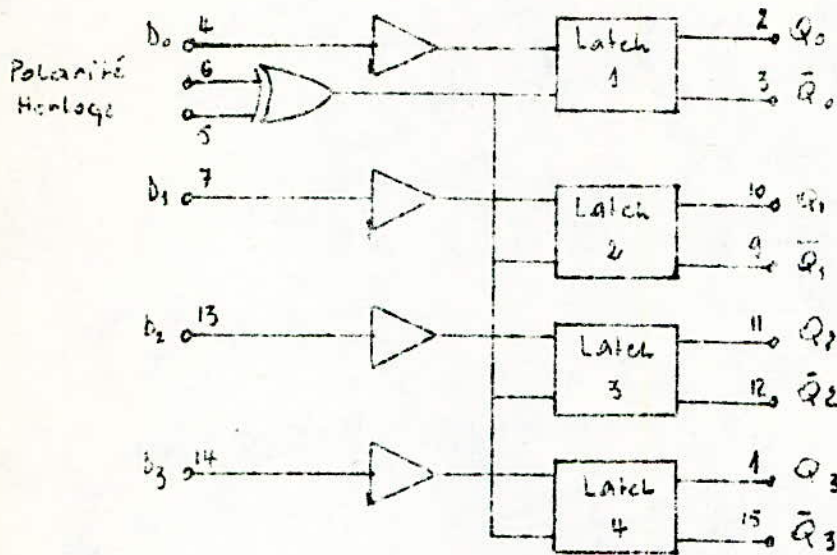
Pour avoir un synchronisme entre l'adressage et l'affichage, ce latch est piloté par le même oscillateur HF (7MHZ). que le registre à décalage. (voir fig)

V / .7 Le clavier alphanumérique :

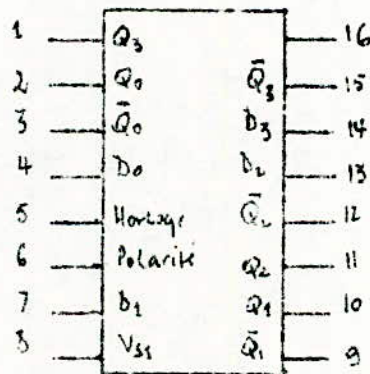
Ce clavier est identique à celui d'une machine à écrire. Il permet l'inscription de textes et l'envoi de paramètres alphanumériques. Le nombre de touches ainsi que la nature des signes disponibles seront fonction du code utilisé: ISO, QWERTY, ASCII, etc... Pour notre cas nous utiliserons un clavier à code ASCII . Le principe de fonctionnement est basé sur l'injection d'un signal d'interruption qui, s'il est accepté, provoquera la lecture dans un registre tampon du code du caractère concerné -



Précisons l'existence, lors d'édition de textes, du repère particulier appelé " curseur " et qui indiquera la position du prochain caractère.

Fig Schéma Fonctionnel du QuadriLatch "MC14042 B"



Q et \bar{Q} sont les Sorties



Data	Horloge ou strobe	Q
0	0	Données
	0	Verrouillé
1	1	Donnée
	1	Verrouillé

Bits 4 à 6	0	1	2	3	4	5	6
0	NUL	DLE	SP	0		P	p
1	SOH	DC1	"	1	A	Q	q
2	STX	DC2	"	2	B	R	r
3	ETX	DC3	#	3	C	S	s
4	EOT	DC4	\$	4	D	T	t
5	ENO	NAK	%	5	E	U	u
6	ACK	SYN	&	6	F	V	v
7	BEL	ETB	'	7	G	W	w
8	BS	CAN	(8	H	X	x
9	HT	EM)	9	I	Y	y
A	LF	SUS	*	A	J	Z	z
B	VT	ESC	+	B	K	[{
C	FF	FS	,	C	L	\	
D	CR	GS	-	D	M]	~
E	SC	RS	.	E	N	^	_
F	SI	US	/	F	O	~	DEL

Figure 3 : Code ASCII

CONCLUSION .

Nous avons l'avantage de profiter des progrès de la technique qui fait que toute la partie synchronisation et balayage du texte sur l'écran TV ainsi que la gestion de la mémoire de rafraichissement et la gestion du curseur, peut tenir dans un circuit intégré unique :

" le controleur d'écran cathodique " (MC 6845).

Ce circuit est complètement programmable par le MPU .Il constitue la partie intelligente d'un terminal alphanumerique .

Le fait de disposer d'un nombre de lignes programmables, de caractères par lignes programmables, d'un curseur programmable, le rendent universel .

Dans notre travail on a étudié une application typique du CRTIC pour l'affichage sur écran TV 625 lignes .

Si on dispose d'un standard différent du notre on peut profiter du travail effectué dans le cadre de ce projet en changeant certaines valeurs de programmation.

Donc pour n'importe quel moniteur utilisé, il suffirait d'initialiser ce CRTIC .

D'autre part le choix de 40 caractères par rangée est assez raisonnable, et les problèmes de haute fréquence liés à l'emploi de circuit MOS sont pratiquement évités .

Il ne faut pas toutefois perdre de vue que tous les circuit utilisés dans notre cas sont digitaux et du même constructeur, d'où une grande fiabilité quand à la réalisation de ce système .

Le fait que tout est programmé dans ce système procure une grande souplesse d'adaptation du " terminal " à différentes applications .

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

- H. LILEN : Du micro processeur au micro ordinateur.
Editions RADIO.
- H. LILEN : Programmation du Microprocesseur .
Editions RADIO .
- P. MORVAN et M. LUCAS : Images et Ordinateurs.
Editions LAROUSSE .

- Documentation MOTOROLA :
 - M 6800 : Microprocessor Programming
MANUAL.
 - M 6800 : Microcomputer System design DATA .
 - Mc MOS HANDBOOK .
 - Mc 6800: CRTIC .

- R. CARRASCO - J. LAURET : Cours fondamental de TV .
Editions RADIOS .

- Revue MINI MICRO .
- Projet de Fin d'études :
Etude et realisation d'une unite de commande d'un Microprocesseur.