

32/83

2ex

FILIERE D'INGENIEUR EN ELECTRONIQUE

C.E.N.

LABORATOIRES D'ETUDES SPATIALES  
DE RAYONNEMENT



**PROJET DE FIN D'ETUDES**

SUJET : II Système interactif

- Contribution a la réalisation d'une console interactive

PROPOSE PAR :

A. ABDELLAOUI

REALISE PAR :

M. HENHEN

M. MOUZAÏ

JANVIER 1983

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE D'ALGER

DEPARTEMENT D'ELECTRONIQUE ET D'ELECTROTECHNIQUE

FILIERE D'INGENIEUR EN ELECTRONIQUE

C.E.N.

LABORATOIRES D'ETUDES SPATIALES  
DE RAYONNEMENT

## **PROJET DE FIN D'ETUDES**

SUJET : II Système interactif

- Contribution à la réalisation d'une  
console interactive

PROPOSE PAR :

A. ABDELLAOUI

REALISE PAR :

M. HENHEN

M. MOUZAÏ

JANVIER 1983

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ  
وَمَا أَوْفَيْتُمُوهَا مِنْ الْعِلْمِ لَأَنْتُمْ سَوَاءٌ  
صَوَّرَ اللَّهُ الْأَشْيَاءَ كَمَا يَشَاءُ

## Remerciements.

Ce Projet a été réalisé au laboratoire d'études Spatiales de rayonnement du C.E.N.

Nous formulons en ces lignes l'expression de notre profonde et sincère gratitude à Monsieur Abdellaoui pour ses précieux conseils qui n'a cessé de nous prodiguer tout au long de ce travail.

Nous tenons également à Remercier Monsieur Menacer Saïd pour les nombreuses discussions critiques et son soutien moral.

Nous Remercions Monsieur BourKEB.

Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail trouvent ici l'expression de nos plus vifs Remerciements.



## - SOMMAIRE -

### - CHAPITRE I -

1. STRUCTURE générale du système de traitement et de visualisation
2. Présentation des traitements réalisés sur ce système
3. objectif de la nouvelle restructuration.

### - CHAPITRE II -

#### A) CARTE DE GESTION

- I - Présentation du microprocesseur (MC6800)
- II - Description des différents éléments de la Carte
  - II-a) des mémoires à lecture-écriture (RAM)
  - II-b) des mémoires ROM - EPROM
  - II-c) interface circuit coupleur PIA - ACIA

#### B) l'interface actif (EP9365) et mémoire d'image

- I - l'interface actif (EP9365)
  - I-a) Généralité
  - I-b) Description des registres internes du coprocesseur
  - I-c) Description des signaux sortant des broches du coprocesseur
- II - Interface microprocesseur - processeur graphique
- III - Gestion de la mémoire d'image
- IV - Registres d'échanges (Tampons)
- V - le séquenceur haute vitesse
- VI - RAM DYNAMIQUE pour mémoire d'image (6F4116)
- VII - Decodage d'adresse -

### - CHAPITRE III -

- I - Role des registres d'échanges.
- II - Accès à la mémoire d'image par une circuiterie externe
- III - Aspect programmation du coprocesseur
- IV - le traitement Zoom.

# CHAPITRE I

I. STRUCTURE GENERALE DU SYSTEME DE  
TRAITEMENT ET DE VISUALISATION

II PRESENTATION DES TRAITEMENTS REALISES  
SUR CE SYSTEME.

III objectif DE LA NOUVELLE RESTRUCTURATION



- 2 -

~ INTRODUCTION ~

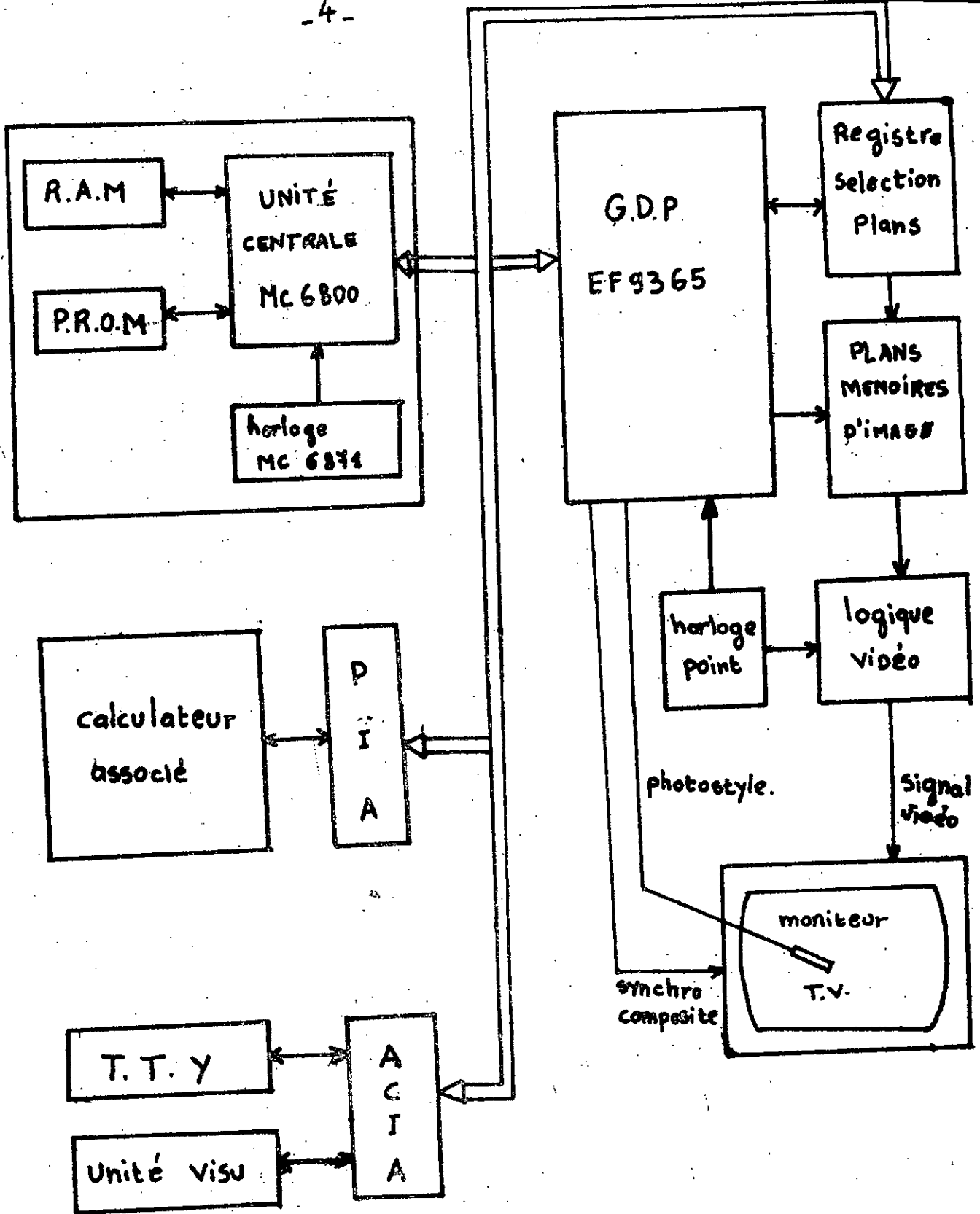
Au cours des vingt dernières années, le traitement numérique d'images a largement bénéficié de l'augmentation des performances et de la diminution de prix des équipements digitaux d'où son utilisation de plus en plus généralisée dans de nombreux domaines: physique nucléaire, médecine, télédétection ....

Notre travail consiste à améliorer le précédent système interactif qui ne permettait que la visualisation d'images numérisées. En plus ce système ne peut fonctionner qu'en liaison permanente avec le calculateur associé ce qui le rendait prohibitif pour une utilisation prolongée en plus de l'indisponibilité du calculateur. Une solution plus souple et plus économique est envisagée: c'est de rendre la console plus autonome et intelligente permettant un certain nombre de traitements simples. Cela nous a amené à restructurer la mémoire d'images et à étudier une carte de gestion bâtie autour du microprocesseur 6800 et d'un coprocesseur graphique EF 9365.

Dans le premier chapitre nous présenterons la structure générale du système de traitement d'images ainsi que les différents traitements.

Dans le deuxième nous détaillerons la structure de la carte de gestion et de la nouvelle mémoire d'image.

Le troisième contiendra les programmes de gestion et d'initialisation du système ainsi que ceux des traitements exécutables au niveau de la carte de gestion ; en annexe les différentes fiches techniques des composants utilisés.



STRUCTURE GENERALE DU SYSTEME INTERACTIF



## I. STRUCTURE GÉNÉRALE DU SYSTÈME DE TRAITEMENT ET DE VISUALISATION

Le système interactif de traitement d'images comprend :

- Une unité de calcul type Mitra 125
- Une carte M.A.V. bâtie autour du  $\mu p$  6800
- Une de gestion d'écran et de conversion numérique-analogique dont l'élément essentiel est un coprocesseur graphique EF 9365.
- d' Une mémoire d'image
- Un moniteur T.V.

### I.1 Unité de calcul :

C'est un mini ordinateur à temps réel grâce à sa conception et sa structure micro programmée ; il permet d'aborder avec efficacité de nombreux domaines d'application : calcul scientifique et automatisation de processus industriels.

Du point de vue structure l'ordinateur Mitra 125 est bâti autour d'une mémoire principale unique munie de quatre accès permettant la connexion d'une ou 4 unités de traitement ou coupleur.

L'unité élémentaire d'information de la mémoire est le mot.

Le temps d'accès de la mémoire est de 400 nanosecondes.  
Un cycle de lecture-écriture est de 800 nanosecondes.  
L'ordinateur travaille en mots de 16 bits c'est à dire que  
les échanges avec la mémoire s'effectue par mot  
mais des microprogrammes permettent d'opérer par octets  
ainsi les adresses sont des adresses d'octets.  
Les mots sont à des adresses paires  
La mémoire est modulaire par blocs de 4096 mots  
Le Mitra 125 contient au maximum 8 blocs soit  
32 Kilo mots.

### I.2 Carte de gestion de la mémoire d'image :

Congue autour du G.P.P EF 9365 qui constitue un  
véritable processeur graphique à haute résolution et  
possédant toutes les fonctions nécessaires à la géné-  
-ration de vecteurs, de points et de caractères à très  
grande vitesse et des signaux logiques nécessaires  
pour l'interfaçage avec un moniteur TV en mode  
entrelacé et non entrelacé compatible aux normes  
CCIR (625 lignes, 50Hz).

La souplesse d'utilisation du G.P.P résulte de son  
interfaçage direct avec n'importe quel microprocesseur 8bits,  
et de ses 11 registres internes.

Le rafraîchissement de la mémoire est totalement sous le contrôle du coprocesseur. Le microprocesseur est seulement concerné par la gestion du registre des caractéristiques et la transmission des commandes d'écriture vers le coprocesseur.

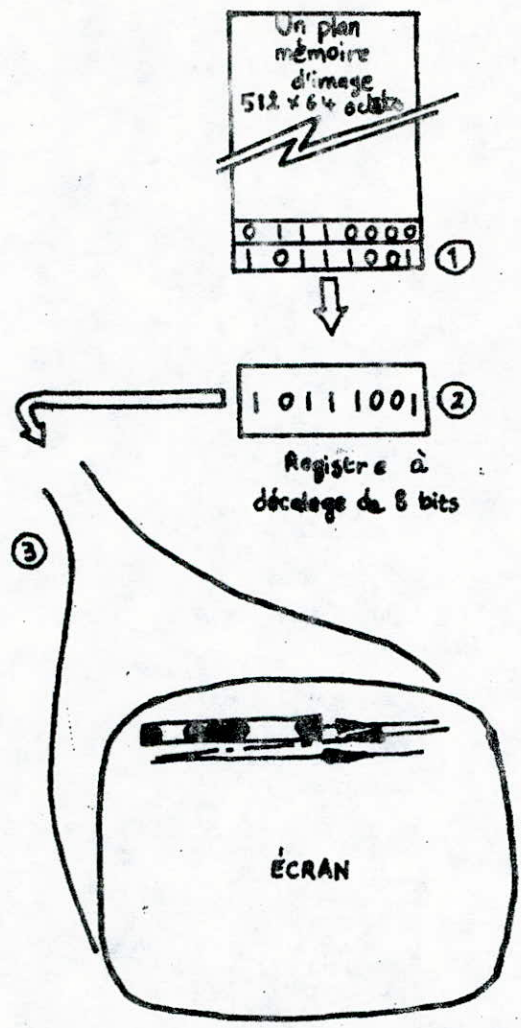
On détaillera son fonctionnement et sa programmation dans les chapitres suivants.

### I.3 MÉMOIRE D'IMAGE

La mémoire d'image est de capacité  $512 \times 512$  bits soit 32 kilo octets, de mémoires vives RAM dynamique EF 4116, par plan. Chaque pixel est stocké sur 8 plans donc pouvant avoir 256 niveaux de gris. Cette mémoire est gérée automatiquement par le G.O.P. Vue par le concepteur, cette mémoire se comporte comme une mémoire statique car rien n'est à prévoir pour la rafraîchir au niveau de la conception du circuit électronique.

Le processeur se charge lui-même de parcourir la mémoire d'image aussi bien au niveau de la visualisation et de la lecture externe et sait sélectionner les bons bits à positionner à "1" ou à "0" lors d'une écriture.





Cheminement et transformation d'un octet de la mémoire d'image.

figure 2.

Pour la structure à 8 plans mémoires le processeur graphique adresse simultanément tous les plans mémoires pour afficher l'image en plusieurs niveaux de gris.

En fait chaque ligne à afficher est ici constituée de 512 pixels et peut être découpée en 64 tranches de 8 pixels adjacents. Dans la mémoire d'image; ces 8 pixels correspondent à 8 bits consécutifs (c'est à dire 1 octet) et ceci dans chaque plan mémoire. Chaque accès charge simultanément les 8 registres à décalage de 8 bits afin qu'ils soient sérialisés et visualisés sur le moniteur TV. Voir figure (2)

#### I.4 Moniteur de Television

Pour dialoguer avec un ordinateur, l'utilisateur a besoin de moyens performants et surtout simples à utiliser. Très vite les consoles « écran-clavier » se sont révélées être le matériel le mieux adapté pour communiquer avec un système informatique elles sont de plus en plus sophistiquées et utilisent en général un tube à rayon cathodique pour l'affichage des informations. Le tube à rayon cathodique n'est pas le seul système qui soit utilisé au niveau de l'affichage alphanumérique,

mais la solution la plus répandue et qui présente le meilleur rapport performance / prix est le tube à rayon cathodique balayé régulièrement par un faisceau d'électrons, analogue à un tube T.V.

Ce mode de balayage, d'ailleurs appelé « balayage télévision » ; est celui qui conduit au maximum de simplicité dans sa réalisation pratique.

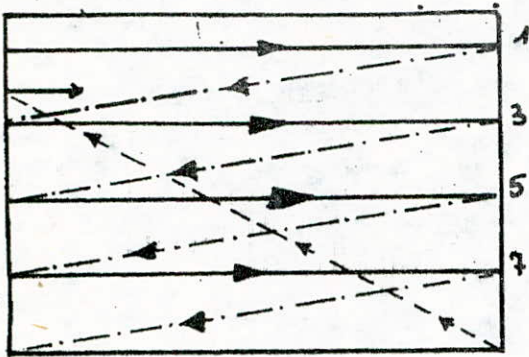
Le balayage se fait ligne par ligne de la gauche vers la droite et de haut en bas. Les parties « aller » du balayage se font à vitesse constante et en modulant les grilles de commande du tube cathodique suivant les points à afficher sur l'écran. Les parties « retour » doivent être aussi rapides que possible.

Cependant il faut considérer la possibilité d'affichage de l'image.

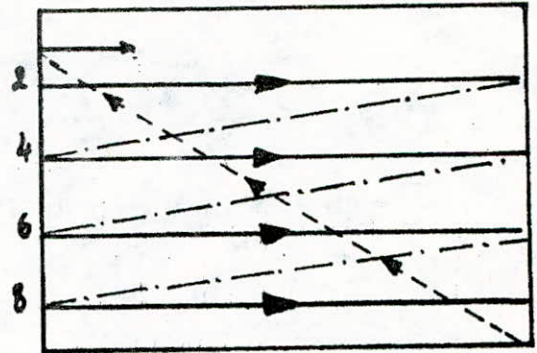
### Le balayage entrelacé

En premier sont transmises les lignes impaires en  $1/50$  de seconde dont l'ensemble constitue une trame puis en  $1/50$  de seconde, également les lignes paires de la 2<sup>e</sup> trame ainsi le balayage ne se fera pas en 25 images par seconde mais en 50 demi-images par seconde ce qui revient au même pour la définition.  
voir figure 3.





trame impaire.



trame paire.

----- retour trame (faisceau éteint)

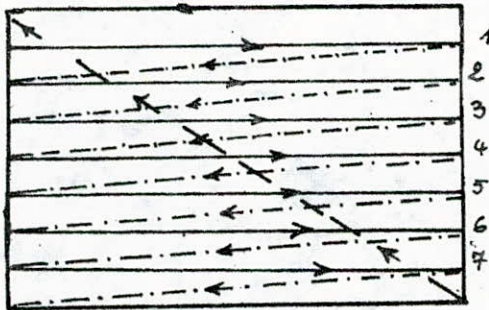
-.-.-.- retour ligne (faisceau éteint)

figure 3.

Pour l'image entrelacée = trame impaire + trame paire.

Le balayage non entrelacé :

L'ensemble des lignes (paires et impaires) formant ainsi une seule trame est scrutée en un seul passage.



----- retour ligne.

-.-.-.- " trame.

balayage non entrelacé : une image = trame complète.

Une trame est donc une suite de « lignes vidéo » qui sont elles mêmes un ensemble de « points vidéo » dans tous les cas, chaque ligne dure  $64 \mu s$ . La fréquence trame étant synchronisée sur la fréquence du secteur, deux standards télévision existent

Durée d'une Ligne	PAYS	nbre de LIGNES	fréquence trame.
$64 \mu s$	USA JAPON CANADA.	525	60 Hz
$64 \mu s$ .	EUROPE	625	50 Hz.

### Standards de balayage t.v.

Ces 2 standards portent sur le balayage vidéo et n'ont rien à voir avec le procédé PAL, SECAM et NTSC de codage de l'image couleur transmise par faisceaux hertziens. Ces procédés ne rentrent donc pas en ligne de compte dans les applications graphiques gérées par micro-processeurs ou des processeurs spécialisés. Les signaux utiles sont les signaux vidéo (exemple: signal de synchronisation composite) aux normes C.C.I.R. (1V, 75 $\Omega$ ).



## II.1 PRESENTATION DES IMAGES DESTINEES AU TRAITEMENT

Les traitements étudiés concernent des images multispectrales prises par le système MSS, la structure de cette image est la même dans tous les canaux.

Dans un canal, l'image est constituée d'un ensemble de 2340 lignes parallèles chacune d'elles représente une distance de 185 km et comprend entre 3000 et 3450 octets d'information vidéo.

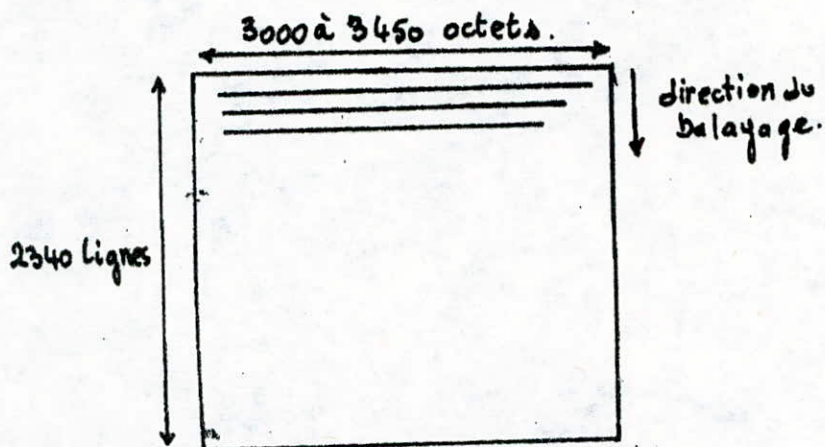


Schéma de l'image MSS

Les 4 types de détecteurs du système MSS travaillent respectivement dans les bandes spectrales suivantes.

Bande 4: 0,5 à 0,6  $\mu\text{m}$  située dans la partie

visible du rayonnement électromagnétique, elle permet l'identification des structures géologiques et la distinction entre les différents types de végétations.

BANDE 5 : 0,6 à 0,7  $\mu\text{m}$  située en partie dans le visible et en partie dans le proche infra-rouge elle est utilisée pour déterminer les caractéristiques topographiques et d'autre part pour classer les différents types de végétations.

BANDE 6 : 0,7 à 0,8  $\mu\text{m}$ . Située dans le proche infra-rouge elle est surtout utilisée pour l'identification des sols.

BANDE 7 : 0,8 à 1,1  $\mu\text{m}$  située entièrement dans l'infra-rouge elle permet l'obtention des renseignements sur l'humidité (hydrologie) et sur les reliefs.

## II.1 But du traitement d'images :

Son rôle est d'extraire à partir des données "images" corrigées des sous ensembles de l'image (objets, zones ayant des caractéristiques communes, caractères etc...) qui seront identifiés et mesurés. Afin de simplifier la recherche de ces paramètres on effectue un certain nombre de traitements à savoir :

Les manipulations d'intensité; Les traitements géométriques, le codage et la compression des informations.

## II.2. Opérations de traitements :

### a) manipulation d'intensité :

Le principe de ce traitement est de remplacer l'intensité  $Z$  d'un point de l'image par  $f(Z)$



$f(z)$  est une fonction non linéaire ne dépendant que de  $z$   
Ce type de traitement permet la correction gamma, correction de non linéarité des capteurs, l'augmentation du contraste la réalisation d'une fenêtre

### b) traitements géométriques

Ces techniques consistent à déformer l'image (ou des parties d'images) en déplaçant les points dans le but de redressement de l'image, correction de l'angle de prise de vue.

effet Zoom permet d'effectuer un agrandissement sur une image ou sur une partie de celle-ci.

### c) codage et compression.

on peut réduire considérablement le nombre de bits nécessaires à la mémorisation (ou à la transmission) si l'on emploie un codage approprié.



### III. objectif de la nouvelle restructuration :

Ainsi décrit ce système n'est conçu que pour visualiser les images qui sont traitées sur le calculateur associé, mais vu la très grande capacité de la mémoire d'image avec ses 8 plans qui autorise 256 niveaux de gris que l'œil ne pourra distinguer on se contentera de 16 niveaux de gris seulement et cela sans altérer la qualité des résultats obtenus après les traitements énumérés auparavant.

On subdivisera la mémoire d'images en 2 feuilles chaque feuille comprendra 4 plans, on utilisera une feuille pour le stockage de l'image issue du calculateur et l'autre servira pour des traitements locaux simples tel que le zoom et le recadrage. Ces traitements se feront au niveau de la console.



---

## CHAPITRE II

---

### - Introduction -

Dans ce chapitre nous présenterons les solutions qui ont été adoptées pour rendre le système plus autonome vis à vis du calculateur. Le système est composé d'une carte de gestion, d'un coprocesseur de visualisation (EF 9365) et d'une mémoire d'image.

Pour un accès externe à la mémoire d'image, nous utiliserons des registres tampons, qui servent d'une part à sélectionner les différents plans et d'autre part à stocker l'information momentanément pour l'envoyer ensuite vers la mémoire d'image ou bien de celle-ci vers le microprocesseur "Opération lecture".

La carte de gestion sera bâtie autour d'un microprocesseur (MC 6800). La mémoire R.A.M. aura une capacité de 1K octet.

On utilisera aussi une mémoire Reprogrammable de 2K octet afin de stocker les programmes des différents traitements qu'on aura à effectuer sur l'image originale.

---



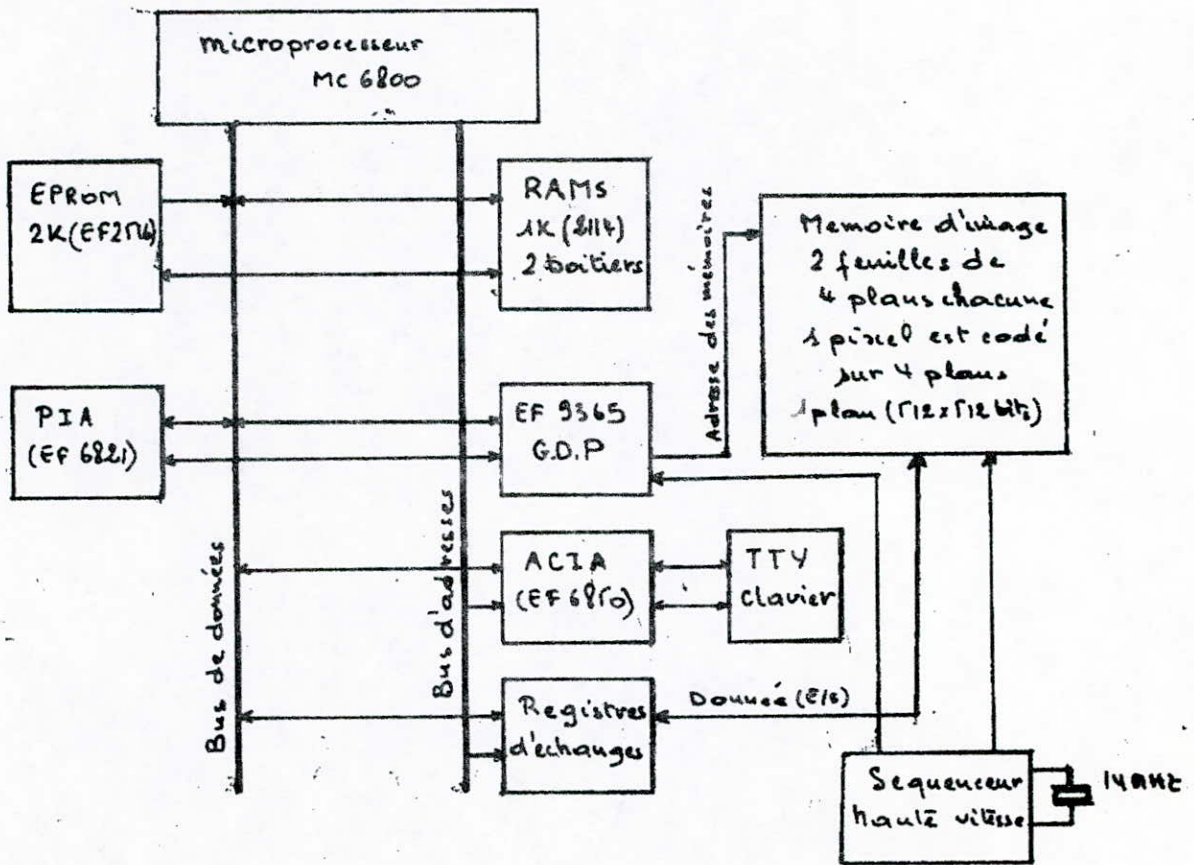
A) CARTE DE GESTION

La carte de gestion est bâtie autour d'un microprocesseur MC6800. Les différents traitements simples effectués sur l'image originale doivent être intégrés dans mémoires EPROM (EF2116) de deux kilo-octets. On a utilisé aussi des mémoires RAMS (EF2114) de capacité - 1 kilo mots de quatre bits - pour le stockage des variables au cours du programme.

Nous avons besoin aussi d'un interface parrallèle pour l'interfaçage de la carte de gestion avec une calculateur. Nous avons utilisés pour cela un PIA (EF 6821)

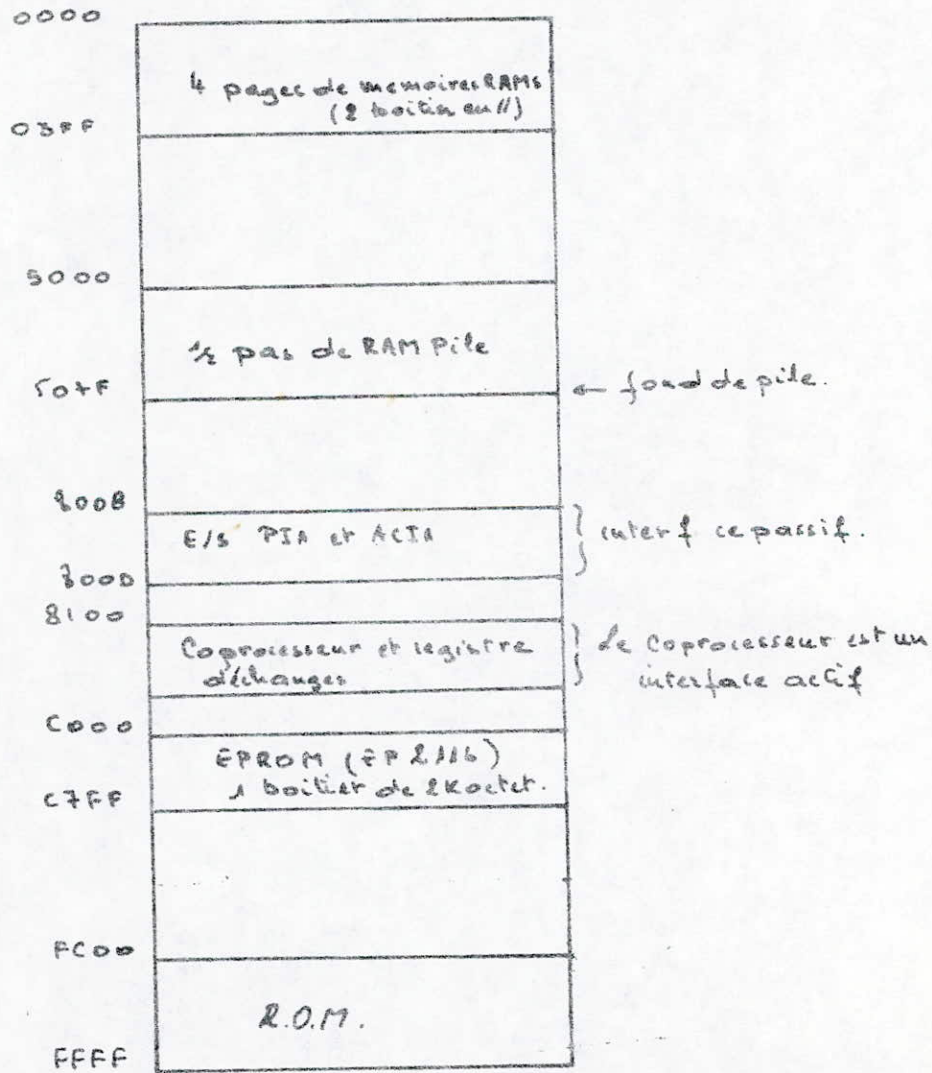
Les différentes commandes adressés au microprocesseur se font à partir d'un clavier TTY, donc nécessité d'un interface serie Asynchrone Type ACIA (EF6810)

Nous donnons ci-dessous le synoptique de la carte de gestion.



L'espace d'adressage alloué aux mémoires RAMS, ROMS, et EPROM et aux différents interfaces est représenté dans la figure ci-dessous et dans le tableau de la page suivante.

La RAM pile est située à l'adresse (5000 - 70FF) et comportera 1 boîtier de 128 bytes. On a utilisé, pour cela, des boîtiers (EP 6810) statiques



- Répartition des adresses dans la carte -



	VMA	R/W	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	ADR
ROM	1	1	1	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.	FC00-FFFF
EPROM	1	.	1	1	0	0	0	0	.	.	.	.	.	.	.	.	.	.	C000-C7FF
PIA	1	x	1	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	8008-800B
ACIA	1	x	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	x	800C-800D
MP E/S	1	x	1	0	0	0	0	0	.	1									
G.O.P	1	x	1	0	0	0	0	0	0	1	0	0	0	0	x	x	x	x	
R.S.F	1	x	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	8110
R.ECRI	1	⊙	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	8111
R.LECT	1	⊙	1	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	8112
RAMS	1	x	0	0	0	0	0	0	.	.	.	x	x	x	x	x	x	x	0000-03FF
RAN. FILE	1	x	0	1	0	1	0	0	0	0	0	.	.	.	.	.	.	.	5000-507F

x = Decodage à l'interieur des registres

. = Decodage des Boitiers

Decoupage de l'espace  
d'adressage

## I] Présentation du microprocesseur (Mc. 6800)

Le microprocesseur tient le rôle de l'unité centrale de traitement d'une carte appelée à gérer un processus quelconque. Ses broches d'adressage et de données déterminent la capacité mémoire de la carte utilisée. La taille du mot et ses broches de commande déterminent le type d'interface qui doit être utilisé.

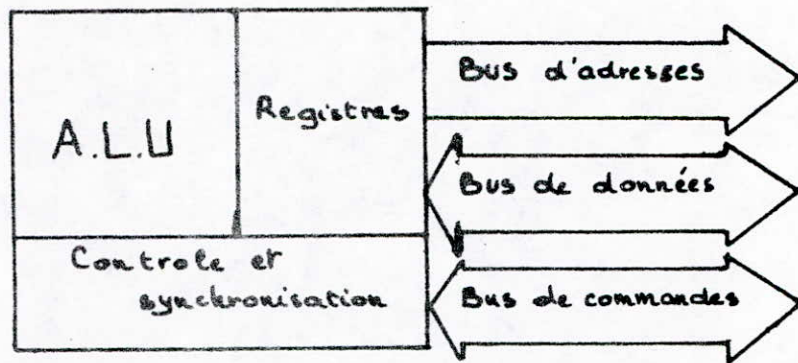
Le microprocesseur réalise un grand nombre de fonctions, comprenant :

- la fourniture des signaux de commande et de synchronisation pour tous les éléments de la carte de gestion.
- la recherche des instructions et des données en mémoire.
- le transfert des données vers les entrées-sorties ou des entrées-sorties vers la MPU.
- la réalisation des opérations arithmétiques et logiques demandées par les instructions.
- le décodage des instructions
- la gestion des signaux de commande générés par les E/S tel que Reset et Interrupt (IRQ)

Le microprocesseur contient tous les circuits logiques permettant de réaliser ces fonctions, mais une grande partie de la logique interne du microprocesseur est inaccessible. Par exemple, nous ne pouvons pas appliquer un signal externe en borne du microprocesseur pour



incrémenter le compteur ordinal (PC). En revanche les éléments du micro-  
 -processeur sont accessibles par software. C'est-à-dire que nous ne pouvons  
 intervenir dans les circuits interne de la M.P.U que par programmation.  
 C'est ce qui rend le microprocesseur si puissant et si souple.  
 Les principales zones fonctionnelles du microprocesseur peuvent se  
 resumer dans la figure ci-dessous.



Les liaisons avec les autres circuits se subdivisent essentiellement en  
 trois bus.

- Un bus de données de 8 bits. Par son intermédiaire les octets sont échangés avec les mémoires et les circuits interface.
- Un bus d'adresse de 16 bits. Il fournit les valeurs d'adresse de  $(0000)_{16}$  à  $(FFFF)_{16}$ , soit  $(65536)$  positions mémoires.
- Un bus de signaux de commande de 12 fils. C'est par ce bus que le microprocesseur reçoit les périodes d'1 horloge externe à deux phases ( $\Phi_1, \Phi_2$ ). Nous y trouvons également les commandes qui



qui permettent d'initialiser (RESET) ou d'arrêter le microprocesseur. A ses signaux s'ajoutent ceux qui proviennent des interfaces avec les périphériques. Les autres signaux servent à l'aiguillage correct de l'information sur les différents bus.

#### \* Registre interne -

La zone de travail en liaison avec les programmes se situe au niveau des 6 registres internes. Ce sont les seules registres mis à la disposition de l'utilisateur.

#### - Accumulateur A ou B.

C'est un registre qui intervient dans la plupart des opérations de l'A.L.U. Il est également le registre dans lequel les résultats sont placés après la plupart des opérations de l'A.L.U. Le mot doit tout d'abord être lu en mémoire et placé dans l'accumulateur. Donc il peut servir ainsi comme registre de stockage des données transmises en sortie ou comme registre recevant les données lues en entrée.

Le microprocesseur MC 6800 possède 2 accumulateurs A et B de 8 bits chacun.

#### - Registre d'état.

Appelé aussi registre condition, le registre état contient différents bits avec des significations différentes. Les bits sont des indicateurs et chaque indicateur permet d'indiquer l'état d'une condition particulière dans le microprocesseur. La valeur de certains indicateurs

peut être testée par programme et ainsi déterminer la séquence d'instruction à suivre.

1	1	H	I	N	Z	V	C
---	---	---	---	---	---	---	---

I = bit masque d'interruption

N = négatif

Z = zéro

V = dépassement en complément à deux.

C = Retenue du bit 7

H = Demi-retenue du bit 3

La valeur de Z indique toujours si l'instruction précédente a donnée ou non un résultat égal à zéro. Normalement un signal de commande du micro-processeur place l'indicateur Z à un, lorsque le résultat d'une instruction est zéro, et met Z à 0 dans le cas contraire. La valeur de C indique toujours si l'instruction précédente a produit un résultat qui dépasse la taille du mot.

L'indicateur I indique la prise en compte d'une interruption ou pas  
si I = 0 le microprocesseur répond à l'interruption

si I = 1 " ignore l'interruption et continue son programme principale.

- Registre d'index : c'est un registre de 16 bits.

De la même manière qu'un registre général qui est utilisé dans d'autres MPU, le registre d'index permet de sauvegarder et aussi de compter. En plus, il a une fonction spéciale qui est d'une grande

dans les programmes où des tableaux de données doivent être manipulés. Dans ce cas, l'indirect permet de déterminer les adresses des données que la M.P.U. saisit.

Cette opération est appelée adressage indirect et représente une forme particulière d'adressage disponible pour le programmeur.

### - Registre du pointeur de pile (16 bits)

Avant de définir la fonction de ce registre, nous devons définir avant tout la pile. La pile est une partie de la mémoire RAM réservée à l'enregistrement temporaire et à la récupération des informations tel que le contenu des registres internes du microprocesseur appelé contexte de programme. Cette zone mémoire constitue une pile en raison de son fonctionnement indiqué ci-dessous.

- 1-) chaque fois qu'un mot doit être enregistré dans cette zone mémoire, il est placé à l'adresse inférieure à l'adresse du mot enregistré précédemment dans la pile.
  - 2-) Les mots enregistrés dans la pile sont lus dans l'ordre inverse à celui dans lequel ils ont été enregistrés.
  - 3-) Une fois qu'un mot est lu, son emplacement dans la pile est disponible pour un nouvel enregistrement.  
Le type de pile est appelé souvent pile LIFO (LAST in, FIRST out)
-



### Mode d'adressage -

Lorsqu'une instruction fait référence à un opérande, elle peut repérer celui-ci en mémoire de différentes façons appelés : mode d'adressage. Les informations (instructions et données) sont stockées dans des cellules des mémoires R.A.M ou R.O.M, chaque cellule est caractérisée par son adresse. Pour atteindre une cellule précise, on dispose de plusieurs manières "mode d'adressage" que l'on examinera un à un.

Ainsi par un choix judicieux, il est possible d'améliorer le programme en réduisant :

- la longueur du programme
- la capacité d'exécution
- le temps d'exécution.

Ces qualités recherchées en micro-informatique confèrent une très grande souplesse à la programmation.

Les modes d'adressages sont les suivants :

- adressage direct.
- " étendu.
- " immédiat.
- " indirect.
- " implicite.
- " inhérent.

#### 1) Adressage direct :

L'instruction contient au moins deux octets ; l'adresse de l'opérande est contenu dans le 2<sup>e</sup> octet de l'instruction ; le 1<sup>er</sup> octet est réservé à l'opération. C'est le mode d'adressage le plus utilisé et le plus simple ; il consiste à utiliser les adresses fournies sur le bus adresse pour

accéder directement à des données contenues dans les positions correspondantes de la mémoire.

- Exemple: `LDA #10` : charger le contenu de l'adresse  $(10)_{16}$  dans l'accumulateur A. Le compteur ordinal nous fournit l'adresse de l'instruction, codée par exemple sur 2 cellules mémoire (2 octets). Mais ce mode d'adressage n'est utilisé que pour la page de base.

2°) Adressage étendu: l'instruction contient 3 octets; l'adresse est contenue dans les 2<sup>es</sup> octets (poids fort) et 3<sup>es</sup> octet (poids faible) de l'instruction. Ce mode d'adressage permet le balayage de toute la mémoire de 0000 à FFFF. Pour lire une adresse sur 16 bits on passera 2 tours de lecture en mémoire.

- Exemple: `LDA #8A03` : supposant que le compteur ordinal indique  $PC = 8000$ , le compteur de programme pointe  $PC = 8000$  ou il ou lit  $(1018)$  puis  $(PC+1) = 8001$  ou il ou lit  $(18A)$  et enfin  $(PC+2)$  ou il ou lit  $(03)_{16}$  et à l'adresse  $8A03$ . On pourra transférer le contenu de cette adresse dans l'accumulateur.

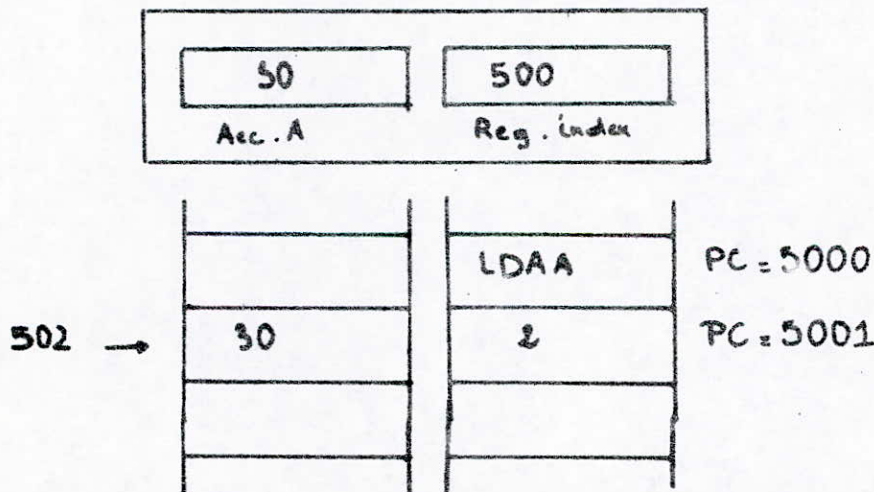
3°) Adressage immédiat: l'opérande est contenu dans le 2<sup>es</sup> et 3<sup>es</sup> octet de l'instruction, selon que l'on s'adresse aux accumulateurs ou au registre. Le champ adresse de l'opérande contient les valeurs de l'opérande. Il est appelé opérande immédiat; donc ce mode d'adressage ne permet de traiter que les opérandes dont les valeurs sont des constantes dans le programme.

- Exemple: `ADDA #7F` cette valeur est additionnée à l'accumulateur

4°) Adressage indexé: L'instruction est composée de l'opération sur le 1<sup>er</sup> octet et de l'adresse sur le 2<sup>e</sup> octet. On fait intervenir le registre d'index de 16 bits; l'adresse effective de l'instruction est obtenue en ajoutant le contenu du registre d'index à la valeur du 2<sup>e</sup> octet de l'instruction.

Le mode d'adressage est particulièrement adapté au traitement de tableau de données par incrémentation ou décrémentation des instructions particulières.

- Exemple: LDAA 2, X : charger l'accumulateur A du contenu de l'adresse mémoire.  
Adresse Mémoire = 2 + Contenu du registre d'index



5°) Adressage Implicite: L'instruction est composée ainsi de 2 ou 3 octets; l'opérande est indiqué par le code opération de l'instruction. Dans l'adressage implicite, on s'adresse implicitement à un registre donné.

- Exemple: PSHA: sauvegarder le contenu de l'accumulateur implicitement dans la pile.



6°) Adressage relatif; le mode d'adressage relatif n'est utilisable que pour les instructions de branchement conditionnel, de branchement inconditionnel (BRA) et de branchement à un sous-programme (BSR). Les instructions ne peuvent pas être utilisées avec un autre mode d'adressage.

Pour que le mode d'adressage relatif soit valide, une règle, limitant la distance, dans le programme objet, entre l'instruction de branchement et la destination du branchement doit être comprise entre les limites suivantes :

$$(Pc+2) - 128 < D \leq (Pc+2) + 127$$

---

## I Description des différents éléments de la carte.

### a) Les mémoires à lecture et écriture (RAM)

Les RAMS sont utilisées pour l'enregistrement temporaire des programmes et des données. Le contenu de ces mémoires change sans arrêt lors de l'exécution d'un programme.

Le principal inconvénient des mémoires RAMS et leur volatilité.

Cela signifie que, lorsque l'alimentation est coupée, la mémoire RAM toutes les informations.

S'agissant de la carte de gestion, les mémoires statiques (EEPROM) de 1 kilo-mots de 4 bits sont utilisées ainsi qu'une mémoire statique de 128 bytes pour la RAM pile.

Avant de ~~decrire~~ les mémoires, rappelons la constitution de mémoire statique

#### a.1) Généralité sur les mémoires statiques:

Les mémoires statiques sont constituées d'une matrice contenant autant de bascules que de bits et chaque bascule mémorise un bit d'information. L'information reste bloquée tant que l'alimentation lui est appliquée.

On peut les relire quand on veut.

- Avantages et inconvénients.

Tant que l'alimentation est présente, la RAM statique conserve l'information et ne nécessite pas de rafraîchissement. L'alimentation étant unique (+5V)



Les inconvénients d'utilisation de ces RAMS sont nombreux, entre autre l'échauffement est très important, nous donnons à la page suivante le schéma de la structure d'une telle cellule mémoire.

i) La RAMS STATIQUE (EF2114)

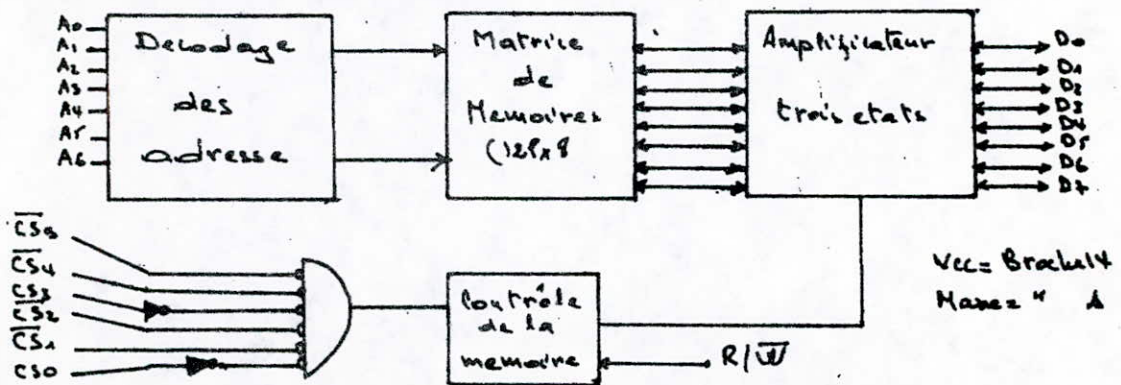
Ce sont des mémoires statique de 1024 mots de 4 bits à accès aléatoire et réalisées en technologie canal N grille silicium, haute densité a grande fiabilité. Les mémoires sont directement compatible TTL et DTL.

Les caractéristiques de ce circuit sont :

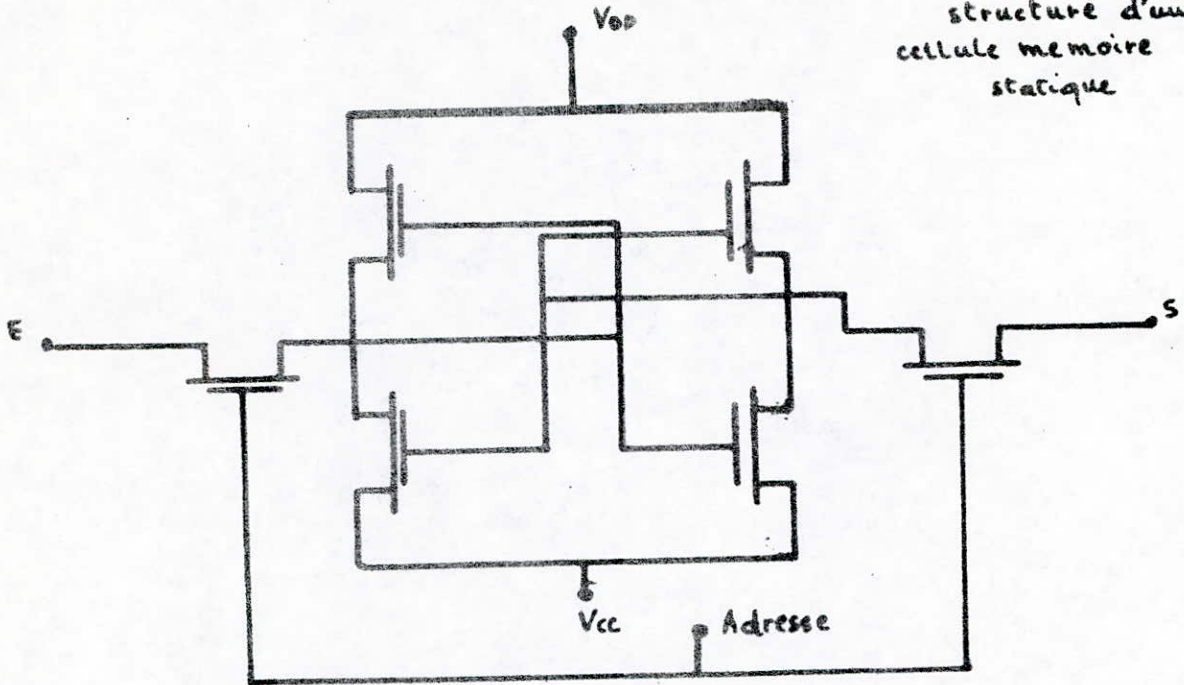
- 1024 mots de 4 bits
- brochage standard 18 broches
- Alimentation unique +5V
- E/S de données bidirectionnelles

ii) La RAM statique (EF 6810)

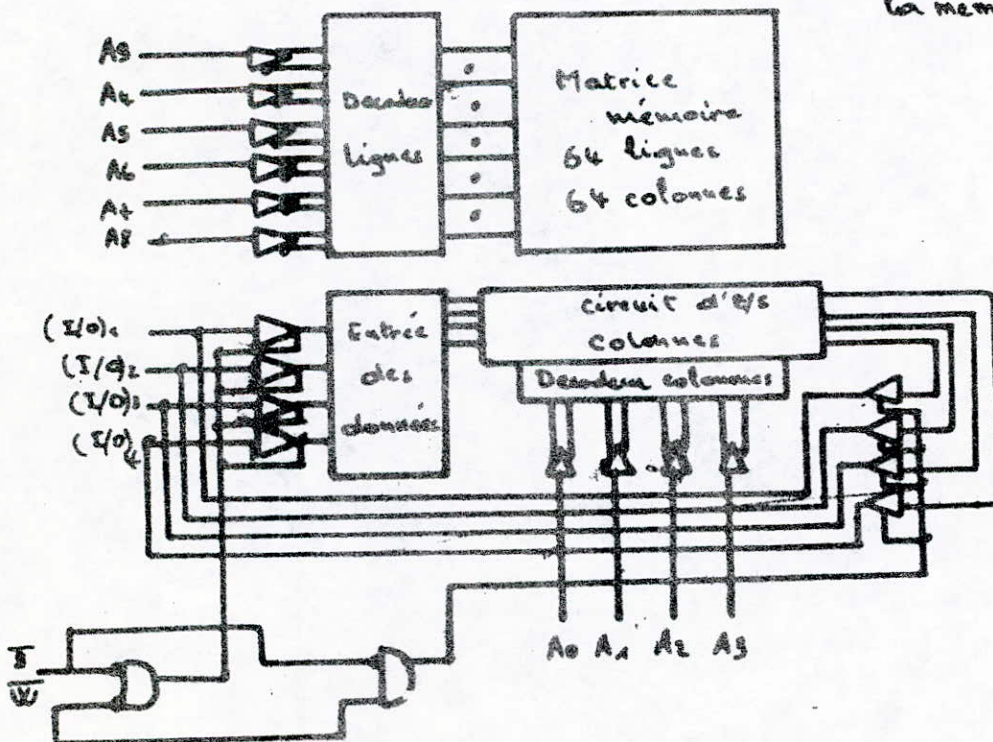
Organisées en mots de 8 bits, sa capacité est de 128 bytes, cette mémoire comporte 6 entrées de sélection. nous donnons ci-dessous le diagramme fonctionnel.



structure d'une  
cellule memoire  
statique



- schema fonctionnelle de  
la memoire (EP 2114) -



b) Mémoire ROM\_EPROM

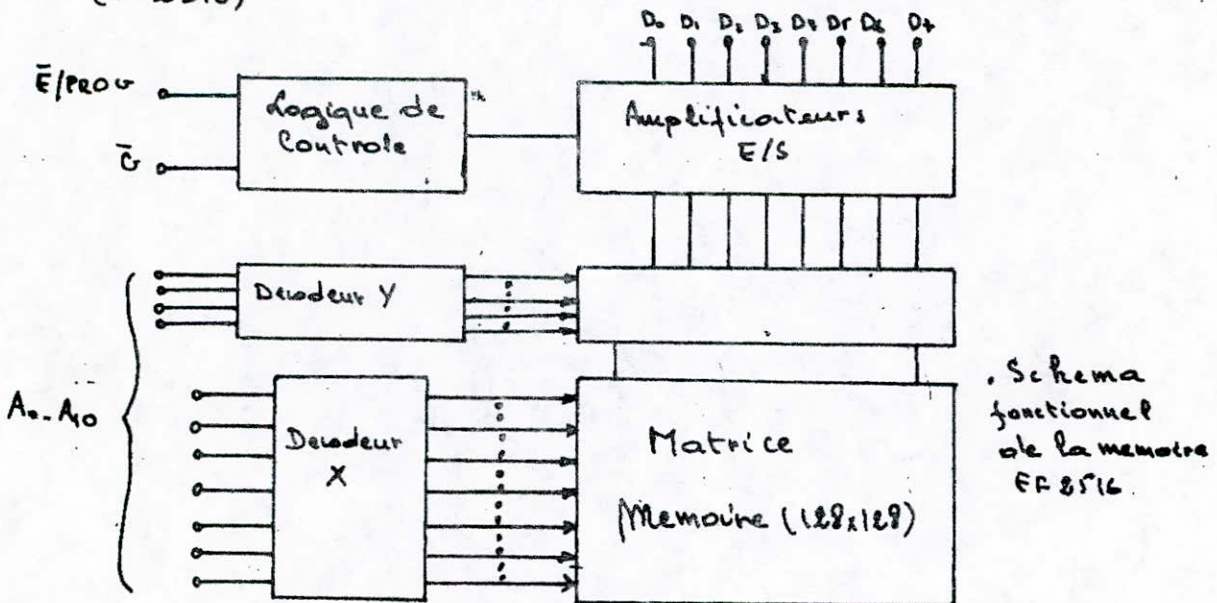
Mentionnons ici l'existence de trois sortes de mémoires mortes : ROM, PROM, REEPROM.

ROM (Read only memory), mémoire à lecture seule dont le contenu est défini par le dessin du masque de fabrication.

PROM (Programmable ROM), mémoire morte dont le contenu est programmé électriquement par l'utilisateur. Toute programmation est définitive.

REEPROM (Reprogrammable ROM). Mémoire morte programmable électriquement. Elle présente les avantages d'un PROM tout en étant réutilisable.

Par un moyen physique (rayons ultra-violet, impulsions électriques), il est possible d'en effacer le contenu que l'on peut programmer à nouveau. La carte de gestion comporte 2 EPROMs de 2 Kilo bytes (EF2516)



- Schema fonctionnel -



### c) Interface PIA et ACTIA.

#### C-1) Interface PIA:

Le circuit d'interface PIA a été conçu pour des applications générales, il assure l'interfaçage entre le microprocesseur et les périphériques qui ne sont pas synchronisés par l'horloge de cette carte. C'est un circuit programmable, destiné pour des transmissions et réception parallèle.

- \* Synoptique ou description interne du PIA : Le PIA se compose de:
  - 2 registres de données de sorties ORA et ORB ayant 8 bits chacun
  - 4 registres de contrôle ; deux de ces registres sont affectés au canal A et les deux autres au canal B.
  - 2 registres de sens de transfert de données le DDRA et le DDRB, associés au canal A et B, permettent de définir le sens de transfert de données. A chaque bit de ces registres est associé 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.
  - Le registre de contrôle et d'état CRA et CRB 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 sont lus seulement et sont modifiés par les interruptions externes sur les lignes de contrôle CA1, CA2, CB1, CB2.
- \* Les différents signaux d'interface entre le PIA et le microprocesseur.  
Le PIA dialogue avec le MPU par :
  - Le bus de données bidirectionnel (D<sub>0</sub>-D<sub>7</sub>)
  - Le bus d'adresse comprenant :
    - les signaux de sélection du boîtier (CS<sub>0</sub>, CS<sub>1</sub>, CS<sub>2</sub>)
    - de 2 signaux de " 0/1 " registre interne (RS<sub>0</sub>, RS<sub>1</sub>)
  - un signal de synchronisation, appliqué sur la broche E, ce signal est généralement  $\Phi_2$
  - Un signal de lecture/écriture qui permet de distinguer une opération d'entrée de donnée d'une opération de sortie.

CS2	CS1	CS0	RS1	RS0	Q <sub>L</sub>	R/W	TYPE D'OPERATION
0	1	1	0	0	0	0	Sortie : Bus données → REG. DBRA
0	1	1	0	0	1	0	Sortie : Bus données → CANAL A
0	1	1	0	0	1	1	Entrée : CANAL A → Bus données
0	1	1	0	1	x	0	Sortie : Bus données → REG. CRA
0	1	1	0	1	x	1	Entrée : REG. CRA → Bus données
0	1	1	1	0	0	0	Sortie : Bus données → REG. ORB
0	1	1	1	0	1	0	Sortie : Bus données → CANAL B
0	1	1	1	0	1	1	Entrée : CANAL B → Bus données
0	1	1	1	1	x	0	Sortie : Bus données → REG. CRB
0	1	1	1	1	x	1	Entrée : REG. CRB → Bus données

- Tableau des opérations de Base du PIA.

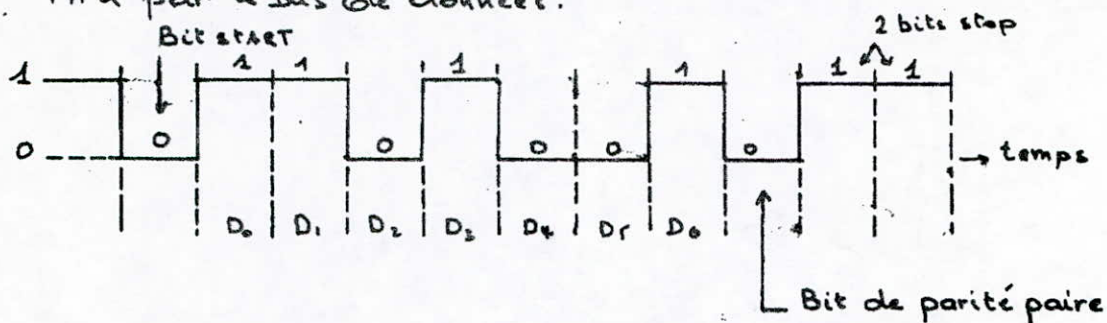
\* Programmation du PIA : Le PIA est un circuit dont la programmation est assez complexe puisqu'il possède quatre registres de contrôle, donc il lui faut plusieurs mots pour définir sa fonction. Par contre il possède une très grande souplesse d'utilisation puisque ces lignes de données sont programmable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

### c.2) Interface série "ACIA"

Lorsqu'un ordinateur ou une unité quelconque doit communiquer avec une unité d'entrée - sortie du type série, un moyen commun de réaliser ceci correspond à la transmission série asynchrone. Elle est asynchrone car l'unité de transmission peut envoyer les informations vers l'unité receptrice à n'importe à n'importe quel instant sans synchronisation avec le récepteur. Ce système réalise deux opérations principales :



- 1°) il prend une information sur 8 bits sous format parallèle provenant du bus de données et la convertit en une information série qui sera envoyée vers l'unité série.
- 2°) il prend un signal série provenant de l'unité série et le convertit en une information parallèle de 8 bits qui sera transféré vers le MPU par le bus de données.



Le circuit ACIA "6850" a principalement 6 fonctions regroupées comme suit :

- 1°) Sélection du boîtier et Commande de lecture et d'écriture
- 2°) Registre tampon du bus de données.
- 3°) Registre de données de l'émetteur (TXDR) et registre à décalage de l'émetteur (TSR)
- 4°) Registre de données du récepteur (RXDR) et registre à décalage du récepteur (RSR)
- 5°) Registre d'état
- 6°) Registre de commande

OPERATION	CS0	CS1	$\overline{CS2}$	RS	R/ $\overline{W}$	VALIDATION
Lect. Reg. ETAT	1	1	0	0	1	1
Ecri. Reg. COMM	1	1	0	0	0	1
Lect. RXDR	1	1	0	1	1	1
ECRI. TXDR	1	1	0	1	0	1

- Sélection de boîtier -

Le microprocesseur peut fonctionner avec de nombreux unités d'E/S. Les trois entrées de sélection (CS0, CS1,  $\overline{CS2}$ ) sont des lignes utilisées pour adresser l'ACIA 6850 quand CS0 et CS1 sont au niveau haut et  $\overline{CS2}$  au niveau bas. (R/W et RS pour sélectionner les registre interne de l'ACIA.



- 18) Lecture du registre d'état: les informations concernant l'état des registres TxDR, RxDR et des circuits de detection d'erreur sont chargés dans le registre d'état. Le MPU obtient ces informations par lecture du registre d'état.
- 28) Lecture du registre RxDR: Quand l'ACIA a reçu une donnée provenant de l'unité d'entrée, le MPU peut acquérir cette donnée en réalisant une opération de lecture sur le registre RxDR.
- 38) Ecriture du registre de Commande: Le registre de commande de l'ACIA est un registre sur 8 bits qui ne peut être qu'écrit et dont le contenu détermine notamment la division de la fréquence d'horloge, la taille du mot, le nombre de bits stop et la parité. Ces caractéristiques sont contrôlées par le MPU qui écrit dans le registre de commande un mot de huit bits.
- 48) Ecriture dans le registre TxDR: Quand le MPU veut envoyer une information vers l'unité de sortie, il réalise cette opération en écrivant la donnée dans le registre TxDR. L'ACIA transmet alors l'information en série vers l'unité de sortie.

- Programmation de l'ACIA.

L'ACIA sort de la mise à zéro par la programmation des bits 0 ou 1 du registre de Commande à l'état logique 1, - 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 fonctionnel de l'ACIA -

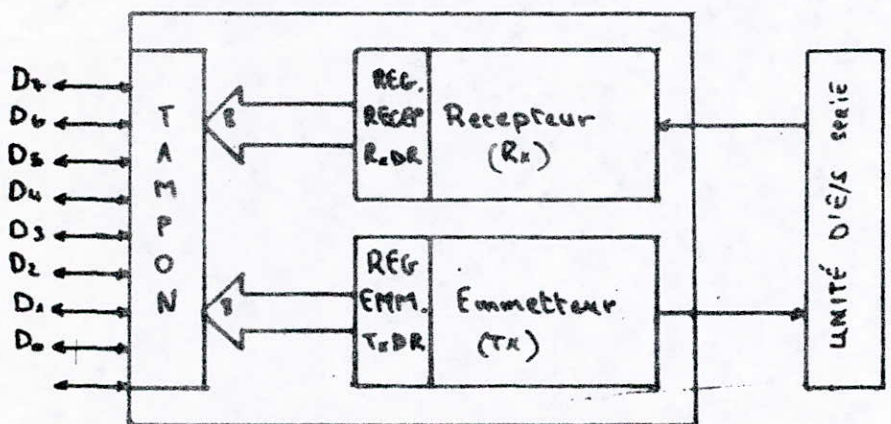


schéma synoptique ou fonctionnel de l'ACIA.









- Le bit 5 à l'état haut autorise l'écriture même en dehors de la fenêtre visualisée.
- Les bits 4, 5, 6 représentent à l'état haut l'autorisation des interruptions émise dans le registre CTRL1.
- Le bit 7 est un ou logique des 3 bits précédent (4, 5, 6.)

#### - Registre CTRL2

C'est un registre de 4 bits à lecture-écriture. Les bits 0 et 1 déterminent le type de trait des vecteurs (continu, pointillé, tireté, ou mixte). Les bits 3 et 2 déterminent le mode d'écriture des caractères, verticale, en italique ou non, horizontale en italique ou non.

#### - Registre EBIZE

C'est un registre de 8 bits à lecture et écriture. Les 4 bits de poids forts déterminent le facteur d'échelle  $P_{en X}$  du caractère écrit, les 4 bits de poids faible déterminent le facteur d'échelle  $P_{en Y}$  du même caractère. Donc ce registre détermine la taille du caractère.

#### - Registre XLP et YLP

Ces registres sont à lecture seule, respectivement de 4 bits et de 8 bits, on retrouve après une action du photostyle l'adresse de visualisation échantillonnée par le premier front montant sur l'entrée (PC K).

### 1.3) Description des signaux, sortant des broches du Coprocesseur.

- + Broche d'alimentation : La broche Vcc correspond à la tension unique +5V, nécessaire au circuit. La broche Vss correspond à la masse.
- + Broche de contrôle de la mémoire d'image : La broche DIN fournit la donnée 0 ou 1 correspondant au point à écrire en mémoire d'image. À l'état haut, elle équivaut à 1 point éteint sur l'écran. Dans une application monochrome, elle est donc la donnée d'entrée du seul plan mémoire d'image. Pour une application avec plusieurs plans ce qui est notre cas, on doit intercaler entre le bus microprocesseur et la mémoire d'image 1 registre appelé "registre écriture". Le signal  $\overline{WE}$  actif à l'état bas est le signal d'écriture dans la mémoire d'image.

la broche MPREE signale que la mémoire d'image n'est pas utilisée elle permet dans ce cas d'accéder à la case mémoire pointée par X,Y de façon externe.

Broches d'adressage de la mémoire d'image :

Les broches MSL0 à MSL3 servent à la sélection des boîtiers mémoires les broches DAD0 à DAD6 fournissent l'adresse de visualisation, elles permettent le rafraîchissement automatique des mémoires dynamiques. Le signal ALL, actif à l'état bas autorise l'accès à tous les boîtiers mémoires pour l'effacement, la visualisation ou le rafraîchissement. A l'état haut, le processeur graphique adresse un seul pixel.

Broches de synchronisation et de suppression vidéo :

La broche sync fournit le signal de synchronisation composite (lignes ou trames) pour le moniteur vidéo.

Le signal BLK (Blanking) est à l'état haut en dehors de la fenêtre de visualisation, il permet d'envoyer le contenu de la mémoire d'image pendant la période d'affichage. En effet la fenêtre de visualisation ne commence pas à la première ligne ni au début de chaque ligne. De même elle ne se termine pas à la fin de chaque ligne ni à la dernière ligne ; le signal VB (Vertical blanking) indique le retour trame.

Broches d'horloge et de paramétrage des fonctionnements :

Sur la broche CK doit être appliquée une horloge qui réquencera le fonctionnement interne du circuit. Par exemple le signal synch du circuit EF 9365 sera aux normes ccir 625 lignes 50Hz, avec une fréquence de 1,750 MHz.

La broche FMAT doit être connectée à VCC pour que le circuit EF 9365 fonctionnera en mode balayage entrelacé (512 lignes). Un signal haut appliqué sur l'entrée WA inhibe la visualisation et le rafraîchissement des mémoires seule l'écriture est possible.

broches de contrôle du photostyle :

Le signal white, actif à l'état bas sert à envoyer un flash à l'écran pour récupérer les coordonnées de la zone pointée par le photostyle.



L'entrée LPCK est connectée directement au photostyle. Elle sert de lien entre le circuit graphique et l'interrupteur du photostyle.

Broches du microprocesseur :

La broche IRQ de demande d'interruption l'entrée  $\bar{E}$  de synchronisation entre le microprocesseur et ses périphériques et l'entrée  $R/\bar{W}$  de lecture-écriture pour les signaux classiques du bus de contrôle issu du microprocesseur.

4 lignes d'adresses de A<sub>0</sub> à A<sub>3</sub> sont nécessaires pour adresser les 11 registres internes du GPP.

Les 8 lignes du bus de données D<sub>0</sub> à D<sub>7</sub> sont présentes pour lire ou écrire les données et les commandes dans les 11 registres.

## 2. Interface microprocesseur - processeur graphique :

Le processeur graphique EF 9365 appelée GPP (graphic display processor) est programmable par l'intermédiaire de ses 11 registres internes. En effet le processeur occupe 16 adresses consécutives dont seulement 11 sont utilisées. Bien entendu ces 11 registres peuvent être modifiés par le circuit lui-même quand il effectue une commande.

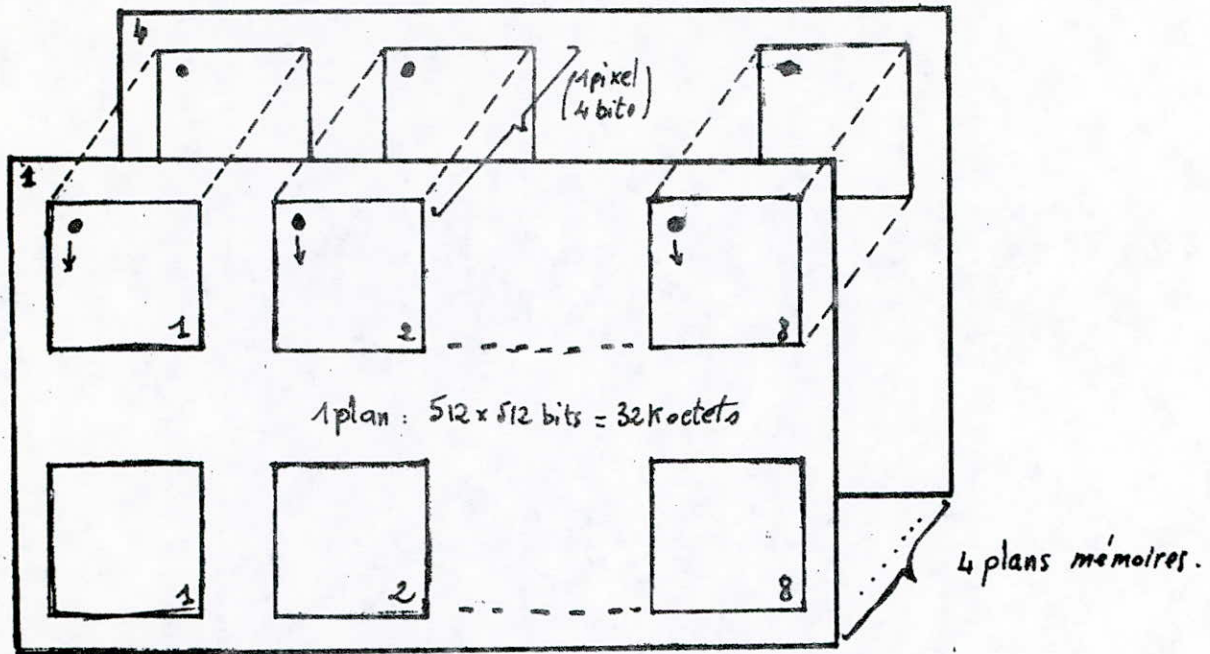
La mémoire d'image gérée par le processeur graphique n'est pas directement adressable par le microprocesseur, seuls 16 registres sont vus du côté du microprocesseur en plus des registres temporaires pour l'accès externe, laissant à celui-ci son espace d'adressage quasi intact. Vu du côté de la sortie et de l'affichage à l'écran, le processeur graphique gère sa mémoire comme le ferait un microprocesseur.

## 3. Gestion automatique de la mémoire d'image :

Le processeur graphique contient une logique de rafraîchissement automatique des différents plans mémoires d'images constitués par les RAM dynamiques. Les baîtes mémoires RAM dynamiques sont de 16 K bits (EF 4116). Le processeur adresse simultanément tous les plans mémoires pour afficher l'image.

En effet chaque ligne à afficher est ici constituée de 512 pixels et peut être découpée en 64 troncous de 8 pixels adjacents. Le pixel est l'élément unitaire affichable et est codé sur 4 bits. La figure suivante nous montre l'organisation de la mémoire d'image.



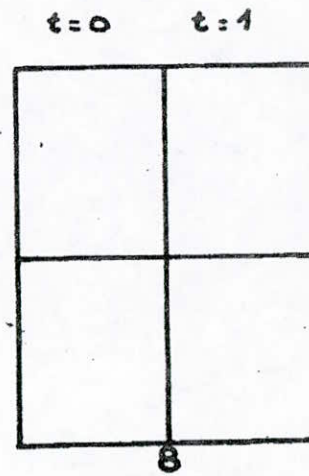
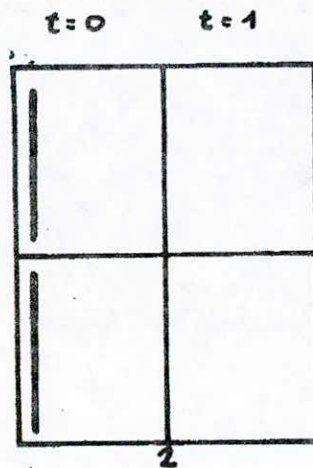
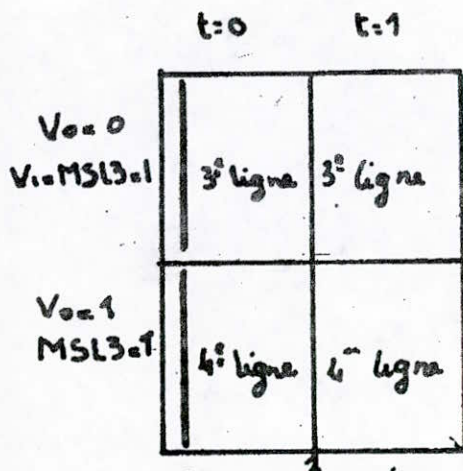
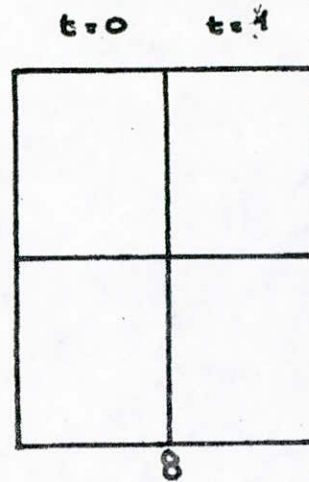
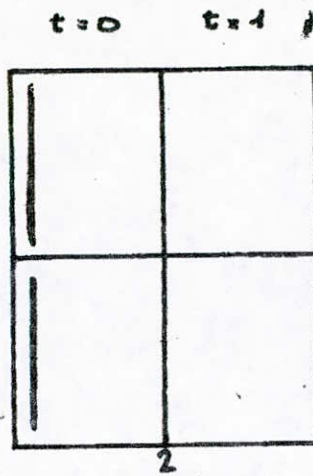
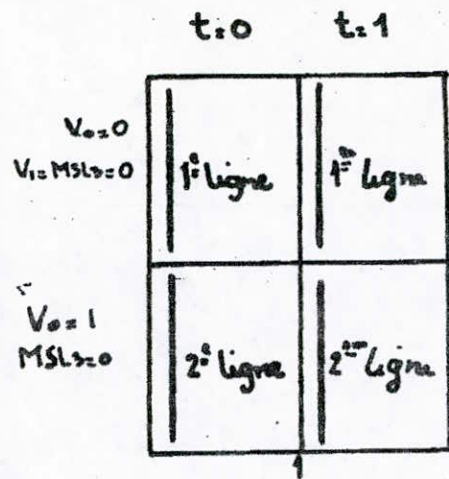


Dans la mémoire d'image, les 8 pixels sont répartis dans les 8 boîtiers mémoires et ceci dans chaque plan mémoire. La mémoire totale contient 512 (nbre de lignes) x 64 (nbre de troncous de 8 bits) x 4 (nbre de plans mémoires) ce qui nous donne une capacité de (128 kilo octets) chaque accès charge simultanément les 4 registres à décalage de 8 bits afin qu'ils soient serialisés.

#### 4. Registres d'échanges avec le microprocesseur :

Il y a au nombre de 3 : 1 registre de sélection feuille.  
1 registre d'écriture dans la mémoire d'image.  
1 " de lecture.

Ces registres permettent au processeur d'accéder à la mémoire d'image en sélectionnant l'une des 2 feuilles cad la feuille de stockage ou de traitement. Mais pour que ces différents registres acceptent ou envoient l'information vers le processeur il faudra que la sortie MFREE du coprocesseur graphique soit à l'état bas et cela à l'aide de la commande (OF)16 du registre de commande CMD



t=0 → trame impaire.

t=1 → trame paire

## ORGANISATION - PLAN - MEMOIRE



#### 4.1. Registre selection feuille :

Ce circuit est composé d'un registre pour stocker le code de selection (Registre de 4 bits) et de 4 buffers à trois états. Ce circuit permet de sélectionner l'une des 2 feuilles de stockage ou de traitement et cela en lecture ou en écriture.

#### 4.2. Registre écriture :

C'est un registre de 8 bits (2AM25LS2518). Ce circuit permet l'écriture dans la mémoire d'image voir schéma.

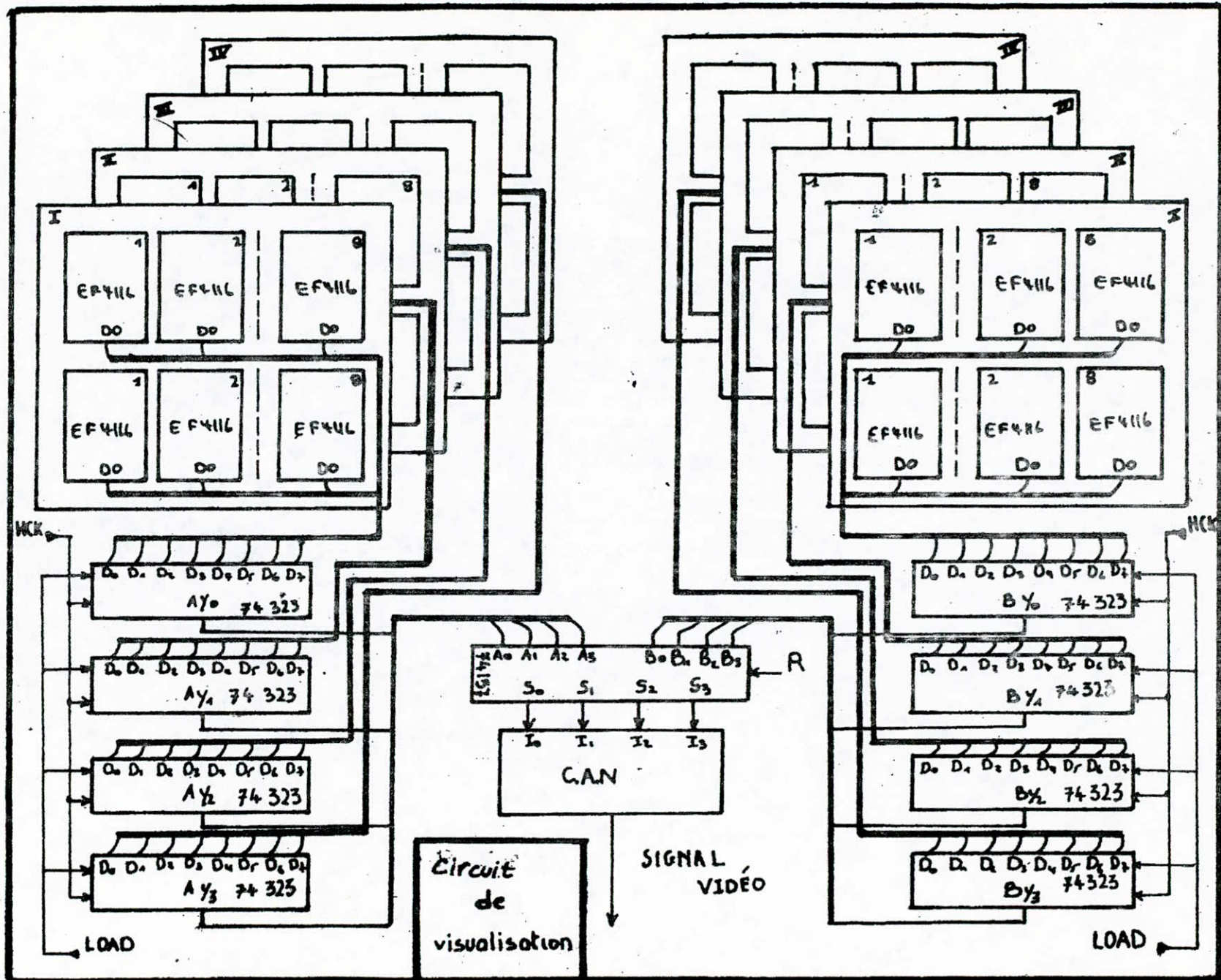
#### 4.3. Registre de lecture :

C'est un registre de 4 bits (74LS245), l'entrée DIR est mise à l'état bas ce qui entraîne le transfert de la donnée du multiplexeur vers le microprocesseur. (voir schéma)

#### 4.4. Registre à décalage :

Pour visualiser des points sur le moniteur T.V à une fréquence de 14 MHz on intercale entre la mémoire d'image et le convertisseur numérique analogique de résolution (4 bits) des registres à décalage de 8 bits qui reçoivent lors de chaque lecture, les données qui correspondent à une même adresse dans plusieurs bannières mémoires. Ces données sont chargées en parallèle dans le registre à décalage à une fréquence de 1,75 MHz tirée par le séquenceur (LOAD) puis pendant le temps qui sépare 2 chargements (8 pixels à chaque chargement), le registre sérialisera les données avec une période de 71 ns imposée par l'horloge du séquenceur à une fréquence de 14 MHz. (HCK) pour visualiser l'une des 2 feuilles de 4 plans mémoires chacune. On intercale 1 multiplexeur 8/4 qui choisira par programmation la quelle des 2 feuilles (de stockage ou de traitement) doit être visualisée. Les sorties du multiplexeur attaquent directement le CAN. Pendant la visualisation le signal AU doit être égal à "zéro".





h.4



## 5. Le pequenceur haute vitesse :

Ce circuit g nere les signaux d'horloge :  $CK$ ,  $\overline{STR}$ ,  $\overline{RAS}$ ,  $\overline{CAS}$   $HCK$  et  $LOAD$  pour respectivement le coprocesseur graphique, le 74LS374 compos  de 8 bascules type D, les m moires RAM composant la m moire d'image enfin  $HCK$  et  $LOAD$  servent respectivement au d calage des donn es   l'int rieur du registre   d calage.

La dur e du point  l mentaire est de 71 ns sur l' cran. Pendant le chargement des donn es dans le registre   d calage il faudra que la fr quence de  $HCK$  soit 8 fois sup rieure    $LOAD$ .

Pour le fonctionnement du circuit on a le 74LS04 mont  en oscillateur   quartz de 14MHz, il d livrera le signal  $HCK$ . L'oscillateur est suivi d'un compteur diviseur par 8 par 4 et 2. Les 3 signaux obtenus sont adress s   1 PROM bipolaire de 256 bits qui donnera   sa sortie 4 signaux  $CK$ ,  $\overline{RAS}$ ,  $\overline{CAS}$ ,  $\overline{STR}$  de m me fr quence (1,75MHz) et d cal s entre eux. Pour  liminer le retard ainsi cr e   la sortie de la Prom, ces 4 signaux sont alors envoy  vers une t tre bascul e (Latch) qui  liminera ainsi ce retard.

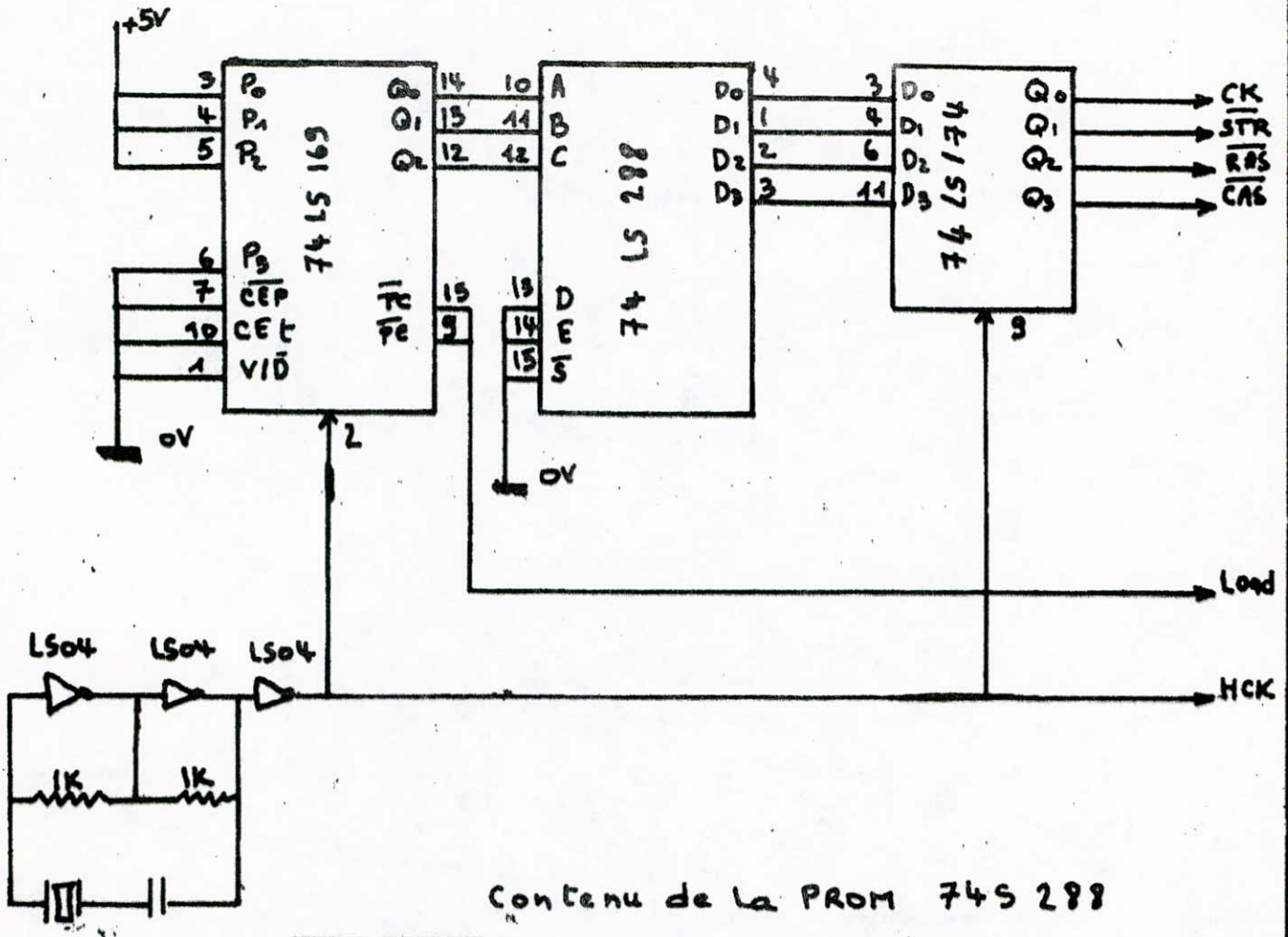
## 6. La RAM dynamique EF 4116 BC 15

1) Pr sentation : Le circuit EF 4116 est une m moire dynamique   acc s al atoire de 16Kbits. Fabriqu e dans une technologie MOS canal N et n cessitant 4 types d'alimentation ( $V_{bb}$ ,  $V_{cc}$ ,  $V_{dd}$ ,  $V_{ss}$ ) - Les 128 lignes d'adresses doivent avoir chacune d'elles un cycle de rafraichissement toutes les 2ms. Les principales caract ristiques sont :

- la grande souplesse d'utilisation par cycle lecture-modification  criture
- Le rafraichissement par la ligne  $\overline{RAS}$  seulement
- boitier standard de 16 broches
- organis e en 16K mots de 1bits.
- enti rement compatible TTL
- sorties 3  tats.
- faible puissance consomm e.

Le sch ma fonctionnel sera donn  en annexe.

Sequenceur



Contenu de la PROM 74S 288

ADRESSE			Contenu				
a	b	c	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	DECIM
0	0	0	1	1	0	0	0
0	0	1	0	0	0	0	1
0	1	0	0	0	0	0	2
0	1	1	0	0	0	0	3
1	0	0	1	0	0	1	4
1	0	1	1	0	1	1	5
1	1	0	1	0	1	1	6
1	1	1	1	1	0	0	7



## organisation interne de la mémoire RAM

Le boîtier présente 2 entrées de sélection RAS et CAS représentant la sélection d'adresse de rangée et la sélection d'adresse colonne. Il est organisé en 16K mots de 16bit. Pour adresser 16bit de la mémoire il faudrait 16 lignes d'adresse ( $2^{14} = 16K$ ). En fait, pour limiter le nombre de broches, l'adresse est multiplexée : il y a 7 lignes d'adresses A0 - A6. La mémoire sera présentée par une matrice de 128 lignes (row)  $2^7$ , par 128 colonnes « column », 16bit étant adressable à l'intersection de chaque ligne et colonne.

Pour lire ou écrire un point de la mémoire, on présentera successivement l'adresse ligne, puis l'adresse colonne, ces deux adresses étant respectivement validées par un front sur 2 lignes de « strobe » RAS (« Row Address Strobe ») et CAS (« Column Address Strobe ») un certain temps après le front de validation de CAS, la mémoire étant adressée, elle prend en compte la donnée présentée sur ligne d'entrée DIN (si la ligne R/W = 0) ou libère la sortie Dout de l'état intermédiaire (3<sup>e</sup> état) en envoyant la donnée vers les multiplexeurs 8/1 (74LS151) si R/W = 1. On a 2 modes de lecture ou d'écriture : mode aléatoire en mode séquentiel ; dans le premier mode, on adresse les points successifs en donnant pour chaque point son adresse ligne puis son adresse colonne. Dans le mode séquentiel, on adresse les points successifs d'une même ligne en n'adressant que les colonnes. Ce mode rapide est effectué pendant la visualisation. Le rafraîchissement est effectué par l'intermédiaire du signal RAS qui actif à l'état bas.

Les signaux RAS et CAS sont délivrés par le séquenceur haute vitesse à la fréquence de 1,75 MHz.

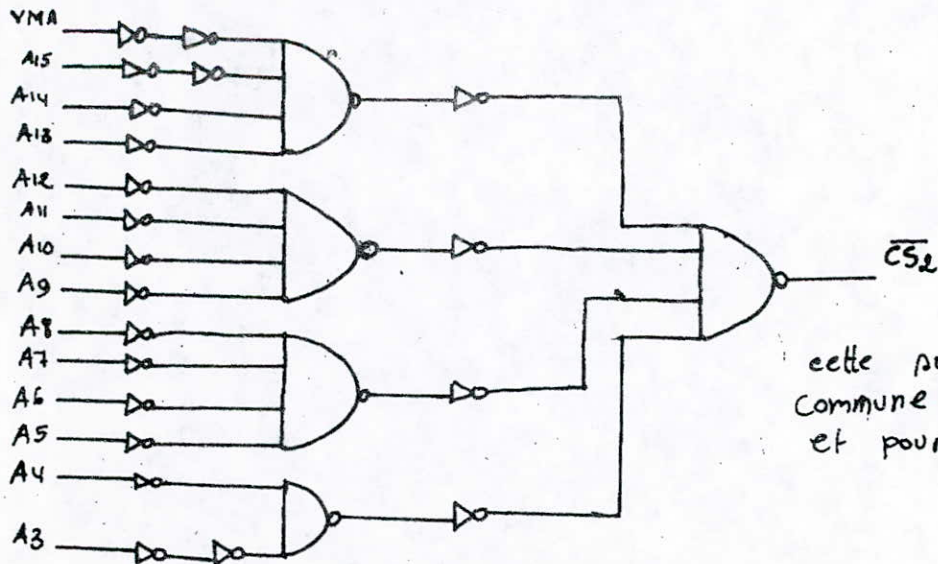
## 7. Décodage d'adresses :

### 7.1: Liaison microprocesseur - circuits coupleurs (PIA, ACIA)

d'après le tableau établi à la page 20 on voit que le PIA s'étend sur l'adresse 8008 à 800B. Le microprocesseur considère le PIA comme 4 positions mémoire, il accède.

aux registres internes du PIA par l'intermédiaire des lignes sélection de boîtier ( $CS_0 - CS_1 - \overline{CS_2}$ ) et sélection du registre ( $RS_0, RS_1$ ) la figure suivante résume la méthode d'adressage des registres internes du PIA.

BUS d'adresse		MPU	A15 ..... A2			A1	A0			
		logique de décodage								
		PIA	$CS_0$	$CS_1$	$\overline{CS_2}$	$RS_1$	$RS_0$	$CRA_2$	$CRB_2$	Adresse
REGISTRES	A	CRA	1	1	0	0	1	-	-	8009
		DDRA	1	1	0	0	0	0	-	8008
		ORA	1	1	0	0	0	1	-	8008
	B	CRB	1	1	0	1	1	-	-	800B
		DDRB	1	1	0	1	0	-	0	800A
		ORB	1	1	0	1	0	-	1	800A



cette sortie est commune pour l'ADDA et pour le PIA.

schéma de décodage.



Selection des registres internes

Le dialogue entre le microprocesseur et la peripherie s'effectue par l'intermediaire des registres internes de l'ACIA. Pour acceder a ces registres, on dispose de trois lignes de selection de boitier ( $CS_0, CS_1, \overline{CS}_2$ ) et d'une ligne de selection du registre ( $RS$ ) (le tableau suivant resume la methode d'adressage des registres internes de l'ACIA).

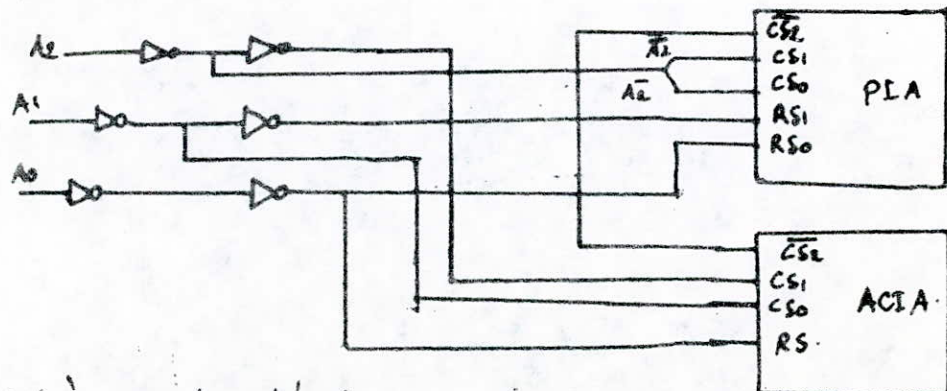
BUS	MPU	A15 - - - - - A1			A0	R/W	Registre selectionne
		Logique de decodage			RS		
		CS0	CS1	$\overline{CS}_2$			
ADR = 800C		1	1	0	0	0	écriture du registre de controle CR
ADR = 800C		1	1	0	0	1	lecture du registre d'etat SR.
ADR+1 = 800D		1	1	0	1	0	écriture dans le registre de transmission
ADR+1 = 800D		1	1	0	1	1	lecture du registre de recepteur.

$A_n$  = ligne adresse du MPU

$CS_n$  = ligne selection de boitier

$RS$  = " " de registre.

ADR = adresse de base resultante de la logique de decodage. Les lignes adresses  $A_2 A_1 A_0$  differentie les registres du PIA et de l'ACIA.



Schema de decodage.

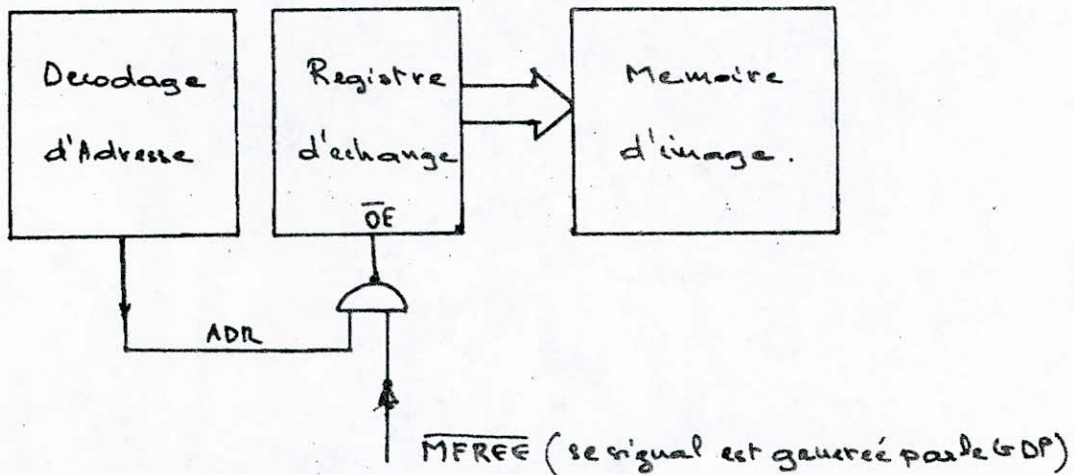


2°) Liaison microprocesseur - G.D.P et registre d'échange.

2.1) Liaison microprocesseur - registre d'échange.

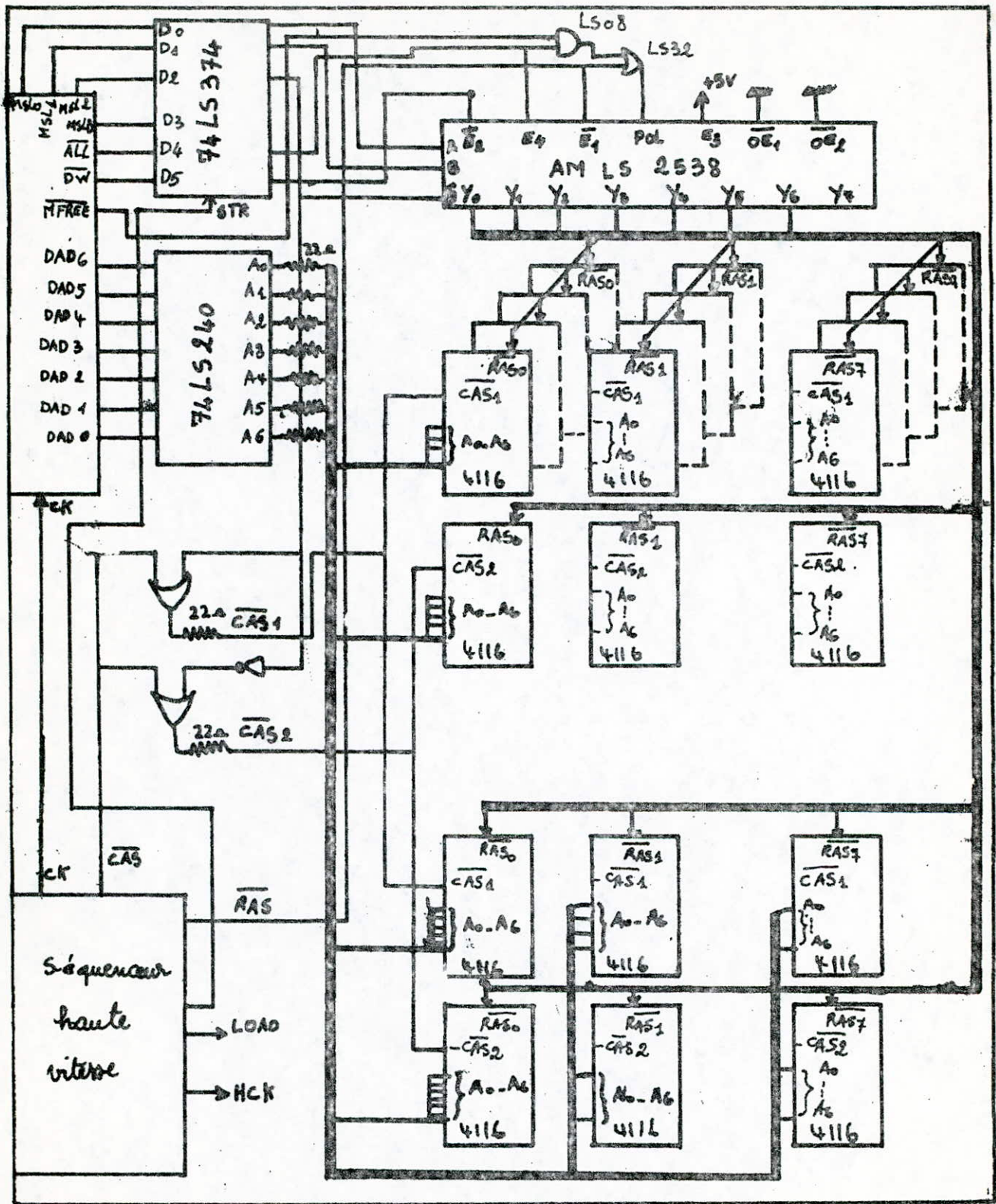
Pendant que le microprocesseur lit ou écrit dans les registres d'échanges, c'est-à-dire les registres de sélection feuille, de lecture et d'écriture, il transmet les données par l'intermédiaire du Buffer d'E/S.

Le tableau 4 donne la répartition des adresses de ces différents éléments. Les registres d'échanges sont à trois états et possèdent une entrée de sélection du boîtier inverse ( $\overline{OE}$ ).



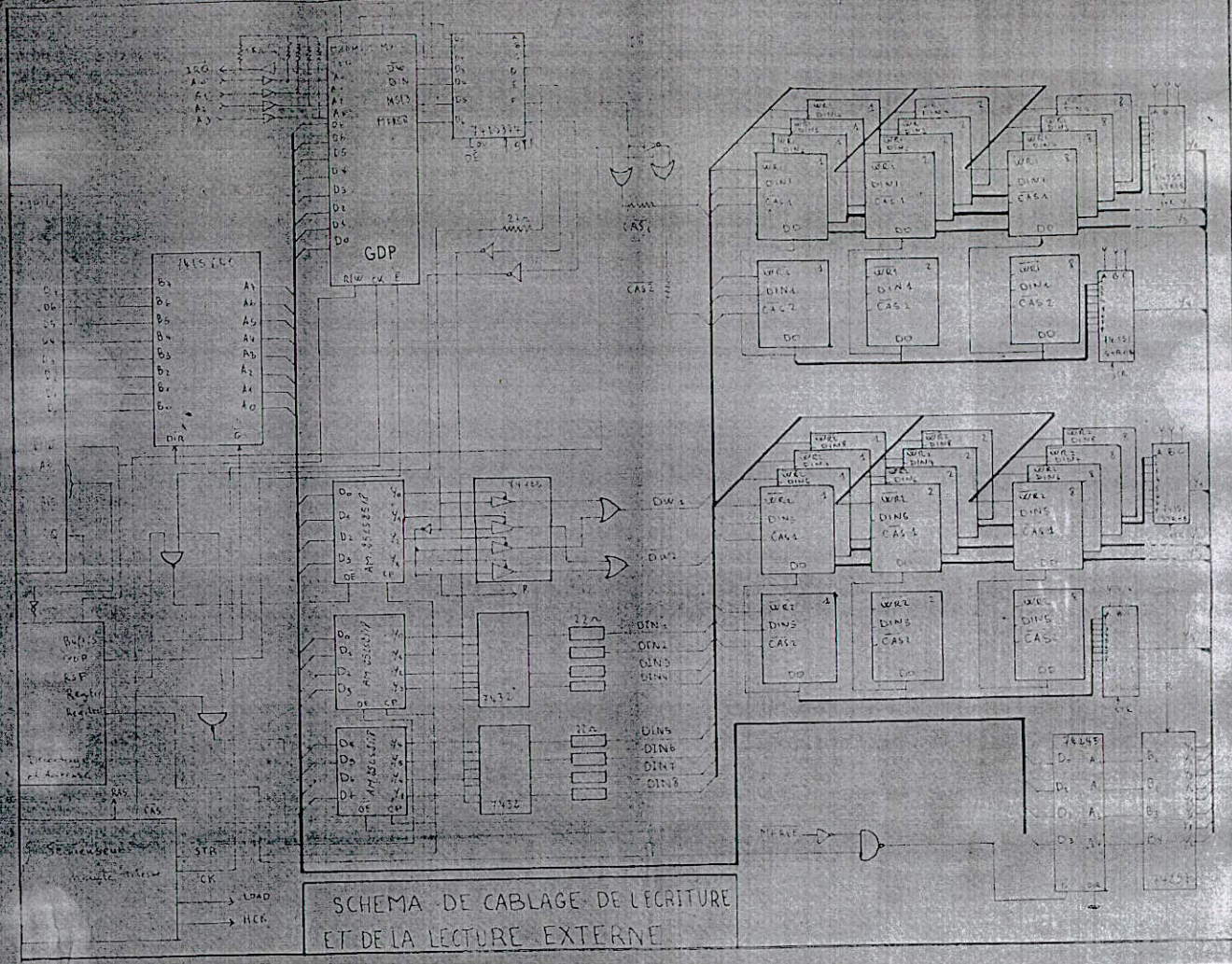
2.2) Liaison microprocesseur - G.D.P

Le Coprocesseur graphique possède une entrée  $\overline{E}$  de sélection du circuit et 4 entrées d'adresse  $A_0, A_1, A_2, A_3$  et cela pour adresser les registres internes du contrôleur d'écran (G.D.P). Ses 4 lignes d'adresse permettent d'adresser 16 registres internes. Il existe aussi une entrée R/W qui pourra être autre différentiel: le registre status du registre de commande qui ont la même adresse (0)16 de registre status est à lecture seule alors que le registre de commande est à écriture seule.



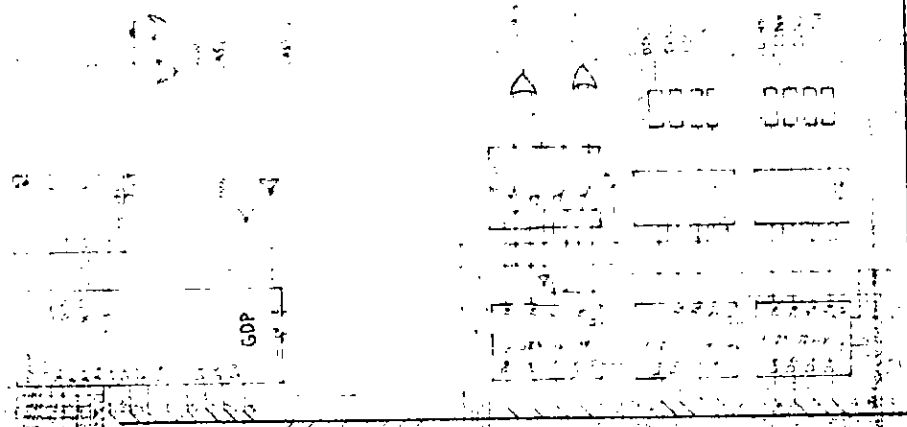
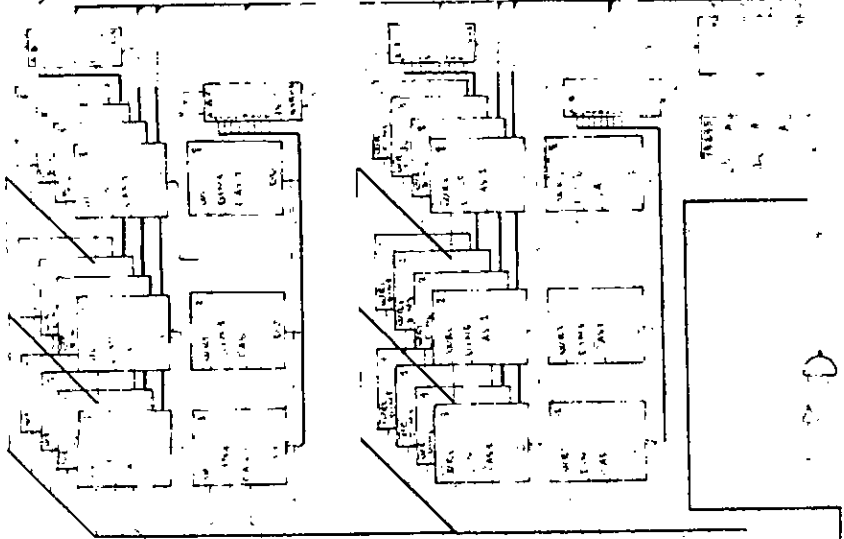
ADRESSAGE DE LA Memoire d'image



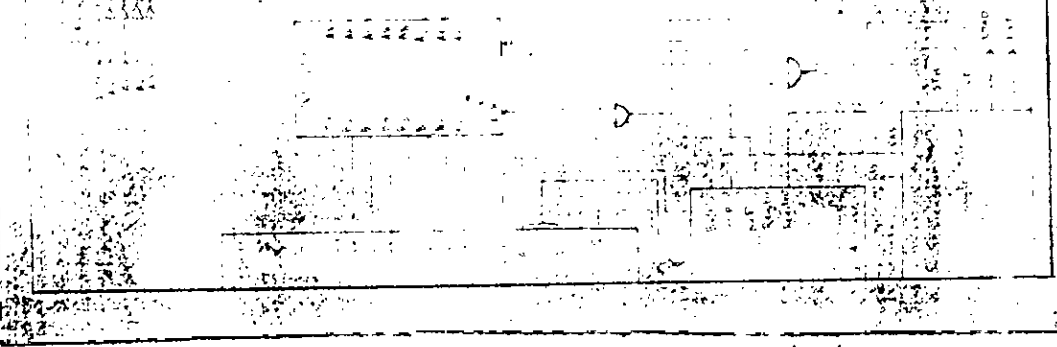


SCHEMA DE CABLAGE DE LECTURE  
ET DE LA LECTURE EXTERNE





SCHEMA DE CABINET DE LABORATOIRE  
 LI DE LA SALLE D'ENTREE



# CHAPITRE 3



## I - Role des registres d'échanges-

### I. 1) Introduction

Pour que le microprocesseur ait un accès externe à la mémoire d'image, il faut utiliser des registres d'échanges entre le microprocesseur et la mémoire d'image. Si on avait à utiliser qu'un seul plan c'est à dire une utilisation graphique ou alphanumérique la gestion de ce plan sera confié seulement au Coprocesseur car celui-ci comporte une mémoire ROM intégrée ayant tout les symboles du Code ASCII et aussi la generation de segments.

Mais dans le cas d'une utilisation plus grande de la mémoire d'image c'est à dire dans notre cas 8 plans mémoires divisés en deux feuilles de quatre plans chaque une, on a ajouté des registres d'échanges qui sont:

- Le registre de selection plan.
- Le registre écriture.
- Le registre lecture.

Ces registres sont des circuits tampons et sont adressés par le microprocesseur. Pendant que le microprocesseur s'adresse à ces registres, il faudra que le signal  $\overline{M\overline{FREE}}$  délivré par le coprocesseur soit à l'état bas.

### I-2) Registre de selection plan.

Ce registre permet d'utiliser chaque plan independamment des autres. Une donnée de 4 bits qui represente en fait un code, adressée au registre permet de selectionner un ou plusieurs plans en ecriture.

### I-3) Registre d'ecriture dans la memoire d'image.

La donnée qui represente le pixel codé en 4 bits donc 16 niveaux de gris est stockée dans ce registre. L'entrée de validation du circuit doit être à l'état bas.

$$\overline{OG} = \overline{M\overline{FREE}} \cdot \overline{ADR}$$

Le transfert de la donnée du registre tampons vers les plans memoires selectionnés se fait pendant le cycle ecriture. Le signal DIN delivré par le G.D.P autorisera à l'état haut ce transfert.

Ce signal est à l'état haut lorsque le registre de controle CTRL1 du G.D.P Contient le code (03)<sub>16</sub>

### I-4) Registre de Lecture.

Pendant ce cycle, il faudra que le signal  $\overline{ALL} = 1$ . C'est un registre de quatre bits car <sup>il</sup> reçoit l'information de quatre plans qui representent la feuille de stockage ou de traitement.

## II - Accès à la memoire d'image par une circuiterie externe.

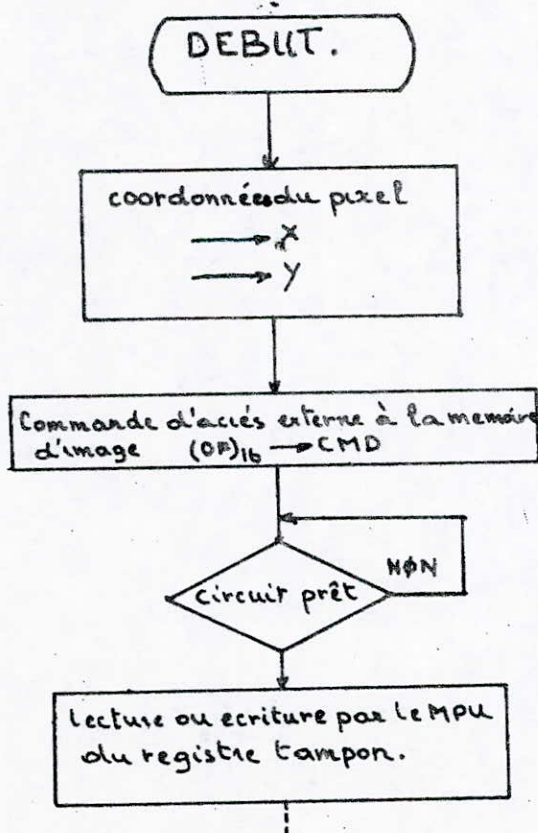
Le processeur graphique gère sa propre memoire d'image et cela à l'aide des signaux DIN,  $\overline{DW}$ ,  $\overline{ALL}$ . Les registres cités en haut



sont ajoutés au montage entre le bus microprocesseur et la mémoire d'image. Ils permettent de lire ou d'écrire dans cette mémoire et de stocker ainsi par exemple le contenu dans la feuille de stockage, faire les traitements choisis auparavant et recevoir le résultat du traitement dans la feuille à visualiser.

La feuille de stockage sera chargée au préalable par le contenu d'un fichier stocké sur une mémoire de masse.

La mise en oeuvre s'effectue comme l'indique l'organigramme de la figure n°-dessous :



\* Coordonnées du pixel.

(4)

Le microprocesseur charge dans les registres X et Y les coordonnées du pixel dont il veut connaître l'état.

\* Commande d'accès à la mémoire d'image.

une commande spéciale déclenche dès que la mémoire est libre, l'accès externe à la mémoire d'image par un registre tampon.

00001111 = \$0F → cmd.

\* Circuit prêt ?

si status (b2) = 1 (continue)

si status (b2) = 0 (attendre)

\* lecture du registre tampon: Avant de recharger de nouvelles coordonnées les registres X et Y, le microprocesseur peut aller lire (ou écrire) les données stockées dans les registres tampons.

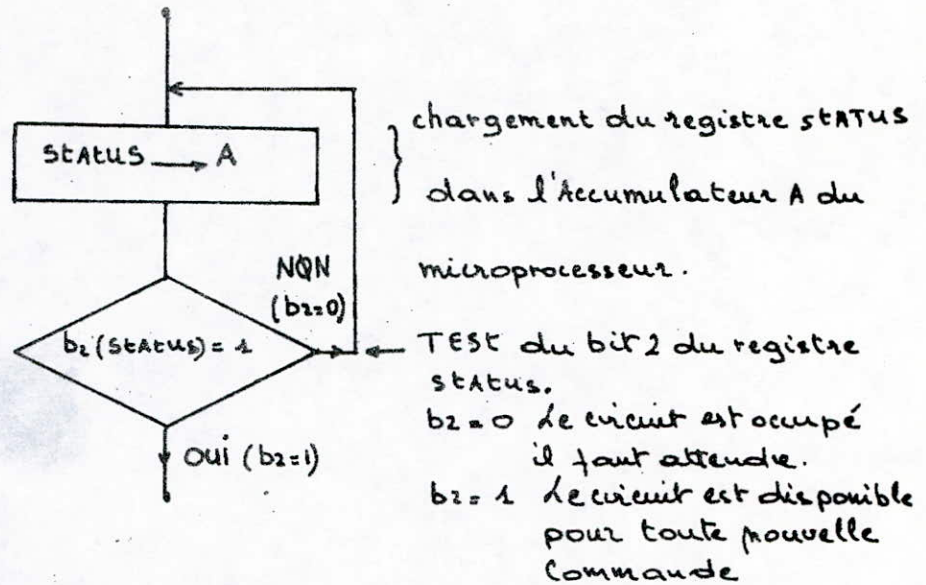
### III. ASPECT Programmation du coprocesseur EF 9365

#### III-1) Généralités

Avant de déclencher une commande, le microprocesseur doit s'assurer que le circuit EF 9365 est correctement initialisé et surtout libre de recevoir un ordre. Pour l'initialisation, le microprocesseur peut éventuellement modifier les registres de contrôle CTRL1 et CTRL2, les registres de coordonnées X et Y, les registres de projection SX et SY ou les registres d'échelle C21 et C22.



Mais il doit lire le registre d'état "STATUS" et attendre que le processeur ait terminé la commande précédente pour lui donner un nouvel ordre.



- Exemple de test de disponibilité du circuit -

dans le déroulement d'une commande, le microprocesseur peut effectuer toute autre opération; mais il ne doit pas lire ni écrire dans les registres internes utilisés par le coprocesseur.

On donnera dans ce qui suit des exemples de mise en œuvre de chaque type de commande.

III.2) TRACÉ DE VECTEUR (Segments)

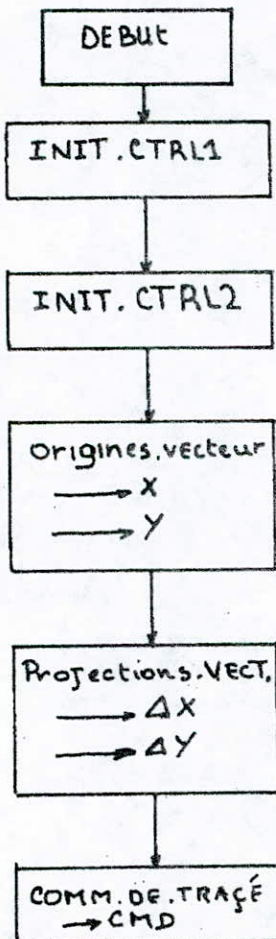
Deux sortes de vecteurs peuvent être tracés : des "petits vecteurs" de projection ( $\Delta x$  et  $\Delta y$ )  $\leq 4$  et des "grands vecteurs". Le tracé de ces derniers met, en jeu les registres CTRL1, CTRL2, X, Y,  $\Delta x$ ,  $\Delta y$  et CMD.

i) Grands vecteurs :

Exemple : Soit à tracer le vecteur d'origine  $\begin{cases} X=220 \\ Y=150 \end{cases}$   
d'accroissement  $\begin{cases} \Delta X=140 \\ \Delta Y=100 \end{cases}$

et cela en trait mixte

L'organigramme ci-dessous définit ou montre les étapes de tracé de ce vecteur.



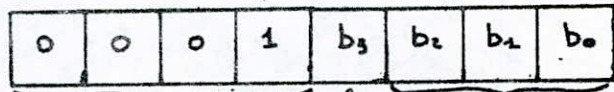
\* Initialisation CTRL1:  $00000011 = (03)_{16}$   
Ce qui correspond d'une part à sélectionner le crayon ( $b_3$  à l'état haut) et à le baisser ( $b_0$  à l'état haut)

\* Initialisation CTRL2:  $00000011 = (03)_{16}$   
Ce qui correspond au tracé de trait mixte.

\* Origine du vecteur:  $(220)_{10} \rightarrow$  registre X  
 $(150)_{10} \rightarrow$  " Y

\* Projection du vecteur:  $(140)_{10} \rightarrow$  registre  $\Delta X$   
 $(100)_{10} \rightarrow$  "  $\Delta Y$   
On met dans ces registres que les valeurs absolues.

\* Commande de tracé: Le registre CMD doit être chargé de cette façon:



orientation du vecteur  
Etat bas  
Indique le tracé d'un grand vecteur



L'orientation du vecteur correspond aux signes des projections  $\Delta X$  et  $\Delta Y$ ; quatre directions sont possibles: deux bits suffiraient donc pour tracer tous les vecteurs à l'écran. Pour plus de souplesse, les quatre directions Nord, Sud, Est et Ouest ont été privilégiées et un code particulière leur est affectée. Trois bits ( $b_2, b_1, b_0$ ) sont donc nécessaires pour exprimer les huit possibilités. Pour <sup>de</sup> plus amples informations il faudra voir le tableau de Commande. Donc le Contenu du registre de Commande sera:  $CMD \leftarrow 00010101 = (15)_{16}$

### (i) Petits vecteurs

Dans le cas des petits vecteurs, les registres  $\Delta X$  et  $\Delta Y$  ne sont plus pris en compte. C'est l'ordre de commande lui-même qui indique ces projections. L'organigramme est plus simple et le nombre d'instruction nécessaires pour exécuter ce tracé se trouve réduit.

Le genre de commande est utilisé pour le tracé de graphe.

• Exemple: Soit à tracer une portion de courbe représentée par quatre petits vecteurs - origine  $X=6$   
 $y=15$

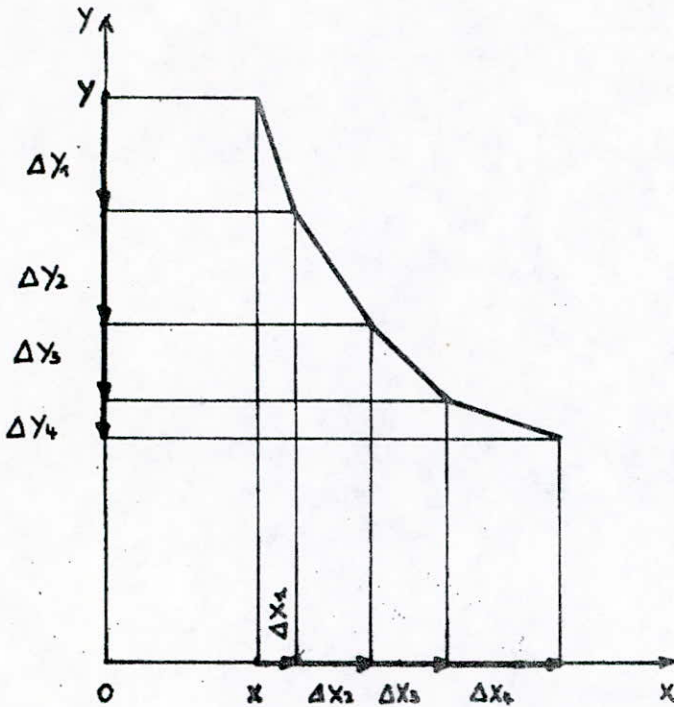
$$\begin{aligned} \text{Vecteur 1 } \Delta X_1 &= 1 \\ \Delta Y_1 &= -3 \end{aligned}$$

$$\begin{aligned} \text{Vecteur 2 } \Delta X_2 &= 2 \\ \Delta Y_2 &= -3 \end{aligned}$$

$$\begin{aligned} \text{Vecteur 4 } \Delta X_4 &= 3 \\ \Delta Y_4 &= -1 \end{aligned}$$

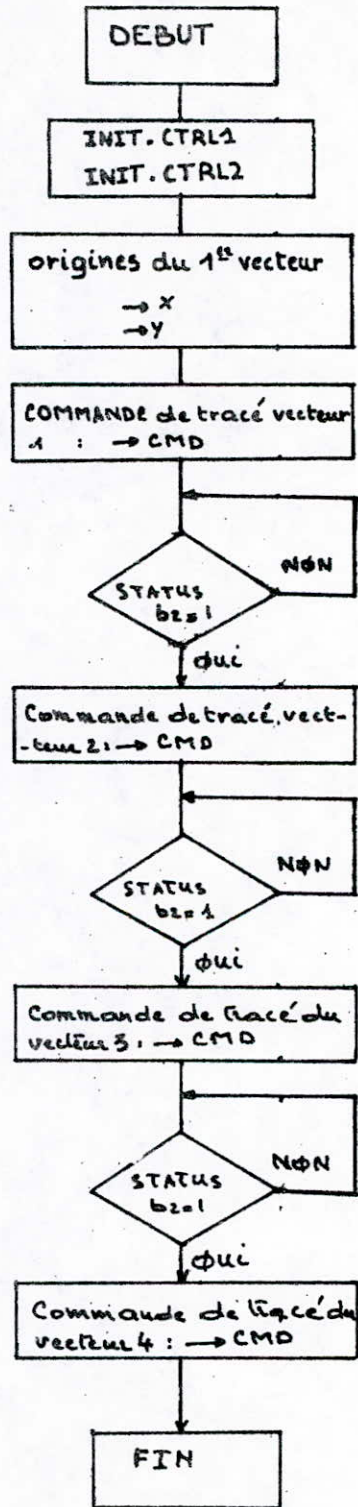
$$\begin{aligned} \text{Vecteur 3 } \Delta X_3 &= 2 \\ \Delta Y_3 &= -2 \end{aligned}$$

Le tracé se fera en trait continu:



a) Description du tracé ↑

b) Organigramme de tracé de quatre petits vecteurs.



\* Initialisation CTRL1 : Peut se faire également par deux commandes :

$(00)_{16} \rightarrow$  CMD choix du crayon

$(02)_{16} \rightarrow$  CMD crayon baissé

Ne pas oublier de tester, entre ces deux commandes, le bit  $b_2$  du registre STATUS

$b_2(\text{status}) = 1 \rightarrow$  continuer.

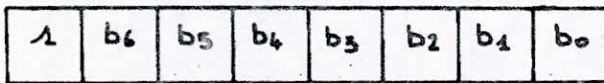
$b_2(\text{status}) = 0 \rightarrow$  attendre.

\* Initialisation CTRL2 :  $(00)_{16} \rightarrow$  CTRL2 ce qui correspond au tracé en trait plein

\* origines du vecteur :  $(6)_{10} \rightarrow X$

$(15)_{10} \rightarrow Y$

\* Commande de tracé des vecteurs  $k$  (1, 2, 3, 4)



avec  $0 \leq \Delta X \leq 3$  et  $0 \leq \Delta Y \leq 3$

L'orientation du vecteur est donnée par les bits ( $b_2, b_1, b_0$ ) de la même façon que pour les grands vecteurs.

### III-3) Tracé d'un caractère.

Le générateur de caractères du processeur graphique se charge de traduire, sous forme de bits à l'état 0 ou 1, dans sa mémoire d'image, les pixels constituant le caractère à afficher. Le code ASCII de ce dernier est transmis dans le registre de CMD.

Mais au préalable, les registres CTRL1, CTRL2, CS16 seront initialisés. L'origine (abscisse et ordonnée) du premier caractère à écrire sera chargée dans les registres X et Y.

\* Registre CTRL2 : Les bits ( $b_2, b_1$ ) donnent les quatre combinaisons d'écriture des caractères d'une part suivant l'axe horizontale ou verticale et d'autre part en caractère droit ou en italique.



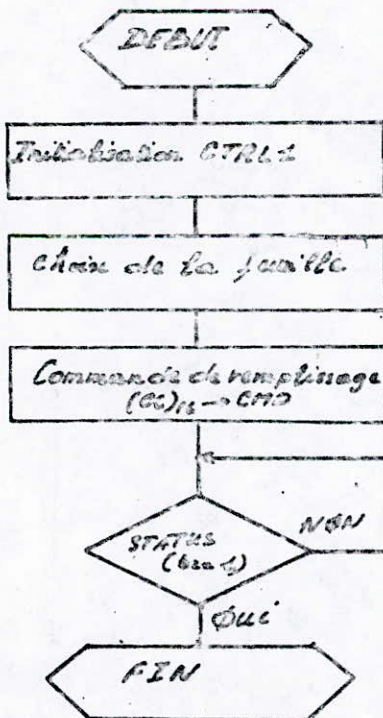
n Le registre CSISE : Comme on l'a dit auparavant, ce registre est un registre d'échelle de caractères selon P et Q qui sont respectivement les facteurs d'échelles suivant l'axe des X et l'axe des Y. Avec  $(P=0, Q=0)$ , le facteur d'échelle vaut 16.

x Le registre CMD : voir le tracé des caractères dans le tableau de Commande.

### III.4) Commande de Remplissage de l'écran.

Une commande permet de positionner à 1 ou à 0 tous les bits de la mémoire d'image. La gomme ou la plume est sélectionnée dans le registre de Contrôle CTRL1, et la commande est déclenchée par l'écriture de la valeur  $(00)_{16}$  dans le registre CMD.

La figure n°-dessous définit l'organigramme de remplissage d'écran.



→ Le choix de la feuille se fera par une écriture dans le registre sélection feuille ou place d'un donnée adressé par le MPU.

## IV. Le traitement Zoom.

### IV.1) Généralité.

On peut effectuer sur l'image originale différents traitements mais on peut les classer de la façon suivante :

#### i) Traitement intensité.

- Seuillage
- Lissage
- Erosion ect...

#### ii) Traitement géométrique.

- Decadrage
- Visualisation d'une fenêtre
- Zoom ect...

Certains traitements sont assez complexes et sont effectués au niveau de l'ordinateur central et cela à cause du nombre de test effectués sur un seul élément de l'image (Pixel) et la nature de son environnement.

### IV.2) Principe - (zoom) -

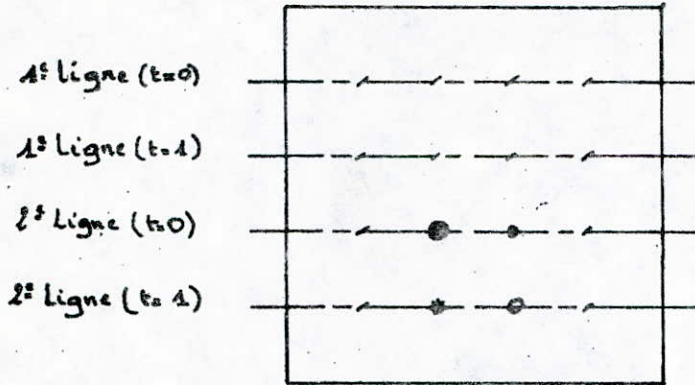
Le traitement zoom est un traitement géométrique qui consiste à agrandir une partie de l'image. La partie intéressée peut se trouver dans n'importe quelle zone de l'écran et elle est composée de lignes paires et impaires qui doivent être pris en compte séparément pendant l'étude de l'Algorithme.

Nous avons étudié le cas du zoom de deux, ce qui revient à prendre une fenêtre de (256x256) pixels.

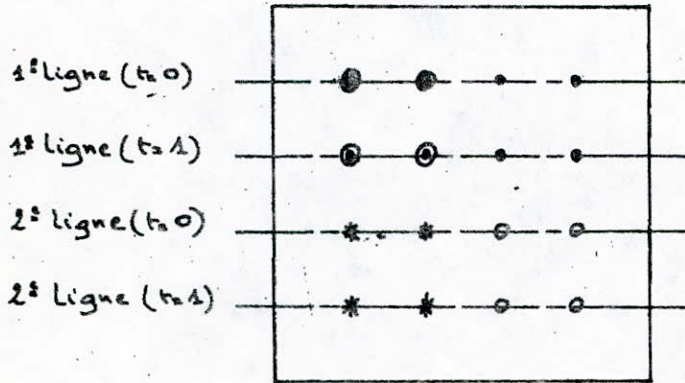
Les principales commandes qu'on adressera au Coprocesseur sont :

- Commande d'accès interne à la mémoire d'image (CMD ← (0F)16)
  - Accès externe mode lecture. (CTRL ← (02)16)
  - Accès externe mode écriture. (CTRL ← (0B)16)
- Commande de balayage de l'écran (CMD ← (0C)16)
  - Avec la plume (CTRL ← (03)16)
  - Avec la gomme (CTRL ← (00)16)

La page suivante montre d'une manière simplifiée, le principe d'un agrandissement de 2 d'une fenêtre 2x2 pixels dans un espace de (4x4) pixels.



a. Image stockée dans la feuille de stockage ici représentée une fenêtre de l'image

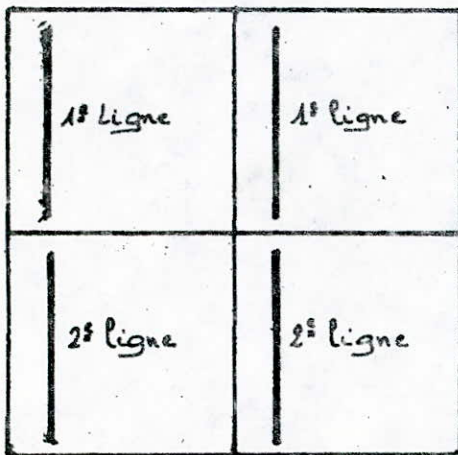


b. Image agrandie dans la feuille de traitement ou de visualisation.

$t=0$  (trame impaire)  
 $t=1$  (trame paire)  
 Les symboles représentés sur l'écran sont des états différents du pixel

$t=0$

$t=1$



c) Représentation des lignes dans le boîtier mémoire.



Avant de faire le Commentaire de l'Algorithme de traitement Zoom, nous donnerons les différents sous programmes de lecture externe, d'écriture externe, et de visualisation.

### IV-2.1) Organigramme lecture externe -

#### i) Commentaire -

Il faudra pour effectuer ce programme, initialiser le registre de Contrôle en le positionnant en pleine levée :  $CTRL1 \leftarrow (02)_{16}$   
Dans ce cas  $ALL = 1$  ou à l'accès pixel à pixel.

Le mode externe doit se faire à l'aide de la commande  $(0F)_{16}$  du registre CMD :  $CMD \leftarrow (0F)_{16}$

Le choix de la feuille à lire (feuille de stockage ou de traitement) et nécessaire, le code de sélection sera envoyé du microprocesseur vers le registre sélection feuille :  $RSF \leftarrow (1F)_{16}$   
→ dans ce cas on a choisi la feuille de stockage à lire.

#### ii) fonctionnement -

Après un cycle de lecture, 8 pixels  $\times$  4 bits sont disponibles à la sortie des 4 plans mémoires de la feuille de stockage l'acheminement vers le microprocesseur s'effectue par mot de 4 bits.

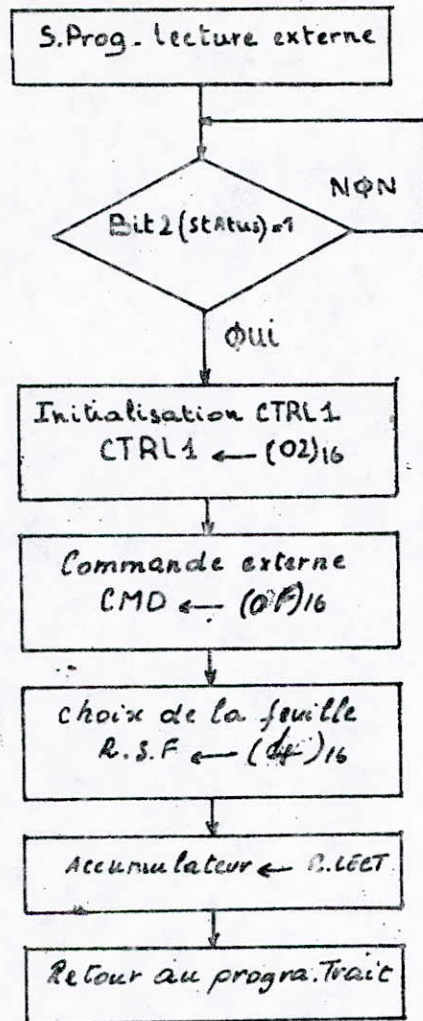
Pour assurer cet acheminement dans de bonnes conditions, nous utilisons 8 multiplexeurs (8 vers 1) "74LS151" avec la répartition suivante : 4 multiplexeurs pour la feuille de stockage.  
" " " " de traitement

A la sortie des multiplexeurs 8 pixels seront validés car ayant le même adresse. A l'aide d'un multiplexeur (8/4) type 74LS157 et suivant le code de sélection choisi, un pixel seulement est disponible à l'entrée du registre tampon type 74LS245 lorsque l'entrée  $\bar{b}$  de ce registre est à l'état bas l'information est transférée vers le bus microprocesseur.

L'entrée DIR de ce registre est mise à la masse car l'échange se fait seulement dans une seule direction.

#### iii) Algorithme -

L'Algorithme est donné page suivante :



- Algorithme de la lecture externe -

IV.2.2) Organigramme écriture externe.

i) Commentaire -

La commande écriture externe est la même que celle de la lecture c'est-à-dire  $(0F)_{16} \rightarrow CMD$ . Mais l'initialisation du registre de contrôle CTRL1 diffère de celle de la lecture externe. L'écriture du code  $(02)_{16}$  dans le registre CTRL1 met la plume "baisée". On peut utiliser aussi la commande  $(0E)_{16}$  qui permet de passer de la



plume levée à la plume baissée, mais après avoir consulté le bit 6 du registre d'état (STATUS).

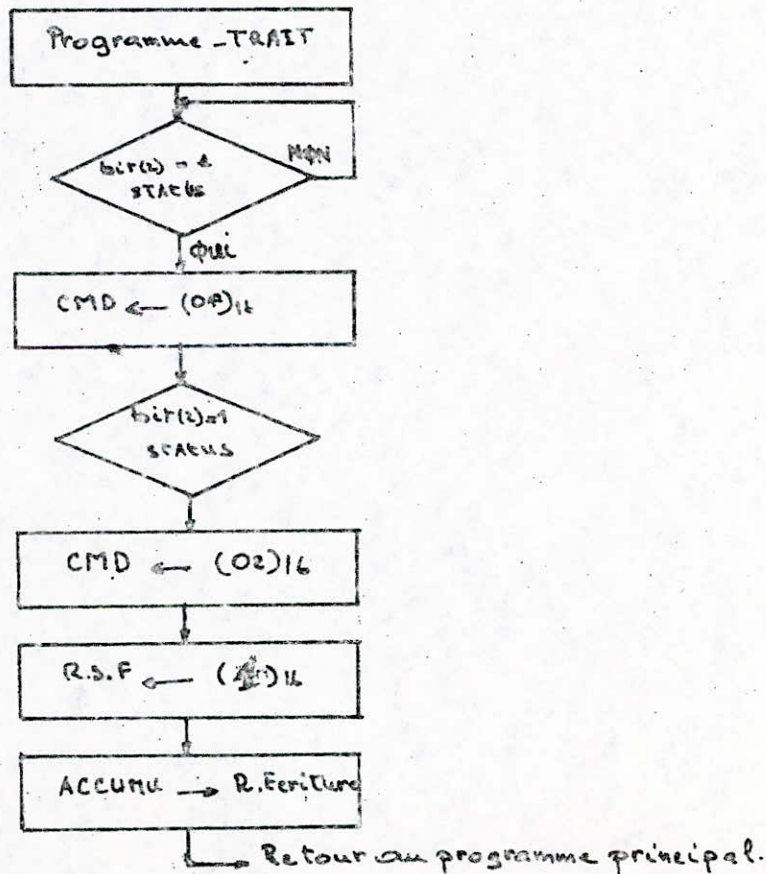
Dans ces conditions le signal  $\overline{MARE}$  délivré par le coprocesseur à l'état bas transmet le contenu du registre écriture aux entrées de la mémoire. Pour un accès pixel à pixel, le signal  $\overline{ALE}$  doit être à l'état haut.

### ii) Fonctionnement -

Pour que le mode écriture externe se fait dans de bonne condition, nous avons utilisé un registre tampon qui permet de mémoriser momentanément les données avant leur transfert dans la mémoire d'image.

Mais il faudra faire le choix de la feuille et cela en écrivant le code  $(1)_{16} \rightarrow R.S.F$  (registre sélection feuilles), ce qui permet de mettre la entrée  $\overline{WR}$  de boîtiers mémoires à l'état bas - celle de la feuille de stockage -

### iii) Algorithme "écriture externe"





#### IV\_3) Organigramme du traitement Zoom

Nous avons utilisé des mémoires fonctionnant en compteurs car il faudra contrôler les registres d'adresses  $x$  et  $y$  du coprocesseur graphique extérieurement. Le microprocesseur positionnera à chaque lecture et écriture dans la mémoire d'image les registres  $x$  et  $y$ .

Lorsque le microprocesseur fait une lecture ou une écriture, les registres  $x$  et  $y$  s'incrémentent de un.

L'adresse d'un pixel - lorsqu'il est lu - est stockée dans les positions mémoires  $x_1$  et  $y_1$  qui occupent chacune d'elles 2 positions mémoires.

L'adresse d'un pixel écrite dans la mémoire d'image et incrémentée de 1 dans les registres  $x$  et  $y$  est stockée dans les mémoires respectivement  $x_2$  (occupant 2 positions mémoires) et  $y_2$  (occupant aussi 2 positions mémoire).

Nous donnons à la page suivante l'Algorithme du zoom de 2 -

Les lignes paires et impaires de la fenêtre choisie et lue à partir de la feuille de stockage seront réécrites dans les lignes paires de la feuille à visualiser puis on refait la même opération mais en écrivant dans les lignes impaires de la feuille à visualiser.

(DEBUT)

Adresse du point P(0,0)  
compteur COMPX  
Compteur ΔX : COMPΔX  
Compteur ΔY : COMPΔY  
Compteur X : COMPX  
Compteur Y : COMPY

retrievée du point initial  
 $X \leftarrow X_0$   
 $Y \leftarrow Y_0$

incrémente COMPX

INCR. COMPLY

INCR. COMPLY

Lecture externe de P(x,y) pointé par les registres d'adresse (x,y)

Transférer le contenu dans VALXP  
" " de  $X_1, Y_1 \rightarrow X_2, Y_2$   
" " de  $X_2, Y_2 \rightarrow X_3, Y_3$

INCRÉMENTE COMPΔX

Ecriture de VALXP dans les adresses pointé par (x,y)

INC. COMPX

COMPX - FACE ≠ 0

NON

COMPΔX = 0

Transférer le contenu de  $X_1, Y_1, X_2, Y_2$   
" " " "  $X_3, Y_3 \rightarrow X_4, Y_4$

COMPΔX - ΔX ≠ 0

NON

COMPΔX = 0

positionner les registres internes X, Y du C.P.P.  
 $X \leftarrow X_0$   
 $Y \leftarrow Y_0$   
 $t \leftarrow t$

COMPLY - FACE ≠ 0

INCR. REGY

COMPLY = 0

COMPΔY - ΔY ≠ 0

NON

COMPΔY = 0

COMPT - FACE ≠ 0

NON

END

Algorithme du traitement  
7 nom

ANNEXE



(Début)

Adresse du 1<sup>er</sup> pixel  $P(x_0, y_0)$   
compteur  $COMPX = COMPY = 1$   
compteur  $\Delta X = COMPX$   
compteur  $\Delta Y = COMPY$   
compteur  $x = COMPX$   
compteur  $y = COMPY$

relecture du pixel  
unité  $x_0 \leftarrow x_0$   
 $y_0 \leftarrow y_0$

incrémente  
COMPX

INCR. COMPY

INCR. COMPY

lecture externe de  $P(x, y)$  pointé  
par les registres d'adresse  $(x, y)$

Transférer le contenu dans VALUP  
de  $x_1, y_1 \rightarrow x_2, y_1$   
de  $x_2, y_2 \rightarrow x_3, y_2$

INCRÉMENTE  
COMPX

Écriture de VALUP dans les  
adresses pointé par  $(x, y)$

INC. COMPX

COMPX - FACE  $\neq 0$

NON

COMPX = 0

Transférer le contenu de  $x_1, y_1 \rightarrow x_2, y_1$   
 $x_2, y_2 \rightarrow x_3, y_2$

COMPX -  $\Delta X \neq 0$

NON

COMPX = 0

positionner les registres internes  $x, y$  de  
G.P.P.  $x \leftarrow x_0$   
 $y \leftarrow y_0$   
 $t \leftarrow L$

COMPY - FACE  $\neq 0$

INCR. REGY

COMPY = 0

COMPY -  $\Delta Y \neq 0$

NON

COMPY = 0

COMPT - FACE  $\neq 0$

NON

END

Algorithme du traitement  
7 nom

## Bibliographie.

### Revue:

- électronique application numéro 19 Aout/Sept 81.
- électronique application numéro 26 oct/NOV 82.

### Ouvrages:

- Microprocesseur et microordinateurs  
Tocci & LASKOWSKI EYROLLES
- Microprocesseur 6809 et ses périphériques  
et le processeur graphique 9365-66.  
Claude DAROANNE et J Boulesteix EYROLLES.
- Microprocesseur et mémoires EFCIS catalogue 80.
- Manuel de Programmation du 6800 Thomson - CSF

### Thèses:

- Système d'aide au traitement numérique d'images  
application à la télédétection  
Par AUF Hourani année 79.
- Système d'affichage d'images avec mémoires  
structurées en feuilles. CERI  
Par M. SMAHAT Année 81.
- Déconvolution des caméras de télévision dans le  
cas des mesures photométriques et géométriques  
Par Benoit Speckel. année 81.
- Conception et réalisation d'un système d'acquisition  
et de restitution d'images numériques  
Par Jean Marc Moser 1981.



**ADRESSE ET FONCTION DES REGISTRES**

REGISTRE D'ADRESSE					FONCTION DES REGISTRES		Nombre de bits
Binaire				Hexa	Lecture	Ecriture	
A3	A2	A1	A0				
0	0	0	0	0	STATUS	CMD	8
0	0	0	1	1	CTRL1 (contrôle de l'écriture et des interruptions)		7
0	0	1	0	2	CTRL2 (orientation des symboles et type de vecteurs)		4
0	0	1	1	3	CSIZE (taille des caractères)		8
0	1	0	0	4	Réservé		—
0	1	0	1	5	DELTA X		8
0	1	1	0	6	Réservé		—
0	1	1	1	7	DELTA Y		8
1	0	0	0	8	X Poids fort		4
1	0	0	1	9	X Poids faible		8
1	0	1	0	A	Y Poids fort		4
1	0	1	1	B	Y Poids faible		8
1	1	0	0	C	XLP	Réservé	7
1	1	0	1	D	YLP	Réservé	8
1	1	1	0	E	Réservé		—
1	1	1	1	F	Réservé		—

**Réservé :** Ces adresses sont réservées pour des versions futures du circuit. En lecture, les amplificateurs de sortie D0 à D7 forcent l'état haut sur le bus de données.

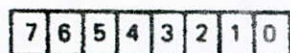
**RESUME DES CODES DE COMMANDE**

b7 b6 b5 b4	b3 b2 b1 b0				b7 b6 b5 b4 b3 b2 b1 b0																																							
	0				1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																									
0 0 0 0	0	Met à 1 le bit 1 de CTRL1 : "Choix de la plume"				Vecteurs standard (cf. petits vecteurs pour b <sub>2</sub> , b <sub>1</sub> , b <sub>0</sub> )	Espace	0	@	#	-	p	<b>PETITS VECTEURS :</b>  <table border="1"> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>1</td><td> ΔX </td><td> ΔY </td><td colspan="4">Direction</td> </tr> </table> Dimension  <table border="1"> <tr> <td>ΔX ou ΔY</td><td>Long. du vecteur</td> </tr> <tr> <td>0 0</td><td>0 pas</td> </tr> <tr> <td>0 1</td><td>1 pas</td> </tr> <tr> <td>1 0</td><td>2 pas</td> </tr> <tr> <td>1 1</td><td>3 pas</td> </tr> </table> Direction  							b7	b6	b5	b4	b3	b2	b1	b0	1	ΔX	ΔY	Direction				ΔX ou ΔY	Long. du vecteur	0 0	0 pas	0 1	1 pas	1 0	2 pas	1 1	3 pas
b7	b6	b5	b4	b3	b2		b1	b0																																				
1	ΔX	ΔY	Direction																																									
ΔX ou ΔY	Long. du vecteur																																											
0 0	0 pas																																											
0 1	1 pas																																											
1 0	2 pas																																											
1 1	3 pas																																											
0 0 0 1	1	Met à 0 le bit 1 de CTRL1 : "Choix de la gomme"				1	1	A	D	e	q																																	
0 0 1 0	2	Met à 1 le bit 0 de CTRL1 : "Passage en plume ou gomme baissée"				"	2	B	#	b	r																																	
0 0 1 1	3	Met à 0 le bit 0 de CTRL1 : "Passage en plume ou gomme levée"				#	3	C	\$	c	s																																	
0 1 0 0	4	Effacement de l'écran.				4	4	D	T	d	t																																	
0 1 0 1	5	Remise à 0 des registres X et Y.				%	5	E	U	e	u																																	
0 1 1 0	6	Effec. de l'écran et remise à 0 de X et Y.				@	6	F	V	f	v																																	
0 1 1 1	7	Effacement de l'écran, positionnement à 11 <sub>16</sub> du registre CSIZE, remise à 0 des autres registres (sauf XLP, YLP).				"	7	G	W	g	w																																	
1 0 0 0	8	Initialisation du photostyle (forçage de la sortie WHITE au niveau bas).				(	8	H	X	h	x																																	
1 0 0 1	9	Initialisation du photostyle.				)	9	I	Y	i	y																																	
1 0 1 0	A	Lancement du tracé du pavé 5 x 8				*	:	J	Z	j	z																																	
1 0 1 1	B	Lancement du tracé du pavé 4 x 4				+	:	K	[	k	{																																	
1 1 0 0	C	Blayage de l'écran avec la plume ou la gomme (suivant CTRL1)				,	<	L	V	l	;																																	
1 1 0 1	D	Remise à 0 du registre X.				-	=	M		m																																		
1 1 1 0	E	Remise à 0 du registre Y.				.	>	N		n	~																																	
1 1 1 1	F	Demande externe d'accès à la mémoire d'image pour le premier cycle libre.				/	?	O	-	o	∞																																	



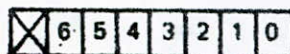
## LES AUTRES REGISTRES

### REGISTRE STATUS (lecture seulement)



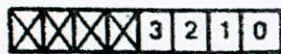
- HAUT = séquence photostyle terminée
  - HAUT = signal retour de trame (VB)
  - HAUT = circuit libre pour une nouvelle commande
  - HAUT = plume en dehors de la fenêtre visualisée : OU logique des bits de poids fort des registres X et Y.
  - HAUT = Interruption pour fin de séquence photostyle
  - HAUT = interruption pour signal VB à l'état haut
  - HAUT = Interruption pour circuit prêt à recevoir une nouvelle commande.
  - IRQ : OU logique des 3 bits précédents (haut quand la sortie  $\overline{IRQ}$  est à l'état bas).
- Origines d'interruptions éventuelles (signaux non échantillonnés et non masqués)
- Le OU logique de ces 3 bits fait passer  $\overline{IRQ}$  à l'état bas.
- Ces 3 bits sont remis à 0 après lecture du registre STATUS.

### REGISTRE CTRL1 (lecture/écriture)



- HAUT = plume ou gomme baissée ; BAS = plume ou gomme levée (contrôle la sortie  $\overline{DW}$ ).
  - HAUT = plume ; BAS = gomme (contrôle la sortie DIN)
  - HAUT = écriture rapide : extinction de la vidéo (BLK au niveau haut), rafraîchissement minimal.
  - HAUT = écran cyclique (écriture même si le bit 3 de STATUS est au niveau haut).
  - HAUT = autorisation d'interruption pour fin de séquence photostyle.
  - HAUT = autorisation d'interruption pour signal VB à l'état haut
  - HAUT = autorisation d'interruption pour circuit prêt à recevoir une nouvelle commande.
  - Non utilisé (toujours 0 en lecture).
- Masque d'interruption

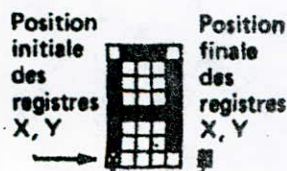
### REGISTRE CTRL2 (lecture/écriture)



(toujours 0 en lecture)

- Type de trait des vecteurs
- HAUT - caractère en italique
- HAUT - caractère tracé suivant l'axe vertical

b1	b0	Types de trait des vecteurs	
0	0	—	continu
0	1	- - - -	pointillé 2 pts allumés, 2 pts éteints
1	0	- - - -	tiré 4 pts allumés, 4 pts éteints
1	1	- · - ·	trait mixte 10 pts allumés, 2 éteints 2 allumés, 2 éteints.



b3 = 0, b2 = 0



b3 = 0, b2 = 1

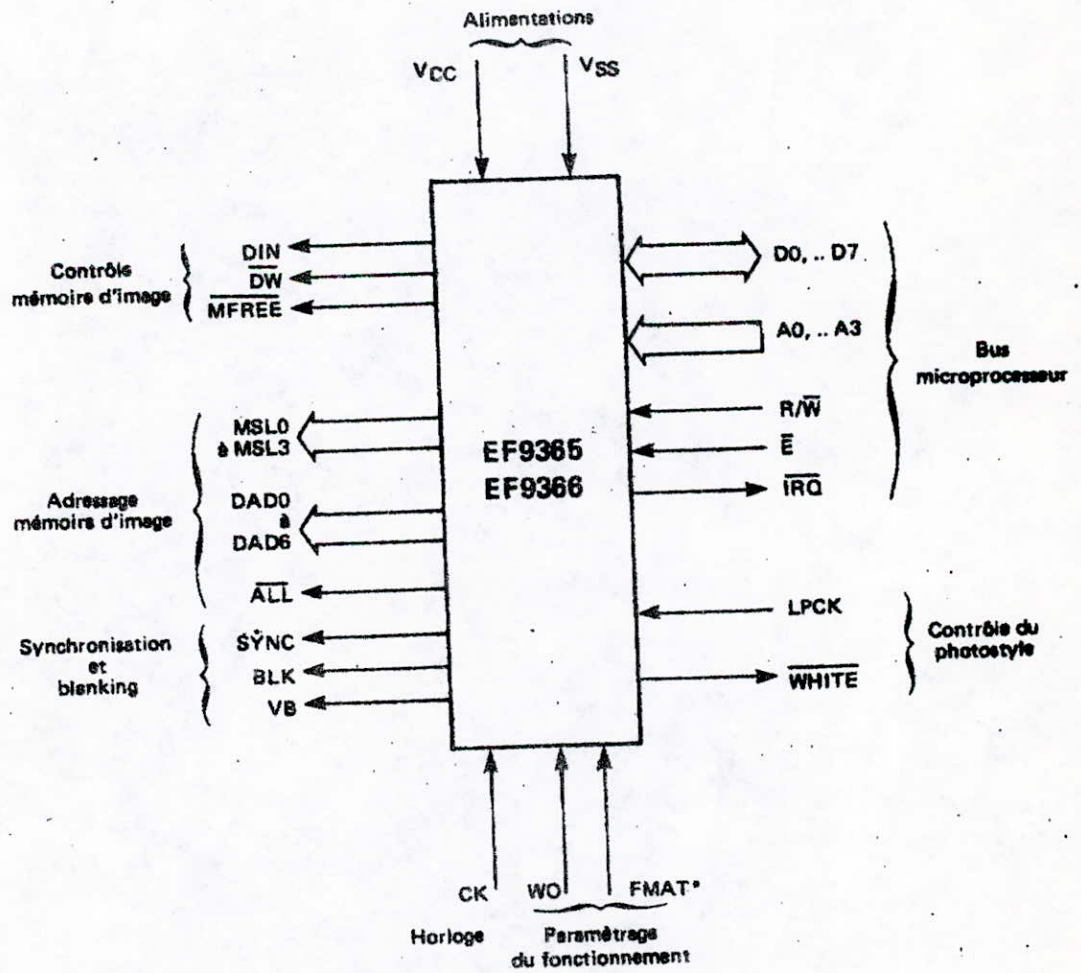


b3 = 1, b2 = 0



b3 = 1, b2 = 1

## DESCRIPTION DES BROCHES



\*  $FMAT$  est à connecter à  $V_{SS}$  pour le EF9366.







## Conclusion.

L'utilisation d'un processeur spécialisé tel que le G.D.P 9365-66 permet de réaliser aujourd'hui des consoles graphiques de très bonne qualité et très bon marché. Ces 2 caractéristiques primordiales sont la raison de notre choix d'un tel composant.

Notre étude consistait d'une part à étudier une configuration de la mémoire d'image qui rendrait le système interactif plus souple et plus autonome vis à vis du calculateur associé et d'autre part à élaborer un logiciel de traitement simple suivant les exigences du support matériel ce qui nous a permis de nous initier au traitement d'images domaine très récent dont les applications sont aussi variées qu'innombrables, mais malheureusement nous déplorons le manque des composants qui a empêché la réalisation et espérons que ce projet sera le sujet d'autres études pour la mise au point complète de la console interactive.

- étude, réalisation et applications d'un ensemble de processeurs destinés au traitement numérique d'images.  
Par ABDEL-MEGED MAHMOUD Année 1980.
- Conception et réalisation du système de traitement numérique d'Image LAE 980.  
par Bernard Keith
- étude et réalisation d'un processeur programmable pour le traitement et l'analyse d'images fixes de TV.  
par Karl H-RACKETTE
- Projets de fin d'étude:
  - M. Belkacem et M. TILIKET Fév 82.
  - A. SOAL et B. HADJ-AÏSSA Juin 82.

Rapport scientifique:

traitement numérique d'images Juin 82.

