

**MEMOIRE**

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

pour l'obtention du diplôme d'ingénieur d'état  
en électronique

Thème

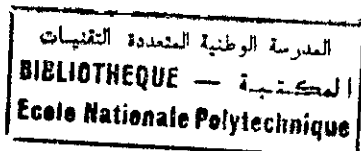
**CONCEPTION ET REALISATION D'UNE  
CARTE GRAPHIQUE COMPATIBLE PC**

Copromoteur :  
Mr L. SAADAoui

Présenté par :  
SALHI Wassila

Proposé par Mr A. ALLAM  
Dirigé par Mr A. ALLAM  
M<sup>me</sup> N. ANANE

**MEMOIRE**



pour l'obtention du diplôme d'ingénieur d'état  
en électronique

Thème

**CONCEPTION ET REALISATION D'UNE  
CARTE GRAPHIQUE COMPATIBLE PC**

Copromoteur :  
Mr L. SAADAOUI

Présenté par :  
SALHI Wassila

Proposé par Mr A. ALLAM  
Dirigé par Mr A. ALLAM  
Mme N. ANANE

## Remerciements

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

Au delà des convenances, je veux assurer les personnes suivantes du respect dans lequel je tiens leur contribution personnelle à la réalisation de ce mémoire.

Qu'il me soit ainsi permis de remercier Mr OUGUINI de m'avoir accueillie au sein du CDTA.

Mes remerciements vont également à Mrs ALLAM et SAADAOUI pour leur encadrement et leur disponibilité.

Je tiens à remercier Mme ANANE pour sa gentillesse et ses précieux conseils.

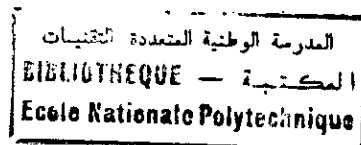
Que tout le personnel du CDTA trouve ici l'expression de ma profonde gratitude pour leur gentillesse et leur merveilleux accueil. Particulièrement Mrs BENDAOUIA, CHALABI, SEHAD, MAAROUF, SALIM et HOUCINI.

Je n'oublierai pas d'adresser mes remerciements au personnel enseignant de l'école nationale polytechnique, particulièrement Mme HAMAMI et Mr BOUSBIA, pour leurs conseils, leur aide et leur contribution à ce travail.

Mes amis de l'INI et de BAB-EZZOUAR savent quel plaisir cela a été de travailler à leurs côtés.

Pour terminer, je réserverai des remerciements très particuliers et très chaleureux à Mr A. BOUZIT pour sa précieuse aide documentaire et sa gentillesse.

# SOMMAIRE



<b>Introduction.....</b>	<b>3</b>
--------------------------	----------

## **Chapitre I GENERALITES**

I.1 Introduction.....	6
I.2 Principe général d'une carte de Visualisation.....	7
1.3 Présentation des cartes de visualisation standard.....	12
I.3.1 Carte d'écran monochrome MDA.....	13
I.3.2 Carte graphique Hercules HGC.....	13
I.3.3 Carte d'écran couleur CGA.....	14
I.3.4 Carte graphique EGA/VGA.....	14

## **Chapitre II LE CONTROLEUR DE VISUALISATION**

II.1 Présentation du MC 6845.....	18
II.1.1 Signaux du MC 6845.....	18
II.1.2 registres du Mc 6845.....	20
II.2 Fonctionnement du MC 6845.....	25

## **Chapitre III PRESENTATION GENERALE DU 80286**

III.1 Description du 80286 et des différents registres.....	28
III.2 Organisation de la mémoire.....	30
III.3 Les Entrées-Sorties de l'IBM PC.....	32
III.4 Le bus AT de l'IBM PC.....	33

## **Chapitre IV CARTE GRAPHIQUE COULEUR COMPATIBLE PC**

IV.1 Aspect conception et réalisation.....	35
IV.2 Décodage et commande du CRTIC.....	41
IV.3 Utilisation de la RAM vidéo.....	46
IV.4 Visualisation de l'image.....	49
IV.5 Tests et programmation de la carte.....	52

## Chapitre V NOTIONS SUR LA GESTION DE LA CARTE

V.1 Langage assembleur.....	58
V.2 Exemple d'application.....	63
<b>Conclusion.....</b>	<b>67</b>

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

### ANNEXES

**Annexe A** Brochage du Bus d'extension de l'IBM PC

**Annexe B** Correspondance entre les emplacements de l'écran et les positions mémoire.

**Annexe C** Schéma de cablage de la carte graphique

**Annexe D** Programme d'affichage des couleurs

يندرج أنشطة ، ويتمثل في تصور وإنتاج خريطة خطية بيانية من أجل الحاسوب (إي. ب. م) مع المتوافقات وتمكينه من أداء وظيفة الإعلان الخطي على شاشة التلفزيون

ستكون هذه الخريطة مسيرة من طرف الحاسوب الإجمالي MOTOROLA 6845 وتسمح بالإعلان الملون من طراز 512 x 256 المخصص للإعلانات الموجهة للعموم ( وصول ومغادرة في المطارات ، ومحطات القطارات ، أسعار العملات في البنوك )

**RESUME :**

Ce projet s'inscrit dans le cadre des activités du centre de développement des technologies avancées (C.D.T.A) et consiste en la conception et la réalisation d'une carte graphique pour IBM-PC et compatibles qui l'enrichiront de la fonction d'affichage graphique sur écran de télévision. Cette carte sera gérée par le coprocesseur graphique 6845 de MOTOROLA et permettra un affichage couleur de 512x256 pixels destiné aux informations grand public (départ-arrivée dans les aéroports et les gares, cours de devises dans les banques).

**SUMMARY**

This report is entering in the CDTA's activities and consists in the conception and the realisation of a graphic adapter for IBM-PC and compatible which will improve it with graphic display on television screen. This adapter will be managed by the graphic processor 6845 of MOTOROLA and will offer a color display (512x256 pixels), it will be devoted to the general public (departure-arrivals in the airports and stations, rate of exchange in banks...).

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

## INTRODUCTION

## INTRODUCTION

Le progrès technologique dans le domaine des circuits intégrés a permis l'usage domestique et l'achat de micro-ordinateurs à des prix de plus en plus abordables. De tels systèmes peuvent être manipulés par des utilisateurs non spécialisés dans le domaine en utilisant des langages évolués et des logiciels spécialisés dont la plupart utilisent le graphisme comme l'une des méthodes les plus naturelles dans la communication homme/machine.

En effet, presque toutes les applications de bureautique, de CAO, de traitement de texte et même les systèmes d'exploitation (tels que celui du MACHINTOSH ou l'environnement windows sur PC ) présentent des interfaces graphiques pour l'utilisateur .

La tendance actuelle pour améliorer les performances de ces systèmes est à l'utilisation de processeurs spécialisés dans le graphisme pour débarrasser le microprocesseur central de toute tâche annexe pouvant le ralentir dans ses exécutions.

Le présent travail s'inscrit dans cette optique ; offrir à un utilisateur pas forcément spécialisé dans le domaine informatique une carte graphique couleur de format 512x256 pixels compatible PC à base du coprocesseur graphique 6845 de MOTOROLA présentant de bonnes performances tout en étant à faible prix .



Des notions générales sur les systèmes graphiques sont données dans le premier chapitre de ce mémoire. Le deuxième et troisième chapitre présentent respectivement le coprocesseur graphique et le microprocesseur 80286 d'INTEL en décrivant leurs différentes parties fonctionnelles .

La partie essentielle qui est la conception et la réalisation de la carte graphique est décrite dans le quatrième chapitre. Le dernier chapitre est réservé à une approche logicielle où l'on parlera du langage assembleur 80286 d'INTEL et l'on développera quelques programmes de gestion de cette carte.

**C H A P I T R E I**

## I.1 INTRODUCTION

Les images prennent une importance croissante de jour en jour dans notre vie, à l'opposé du texte car elles permettent un plus grand débit d'informations et elles sont comprises globalement et non séquentiellement par l'être humain .

De plus, le développement prodigieux qu'a connu le domaine de la microélectronique ces dernières années, a permis la diversité des circuits spécialisés tels que coprocesseurs graphiques - appelés à travailler en conjonction avec le microprocesseur principal et optimisés à une tâche particulière - et de ce fait a imposé le graphisme dans le domaine de l'informatique professionnelle et domestique.

Cette présence se confirme par l'existence de dizaines de cartes de visualisation sur le marché des IBM-PC et compatibles mais pratiquement toutes respectant l'un des standards suivants :

- \* La carte d'écran monochrome **MDA**.
- \* La carte d'écran couleurs **CGA**.
- \* La carte graphique Hercules **HGC**.
- \* La carte graphique couleur **EGA**.

Ceci s'applique aussi aux cartes **VGA** et **superVGA** dernière génération des cartes de visualisation initialement conçues pour les nouveaux systèmes PS, elles présentent une totale compatibilité avec les cartes **EGA** qui les ont précédées et par rapport auxquelles elles comportent simplement quelques

possibilités supplémentaires.

Toutes ces cartes possèdent des caractéristiques et des performances différentes mais elles ont quelque chose en commun : un même principe de fonctionnement que nous allons d'ailleurs développer dans le paragraphe suivant .

## I.2 PRINCIPE GENERALE D'UNE CARTE DE VISUALISATION

Un système de visualisation contient en général quatre éléments (voir fig I.2.1).

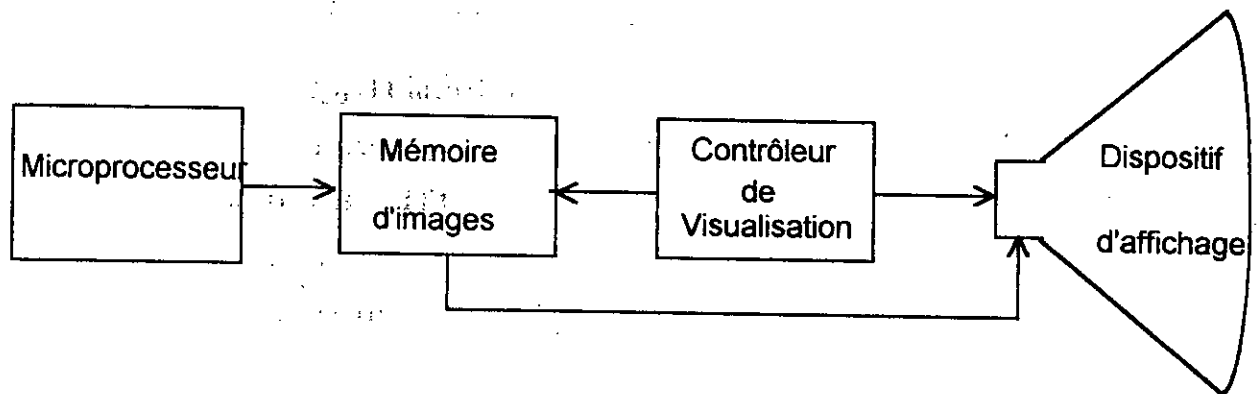


fig I.2.1 Schéma bloc d'un système de visualisation

- Le **microprocesseur** se charge de l'allumage des points images et du stockage dans la mémoire vidéo.
- la **mémoire** contient l'image sous forme d'informations relatives à chaque point de l'écran
- Le **contrôleur** de visualisation balaye l'écran tout en affichant le contenu de la mémoire vidéo.
- Le **moniteur** est le dispositif d'affichage de l'image (écran).

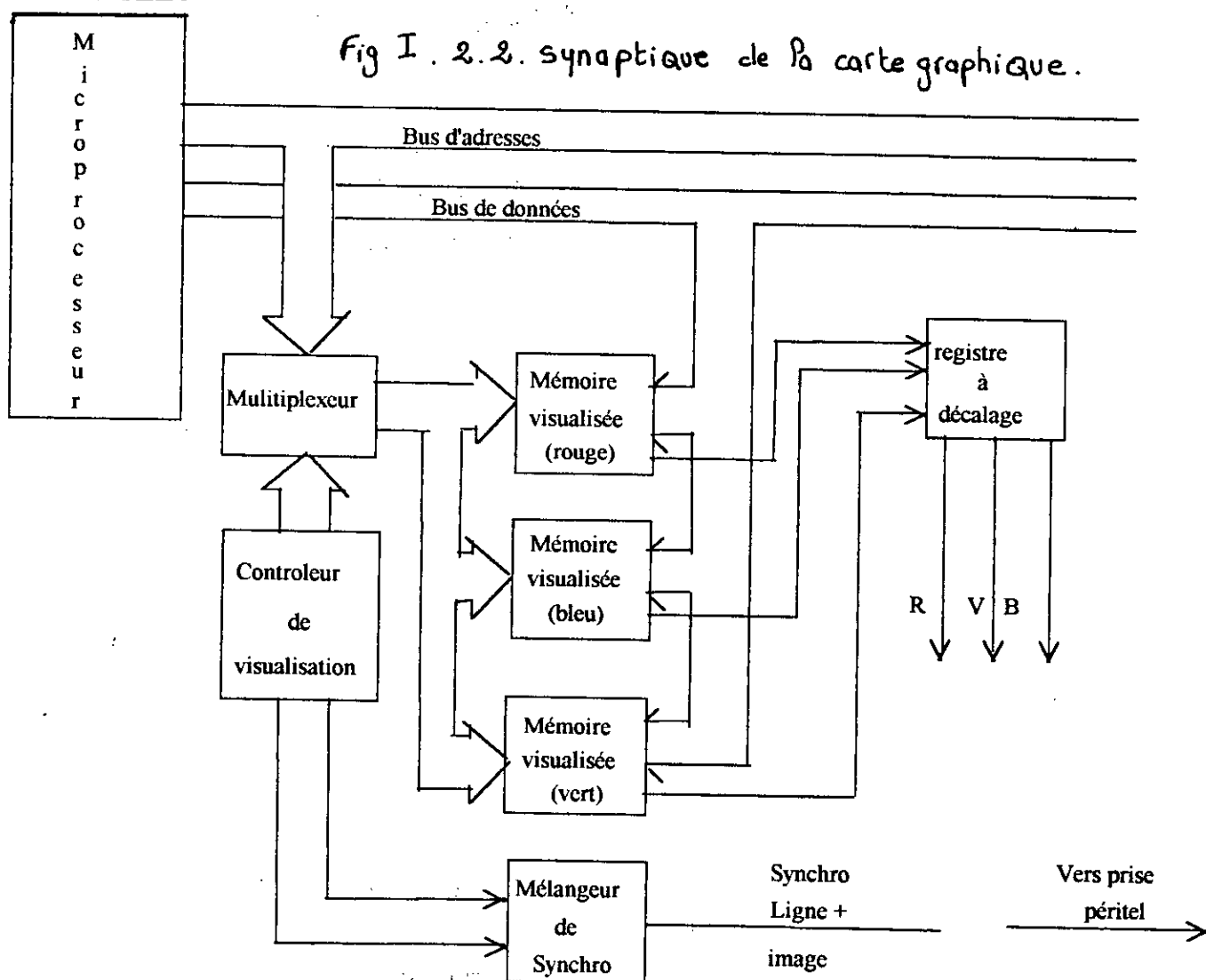
Les systèmes graphiques sont caractérisés par le nombre de couleurs disponibles ainsi que par le nombre de points -résolution- qui peuvent être tracés sur l'écran.

La carte graphique que nous avons réalisée a une résolution de 512x256 pixels et permet un affichage simultané de huit couleurs, son synoptique est donné (fig I.2.2). Sur ce schéma nous retrouvons les quatre éléments essentiels d'un système de visualisation :

Le dispositif d'affichage est un écran de télévision couleur.

Le contrôleur de visualisation est le coprocesseur graphique 6845 de **MOTOROLA**

Le processeur de contrôle est un microprocesseur **INTEL**.



Quant à la mémoire, elle est constituée de trois plans, de sorte qu'à chaque point de l'image correspond 3 bits de la mémoire reflétant l'état des signaux de chrominance suivant les codes donnés fig(I.2.3).

ROUGE	BLEU	VERT	RESULTANTE
0	0	0	NOIR
0	0	1	VERT
0	1	0	BLEU
0	1	1	CYAN
1	0	0	ROUGE
1	0	1	JAUNE-BRUN
1	1	0	MAGENTA
1	1	1	BLANC

FIG I.2.3 LE CODAGE DES COULEURS

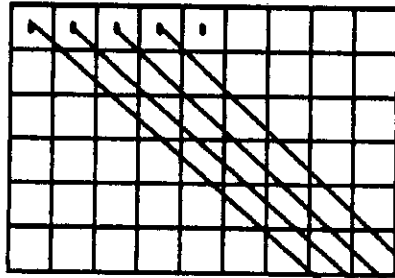
Le principe d'une carte de visualisation à base du 6845 qu'elle soit alphanumérique ou bien graphique et quelque soit sa résolution reste le même et repose sur une mémoire à double accès.

Dans cette mémoire, le microprocesseur vient écrire les points à allumer (fig I.2.4) et indépendamment de cela et en permanence le contrôleur de visualisation adresse cette mémoire au rythme des signaux d'adressage qu'il produit.

**Remarque :**

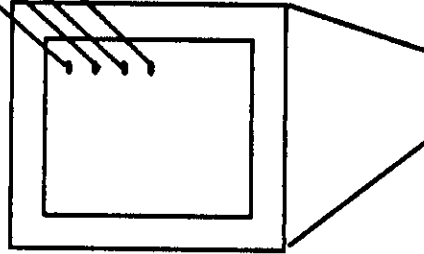
Une carte de visualisation alphanumérique ne sait qu'afficher des caractères prédéfinis respectant des dessins établis à l'avance. Son domaine de représentation est donc limité tandis qu'une carte de visualisation graphique est capable d'afficher n'importe quelle forme sur un écran car elle sait piloter individuellement chaque point de ce dernier en sachant lequel doit être allumé et dans quel emplacement de la mémoire il se trouve.

**CONTENU DE LA MEMOIRE**



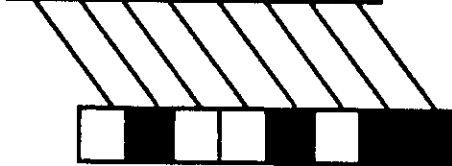
**4.1 PRINCIPE GENERAL**

**POINTS- IMAGE SUR L'ECRAN**



1 0 1 1 0 1 0 0

**OCTET DE CARACTERES BINAIRES**



**POINTS-IMAGE**

**figI.2.4 Relation entre le contenu de la mémoire et l'image.**

La carte graphique que nous avons réalisée est une carte graphique couleur dont l'affichage se fait sur écran de télévision.

Premier aspect - **l'affichage d'information couleur** - nécessite une approche matérielle qui consiste à structurer la mémoire en plans couleur comme il a été montré (fig I.2.2) le processeur se charge de stocker les informations relatives à chaque plan et le coprocesseur graphique adresse simultanément les trois plans mémoire.

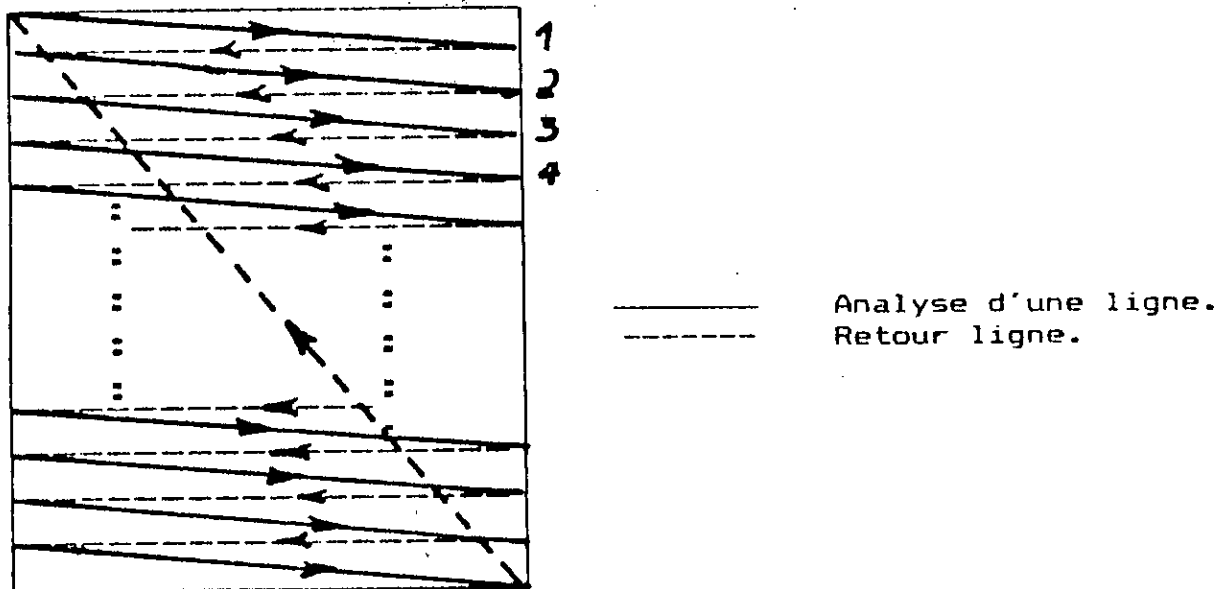
Deuxième aspect - **L'affichage sur écran de télévision** - nécessite une approche logicielle qui consiste en la programmation du contrôleur de visualisation de manière à ce qu'il produise des signaux de contrôle conformes aux normes de télévision qui sont la durée d'une ligne et celle d'une trame:

Le déplacement du spot de l'écran commence toujours par le coin supérieur gauche de l'écran et se fait de gauche à droite à une vitesse bien définie pour constituer une ligne, puis le spot retourne très rapidement à gauche et un peu en dessous de la ligne ainsi balayée pour constituer la ligne suivante et ce jusqu'en bas de l'image(trame), il remonte ensuite très rapidement en haut à gauche pour pouvoir former l'image suivante (fig I.2.5).

Compte tenu des phénomènes de persistance rétinienne et de la rémanence du phosphore, l'écran doit être entièrement balayé de nombreuses fois par seconde pour produire à nos yeux l'impression d'une image fixe exempt de scintillement, cette répétition dépend de la fréquence du secteur. Dans notre cas, elle est de 25



images par seconde, ce qui correspond à une trame de durée 20ms et à une durée de ligne de 64  $\mu$ s (une image contient 625 lignes).



**fig. I.2.5 Balayage télévision**

Ce sont ces deux valeurs que le coprocesseur doit respecter afin d'obtenir un affichage correct sur l'écran de télévision et nous verrons dans le chapitre suivant comment programmer le 6845 pour les avoir.

### **I.3 PRESENTATION DES CARTES DE VISUALISATION STANDARDS**

Comme il a été dit précédemment toutes les cartes vidéo existant sur le marché des IBM-PC respectent un des quatre standards cités en introduction. Nous présenterons dans ce paragraphe les principales caractéristiques de ces cartes .

### I.3.1 CARTE D'ECRAN MONOCHROME **MDA**:

Cette carte est apparue en même temps que le PC en 1981. Elle permet un affichage de texte de très bonne qualité, c'est donc une carte alphanumérique qui ne peut afficher qu'un nombre limité de caractères prédéfinis et représentés sous forme d'une matrice de neuf points horizontalement sur quatorze verticalement.

Ces informations sont stockées dans une ROM de 4 K octets, quant à sa RAM vidéo, sa capacité est de 8K octets et commence à l'adresse B0000H.

### I.3.2 CARTE GRAPHIQUE HERCULES **HGC**:

Cette carte créée par la société **HERCULES** en 1982, eut un franc succès car elle permettait de combler une lacune des systèmes vidéo **IBM**. En effet à cette époque il n'existait que deux systèmes vidéo pour le PC , l'un permettait d'afficher du texte de bonne qualité à l'exclusion de tout graphisme (**MDA**) et l'autre permettait les deux mais avec une qualité nettement inférieure (**CGA**). C'est pourquoi on imagina chez **HERCULES** de combiner un affichage de texte excellent avec la possibilité graphique .

Sur un moniteur monochrome ordinaire, cette carte offre un mode de texte de 80 caractères horizontalement sur 25 verticalement et un mode graphique d'une résolution de 720x348 points. Sa mémoire vidéo est de 64K octets de RAM répartie en deux pages écran:la première s'étend de l'adresse B0000H à B7FFFH pour le mode texte et la deuxième de B8000H à BFFFFH pour le mode graphique.

### I.3.3 CARTE D'ECRAN COULEUR (CGA) :

La carte couleur **IBM** soutient deux modes de texte et trois de graphisme.

Outre le mode normal qui affiche 25 caractères verticalement sur 80 horizontalement, un deuxième mode de 25x40 est également disponible. les caractères sont représentés par une matrice de 8x8, ce qui entraine une qualité d'affichage nettement inférieure à celle des cartes monochromes .

En mode graphique, la carte **CGA** offre trois possibilités de représentation

- 160x100 points avec 16 couleurs (rarement utilisé).
- 320x200 points avec quatre couleurs.
- 640x200 points avec deux couleurs .

La RAM vidéo est de 16 K octets et débute à l'adresse B8000H.

### I.3.4 CARTES GRAPHIQUE **EGA/VGA**

Les cartes **VGA** ont été initialement conçues pour les systèmes PS mais une version compatible pour PC existe. Ces cartes suivent le standard des cartes **EGA** et surclassent largement les autres cartes vidéo par leurs possibilités d'affichage graphique et leur aptitude à émuler d'autres cartes vidéo. IL est évident que ces caractéristiques ne peuvent pas être obtenues par un contrôleur vidéo traditionnel "6845 de MOTOROLA" comme pour les autres cartes vidéo . IL est donc remplacé sur les cartes **EGA/ VGA** par un circuit spécifique LSI qui a été développé tout spécialement pour la carte **EGA** .

Les cartes SVGA (dernière génération des cartes graphiques) ont carrément éclipsé les précédentes.

En effet, depuis les années 92 presque tous les PC sont équipés de cartes SVGA et rares sont les cas où les cartes **EGA** sont encore utilisées. Quant aux cartes **MDA**, **CGA** et **HGC** elles ne sont plus commercialisées.

*le tableau suivant regroupe les caractéristiques des quatre cartes.*

Cartes	MDA	HGC	CGA	EGA/VGA
Caractéristiques				
Mode de Visualisation (1)	AN	AN*GR	AN+GR	AN+GR
Résolution	-	720*348	160*100 320*200 640*200 640*200	320*200 640*200 640*350 640*480
Format de l'écran	80*25	80*25	80*25 80*40	80*25 80*40
Couleurs	monochrome	monochrome	16 4 2	16 16 monochrome+16 monochrome+16
Matrice du caractère	9*14	8*16	8*8	9*14 8*8
Contrôleur de visualisation	6845	6845	6845	Circuit spécialisé LSI
Adresses RAM	B000:0000	B000:0000	B800:0000	A000:0000
Vidéo HEX	B000:0FFF	B000:FFFF	B800:4000	A000:FFFF
Capacité	4KO	64KO	16KO	64KO

(1) AN = alphanumérique  
GR = graphique

Le tableau suivant indique le codage des modes vidéo de la fonction 00H du BIOS vidéo. Cette fonction permet d'activer tous les modes standards des cartes MDA jusqu'aux cartes VGA grâce à ce codage.

Codage des modes vidéo par la fonction 00h du BIOS vidéo		
Code	Mode	Carte
00h	40*25 caractères de texte, 16 couleurs, pas d'affichage couleur	CEV
01h	40*25 caractères de texte, 16 couleurs	CEV
02h	80*25 caractères de texte, 16 couleurs, pas d'affichage couleur	CEV
03h	80*25 caractères de texte, 16 couleurs	CEV
04h	320*200 points graphiques, 4 couleurs	CEV
05h	320*200 points graphiques, 4 couleurs	CEV
06h	640*200 points graphiques, 2 couleurs	CEV
07h	80*25 caractères, monochrome	MH E*
08h	réservé	---
09h	réservé	---
0Ah	réservé	---
0Bh	réservé	---
0Ch	réservé	---
0Dh	320*200 points graphiques, 16 couleurs	EV
0Eh	640*200 points graphiques, 16 couleurs	EV
0Fh	640*350 points graphiques, monochrome	E*
10h	640*350 points graphiques, 16 couleurs	EV
11h	640*480 points graphiques, 2 couleurs	V
12h	640*480 points graphiques, 16 couleurs	V
13h	320*200 points graphiques, 256 couleurs	V
* Carte EGA sur moniteur MDA M = MDA H = Hercules C = CGA E = EGA V = VGA		

**C H A P I T R E   I I**

## **LE CONTROLEUR DE VISUALISATION**

Le contrôleur de visualisation contrôle le déroulement des opérations à l'intérieur de la carte vidéo et génère les signaux dans le temps dont le moniteur a besoin pour construire l'image. Le contrôleur que nous avons utilisé pour la réalisation de notre carte est le 6845 de Motorola. Il présente deux atouts très importants : un excellent rapport qualité prix et une très grande souplesse d'emploi.

### **II .1 PRESENTATION DU MC 6845**

Le MC 6845 est un circuit conçu par Motorola pour l'interfaçage entre le microprocesseur et l'écran de visualisation .

Les différents signaux qui constituent son brochage voir (fig II.1.1) permettent au CRTC (cathode ray tube controler) de gérer le moniteur, d'adresser la RAM écran et de communiquer avec le microprocesseur.

En plus de ces différents signaux, le 6845 possède 18 registres internes qui par leurs nombres et fonction justifient les nombreuses performances du CRTC.

#### **II.1.1 SIGNAUX DU 6845**

Le MC 6845 se présente sous forme d'un boîtier de 40 broches voir (fig II.1.1). Suivant leurs fonctions, ces signaux peuvent être représentés en trois catégories.

##### **II.1.1.a INTERFACE AVEC LE MICROPROCESSEUR**

Le CRTC s'interface sur le bus d'un microprocesseur par le bus de données bidirectionnelles (D0-D7) et en utilisant les signaux de contrôle.

Le bus de données bidirectionnelles permet le transfert des données entre les registres internes du CRTC et le processeur. Quant aux signaux de contrôle, ils autorisent et valident ces transferts.

Ces signaux de contrôle sont constitués de deux lignes ( $\overline{\text{CS}}$  et RS) pour la sélection du CRTC et de ses registres, d'une ligne  $\text{R}/\overline{\text{W}}$  pour spécifier la direction de transfert des données et finalement d'une entrée d'horloge E utilisée pour la validation du CRTC et la synchronisation de tout le système en réglant le problème du conflit d'accès mémoire entre les lignes d'adresse du CRTC et celles du microprocesseur.

#### **II.1.1 B CONTROLE DE VISUALISATION**

Le 6845 délivre des signaux qui permettent de commander l'écran de visualisation et sont au nombre de trois.

**VSYN** et **HSYN** : déterminent respectivement la position verticale et horizontale du spot électronique du tube cathodique sur l'écran et le troisième signal **DE** va délimiter une zone active au niveau de l'écran où la visualisation sera autorisée.

#### **II.1.1 C ADRESSAGE DE LA MEMOIRE**

Le CRTC fournit les adresses mémoires grâce aux quatorze lignes d'adresses (MA0-MA13) pour balayer les mémoires RAM d'écran et cinq lignes (RA0-RA4) pour la mémoire ROM.

Le 6845 présente l'avantage d'un double mode de visualisation, graphique et alphanumérique.

En mode alphanumérique, l'écran est géré en pavés dont la sélection se fait grâce aux lignes (MA0 - MA13), ce qui donne une possibilité d'adressage de 16 kilo-



pavés. Les lignes (RA0-RA4) vont déterminer le format de ces pavés ou plus exactement leurs hauteurs qui peut atteindre dans le cas du CRTC jusqu'à 32 lignes.

En mode graphique, les lignes RA0-RA4 et MA0-MA13 sont utilisées en série pour adresser jusqu'à 512 kilo-octets de RAM de visualisation et l'écran n'est plus géré en pavés mais en octets.

Le reste des signaux permet l'alimentation du CRTC, sa réinitialisation, la visualisation d'un curseur ainsi que la gestion du registre light-pen et finalement une entrée d'horloge CLK utilisée par le CRTC comme référence temporelle.

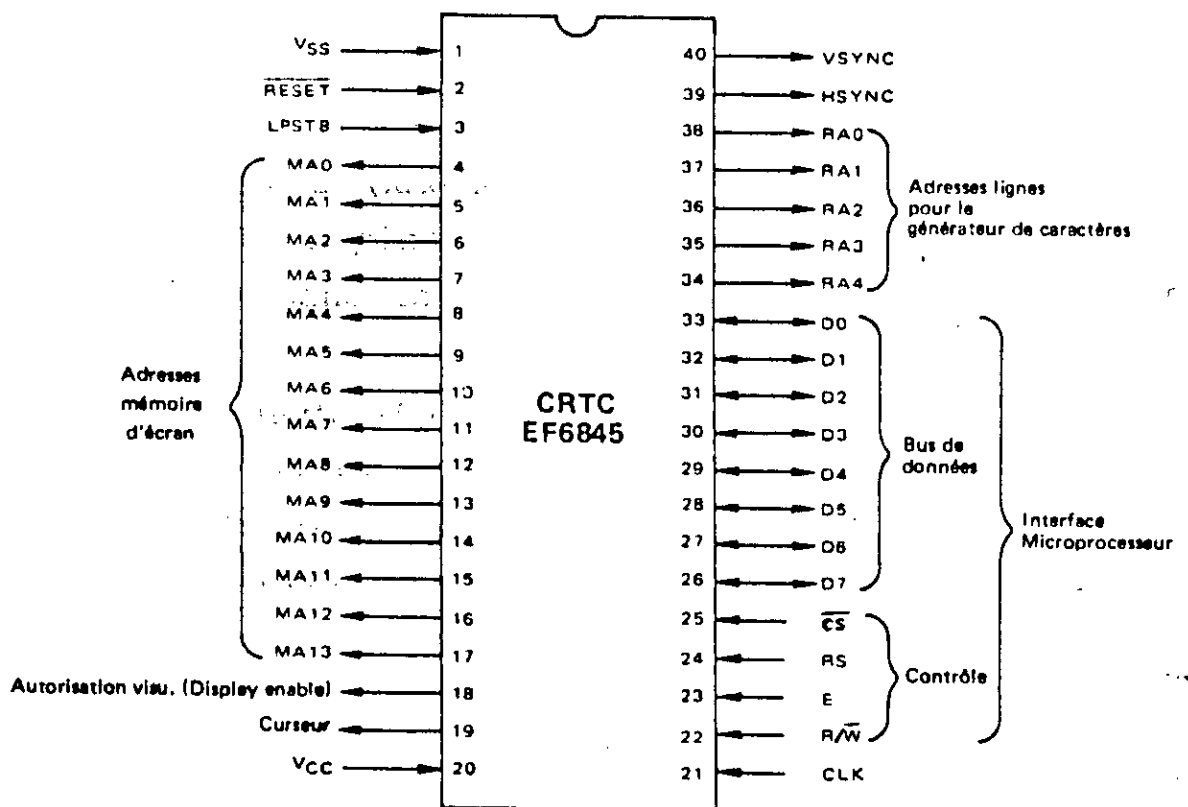


fig II.1.1 Brochage du 6845

## II. 1.2 REGISTRES DU 6845

Dans le CRTC 18 registres sont accessibles à partir du bus de données. L'adresse et la largeur de

Fig II.1.2

## AFFECTATION DES REGISTRES INTERNES

CS	RS	Adresse du registre					Registre =	Nom des registres	Unité de programm.	Lecture	Ecriture	Nombre de bits											
		4	3	2	1	0						7	6	5	4	3	2	1	0				
1	X	X	X	X	X	X	X	—	—	—	—	/	/	/	/	/	/	/	/	/	/	/	/
0	0	X	X	X	X	X	X	Registre d'adresses	—	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	0	0	0	0	R0	Horizontal total	Caractères	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	0	0	0	1	R1	Visualisation H.	Caractères	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	0	0	1	0	R2	Position synchro H.	Caractères	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	0	0	1	1	R3	Largeur synchro H.	Caractères	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	0	1	0	0	R4	Vertical total	Rangées car.	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	0	1	0	1	R5	Ajustement vertical	Lignes bal.	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	0	1	1	0	R6	Visualisation verticale	Rangées car.	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	0	1	1	1	R7	Position synchro vert.	Rangées car.	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	1	0	0	0	R8	Modes entrelacés	—	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	1	0	0	1	R9	Lignes adressées max.	Lignes bal.	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	1	0	1	0	R10	Début du curseur	Lignes bal.	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	1	0	1	1	R11	Fin du curseur	Lignes bal.	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	1	1	0	0	R12	Adresse de départ (fort)*	—	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	1	1	0	1	R13	Adresse de départ (faible)*	—	Non	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	1	1	1	0	R14	Curseur (fort)*	—	Oui	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	0	1	1	1	1	R15	Curseur (faible)*	—	Oui	Oui	/	/	/	/	/	/	/	/	/	/	/	/
0	1	1	0	0	0	0	R16	Light-pen (fort)*	—	Oui	Non	/	/	/	/	/	/	/	/	/	/	/	/
0	1	1	0	0	0	1	R17	Light-pen (faible)*	—	Oui	Non	/	/	/	/	/	/	/	/	/	/	/	/

Note (1) - Le bit 5 du registre départ du curseur est utilisé pour la période de clignotement et le bit 6 est utilisé pour sélectionner le clignotement ou non.

\* : fort : octet de poids fort

faible : octet de poids faible

chaque registre sont données en (Fig II.1.2).

Pour accéder à l'un des registres du CRTC, deux adresses sont nécessaires:

- La première positionne les deux signaux RS et  $\overline{CS}$  à l'état bas afin d'accéder au registre d'adresse et d'y inscrire l'adresse du registre désiré.
- La deuxième maintient  $\overline{CS}$  à l'état bas et fait passer RS à l'état haut afin d'accéder au registre désiré.

Exemple: Si l'on veut mettre la valeur 02H dans le registre de visualisation verticale, on doit charger la valeur 06H dans la première adresse ensuite la valeur 02H dans la deuxième adresse.

Les registres du CRTC peuvent être regroupés en deux catégories .

## **II .1.2 A REGISTRES HORIZONTAUX**

Ces registres sont au nombre de quatre et définissent les caractéristiques horizontales de l'écran, leurs contenus sont programmés en périodes de l'horloge CLK .

Le premier registre R0 détermine la fréquence horizontale du balayage. Pour cela, on mémorise dans ce registre la durée d'une ligne comme étant un multiple de la période caractère .

Le registre R1 contient le nombre de caractères qu'on veut visualiser, ce qui va déterminer une sorte de zone active sur l'écran. Cette zone peut se déplacer horizontalement grâce à la modification du contenu du registre R2, le dernier registre horizontal R3 détermine la durée du retour horizontal du rayon électronique .

Toutefois, ce registre ne reçoit pas directement la durée en temps mais le nombre de caractères qui auraient pu être sortis dans cet intervalle de temps, généralement cette durée est de 6 à 8 périodes caractères.

## II. 1.2.B REGISTRES VERTICAUX

Le CRTC possède six registres verticaux qui déterminent certaines caractéristiques verticales de l'image, ces registres sont programmés en nombre de rangée de caractères ou de lignes balayées .

Les deux premiers registres R4 et R5 déterminent la fréquence verticale. Pour cela, on mémorise la durée d'une image dans ces deux registres comme étant le nombre de rangées de caractères qui auraient pu être sortis dans cet intervalle de temps. Généralement, ce nombre n'est pas entier et donc la partie entière sera contenue dans R4 et la partie décimale dans R5 .

Comme pour les registres horizontaux, deux registres verticaux R6 et R7 vont définir respectivement une zone verticale visualisable et un déplacement vertical de cette zone .

Les deux autres registres R8, R9 vont déterminer le mode de balayage et le nombre de lignes maximum dans une rangée de caractères.

Une illustration de l'espace visualisable dans un moniteur vidéo en fonction du contenu des registres horizontaux et verticaux est donnée voir fig II.1.3.

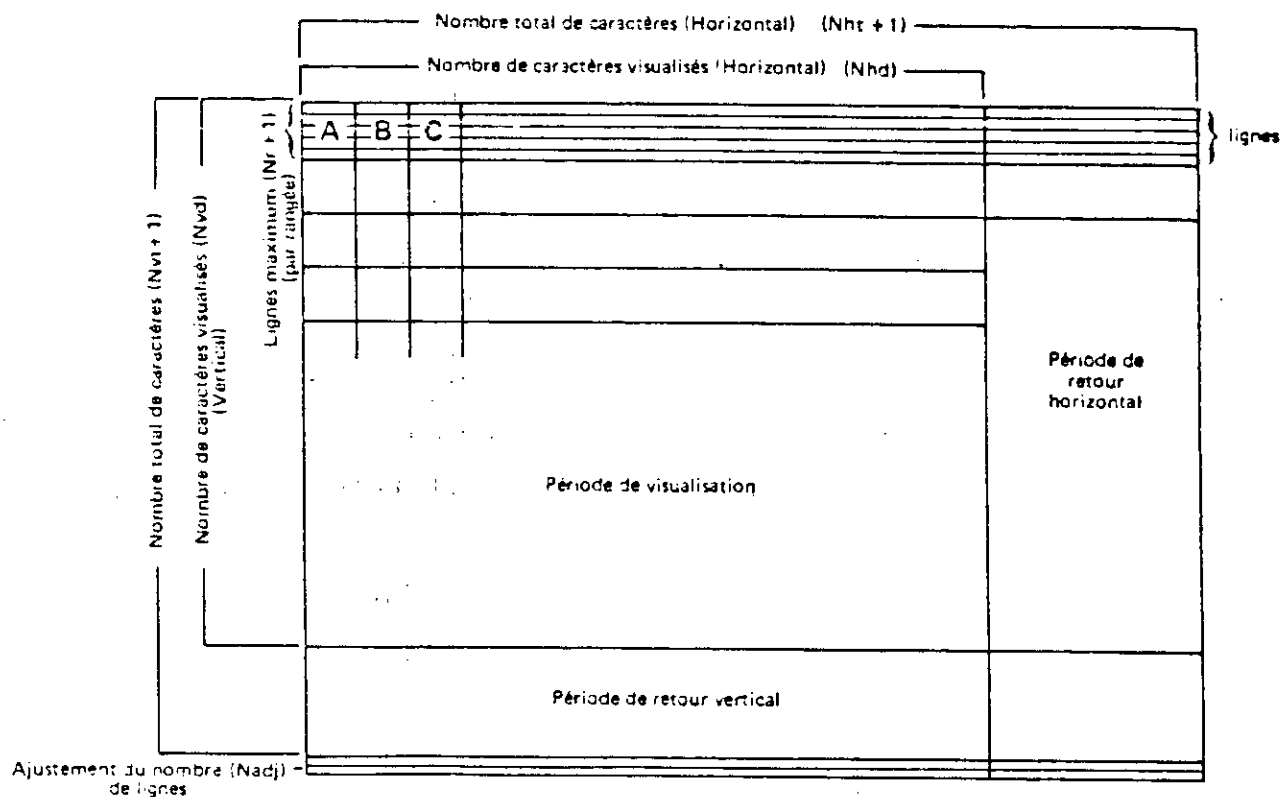


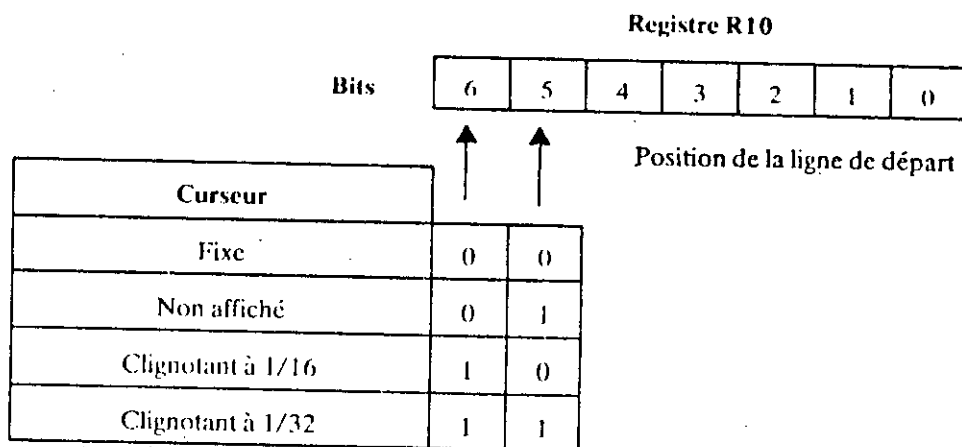
fig II.1.3 Illustration du format de l'écran du CRTC

Les huit autres registres restant permettent de doter le CRTC des fonctions supplémentaires suivantes:

\* **Le curseur**: les registres R10 et R11 permettent la visualisation d'un curseur pouvant aller jusqu'à 32 lignes de haut et ayant une fréquence de clignotement de  $1/16$  et  $1/32$  de la fréquence trame (FIG II.1.4). Le curseur peut être placé à n'importe quelle position de l'écran en changeant le contenu des registres R14 et R15.

\* **light-pen**: Les registres R16, R17 mémorisent l'adresse de la mémoire au moment où le signal LPSTB est validé.

**\*Adresse de départ:** Le contenu des registres R12 et R13 détermine l'adresse mémoire qui donnera le premier caractère visualisé sur l'écran.



Le registre R11 ne contient que la ligne d'arrivée.

**fig II.1.4. Interprétation du contenu des registres**

## II.2 FONCTIONNEMENT DU CRTC:

Pour comprendre le fonctionnement du 6845 il faudrait d'abord se pencher sur sa composition interne.

Le CRTC contient différents compteurs dont le contenu est continuellement comparé (dans des circuits de coïncidence) à celui des registres internes R0-R17.

Cette continuelle comparaison permet la génération des temps horizontaux et verticaux ainsi que la gestion du générateur d'adresses conformément à la programmation des registres internes et les maintient actives tant que le CRTC est alimenté.

Pour visualiser une image sur l'écran, le CRTC procède de la façon suivante :

Il adresse la RAM d'écran grâce aux lignes (MA0-MA13) en commençant à partir de l'adresse indiquée dans les registres R12-R13 et incrémente les adresses au rythme de l'horloge caractère CLK jusqu'à la fin de la première ligne.

Cet instant lui est indiqué par le registre R1, le générateur d'adresses est alors remis à son adresse de départ et le compteur de lignes (commande RA0-RA4) est incrémenté.

Le CRTC est donc prêt pour générer la deuxième ligne sur l'écran, il reprend le même processus pour chaque ligne jusqu'à une égalité entre le contenu du compteur ligne et celui du registre R9 (nombre de lignes adressées max). Cette égalité va remettre à zéro le compteur de lignes et incrémenter le compteur vertical ce qui va indiquer que le CRTC a visualisé une rangée de caractères.

Toutes les étapes précédentes vont être reprises pour chaque rangée de caractères, seulement cette fois le générateur d'adresses n'est pas remis à son adresse de départ comme pour chaque début de ligne mais est incrémentée à partir de la dernière adresse.

Quand toutes les rangées de caractères sont visualisées (ce qui est indiqué par le registre R6) tous les compteurs internes sont remis à leur état initial pour recommencer la visualisation d'une autre image.

**C H A P I T R E I I I**



## PRESENTATION GENERALE DU 80286

La carte graphique compatible PC que nous avons réalisée a été testée sur le système de développement **I2ICE d'INTEL** à base du 80286.

Une présentation générale de ce microprocesseur sera donnée dans ce chapitre.

### III.1 Description du 80286 et des différents registres:

Le microprocesseur 80286 a été créé par la société **INTEL** en 1981. Il est constitué de 24 lignes (A0-A23) et 16 lignes de données (D0-D15).

Il présente par rapport au 8088/8086 l'avantage d'un double mode d'adressage qui sont : le mode **réel** et le mode **virtuel** (protégé).

Le mode réel lui assure une parfaite compatibilité ascendante en code objet avec les 8086/8088, il peut adresser jusqu'à un Méga-octets de mémoire.

Le mode protégé ne lui permet une compatibilité qu'en code source et l'espace d'adressage peut atteindre 16 Méga-octets de mémoire virtuelle.

En plus de son double mode de fonctionnement, le 80286 est un microprocesseur à "pipeline" c'est à dire que les cycles de recherche et d'exécution ne sont pas séparés, ce qui accroît la vitesse de traitement.

Le 80286 possède 15 registres internes de 16 bits que l'on peut regrouper en 4 sous ensembles (fig III.1.1).

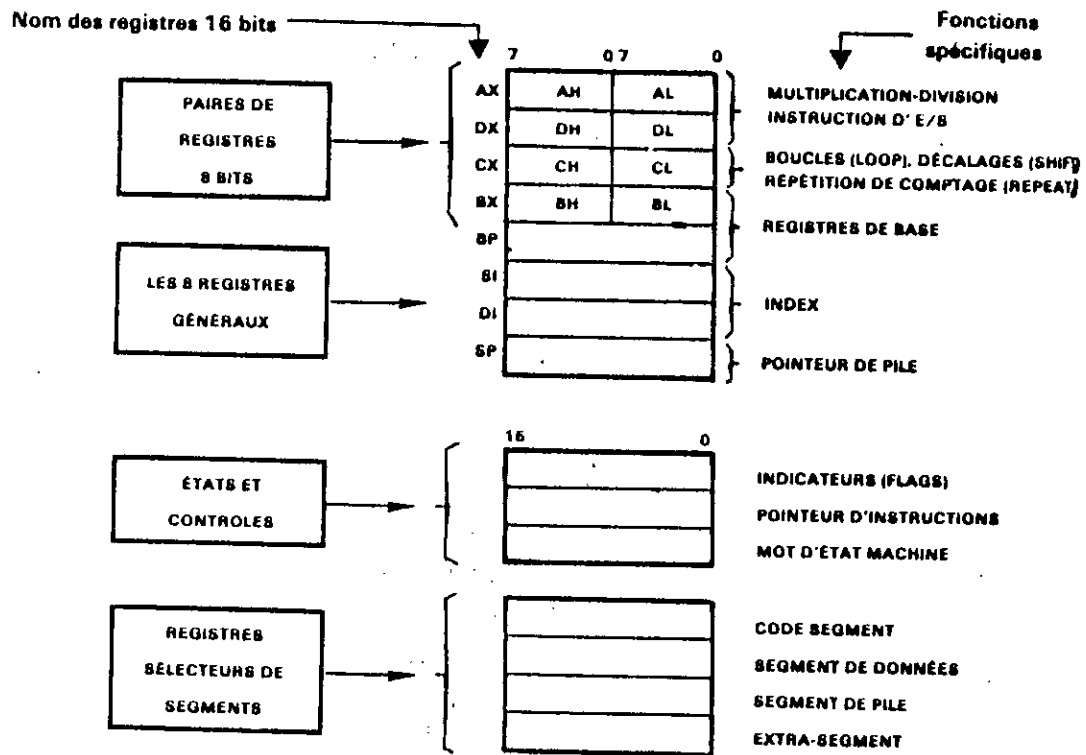


fig III.1.1 Le jeu de registres du 80286

- .Les registres généraux pour les données: AX, BX, CX, DX.
- .Les registres généraux pour les pointeurs et les index: SP, BP, SI, DI.
- .Les registres sélecteurs de segments : CS, DS, SS et ES.
- .Les registres d'état et de contrôle: F, IP, MSW.

## III.2 ORGANISATION DE LA MEMOIRE

Le 80286 possède en mode réel un champ d'adressage de un Méga-octets de mémoire, ce qui nécessite un décodage sur vingt bits. Or nous avons vu au paragraphe précédent que tous les registres du 80286 sont à seize bits au maximum donc pour résoudre ce problème, on divise l'espace total adressable en seize segments de soixante Kilo-octets. Chaque position mémoire sera localisée par deux adresses sur seize bits, la première spécifiera dans quel segment elle se trouve et la deuxième, le déplacement (OFFSET) par rapport au début de ce segment fig(III.2.1).

Référence mémoire	Registre de segment		Offset (décalage)
	Par défaut	Remplacement possible par	
Recherche d'instruction	CS	Aucun	IP
Opération sur pile	SS	Aucun	SP
Variable (les exceptions suivent)	DS	CS, ES, SS	Adresse effective
Source, par une chaîne	DS	CS, ES, SS	SI
Destination, pour une chaîne	ES	Aucun	DI
Registre BP remplaçant le registre de base	SS	CS, DS, ES	Adresse effective
Registre BX remplaçant le registre de base	DS	CS, ES, SS	Adresse effective
Pour des données externes globales	ES	Mode protégé uniquement	

**Fig III.2.1 Registres pour le calcul de l'adresse physique**

La première adresse sera chargée dans un registre segment et la deuxième, dans un des registres de déplacement.

Pour reconstituer l'adresse réelle sur vingt bits, le microprocesseur multiplie le contenu du registre segment par seize (ce qui équivaut à ajouter quatre bits à zéro en faible poids) et à ce résultat il additionne le contenu du registre déplacement. Les seize segments de la mémoire du PC sont indiqués (FIG III.2.2).

Organisation de la mémoire du PC		
Bloc	Adresse	Contenu
15	F000:0000 - F000:FFFF	BIOS
14	E000:0000 - E000:FFFF	libre pour Insertion de ROM
13	D000:0000 - D000:FFFF	libre pour Insertion de ROM
12	C000:0000 - C000:FFFF	ROM BIOS supplémentaire
11	B000:0000 - B000:FFFF	RAM d'écran
10	A000:0000 - A000:FFFF	RAM d'écran supplément (EGA/VGA)
9	9000:0000 - 9000:FFFF	RAM de 576 Ko à 640 Ko
8	8000:0000 - 8000:FFFF	RAM de 512 Ko à 576 Ko
7	7000:0000 - 7000:FFFF	RAM de 448 Ko à 512 Ko
6	6000:0000 - 6000:FFFF	RAM de 384 Ko à 448 Ko
5	5000:0000 - 5000:FFFF	RAM de 320 Ko à 384 Ko
4	4000:0000 - 4000:FFFF	RAM de 256 Ko à 320 Ko
3	3000:0000 - 3000:FFFF	RAM de 192 Ko à 256 Ko
2	2000:0000 - 2000:FFFF	RAM de 128 Ko à 192 Ko
1	1000:0000 - 1000:FFFF	RAM de 64 Ko à 128 Ko
0	0000:0000 - 0000:FFFF	RAM de 0 Ko à 64 Ko

**Fig III.2.2 Organisation de la mémoire**

Les dix premiers segments sont réservés à la RAM de la mémoire centrale, le premier segment est destiné à contenir le système d'exploitation et le reste de cette RAM est réservée aux programmes de l'utilisateur (voir chapitre V).

### III.3 LES ENTREES - SORTIES DU 80286 :

Le 80286 dispose de quelques instructions spécifiques d'E/S SUR HUIT OU SEIZE BITS (IN et OUT). Cette adressage distinct de celui de la mémoire permet au micro-processeur de gérer 64 Kilo-octets d'E/S sans chevaucher les positions mémoire. Le tableau suivant (FIG III.3.3) indique les adresses des ports attribués aux principaux circuits du PC.

Circuit	PC/XT	AT
Contrôleur DMA (8237A-5)	000-00F	000-01F
Contrôleur d'interruption (8259A)	020-021	020-03F
Temporisateur	040-043	040-05F
Interface périphérique programmable (PPI 8255A-5)	060-063	néant
Clavier (8042)	néant	060-06F
Horloge en temps réel (MC146818)	néant	070-07F
Registre de page DMA	080-083	080-09F
Contrôleur d'interruption n2 (8259A)	néant	0A0-0BF
Contrôleur DMA n2 (8237A-5)	néant	0C0-0DF
Coprocasseur arithmétique	néant	0F0-0F1
Coprocasseur arithmétique	néant	0F8-0FF
Contrôleur de disque dur	320-32F	1F0-1F8
Manette de jeux (Joysticks)	200-20F	200-207
Unité d'extension	210-217	néant
2nde Imprimante parallèle	néant	278-27F
Seconde interface série	2F8-2FF	2F8-2FF
Carte de prototype	300-31F	300-31F
Carte de réseau	néant	360-36F
1ère Imprimante parallèle	378-37F	378-37F
Carte d'écran monochrome et 1re Imprimante parallèle	3B0-3BF	3B0-3BF
Carte vidéo couleur/graphique	3D0-3DF	3D0-3DF
Contrôleur de disquette	3F0-3F7	3F0-3F7
Première interface série	3F8-3FF	3F8-3FF

■ (adresses exprimées en hexadécimal)

fig III.3.1 Adresses des principaux circuits du PC

### III.4 LE BUS AT DE L'IBM PC:

Pour donner plus de puissance à un micro-ordinateur et de confort à l'utilisateur, la configuration de base d'un système est souvent améliorée (ajout d'une deuxième unité de lecture de disquette, extension de la mémoire centrale, remplacement de l'écran monochrome de base ne pouvant afficher que du texte par un écran haute résolution permettant de travailler avec du graphique, etc...).

Une autre façon d'améliorer les performances d'un micro-ordinateur est de lui adjoindre des cartes d'extensions. Ainsi, sur la carte mère de l'IBM PC /AT et compatibles, huit connecteurs internes sont disponibles à cet effet. Tous les signaux nécessaires au fonctionnement des cartes d'extension sont décrits en Annexe I.

**C H A P I T R E   I V**

## CARTE GRAPHIQUE COMPATIBLE PC

La carte graphique a été réalisée en utilisant la technique du **WRAPPING** car cette dernière présente une très grande souplesse d'emploi. Nous présenterons dans ce chapitre les différents modules de la carte.

### IV.1 ASPECT REALISATION ET CONCEPTION

Au chapitre I, nous avons décrit les quatre éléments du schéma bloc d'un système de visualisation. La carte graphique comprend la RAM vidéo et le contrôleur de visualisation, elle communique avec le microprocesseur grâce au bus d'extension de l'IBM PC et avec le dispositif d'affichage (Télévision) à travers un connecteur femelle de neuf broches reliant la carte à la prise péritélévision.

La carte a un format de 512x256 pixels, ce qui nécessite une mémoire de 16 Kilo-octets. Comme chaque pixel doit être codé sur 3 bits, la capacité mémoire totale sera de 48 kilo-octets répartis en 3 plans. Ces trois plans de 16 kilo-octets chacun seront vus par le CRTC comme un seul de 16 kilo-octets et seront adressés par le même bus d'adresses constitué de 14 lignes RA0-RA3 et MA0-MA9.

Ces 16 kilo-octets vont être répartis au niveau de l'écran de la façon suivante:

**Horizontalement:** les 512 pixels vont être regroupés en 64 positions de 8 pixels chacune et sera donc représentée



au niveau de la mémoire par un octet.

**Verticalement:** les 256 pixels vont être regroupés en 16 lignes pour former 16 rangées.

Une illustration de cette interprétation est donnée (fig IV.1.1) et les adresses de chaque position sont données en annexe.

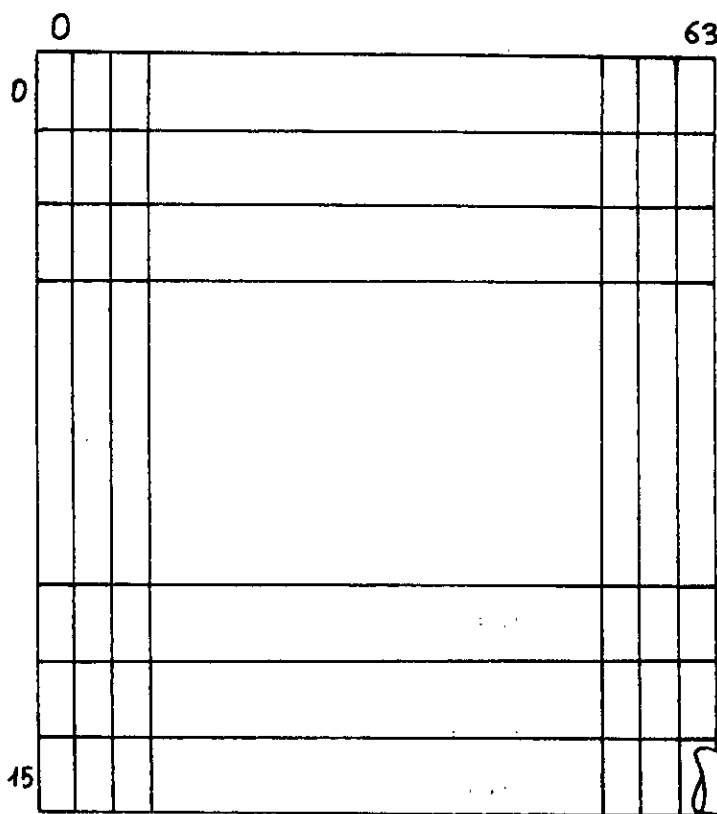


Fig IV. 1.1. a. représentation de P'écran.

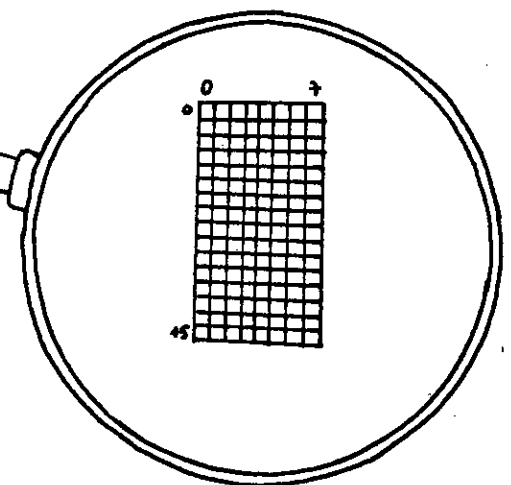


Fig IV. 1.1. b. représentation d'un caractère.

Cette RAM va constituer un des modules de la carte graphique.

En effet, la représentation modulaire (fig IV.1.2) peut-être donnée par :

- \* Module de visualisation
- \* Module mémoire
- \* Module CRTC

Le CRTC effectue des transferts avec le microprocesseur à travers un buffer commandé par un module de décodage.

Le CRTC va produire les signaux de contrôle de visualisation qui vont attaquer directement le module de visualisation.

Quant aux signaux d'adressage, ils seront multiplexés avec le bus d'adresses en provenance du microprocesseur avant d'attaquer le module mémoire constitué de trois RAMS de 16 kilo-octets. Le transfert de données se fait à travers trois buffers commandés par un module de décodage. Les trois bus de données de la mémoire vont subir au niveau du module de visualisation une sérialisation à travers des registres à décalage pour constituer les trois signaux de chrominance.

Ces différents modules sont synchronisés grâce à un séquenceur (fig IV.1.3).

La synchronisation de ces différents modules est liée aux contraintes de balayage télévision. En effet, la durée d'une ligne est de 64  $\mu$ s. Durant ce laps de temps le CRTC doit faire sortir de la mémoire 64 caractères et se positionner en début de la ligne suivante.

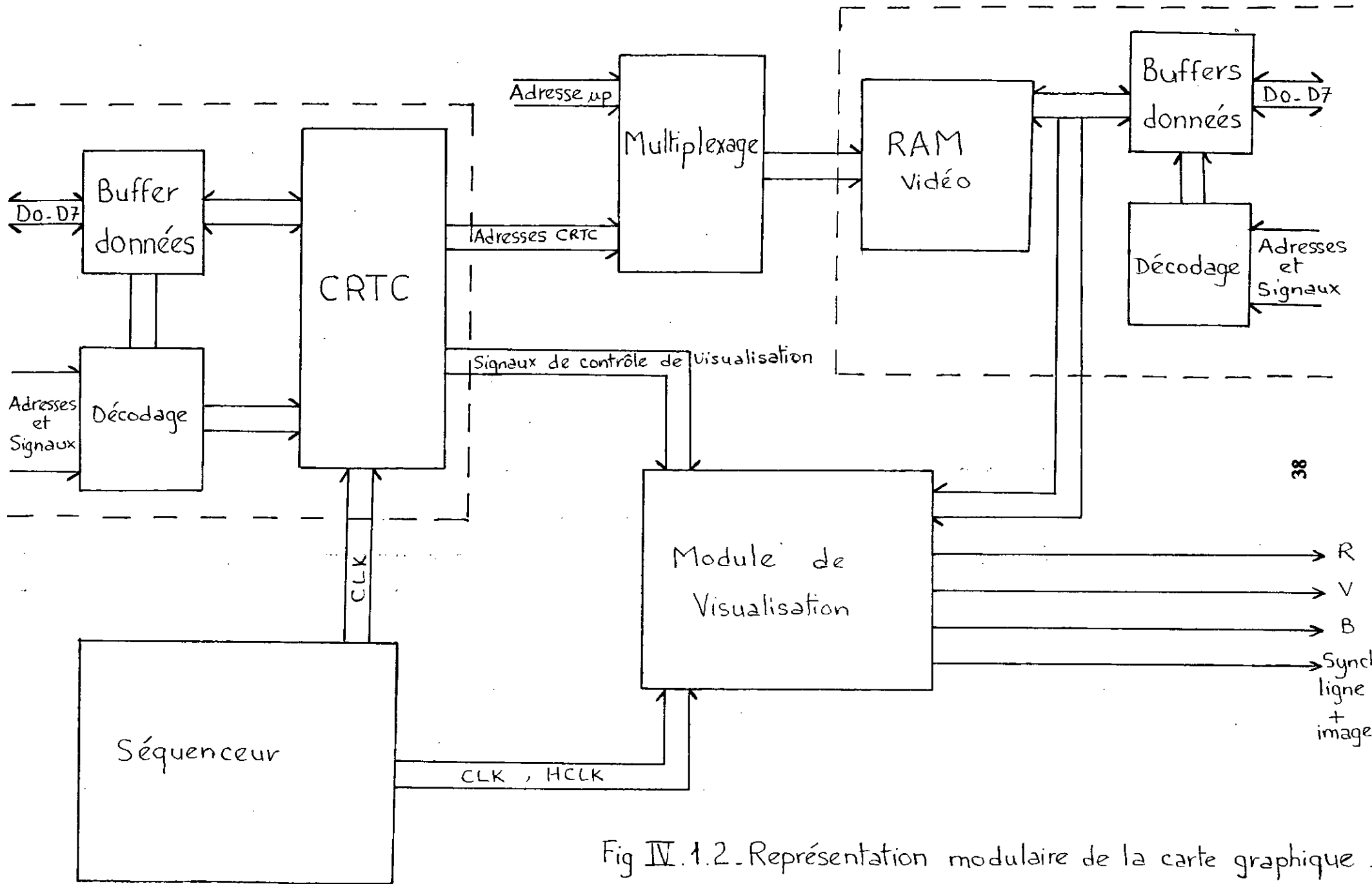
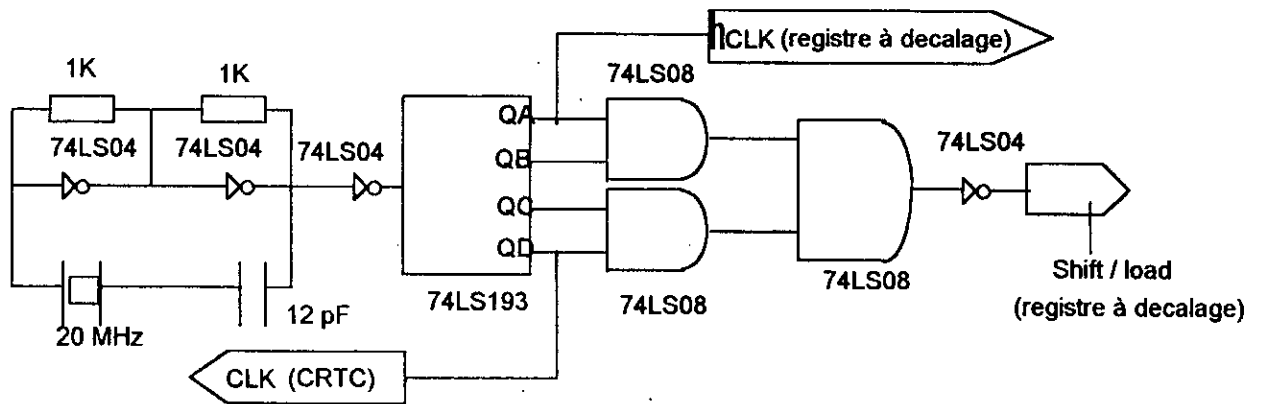


Fig IV.1.2. Représentation modulaire de la carte graphique.

Ce qui lui laisse un peu moins de 1µs pour chaque caractère. Nous avons donc choisi comme période caractère 0,8 µs (fréquence 1,25 MHz) ce qui donne pour les 64 caractères 51,2 µs (fréquence pixels 10MHz).



**fig IV.1.3. Séquenceur**

Ces fréquences sont obtenues à partir de 20MHz et d'un compteur-décompteur 74LS193 utilisé comme un diviseur de fréquences. Une capacité de 12 PF, deux résistances de 1Kohms et trois inverseurs sont utilisés pour stabiliser le quartz qui va attaquer le 74LS193 ce qui va permettre une division successive de la fréquence (fig IV.1.4). Le signal de fréquence 1,25 MHz va être relié à l'entrée horloge du CRTC et celui de 10 MHz à l'entrée horloge du registre à décalage pour lui spécifier à quelle fréquence il doit sérialiser les données. Quant au signal shift/load, c'est un signal de fréquence 1,25 MHz et il spécifie au registre à décalage à quel moment il doit commencer à sérialiser.

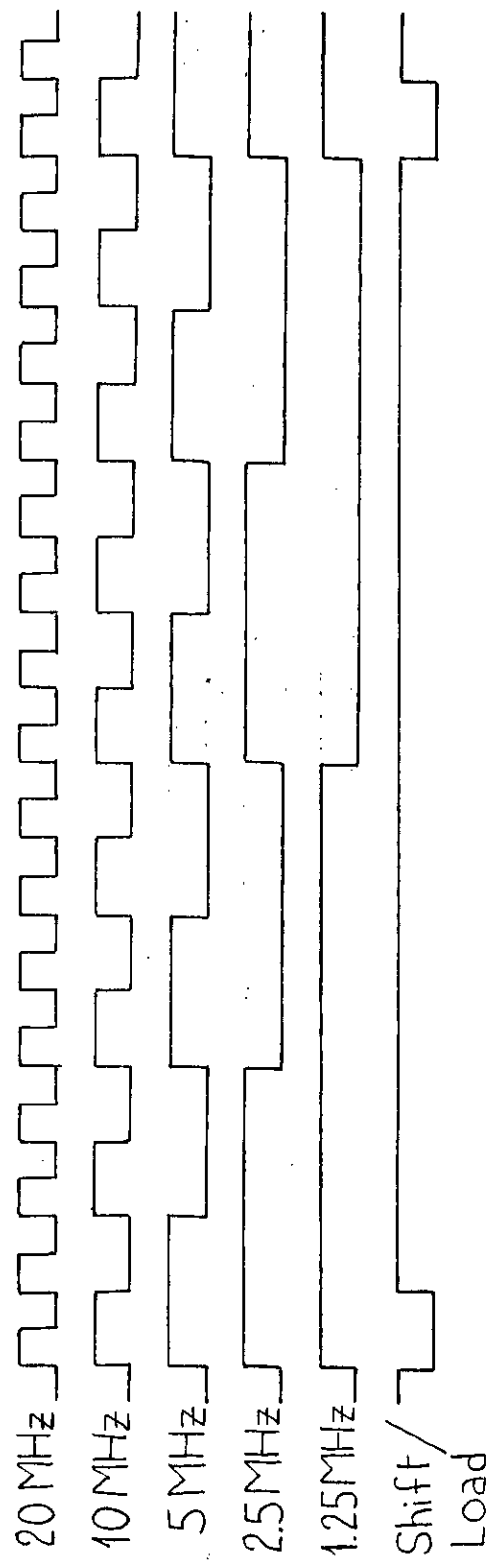


Fig IV.1.4 - Diagramme des temps.

## IV.2 DECODAGE ET COMMANDES DU CRTC:

Pour dialoguer avec le microprocesseur, le CRTC nécessite deux adresses.

La première doit sélectionner le registre d'adresses ( $\overline{CS}=0$ ) et ( $RS=0$ ).

La deuxième doit maintenir la sélection du CRTC ( $\overline{CS}=0$ ) et positionner ( $RS=1$ ).

Pour que cette carte puisse être intégrée facilement dans un système existant sans qu'il y ait de conflit avec d'autres cartes d'extension. Un décodage d'adresse pour le CRTC est nécessaire et doit tenir compte des contraintes suivantes:

- \* Deux adresses différentes de celles attribuées aux circuits de l'IBM-PC sont nécessaires au CRTC.
- \* L'adressage des entrées/sorties est distinct de celui de la mémoire.
- \* L'adressage microprocesseur est distinct de celui d'une DMA.

Ces contraintes ont été respectées en utilisant le module de décodage suivant fig(IV.2.1).

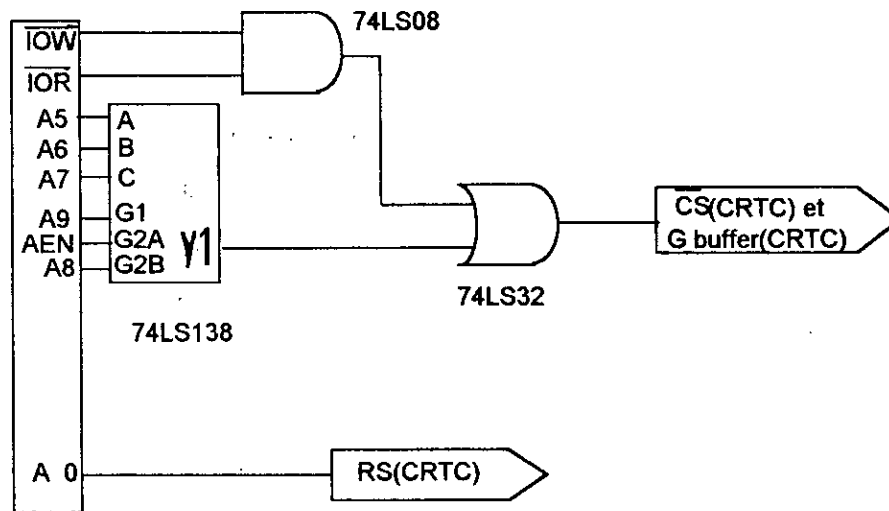


Fig IV.2.1 Décodage du CRTC

La sortie Y1 du 74LS138 est validée lorsque ces signaux d'entrée ont la configuration suivante:

AEN = 0	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
accès	1	0	0	0	1	x	x	x	x	x

microprocesseur

Avec cette configuration deux contraintes ont été respectées:

Les adresses validées 220 à 23F sont non utilisées et AEN est positionné de manière à ne valider que les accès microprocesseur.

La dernière contrainte est satisfaite en combinant les signaux  $\overline{IOW}$ ,  $\overline{IOR}$  dans une porte AND.

Cette dernière passe à l'état bas (donc valide le CRTC) quand le microprocesseur effectue soit une lecture ou une écriture E/S.

La ligne A0 à son état bas valide le registre d'adresse et à son état haut permet d'accéder à l'un des registres internes du CRTC.

Une adresse paire va donc adresser le registre d'adresses et une adresse impaire adressera l'un des registres internes.

L'entrée  $\overline{CS}$  du CRTC est reliée à l'entrée de validation d'un circuit 74LS245 (fig IV.2.2). Ce dernier est un buffer bidirectionnel et permet d'une part d'amplifier les signaux en provenance du microprocesseur et d'autre part d'aiguiller le bus de données vers le circuit qui lui est destiné. La direction de transfert de données (DIR) est déterminée par IOR.

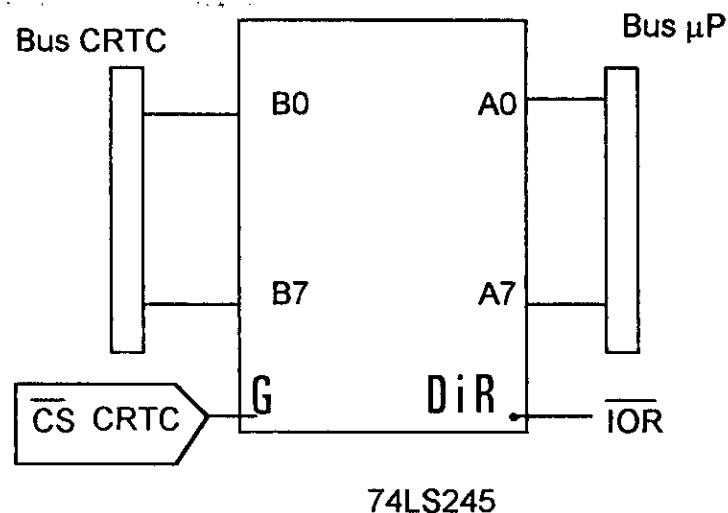


fig IV.2.2 Transfert des données vers le CRTC



Pour **DIR=0** le transfert se fait de (B0-B7) vers (A0-A7).

Pour **DIR=1** le transfert se fait dans l'autre sens.

\* Quand le microprocesseur effectue une lecture (E/S), il positionne  $\overline{\text{IOR}}$  à l'état bas (DIR sera à l'état bas aussi) ce qui permet le transfert des données du bus du CRTC vers celui du microprocesseur.

\* Quand le microprocesseur n'effectue pas une lecture (E/S), il positionne  $\overline{\text{IOR}}$  à 1 et donc automatiquement DIR=1. Deux cas sont possibles:

$\overline{\text{IOW}}=1$  Le microprocesseur n'effectue pas d'écriture entrées/sorties. Ce cas est à éliminer puisque cette configuration ne valide pas le buffer.

$\overline{\text{IOW}}=0$  Le microprocesseur effectue une écriture (E/S) et le transfert des données se fera du bus de données du microprocesseur vers celui du CRTC.

En ce qui concerne les deux signaux de contrôle du CRTC,  $\overline{\text{R/W}}$  et  $\overline{\text{E}}$  (fig IV.2.3), ils sont déterminés respectivement par:

**$\overline{\text{R/W}}$  est relié au signal  $\overline{\text{IOW}}$**

Quand le microprocesseur effectue une écriture ( $\overline{\text{IOW}}=0$ ), le signal  $\overline{\text{R/W}}$  est à zéro lui aussi et le CRTC sera informé que ses registres internes vont être écrits.

### En ce qui concerne le signal d'horloge E

une petite astuce est nécessaire. En effet, par construction, les coprocesseurs MOTOROLA ont une entrée d'horloge qui va leur permettre de travailler en conjonction avec les microprocesseurs. Cette entrée horloge E à son état haut autorise le microprocesseur MOTOROLA à faire des lectures ou écritures et à son état bas le microprocesseur est au repos et pour le coprocesseur cette situation est inversée. De cette façon, les deux peuvent travailler sans conflit.

Cette particularité n'existe pas chez les circuits INTEL, un microprocesseur INTEL peut à n'importe quel moment adresser une mémoire ou une E/S. Il lui suffit pour cela de valider un des quatre signaux  $\overline{\text{MEMR}}$ ,  $\overline{\text{MEMW}}$ ,  $\overline{\text{IOR}}$ ,  $\overline{\text{IOW}}$ .

Nous allons donc combiner un signal E tel que à son état bas le microprocesseur intel soit (écrit ou lit) dans les mémoires soit (lit ou écrit) dans les E/S. Cette combinaison est donnée fig(IV.2.3).

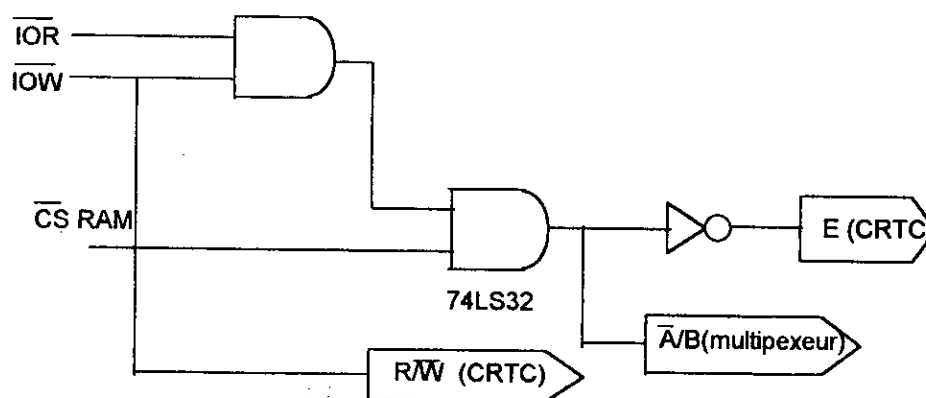


Fig IV:2.3: Signaux de commande du CRTC

Nous verrons dans le paragraphe suivant comment on a obtenu la sélection de la RAM.

### IV.3 UTILISATION DE LA RAM GRAPHIQUE

La capacité mémoire de cette carte est de 48 Kilo-octets et elle est répartie en trois plans mémoire. Comme pour le décodage du CRTIC, trois contraintes sont à respecter.

L'accès microprocesseur, l'accès mémoire, un adressage de 48 kilo-octets libres.

Les deux premières contraintes ont été satisfaites comme pour le CRTIC (seulement les signaux de lecture et d'écriture ne sont plus  $\overline{IOR}$  et  $\overline{IOW}$  mais  $\overline{MEMR}$  et  $\overline{MEMW}$ ).

Pour la troisième contrainte, il a fallu trouver un emplacement de 48 kilo-octets de mémoire libres. Or, nous avons vu (chapitre III) qu'il y avait très peu de place mémoire non utilisée. Nous avons donc opté pour l'utilisation de la zone mémoire destinée à accueillir la RAM vidéo de la carte HERCULES. Ces positions mémoire sont données par les adresses :

B0000 à B3FFF pour le plan vert  
B4000 à B7FFF pour le plan rouge  
B8000 à BBFFF pour le plan bleu

L'occupation de cette zone ne va pas restreindre le domaine d'utilisation de notre carte car celles qui occupent cette zone mémoire (CGA, MDA, HERCULES) ne sont plus commercialisées.

Le décodage de cette zone est illustré (fig IV.3.1)

	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
b 0 0 0 0 H	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b 3 F F F H	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
b 4 0 0 0 H	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b 7 F F F H	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
b 8 0 0 0 H	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b b F F F H	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Les lignes A16, A17, A18, A19 sont utilisées pour décoder toute la zone mémoire. Pour distinguer entre ces plans, on utilisera respectivement la ligne A15 pour différencier le plan bleu des deux autres plans puis la ligne A14 pour ces deux derniers.

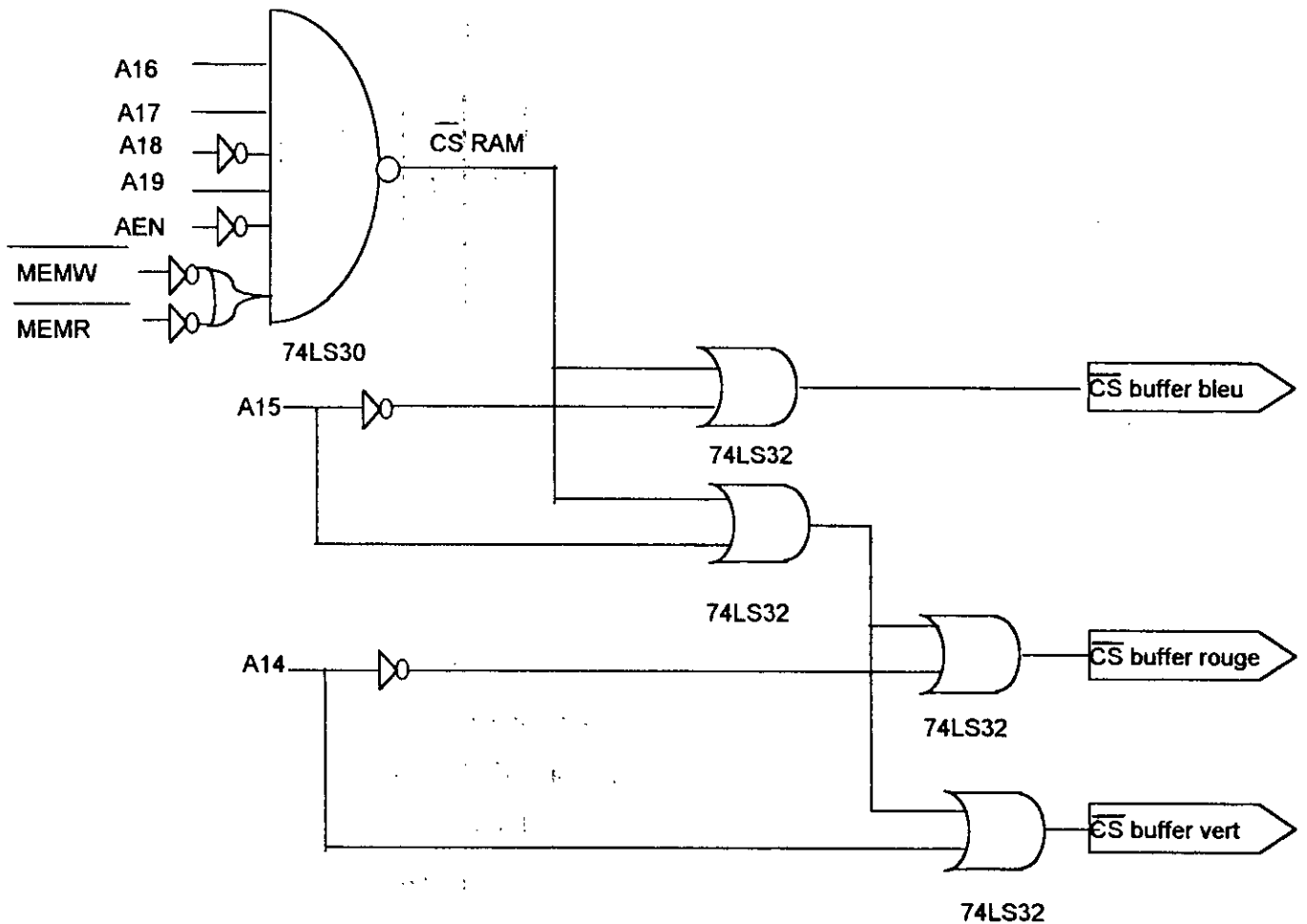
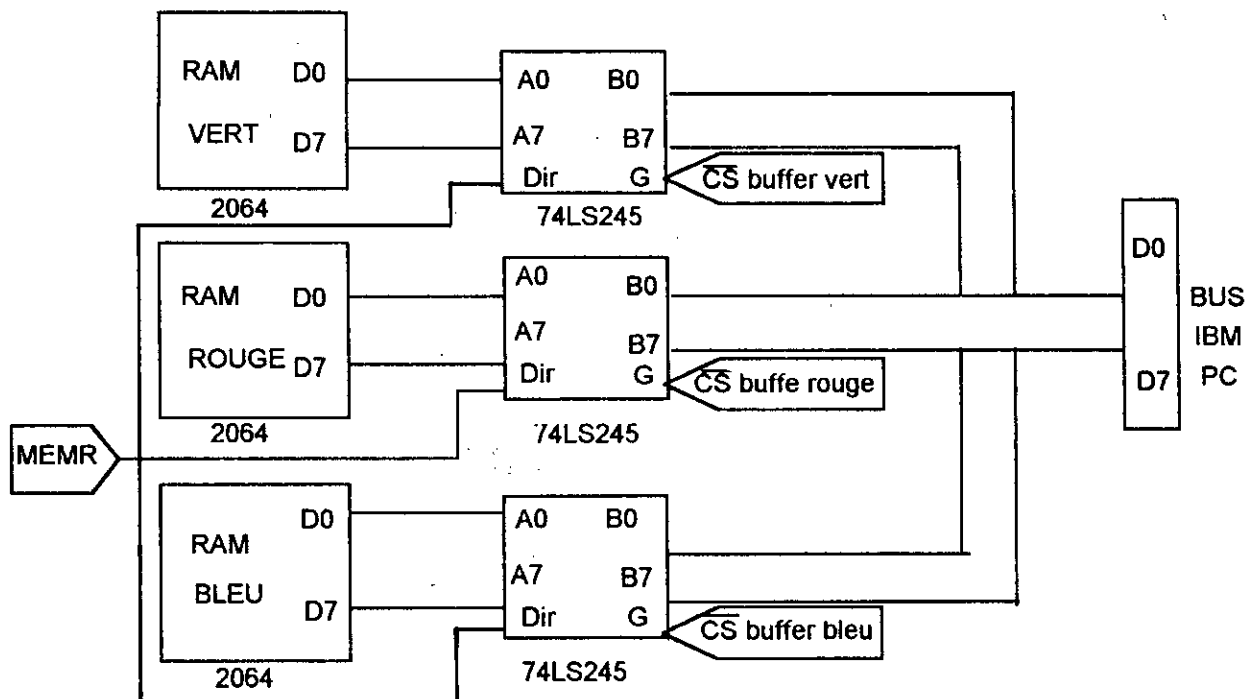


Fig IV.3.1 Décodage de la zone mémoire

Ces sélections vont valider les buffers reliés aux bus de données des mémoires. La direction de ces buffers est déterminée par MEMR (fig IV.3.2).



**Fig IV.3.2 Transfert des données vers la mémoire**

Les trois plans mémoire sont adressés par le même bus d'adresses qui provient de quatre multiplexeurs 74LS157 (voir schéma complet Annexe).

Ce multiplexage est nécessaire car dans notre cas, la mémoire est à double accès.

Quand le CRTC est autorisé à travailler ( $E=0$ ), les multiplexeurs commandés par  $\overline{E}$  vont connecter le bus d'adresses du CRTC à celui des trois RAMS. De cette façon, le CRTC verra un seul plan mémoire de 16 Kilo-octets et ces trois mémoires recevront au niveau de leurs entrées  $R/\overline{W}$  un signal à l'état haut leur spécifiant qu'elles seront lues et transférées vers des registres à décalage.

Quand le CRTC n'est pas autorisé à travailler (E=1), le microprocesseur est donc en train d'effectuer des transferts mémoire ou d'entrées/sorties.

Dans ce cas, les multiplexeurs vont connecter le bus d'adresses du microprocesseur à celui des trois RAMS par contre seul un des trois buffers est validé à la fois de sorte que le microprocesseur ne communique qu'avec un seul plan mémoire à la fois et la mémoire pour lui occupera 48 kilo-octets.

Le signal  $R/\overline{W}$  de la RAM est connecté dans ce cas à  $\overline{MEMW}$  pour déterminer si le microprocesseur va lire ou écrire dans la mémoire.

#### REMARQUE:

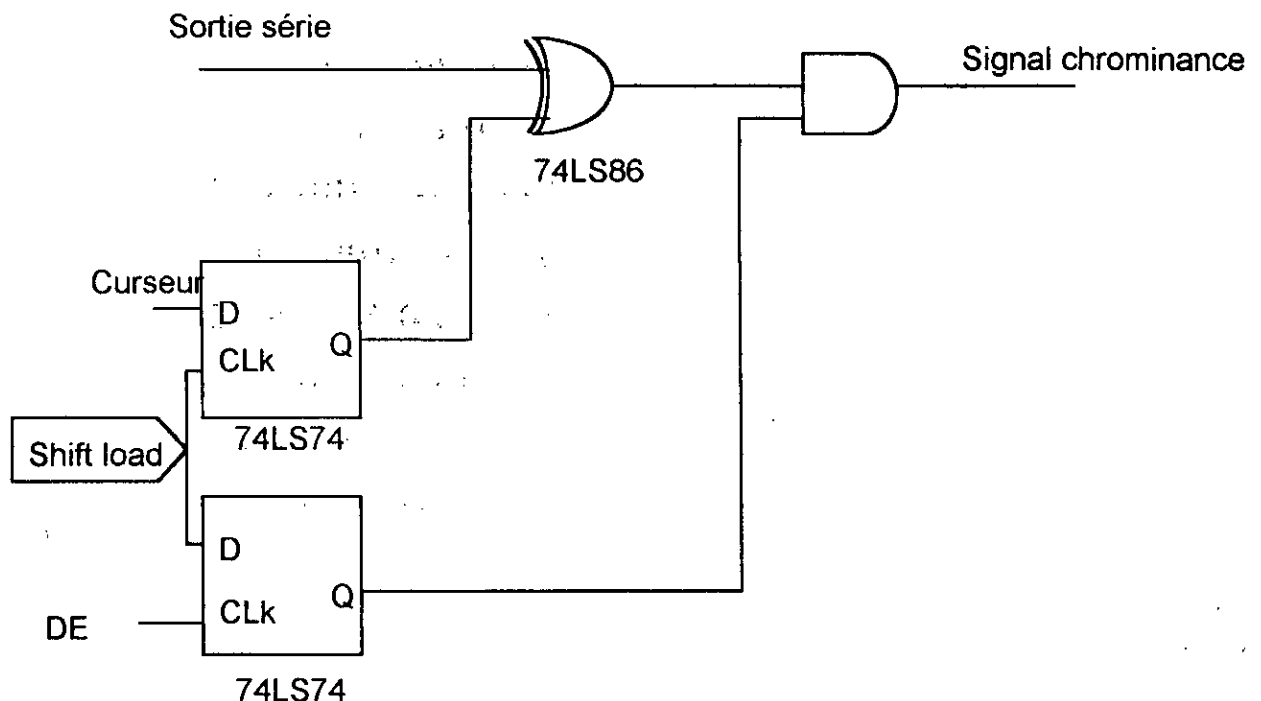
Au niveau du schéma complet, on remarque que chaque plan mémoire de 16 kilo-octets est constitué de deux RAMS chacune de 8 kilo-octets. Cette situation n'est pas voulue pour une raison spécifique mais est tout simplement due à un manque de composants.

#### IV.4 VISUALISATION DE L'IMAGE

Pour constituer l'image, le dispositif d'affichage doit recevoir d'une part les signaux de synchronisation et d'autre part les signaux de chrominance.

**Les signaux de chrominance** proviennent des registres à décalage connectés au bus de données des RAMS. Quand le CRTC adresse une position mémoire, il obtient sur le bus de données de chaque RAM une information binaire

parallèle décrivant l'état d'une suite de 8 pixels. Pour que le spot électronique du tube cathodique puisse interpréter ces informations il faut les lui présenter en série et non en parallèle. Le registre à décalage, un 74LS165 se charge de l'opération. Pour cela, il faut lui fournir les instants où il doit sérialiser les informations (shift/load) et la fréquence de sérialisation (HCLK fréquence pixels). A la sortie de chaque registre à décalage, on obtient des informations série relatives à chaque couleur. A ce signal on va additionner les informations curseur fig (IV.4.1).



**Fig IV.4.1 Génération du signal de chrominance**

La sortie curseur en provenance du CRTC va indiquer l'état du curseur à chaque position de l'écran. C'est pour qu'il n'y ait pas un déphasage entre les informations curseur et celles de l'image qu'une synchronisation s'impose, elle est obtenue grâce à une bascule D commandée par le shift/load. La sortie DE du CRTC est elle aussi synchronisée grâce à une bascule D avant d'être reliée à une porte AND.

**Les signaux de synchronisation** proviennent du CRTC et sont HS et VS, actifs à l'état bas. Ces signaux sont additionnés dans une porte EXOR avant de passer par un amplificateur qui est utilisé pour stabiliser le signal de synchronisation composite et l'inverser (fig IV.4.2).

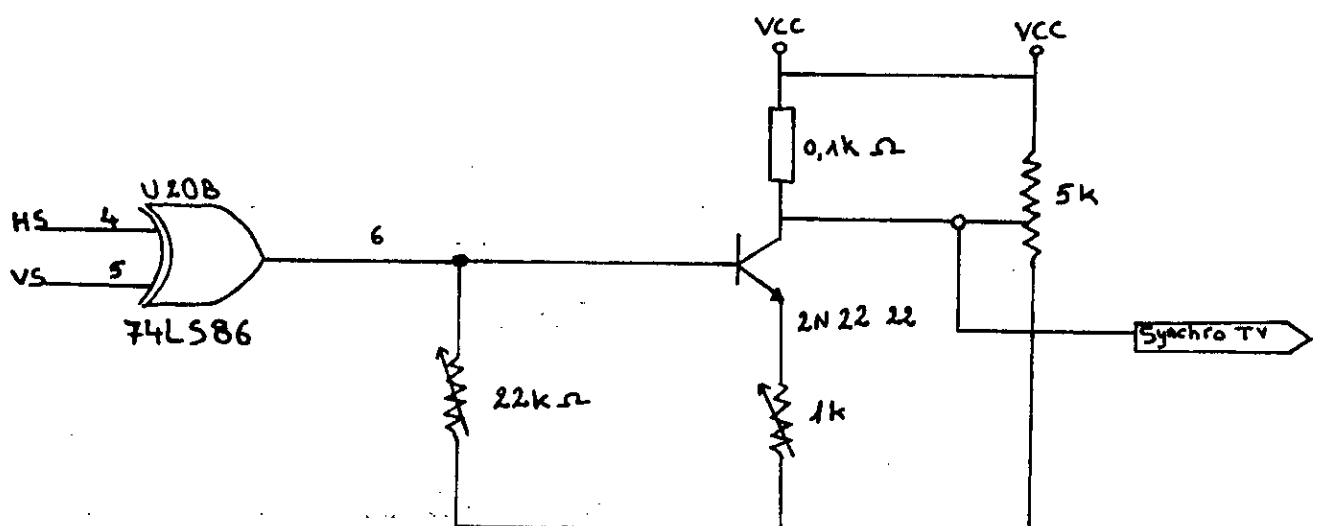


Fig IV.4.2 Signal de synchronisation composite



## IV.5 TEST ET PROGRAMMATION DE LA CARTE

Les tests ont été effectués sur les différentes parties de la carte suivant cet ordre:

### **Le séquenceur:**

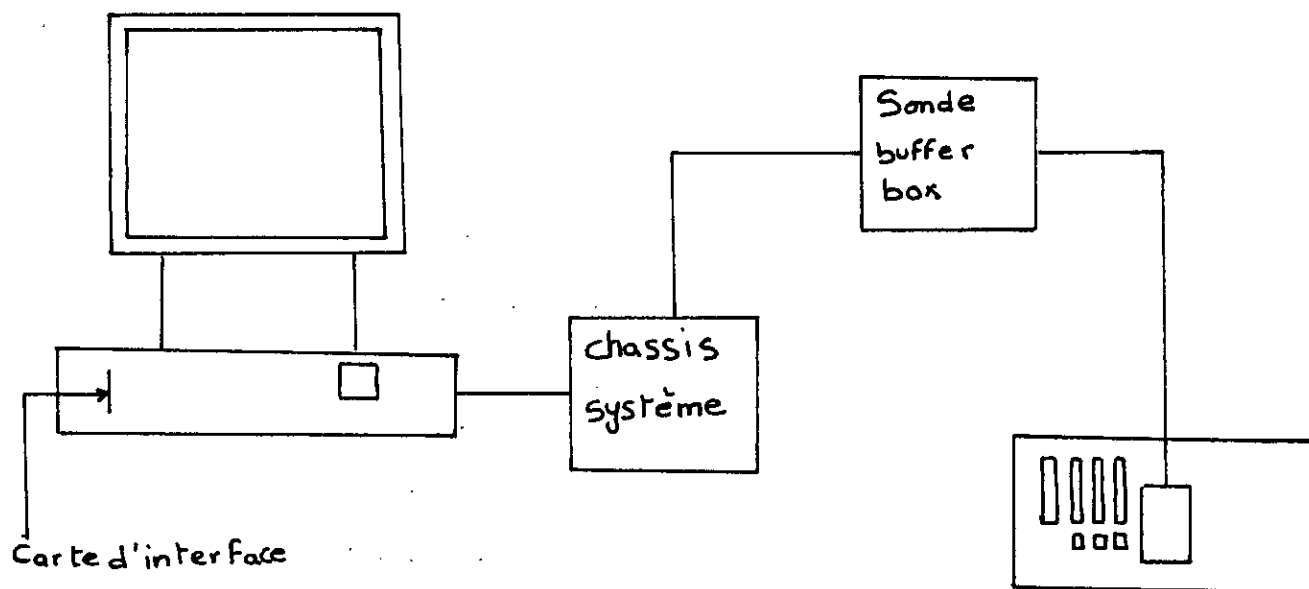
Les différents signaux issus 74LS193 ont été visualisés à l'oscilloscope et étaient conformes aux signaux décrits fig (IV.1.4).

### **Le contrôleur de visualisation:**

A partir de cette étape, la carte a été testée au niveau du système de développement I2ICE d'INTEL.

Ce système permet l'émulation en temps réel et comporte les modules suivants (fig IV.5.1):

- Une **carte d'interface** avec l'ordinateur hôte (IBM PC et compatibles).
- Un **chassis système I2ICE** contenant la carte de communication avec l'ordinateur hôte, une carte mémoire à accès rapide de 32 Kilo-octets et une carte buffer trace contenant les informations et résultats issus de l'exécution.
- Une **sonde** (probe buffer box): celle-ci connecte le prototype hardware de l'utilisateur au chassis.



**Fig IV.5.1 Schématisation du système de développement I2ICE**

Le CRTC a été testé en programmant ses 16 registres internes et en vérifiant à l'oscilloscope que les trois signaux de contrôle de visualisation étaient conformes à la programmation.

Remarque: La programmation des registres est donnée en annexe.

### **La mémoire:**

Le test de la mémoire a consisté à écrire dans des positions mémoire et lire ces mêmes positions. Les valeurs étant les mêmes nous sommes passés à la dernière étape.

## Visualisation:

Sur l'écran de télévision connecté grâce à la prise péritel à la carte d'interface, nous avons pu visualiser les huit couleurs données dans (chapitre I code des couleurs) ainsi que quelques lettres de l'alphabet.

Pour l'obtention des couleurs nous avons procédé de la sorte:

\*Chaque couleur a un segment

Vert: 0B000      Rouge:0B400      Bleu:0B800

\*Si l'on veut obtenir la couleur verte sur tout l'écran:

-On positionne tout le plan mémoire réservé à la couleur verte à FF (ce qui équivaut à mettre tous les bits à 1).

-On positionne tout les plans mémoire réservés aux couleurs rouge et bleu à 00 (ce qui équivaut à mettre tous les bits à 0)

On procède de la même manière pour toutes les autres couleurs.

Pour ce qui est de l'affichage de lettres, nous avons visualisé quelques lettres de l'alphabet en utilisant le principe suivant (fig IV.5.2).

On représente le caractère que l'on veut afficher dans une grille constituée de 8 pixels horizontalement et verticalement. Chaque ligne de cette grille aura une valeur binaire reflétant l'état de ces pixels. il suffira de mettre à l'adresse de chaque ligne la valeur

correspondante pour voir afficher le A.

D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>	
								3E
								7F
								41
								41
								41
								41
								41
								7F
								7F
								41
								41
								41
								41
								41
								00
								00

Fig IV.5.2 Représentation du A sur une grille 8x16

Remarque:

Les dimensions de la grille ne sont pas obligatoirement 8x16 pixels et peuvent être prises plus grandes ou plus petites selon les besoins de l'utilisateur et il appartiendra à ce dernier de faire la correspondance entre les lignes et les positions mémoire (fig IV.5.3).

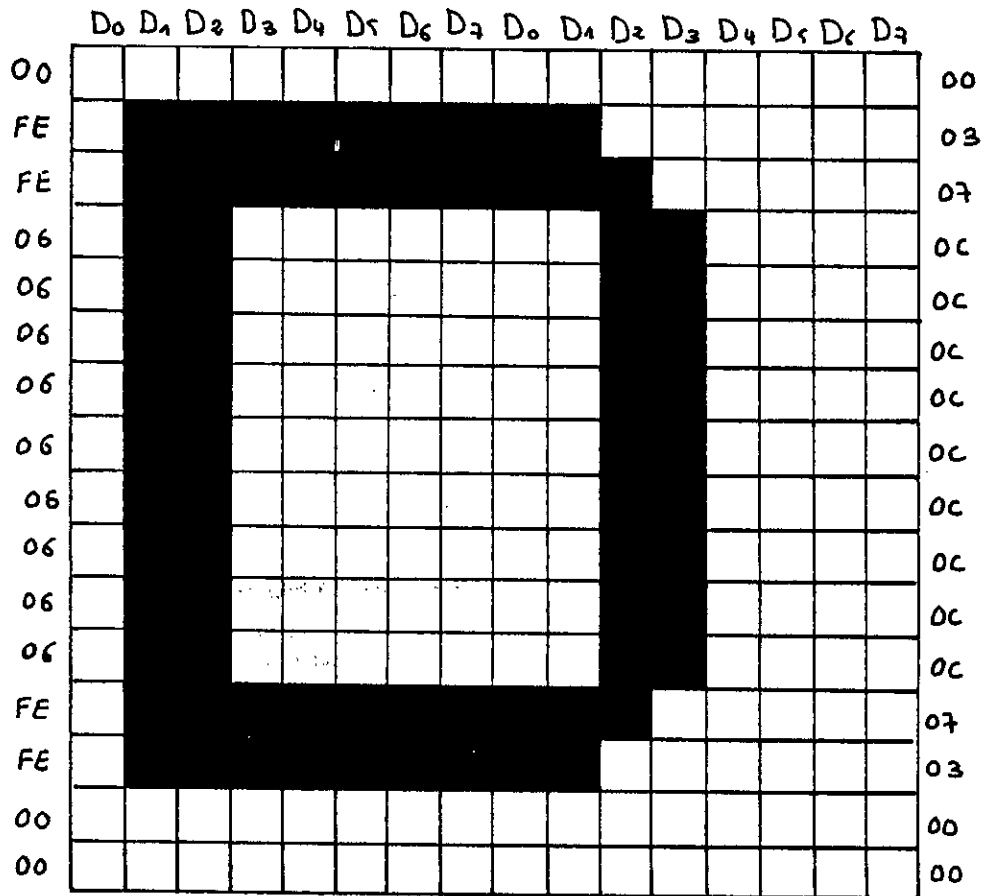


Fig IV.5.3 Représentation du D sur une grille 16x16

**C H A P I T R E V**

## NOTIONS SUR LA GESTION DE LA CARTE

La carte graphique a été testée grâce au système de développement **I2ICE**, ce dernier comporte une partie software qui dote l'utilisateur d'instructions permettant la modification du contenu de zones mémoire ou de ports d'entrées/sorties. Ces instructions, du fait de leur pouvoir limité, ne sont généralement utilisées par les constructeurs que pour les tests.

Cette carte destinée à l'affichage d'informations dans les gares et les aéroports et donc à être utilisée par des utilisateurs non spécialisés dans le domaine, un programme de gestion lui donnera plus de puissance et limitera le travail de l'opérateur à l'exécution de ce programme et éventuellement à l'introduction d'informations complémentaires (emplacement désiré, couleur,...).

Dans ce chapitre, nous aborderons cette gestion, nous parlerons d'abord du langage assembleur et nous donnerons quelques principes concernant la visualisation d'images.

### V.1 LANGAGE ASSEMBLEUR :

L'assembleur est un langage intimement lié au microprocesseur et aux circuits d'un ordinateur à l'opposé des langages évolués qui isolent en fait le programmeur du processeur. De plus, l'assembleur permet de développer des programmes qui occupent moins d'espace mémoire et s'exécutent plus rapidement que s'ils avaient été développés en langage évolué.

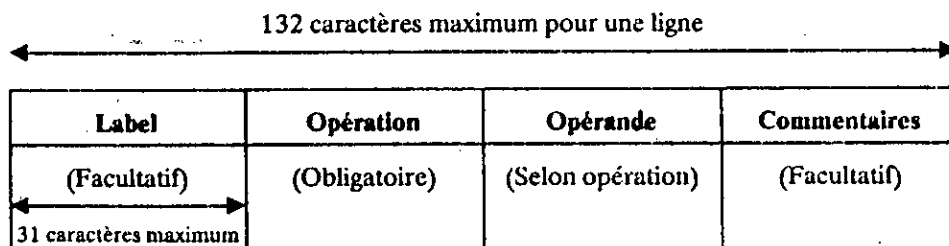
Les instructions du 80286 sont conformes à celles du 8086/8088 mais certaines ont pu être modifiées dans leur forme afin de mener à des programmes plus courts ou accroître leur efficacité, elles s'appliquent aux modes ;réel et protégé de la même façon. Le jeu de base regroupe plus d'une centaine d'instructions multipliés par les modes d'adressage (voir fig V.1.1) et les différents types de variables, on aboutit à un total de plus de 250.000 combinaisons.

Mode	Format de l'opérande	Registre de segment
A registre Immédiat à registre Implicite	Un registre Donnée Rien	Aucun
Immédiat à mémoire	Donnée	DS
Direct	Constante Label	DS DS
Indirect par registre	[BX] [BP] [DI] [SI]	DS SS DS DS
Par base	[BP+Déplacement] [BX+Déplacement]	SS DS
Indexé	[DI+Déplacement] [SI+Déplacement]	DS DS
Indexé par base	[ BX+DI+Déplacement ] [ BX+SI+Déplacement ] [ BP+DI+Déplacement ] [ BP+SI+Déplacement ]	DS DS SS SS
Chaines · Source · Destinataire	[SI] [DI]	DS ES

Fig V.1.1 Modes d'adressage du 80286



Les instructions écrites en mnémoniques restent toujours très courtes et tiennent aisément sur une ligne et pour plus de commodité, on la scinde en quatre zones appelées champs fig (V.1.2) et l'on distingue:



**Fig V.1.2 Représentation d'une ligne écrite en assembleur**

**.Champs symbole "champ label":** On attribue à une instruction une étiquette "un nom" qui permet de la distinguer et de l'appeler en usant de ce nom.

**.Code opération :** c'est le code qui définit l'opération à exécuter (par exemple, on écrira ADD pour exécuter une addition).

**.Le champ opérande:** Ce champ est obligatoire avec certaines instructions, interdit avec d'autres (adressage implicite), il indique sur quoi porte l'opération ou encore à quel registre ou à quelle cellule mémoire on se réfère.

**.Le champ commentaire:** Il est facultatif et est utilisé pour apporter quelques explications sur ce qui le précède. Il n'est utile qu'au lecteur du programme mais strictement inutile à l'exécution.

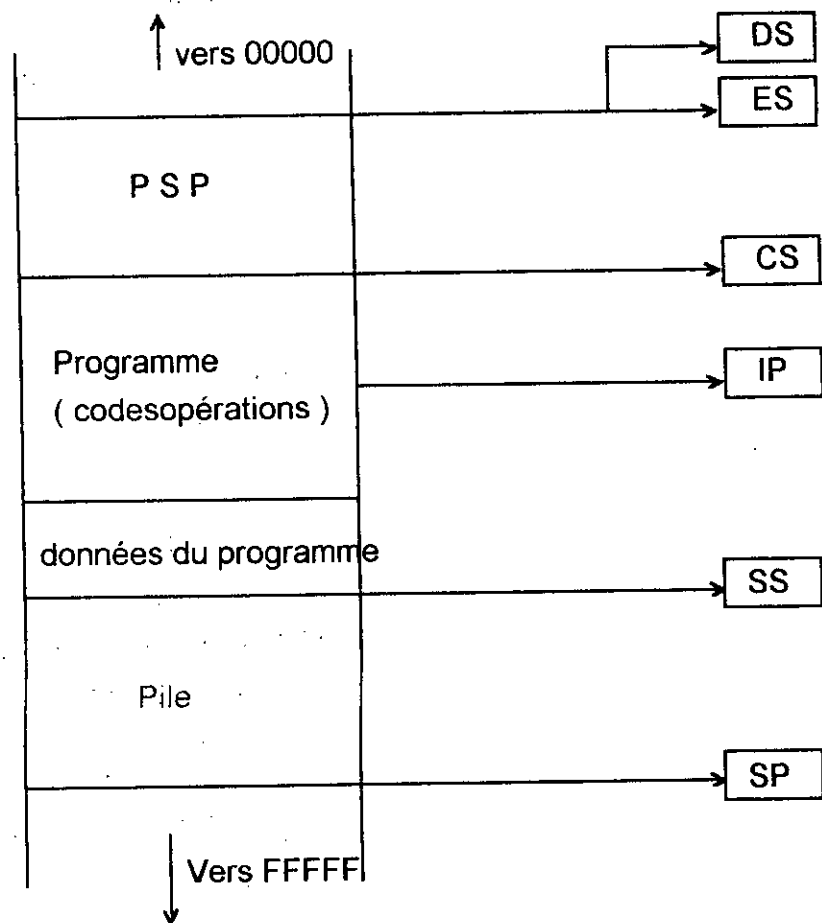
Il existe trois règles très importantes à respecter afin de mener à bien un programme en assembleur.

- . Il est impossible de faire un transfert de mémoire à mémoire, il faut obligatoirement passer par un registre du 80286.
- . Il est interdit de charger une valeur immédiate dans un registre de segment, il faut obligatoirement passer par un registre général.
- . Il est interdit d'utiliser le registre IP comme source ou destination.

L'assembleur d'IBM version 2 porte le nom de macro-assembleur "MASM", il présente deux avantages par rapport à ASM; en cas d'erreur il délivre un message d'erreur complet et non pas un numéro d'erreur et il sait traiter les macro-instructions.

Le développement d'un programme en assembleur sous MASM se fait de la façon suivante: on fait appel à un éditeur de texte ou à n'importe quel traitement de texte et on rédige le programme en mnémoniques. L'assembleur MASM fournira un fichier dont l'extension sera .obj (fichier objet) non encore exécutable. C'est l'éditeur de lien (link) qui transformera le fichier (.obj) en (.exe).

Dès le chargement d'un programme, il va se loger dans la zone réservée à cet effet (CHAPITRE III) et l'assembleur placera à la suite les quatre sections suivantes (fig V.1.3).



**Fig V.1.3 Initialisation des registres**

**. PSP: Préfixe Segment de Programme**

Il s'agit d'un bloc de contrôle élaboré par le DOS qui contient des informations relatives au programme qui suit et qui assure la liaison avec le DOS. L'adresse du début de ce bloc est automatiquement chargée dans les registres segments DS et ES.

**. Le programme:**

Représente la séquence d'instructions sous forme de code opération. L'adresse du début de ce bloc est elle aussi automatiquement chargée dans le registre segment CS.

### **. Les données du programme :**

Les données nécessaires à l'exécution du programme sont stockées dans cette partie de la mémoire. L'adresse de départ de ce segment n'est pas automatiquement stockée dans DS, il appartiendra donc à l'utilisateur de la lui fournir.

### **. Pile de sauvegarde :**

La pile peut-être considérée comme une zone mémoire destinée à recevoir des informations temporaires. L'adresse de départ est stockée dans le registre SS et l'adresse de fin de la pile dans SP (pile à fonctionnement LIFO).

## **V.2 EXEMPLES D'APPLICATION**

Le programme de gestion de la carte devra offrir à l'utilisateur des fonctions standards telles que: Affichage de caractères avec couleur désirée aux emplacements voulus, Représentation de figures géométriques (cercles et rectangles), Fonction zoom.

-Pour l'affichage de caractères tels que : lettres de l'alphabet, chiffres ou autres représentations, l'utilisateur n'aura pas à représenter le caractère sur une grille et à calculer la valeur de chaque ligne (comme nous l'avons fait CHAPITRE IV). Ces valeurs seront stockées dans la zone mémoire réservée aux données et l'utilisateur n'aura qu'à spécifier quel caractère il désire.

-Pour ce qui est de l'emplacement sur l'écran, l'utilisateur n'aura pas à connaître l'adresse correspondante au niveau de la mémoire mais juste préciser la position sur l'écran en donnant le numéro de la rangée et celui du caractère et éventuellement le numéro de la ligne au niveau de la rangée.

L'adresse trouvée (voir annexe ) correspondra à un **OFFSET** c'est-à-dire un déplacement à l'intérieur d'un segment et c'est le choix de la couleur qui déterminera les segments adéquats.

-Pour la fonction zoom, l'utilisateur devra préciser la position sur l'écran comme précédemment, le microprocesseur lira l'adresse correspondante et testera bit par bit le contenu de cette position mémoire et suivant l'agrandissement demandé, il dupliquera ces bits.

Exemple: Zoom de 8 fois.

Si le microprocesseur lit la valeur 15H, il mettra dans huit positions mémoire (correspondant à huit emplacements sur la même ligne de l'écran) les valeurs suivantes: 00,00,00,FF,00,FF,00,FF.

REMARQUE:

Il est possible de trouver l'OFFSET sans que l'utilisateur ait à estimer le numéro du caractère et de la rangée. Pour cela, il suffit d'utiliser un crayon optique et de le pointer sur la position désirée. L'adresse correspondante dans la mémoire sera stockée dans les registres R16 et R17 et il suffira au microprocesseur de les lire.

**C O N C L U S I O N**

Ce travail nous a permis d'approfondir nos connaissances des circuits programmables, du langage assembleur 80286 et des techniques d'interfaçage des PC. De plus, les tests effectués au cours de la phase réalisation nous ont familiarisé avec l'utilisation du système de développement **I2ICE** d'INTEL.

En plus de ces apports théoriques et pratiques, la réalisation de la carte graphique nous a confronté à des problèmes qui bien souvent n'apparaissaient pas lors de la conception. C'est alors que l'on réalise l'importance du bon choix des composants et l'utilité d'une conception matérielle modulaire.

L'architecture conçue et réalisée a subi tous les tests avec succès. Les tests ont été effectués sur les différentes parties grâce au système de développement qui a été dans bien des cas d'un grand secours pour la localisation des pannes.

La carte graphique réalisée n'étant qu'un prototype, diverses améliorations peuvent être envisagées. Parmi celles-ci, l'utilisation de RAMS de 16K au lieu de 8K et l'intégration des portes logiques dans des circuits programmables



PAL réduira à coup sûr l'encombrement de la carte.

Une autre amélioration envisageable serait l'utilisation de mots de 16 bits, cette utilisation permettra de réduire le temps d'écriture dans la mémoire mais restreindra le domaine d'utilisation de la carte aux PC-AT et compatibles.

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

- [1] **C.DARDANNE** "LE MICROPROCESSEUR 6809"  
Edition EYROLLES 1982
- [2] **H.LILEN** "80286 ET SES PERIPHERIQUES"  
Edition RADIO 1987
- [3] **H.LILEN** "80286 ASSEMBLEUR IBM PC ET COMPATIBLES"  
Edition RADIO 1987
- [4] **C.VIEILLEFOND** "80286 MISE EN OEUVRE ET  
PROGRAMMATION"  
Edition CYBEX 1989
- [5] **L.PADJASEK** " ASSEMBLEUR, LE CALCUL NUMERIQUE"  
Edition CYBEX 1988
- [6] **M.TISCHER** "BIBLE DU PC"  
Edition MICRO APPLICATION 1992
- [7] **D.MARTIN** "CLEFS POUR IBM PC"  
**F.PIETTE** Edition PSI 1988
- [8] **J.C.CHAUVEAU** "MEMOTECH"  
Edition CASTEILLA 1989

#### **MANUELS**

- [9] **BULL MICRAL** "SYSTEME D'EXPLOITATION MS-DOS"  
Version 3.3 1987
- [10] **I2ICE** "SYSTEM REFERENCE MANUEL"

## REVUES

- [11] C.TAVERNIER **Revue SOFT&MICRO**  
"ASSEMBLEURS DES PC"  
Dec, Jan, Fev, Mar 1987
- [12] P.GUEULLE **Revue RADIO PLAN**  
"DECODEUR POUR CARTES D'EXTENSION PC"
- [13] C.TAVERNIER **Revue SOFT&MICRO**  
"LES INTERFACES DE VISUALISATION"  
DEC 1988
- [14] CDTA **BULLETIN D'INFORMATION TRIMESTRIEL**  
N°6 1er trimestre 1992

## MEMOIRES

- [15] A.KECHID .."Contribution à la conception et à la  
réalisation d'une station de travail  
multi-utilisateur"  
  
Mai 1991
- [16] H.Hamdi "réalisation d'une carte d'acquisition  
A.Hemadi d'images"

**A N N E X E S**

## ANNEXE A BUS D'ENTREES/SORTIES DES PC ET COMPATIBLES

Sur la carte mère des IBM PC et compatibles, cinq connecteurs d'extension (huit pour les PC/XT et PC/AT) sont disponibles pour réaliser une multitude de configurations.

Ces connecteurs sont montés sur un bus comportant soixante deux broches. C'est le bus d'entrées/sorties.

Les cartes d'extension seront enfichées dans un de ces connecteurs sur lesquels on trouve tous les signaux nécessaires au fonctionnement.

La figure 1.a montre le connecteur d'un PC. Ce connecteur se retrouve sur les trois types de PC, avec de légères modifications chez l'AT, comme nous le verrons plus loin.

Les signaux disponibles sur ce connecteur :

**.SD0 à SD7:** Qui sont les lignes bidirectionnelles permettant l'échange des données entre la mémoire, le processeur et les entrées/sorties (E/S). Dans le cas d'un cycle d'accès direct en mémoire (Direct Access Memory :DMA), le microprocesseur est complètement déconnecté et sa fonction est partiellement remplacée par le contrôleur DMA.

**.SA0 à SA19 :** Qui sont utilisées pour adresser la mémoire du système ainsi que les différents ports d'E/S. Elles sont générées par le microprocesseur ou

le contrôleur de DMA. A16 à A19 sont maintenues inactives durant le cycle d'E/S.

. **AEN** (**A**dress **E**nable: Validation d'adresse) qui ne peut être issu que de la logique du contrôleur de DMA. Il signale aux différentes extensions qu'un cycle DMA est en cours. Il bloque les circuits de décodage des ports d'E/S pour éviter la confusion au moment de la génération des adresses mémoire et des E/S à l'intérieur d'un cycle DMA.

. **I/O CH CK** (**I/O C**hannel **C**heck: contrôle voie d'E/S):  
Cette ligne sert à informer le processeur d'un mauvais fonctionnement ou d'une erreur. Si cette ligne est mise à un niveau bas, elle génère une interruption de type **NMI** (**N**on **M**askable **I**nterrupt: interruption non masquable). En fait, les sources susceptibles de déclencher des interruptions **NMI** étant nombreuses, il a fallu faire franchir à cette ligne un port d'E/S. Celui-ci permet de déterminer plus facilement l'origine de cette **NMI**.

- . **I/O CH RDY** (**I/O CHannel Ready**: canal d'E/S prêt):  
Cette ligne n'est utilisée que dans le sens extension-microprocesseur et sert à la demande d'extension de cycles. Si une extension a du mal à suivre le rythme du processeur, elle peut lui demander de placer des cycles d'attente (wait states). Si cette ligne est maintenue active, le nombre de cycles additionnels peut aller jusqu'à dix.
  
- . **Reset DRV** (**Reset DRiVer**: commande de réinitialisation): Permet la remise à zéro des circuits électroniques lors d'une mise sous tension ou d'une chute de tension. Cette commande est synchronisée sur le flanc négatif du signal d'horloge et elle est active au niveau haut.
  
- . **OSC** : Est le signal directement issu du circuit oscillateur. Ce signal représente la fréquence disponible la plus haute dans le PC.
  
- . **CLK** (horloge) : est également issu de l'oscillateur mais a subi une division par trois. Ce n'est pas un signal symétrique : sa période positive est égale à la moitié de sa période négative.



Ce signal est directement lié, sans décalage, aux cycles de lecture et d'écriture de la mémoire ou des E/S et peut parfaitement servir à la génération des cycles d'attente : par le biais d'I/O CH RDY :

- . **T/C** (**T**erminal **C**ount: compte des terminaux pour DMA): Est généré par le contrôleur de DMA lorsque le compteur d'octets d'un des canaux atteint la limite des mots à transférer. Pour spécifier le canal concerné, le contrôleur active DACK correspondant au canal.
  
- . **BALE** (**B**us **A**dress **L**atch **E**nable: validation d'adresse) : Ce signal indique que l'adresse disponible sur le bus est valide. Il est utilisé dans l'électronique périphérique du processeur pour maintenir les adresses. Il ne joue aucun rôle en cycle DMA.
  
- . **DRQ1 à DRQ3** (**D**irect memory access **R**e**Q**uest: demande de DMA) : Ces trois lignes, actives au niveau haut, sont générées par l'extension pour demander un cycle de transfert DMA entre la mémoire et elle-même. Ces lignes sont directement reliées au contrôleur de DMA. Si la priorité des lignes n'a pas été modifiée par le programmeur, DRQ14 est la ligne prioritaire.

. **DACK0 à DACK3** (Direct memory access **ACK**nowledge: accusé de réception de DMA) : Ces quatre signaux sont issus du contrôleur de DMA pour indiquer que la demande de cycle DMA a été honorée. La ligne DACK0 est envoyée à titre d'information puisqu'elle sert au cycle de rafraichissement de la mémoire et que la ligne DRQ0 n'est pas accessible au bus. Ce signal permet de détecter le cycle de rafraichissement de la mémoire. Il peut également servir de synchronisation, sa période d'apparition étant systématiquement de soixante-douze cycles d'horloge.

. **IRQ1 à IRQ3** (Interrupt **ReQ**uest : demande d'interruption): Ces signaux servent aux demandes d'interruptions et sont donc directement reliés au contrôleur d'interruption. Il existe une priorité entre ces lignes, elle est déterminée par le BIOS (Block Input Output **S**ystem : module d'E/S du système d'exploitation ) à l'initialisation du micro-ordinateur. Sur le bus d'E/S, la ligne IRQ2 est prioritaire. La demande d'interruption se fait en passant la ligne d'un niveau bas vers un niveau haut. Elle doit être maintenue jusqu'à ce que la demande soit honorée.

. **IOR** (I/O Read: lecture périphérique): ce signal est actif au niveau bas. Il demande à l'extension de présenter ses données sur le bus. Il est généré par le microprocesseur ou le contrôleur de DMA.

. **IOW**(I/O Write: écriture périphérique): actif au niveau bas, il demande à l'extension de lire les données disponibles sur le bus. Cette ligne est pilotée par le microprocesseur ou le contrôleur de DMA.

. **MEMR** (MEMory Read: lecture mémoire) demande à la mémoire de présenter ses données sur le bus. Ce signal est actif au niveau bas. Il est issu du microprocesseur ou du contrôleur de DMA.

. **MEMW** (MEMory Write: écriture mémoire) demande à la mémoire de lire les données disponibles sur le bus. De même que MEMR, ce signal est actif au niveau bas et est généré par le microprocesseur ou le contrôleur de DMA.

. **Les alimentations** : on trouve sur le bus d'E/S les alimentations suivantes : +5v précis à 5% sur les broches B3 et B29, -5v précis à 10% sur la broche B5, +12v précis 5% sur la broche B9 et -12v

précis à 10% sur la broche B7. B8 est une broche réservée : inutilisée sur le PC, elle sert à la sélection des cartes sur l'XT.

En plus de ce connecteur soixante-deux broches, le PC/AT dispose d'un connecteur supplémentaire (fig 1.b) de trente-six broches pour six des huit connecteurs d'extension dont il est équipé. Les signaux disponibles sur ce connecteur sont:

**MEM CS 16** : actif à l'état bas, il indique un transfert mémoire de seize bits avec un cycle d'attente.

**I/O CS 16** : De même nature que MEM CS 16 mais indique un transfert d'E/S.

**DRQ0** : Qui sert aux demandes de DMA pour les transferts sur huit bits tout comme DRQ1 à DRQ3 qui se trouvent sur le connecteur soixante-deux broches.

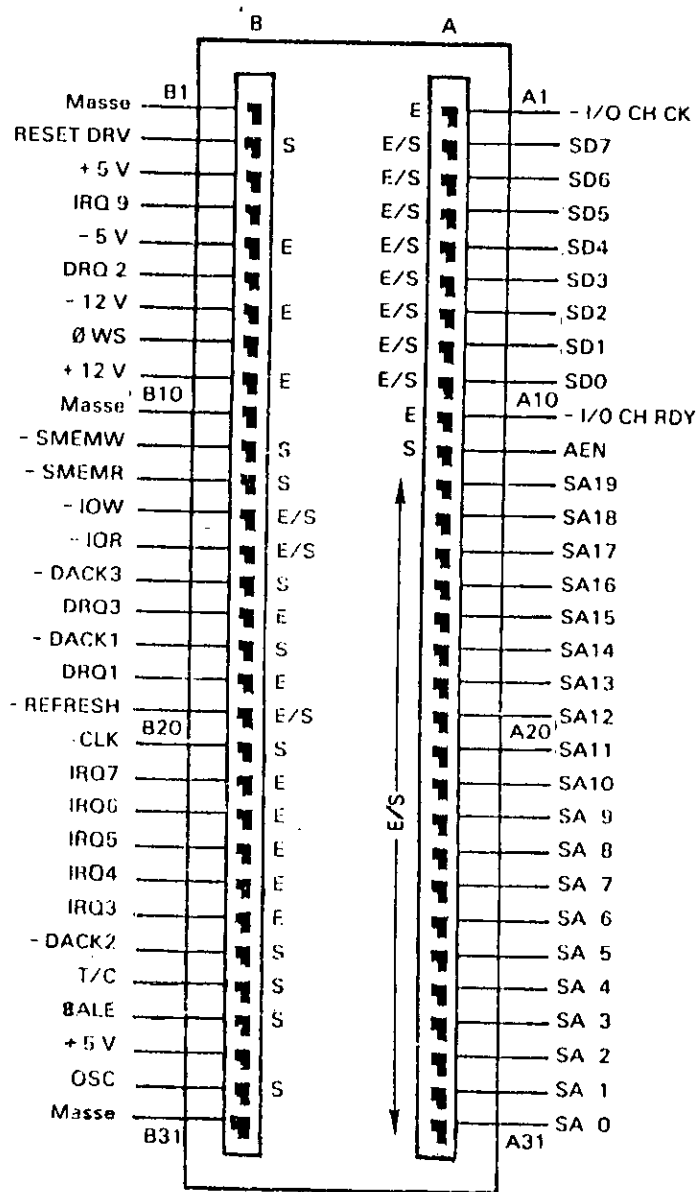
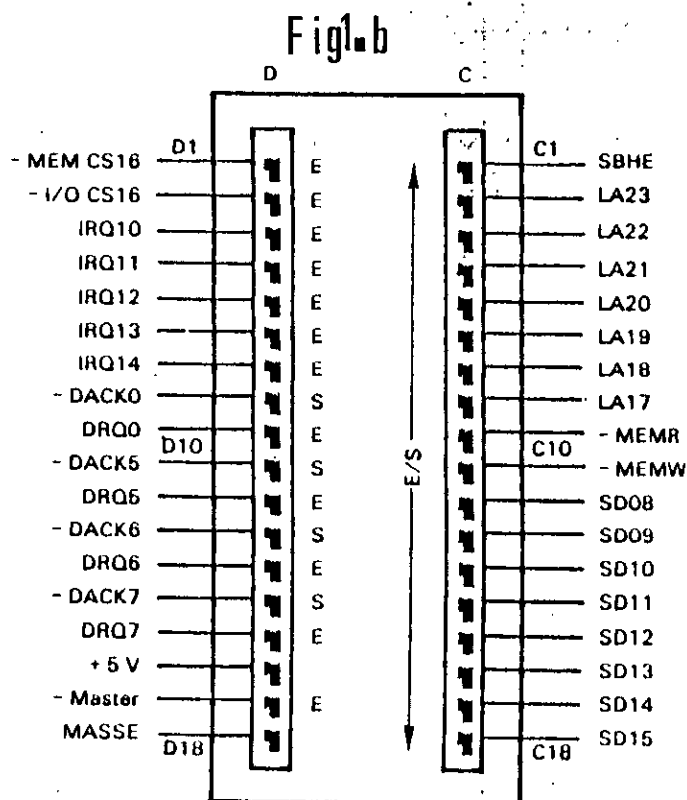
**DRQ5 à DRQ7** : servent quant à eux aux demandes de DMA sur seize bits.

**Master** (maître) : est une entrée utilisée avec un DRQ pour commander le système.

**SHBE** (System Bus High Enable) : est une ligne de validation pour le transfert d'octets sur SD08 à SD15. Elle est active à l'état haut.

**LA17 à LA23**: sont les lignes de poids fort pour l'adressage des mémoires et des E/S. Les adresses sont valides lorsque le signal ALE est à l'état haut. Elles ne sont pas verrouillées.

**SD08 à SD15** : sont les huit lignes de données pour l'octet de poids fort.



**Fig 1.a**

0 1 2 3 4 5 6 7 8 9 10

54 55 56 57 58 59 60 61 62 63

0000	0010	0020	0030	0040	0050	0060	0070	0080	0090	00A0
000F	001F	002F	003F	004F	005F	006F	007F	008F	009F	00AF
0400	0410	0420	0430	0440	0450	0460	0470	0480	0490	04A0
040F	041F	042F	043F	044F	045F	046F	047F	048F	049F	04AF
0800	0810	0820	0830	0840	0850	0860	0870	0880	0890	08A0
080F	081F	082F	083F	084F	085F	086F	087F	088F	089F	08AF
0C00	0C10	0C20	0C30	0C40	0C50	0C60	0C70	0C80	0C90	0CA0
0C0F	0C1F	0C2F	0C3F	0C4F	0C5F	0C6F	0C7F	0C8F	0C9F	0CAF
1000	1010	1020	1030	1040	1050	1060	1070	1080	1090	10A0
100F	101F	102F	103F	104F	105F	106F	107F	108F	109F	10AF
1400	1410	1420	1430	1440	1450	1460	1470	1480	1490	14A0
140F	141F	142F	143F	144F	145F	146F	147F	148F	149F	14AF
1800	1810	1820	1830	1840	1850	1860	1870	1880	1890	18A0
180F	181F	182F	183F	184F	185F	186F	187F	188F	189F	18AF
1C00	1C10	1C20	1C30	1C40	1C50	1C60	1C70	1C80	1C90	1CA0
1C0F	1C1F	1C2F	1C3F	1C4F	1C5F	1C6F	1C7F	1C8F	1C9F	1CAF
2000	2010	2020	2030	2040	2050	2060	2070	2080	2090	20A0
200F	201F	202F	203F	204F	205F	206F	207F	208F	209F	20AF
2400	2410	2420	2430	2440	2450	2460	2470	2480	2490	24A0
240F	241F	242F	243F	244F	245F	246F	247F	248F	249F	24AF
2800	2810	2820	2830	2840	2850	2860	2870	2880	2890	28A0
280F	281F	282F	283F	284F	285F	286F	287F	288F	289F	28AF
2C00	2C10	2C20	2C30	2C40	2C50	2C60	2C70	2C80	2C90	2CA0
2C0F	2C1F	2C2F	2C3F	2C4F	2C5F	2C6F	2C7F	2C8F	2C9F	2CAF
3000	3010	3020	3030	3040	3050	3060	3070	3080	3090	30A0
300F	301F	302F	303F	304F	305F	306F	307F	308F	309F	30AF
3400	3410	3420	3430	3440	3450	3460	3470	3480	3490	34A0
340F	341F	342F	343F	344F	345F	346F	347F	348F	349F	34AF
3800	3810	3820	3830	3840	3850	3860	3870	3880	3890	38A0
380F	381F	382F	383F	384F	385F	386F	387F	388F	389F	38AF
3C00	3C10	3C20	3C30	3C40	3C50	3C60	3C70	3C80	3C90	3CA0
3C0F	3C1F	3C2F	3C3F	3C4F	3C5F	3C6F	3C7F	3C8F	3C9F	3CAF

المعهد الوطني للتكنولوجيا  
 DIBLOUPTIQUE - المكنة  
 Ecole Nationale Polytechnique

$$\text{OFFSET}_{\text{HEXA}} = 400.R + 10.C + L$$

R : RANGEE  
 C : CARACTERE  
 L : LIGNE

0360	0370	0380	0390	03A0	03B0	03C0	03D0	03E0	03F0
036F	037F	038F	039F	03AF	03BF	03CF	03DF	03EF	03FF
0760	0770	0780	0790	07A0	07B0	07C0	07D0	07E0	07F0
076F	077F	078F	079F	07AF	07BF	07CF	07DF	07EF	07FF
0B60	0B70	0B80	0B90	0BA0	0BB0	0BC0	0BD0	0BE0	0BF0
0B6F	0B7F	0B8F	0B9F	0BAF	0BBF	0BCF	0BDF	0BEF	0BF
0F60	0F70	0F80	0F90	0FA0	0FB0	0FC0	0FD0	0FE0	0FF0
0F6F	0F7F	0F8F	0F9F	0FAF	0FBF	0FCF	0FD	0FEF	0FFF
1360	1370	1380	1390	13A0	13B0	13C0	13D0	13E0	13F0
136F	137F	138F	139F	13AF	13BF	13CF	13DF	13EF	13FF
1760	1770	1780	1790	17A0	17B0	17C0	17D0	17E0	17F0
176F	177F	178F	179F	17AF	17BF	17CF	17DF	17EF	17FF
1B60	1B70	1B80	1B90	1BA0	1BB0	1BC0	1BD0	1BE0	1BF0
1B6F	1B7F	1B8F	1B9F	1BAF	1BBF	1BCF	1BDF	1BEF	1BF
1F60	1F70	1F80	1F90	1FA0	1FB0	1FC0	1FD0	1FE0	1FF0
1F6F	1F7F	1F8F	1F9F	1FAF	1FBF	1FCF	1FD	1FEF	1FFF
2360	2370	2380	2390	23A0	23B0	23C0	23D0	23E0	23F0
236F	237F	238F	239F	23AF	23BF	23CF	23DF	23EF	23FF
2760	2770	2780	2790	27A0	27B0	27C0	27D0	27E0	27F0
276F	277F	278F	279F	27AF	27BF	27CF	27DF	27EF	27FF
2B60	2B70	2B80	2B90	2BA0	2BB0	2BC0	2BD0	2BE0	2BF0
2B6F	2B7F	2B8F	2B9F	2BAF	2BBF	2BCF	2BDF	2BEF	2BF
2F60	2F70	2F80	2F90	2FA0	2FB0	2FC0	2FD0	2FE0	2FF0
2F6F	2F7F	2F8F	2F9F	2FAF	2FBF	2FCF	2FD	2FEF	2FFF
3360	3370	3380	3390	33A0	33B0	33C0	33D0	33E0	33F0
336F	337F	338F	339F	33AF	33BF	33CF	33DF	33EF	33FF
3760	3770	3780	3790	37A0	37B0	37C0	37D0	37E0	37F0
376F	377F	378F	379F	37AF	37BF	37CF	37DF	37EF	37FF
3B60	3B70	3B80	3B90	3BA0	3BB0	3BC0	3BD0	3BE0	3BF0
3B6F	3B7F	3B8F	3B9F	3BAF	3BBF	3BCF	3BDF	3BEF	3BF
3F60	3F70	3F80	3F90	3FA0	3FB0	3FC0	3FD0	3FE0	3FF0
3F6F	3F7F	3F8F	3F9F	3FAF	3FBF	3FCF	3FD	3FEF	3FFF

**ANNEXE B CORRESPONDANCE ENTRE LES EMBLEMES DE L'ECRAN  
ET LES POSITIONS MEMOIRE**

**ANNEXE C SCHEMA DE CABLAGE DE LA CARTE GRAPHIQUE**



PAGE 50,132

```

;*****
;*****ce programme permet d'afficher sur un ecran de *****
;*****television grace ala carte grahique 8 bandes horizontales*
;*****de couleurs suivantes noir bleu rouge magenta vert *****
;*****cyan jaune blanc et la lettre A le chiffre 1*****
;***** et le symbole #*****
    
```

.286c

= B100  
= B300  
= B000

```

rouge equ 0b400h
bleu  equ 0b300h
vert  equ 0b000h
    
```

```

0000      pile segment para stack 'pile';resrve 256 octets
0000 0100f      db 256 dup(0)          ; en memoire pour
           00
    
```

1

```

0100      pile ends                ; la pile
    
```

```

0000      donnees segment para public 'donnees'
0000 4F 40 43 06 12 0E 10      val_reg db 4fh,40h,43h,06h,12h,0eh,10h,11h,00h,0fh,20h,00h,00h,00h
           11 00 0F 20 00 00 00
000E 3E 3E 41 41 41 41 7F      let_A  db 3eh,3eh,41h,41h,41h,41h,7fh,7fh,41h,41h,41h,41h
           7F 41 41 41 41
001A 1C 1E 1B 19 18 18 18      chi_1  db 1ch,1eh,1bh,19h,18h,18h,18h,18h,18h,18h,11h
           18 18 11
0024 18 18 18 18 18 18 18      symbol db 18h,18h,18h,18h,18h,18h,18h,18h,18h,00h,00h,18h,18h
           18 18 00 00 18 18
    
```

```

0031      donnees ends
    
```

```

0000      extra segment
0000      extra ends
    
```

```

0000      code segment para public 'code'
           assume cs:code ,es:extra ,ds:donnees
    
```

```

0000      affichage proc far
           ;*****
           ;*****initialisation du CRT*****
           ;*****
    
```

```

0000 B8 ---- R      mov ax , donnees
0003 8E D8          mov ds ,ax
0005 B9 000E        mov cx ,000eh
0008 8D 3E 0000 R   lea si,val_reg
000C E0 00          mov al ,00h
    
```

```

000E BA 0220      debut: mov dx ,220h      ; acces au registre
0011 EE          out dx ,al        ; d adresse
0012 BA 0221      mov dx ,221h      ; programmation des
0015 6E          outsb          ; registres internes
0016 FE C0        inc al
0018 45          inc si
0019 E2 F3        loop debut

```

```

;*****
;*****effacement de l ecran *****
;*****

```

```

001B B8 B400      mov ax ,rouge    ; ecriture dans tout
001E 8E C0        mov es ,ax       ; le plan memoire
0020 BF 0000      mov di ,0h       ; rouge et mise
0023 B0 FF        mov al ,0ffh    ; a 1
0025 FC          cld          ; de tous les
002E B9 3FFF      mov cx ,3fffh   ; bits
0029 F3/ AA      rep stosb      ;

```

```

002B B8 B000      mov ax ,vert     ; ecriture dans tout
002E 8E C0        mov es ,ax       ; le plan
0030 BF 0000      mov di ,00h     ; memoire vert et
0033 B0 FF        mov al ,0ffh    ; mise a 1
0035 FC          cld          ; de tous les bits
0036 B9 3FFF      mov cx ,3fffh   ;
0039 F3/ AA      rep stosb      ;

```

```

003B B8 B800      mov ax ,bleu     ; ecriture dans tout le
003E 8E C0        mov es ,ax       ; plan memoire bleu
0040 BF 0000      mov di ,00h     ; et mise a 1
0043 B0 FF        mov al ,0ffh    ; de tous les bits
0045 FC          cld          ;
0046 B9 3FFF      mov cx ,3fffh   ;
0049 F3/ AA      rep stosb      ;

```

```

;*****
;***** affichage de la mire *****
;*****

```

```

004B B8 B000      mov ax ,vert
004E 8E C0        mov es ,ax
0050 BF 0000      mov di ,00h

```

```

0053 B0 00      mov al ,00h
0055 B9 1FFF    mov cx ,1fffh
0058 FC        cld
0059 F3/ AA     rep  stosb
005E B8 B100    mov ax ,rouge
005E 8E C0      mov es ,ax
0060 BF 0000    mov di ,00h
0063 B9 0FFF    mov cx ,0fffh
0066 FC        cld
0067 F3/ AA     rep  stosb
0069 BF 2000    mov di ,2000h
006C B9 0FFF    mov cx ,0fffh
006F FC        cld
0070 F3/ AA     rep  stosb
0072 B8 B800    mov ax ,bleu
0075 8E C0      mov es ,ax
0077 BF 0000    mov di ,00h
007A B9 07FF    mov cx ,7ffh
007D FC        cld
007E F3/ AA     rep  stosb
0080 BF 1000    mov di ,1000h
0083 B9 07FF    mov cx ,7ffh
0086 FC        cld
0087 F3/ AA     rep  stosb
0089 BF 2000    mov di ,2000h
008C B9 07FF    mov cx ,7ffh
008F FC        cld
0090 F3/ AA     rep  stosb
0092 BF 3000    mov di ,3000h
0095 B9 07FF    mov cx ,7ffh
0098 FC        cld
0099 F3/ AA     rep  stosb

```

```

;*****
;*****tamponisation pour maintenir la mire*****
;*****

```

```

009B B0 FF      mov al ,0ffh
009D B9 FFFF    mov cx ,0ffffh
00A0 E2 FE      grande: loop petite
00A2 FE C8      petite: dec al
00A4 75 F7      jnz grande

```

```

;*****
;***** affichage du A en blanc sur un fond magenta*****
;*****

```

```

00A6 8D 3E 000E R      lea si ,let_A
00AA BF 0741          mov di ,1860
00AD B8 B000          mov ax ,vert
00B0 8E C0            mov es ,ax
00B2 FC            cld
00B3 B9 000C          mov cx ,0ch
00B6 F3/ A1          rep  movsb
                    ;*****
                    ;***** affichage de l en blanc sur un fond turquoise****
                    ;*****
00B8 8D 3E 001A R      lea si ,chi_1
00BC BF 0B18          mov di ,2840
00BF B8 B400          mov ax ,rouge
00C2 8E C0            mov es ,ax
00C4 FC            cld
00C5 B3 000A          mov cx ,0ah
00C8 F3/ A1          rep  movsb
                    ;*****
                    ;*****affichage du symbole # en jaune sur un fond rouge **
                    ;*****
00CA 8D 3E 0024 R      lea si ,symbol
00CE BF 0421          mov di ,10E0
00D1 B8 B300          mov ax ,vert
00D4 8E C0            mov es , ax
00D6 FC            cld
00D7 E3 000D          mov cx ,0dh
00DA F3/ A1          rep  movsb

00DC B4 4C            mov ah ,4ch ; retour au dos
00DE CB 21            int 21h ;
affichage endp
code ends
00E0                end affichage

```

Symbols-1

Segments and Groups:

Name	Size	Align	Combine	Class
CODE . . . . .	00E0	PARA	PUBLIC	'CODE'
DONNEES . . . . .	0031	PARA	PUBLIC	'DONNEE'
EXTRA . . . . .	0000	PARA	NONE	
PILE . . . . .	0100	PARA	STACK	'PILE'

Symbols:

Name	Type	Value	Attr
AFFICHAGE . . . . .	F PROC	0000	CODE Length = 00E0
BLEU . . . . .	Number	B300	
CHI_1 . . . . .	L BYTE	001A	DONNEES
DEBUT . . . . .	L NEAR	000E	CODE
GRANDE . . . . .	L NEAR	009D	CODE
LET_A . . . . .	L BYTE	000E	DONNEES
PETITE . . . . .	L NEAR	00A0	CODE
ROUGE . . . . .	Number	B400	
SYMBOL . . . . .	L BYTE	0024	DONNEES
VAL_REC . . . . .	L BYTE	0000	DONNEES
VERT . . . . .	Number	B000	

165 Source Lines  
 165 Total Lines  
 36 Symbols

49424 Bytes symbol space free

0 Warning Errors  
 0 Severe Errors