

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

12/88

وزارة التعليم و البحث العلمي
Ministère de l'Enseignement et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE

المدسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

DEPARTEMENT : **Électronique**

PROJET DE FIN D'ETUDES

SUJET

**CARTE D'ÉVALUATION POUR
LE COPROCESSEUR GRAPHIQUE
I 82786 D'INTEL**

Proposé par : A. HAMAMI

Etudié par : S. ILES

Dirigé par : A. HAMAMI

N. MAHAMDI

PROMOTION Janvier 88

ECOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

CARTE D'EVALUATION
POUR LE COPROCESSEUR GRAPHIQUE
I82786 D'INTEL

PRESENTE PAR :

N . MAHAMDI
S . ILES

R E M E R C I M E N T S

Nous voudrions à travers ce document , humblement mais chaleureusement, remercier Monsieur et Madame Ahmed HAMAMI, pour leurs dévouements, leurs attentions, et leurs assistances, tout le long de notre projet.

Nous voudrions remercier également toute l'équipe du laboratoire, a savoir:

- N. BOUZA
- Z. TOUATI
- F. EL-MANSALI
- M. CHAOUCH
- N. REZZOUG

pour leurs accueils , orientations , suggestions , aides.....

Enfin que soit remerciés tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce projet.

TABLES DES MATIERES

Page

PRESENTATION

Première partie : GENERALITES

1 - Introduction	1
2 - Le terminal de visualisation	2
3 - Les controleurs graphiques VLSI.	4

Deuxieme partie : **DESCRIPTION DU GDC**

1 -	INTRODUCTION	5
2 -	ARCHITECTURE INTERNE	5
	2.1 - LE PROCESSEUR GRAPHIQUE	6
	2.1.1 Introduction	6
	2.1.2 Registres du processeur graphique	6
	2.1.3 Structure des commandes du processeur graphique	9
	2.1.4 Clipping rectangle	11
	2.1.5 Le mode pick	11
	2.1.6 Jeu de caractere	11
	2.1.7 Operations de transfert de blocs	12
	 2.2 - LE PROCESSEUR DE VISUALISATION	12
	2.2.1 Introduction	12
	2.2.2 Registre du processeur de visualisation	12
	2.2.3 Organisation du bit-map	18
	2.2.4 Fenetrages	18
	2.2.5 Le curseur	18
	2.2.6 Controle du CRT	18
	2.2.7 Grossissement d'une figure	18
	2.2.8 Descripteurs de fenetre	19
	 2.3 - L'UNITE D'INTERFACE DE BUS (BUI) AVEC LE CONTROLEUR DE DRAM	21
	2.3.1 Internal Relocation	21
	2.3.2 BUI Control Register	21
	2.3.3 Refresh Control Register	22
	2.3.4 DRAM Control Register	22
	2.3.5 Registre de priorite	23
3 -	JEU D'INSTRUCTIONS	
	3.1 Commandes non graphiques	24
	3.2 Commandes de controle graphique	25
	3.3 Commandes géométriques	26
	3.4 Commandes de transfert	27

Troisième partie : **CONCEPTION DU MODULE GRAPHIQUE**

1 - INTRODUCTION	29
2 - ETUDE DE LA CARTE	30
2.1 Horloges du GDC	30
2.1.1 Horloges systeme Clk	30
2.1.2 Horloges video Clk	30
2.2 Bus d'adresse	30
2.2.1 Décodage de la carte	31
2.2.2 Réalisation du circuit d'adressage	31
2.3 Signaux de controles.	31
2.3.1 Le signal RESET.	31
2.3.2 Le signal BHE.	32
2.3.3 Le signal MIO.	32
2.3.4 Le signal SEN.	33
2.4 Analyse des échanges CPU -GDC.	33
2.4.1 Analyse des opérations d'écriture/lecture du GDC.	33
2.4.1.1 Signaux de controles	33
2.4.1.2 Bus de donnée.	34
2.4.2 Analyse des opérations d'écriture/lecture du CPU.	35
2.4.2.1 Cycle de lecture	35
2.4.2.2 Cycle d'écriture	35
2.4.3 Conclusion	36
2.5 Conception	37
2.5.1 Circuit d'écriture	37
2.5.2 Ecriture vue du cote CPU	37
2.5.3 Circuit de lecture	37
2.5.4 Lecture vue du cote CPU.	38
2.6 Mémoires	38
2.6.1 Mémoire d'écran	38
2.6.2 Rafraichissement de la DRAM.	38
3 - ADAPTABILITE DE LA CARTE:	39

**Quatrième partie : REALISATION ET MISE AU POINT
DE LA CARTE**

4.1	SYSTEME DE DEVELOPPEMENT	41
4.2	INTRODUCTION	42
4.3	INITIALISATION DE LA CARTE.	42
4.3.1	Registres de controles	42
4.3.2	Registres de visualisation	43
4.3.2.1	Calculs des parametres de synchronisation.	43
4.3.2.2	Séquences d'initialisation.	46
4.4	LE MONITEUR GRAPHIQUE ET SA STRUCTURE	47
CONCLUSION		49
BIBLIOGRAPHIE		50
ANNEXE :	Schéma général de la carte	

P R E S E N T A T I O N

Le département Péri-informatique de L'UR / ENSI s'occupe de la réalisation des terminaux graphiques et de l'arabisation des imprimantes .

L'activité du développement des terminaux graphiques , nécessite l'utilisation de processeurs graphiques , ce qui aboutit donc à un problème de choix de ceux-ci, selon leur performances .

C'est ainsi que le laboratoire est amené à concevoir une carte d'évaluation du coprocesseur graphique I82786 d'Intel, mis sur le marché dans le courant de l'année 1986.

le projet consiste, d'une part en la conception et la réalisation d'une carte d'évaluation à base de ce nouveau produit, interfaçé avec un micro-contrôleur 8 bits, le Intel 8031, et d'autre part à implementer un software minimum permettant le dialogue entre le C.P.U et le G.D.C , ainsi que réaliser, autant que possible, les fonctions de traçé.

premiere partie

GENERALITES

- 1 - Introduction.
- 2 - Le terminal de visualisation.
- 3 - Les controleurs graphiques VLSI.

1: INTRODUCTION

Le graphique a tendance à prendre de plus en plus d'importance dans le domaine de l'informatique. Jusqu'à présent cela a été l'une des branches les plus spectaculaires des technologies informatiques.

Outre l'attrait de nouveauté, l'image informatique offre un support de représentation synthétique et universel de l'information.

C'est la raison pour laquelle ses domaines d'applications sont multiples. On peut regrouper ces domaines en cinq principaux secteurs: Arts graphiques, Modélisation, Simulation, Graphisme de gestion, conception des produits industriels (CAO, CFAO).

Le premier terminal d'affichage graphique est apparu au début des années 50 utilisant un tube cathodique (CRT) pour générer des figures simples, à des vitesses relativement élevées.

Avant l'apparition des coprocesseurs graphiques, la mise au point des terminaux graphiques nécessitait beaucoup d'efforts pour réaliser les fonctions de base du traitement d'image, au niveau matériel.

Aujourd'hui le graphique est un domaine en pleine expansion dans tous les pays. Son succès populaire est dû principalement à la fonction de communication que lui ont donné les medias (films publicitaires, génériques, etc...)

2 : LE TERMINAL DE VISUALISATION

Un terminal de visualisation est généralement composé de quatre blocs fonctionnels :

- Un processeur maître et ses périphériques (port seriel , interface clavier).
- Une mémoire de travail pour le processeur maître.
- Un contrôleur d'affichage qui, comme son nom l'indique, permet le contrôle d'affichage et le transfert d'informations de la mémoire d'écran vers le CRT .
- Un tube cathodique (cathod Ray Tube), qui constitue le support d'affichage (fig 1)

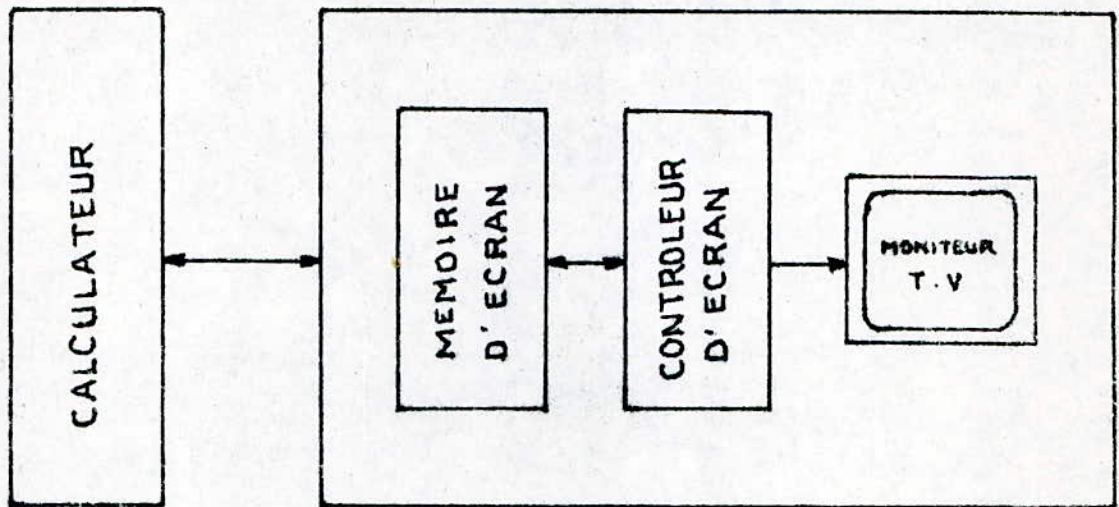


fig 1 : Terminal de visualisation

Le tube cathodique

Le tube cathodique est utilisé pour permettre la conversion des différents signaux électriques de l'ordinateur en une image.

Généralement, le tube est composé d'un canon à électrons (trois canons dans le cas d'un tube trichrome), d'un système de concentration de ces électrons pour en faire un faisceau qui attaque une couche de luminophores sur l'écran monochrome. Le point de l'écran attaqué par le faisceau d'électrons sera illuminé et visible de l'extérieur du tube. Un système de déflexions horizontale et verticale du faisceau permettra de balayer l'ensemble des points de l'écran (fig 2).

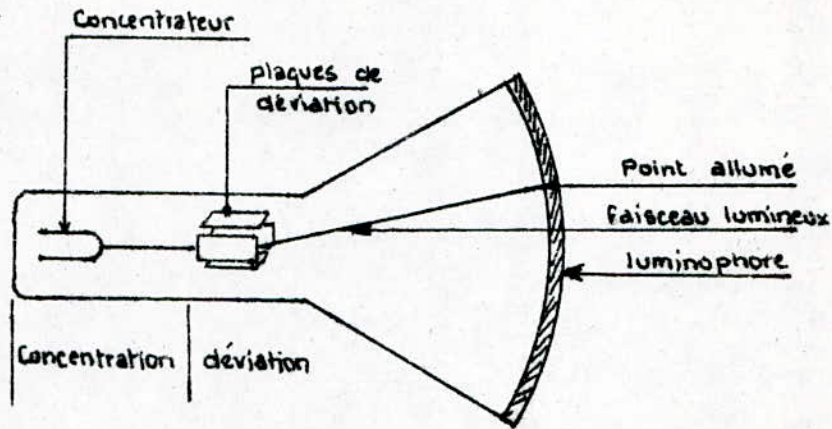


fig 2 : Tube cathodique

Si l'on associe à chaque point de l'écran un bit de la mémoire, on peut ainsi commander l'affichage du point en l'allumant, si ce bit est à l'état logique "1", et en l'éteignant s'il est à l'état "0".

En utilisant le principe de balayage télévision et en balayant d'une manière séquentielle la mémoire d'image on peut aisément produire une image sur l'écran.

Les différents signaux de synchronisation pour le contrôle de l'affichage sont fournis par le contrôleur d'écran.

Les contrôleurs d'écran sont des processeurs spécialisés suivant le type d'application, alphanumérique (CRTC) ou graphique (GDP) leur rôle principal est de lire l'information contenue dans la mémoire, d'effectuer certains traitements éventuels et de renvoyer l'information sous forme de bits en parallèle dans un registre à décalage appelé registre vidéo. Ce registre sérialise l'information vers le tube pour l'affichage (fig 3).

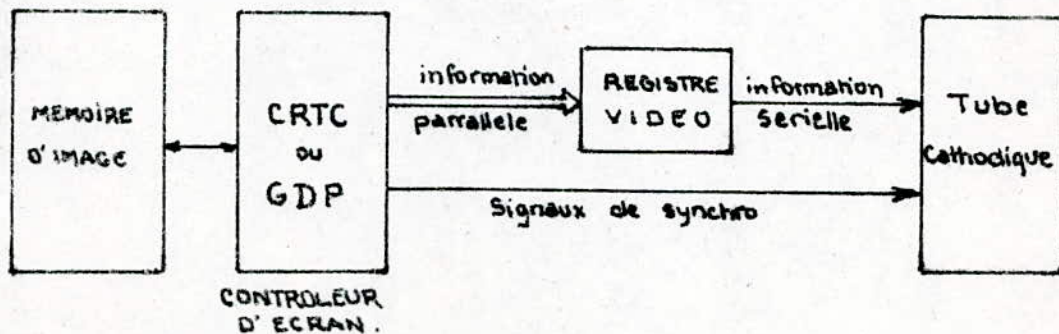


fig 3 : Contrôleur d'écran

3: LES CONTROLEURS GRAPHIQUES VLSI

Une application graphique sans un processeur spécialisé implique des temps de traitement très longs. C'est pourquoi des circuits, permettant de réaliser les fonctions de base du traitement d'image au niveau matériel, ont été développés.

En effet ils assureront l'interface entre l'écran graphique et le calculateur associé, le libérant ainsi des calculs de bas niveaux (gestion de mémoire spécialisée, rafraichissement de l'écran, sérialisation des pixels à afficher, fonctions graphiques répétitives, gestion des fonctions de synchronisation de l'écran, etc).

Aujourd'hui lorsqu'on parle de processeurs graphiques plusieurs noms viennent s'insérer dans notre liste : l'Am 95c60 ; Le NS DP 8500 ; Le Thomson TS 68483 ; Le Nec (micro PD 7220) ; Le Motorola (MC68486) et enfin le INTEL (I82786)

Ils présentent trois avantages principaux :

- Leur cout est relativement peu élevé, par rapport aux controleurs cablés ;
- Ils s'intègrent facilement dans presque tout système bâti autour d'un microprocesseur universel de 8, 16 ou 32 bits ;
- Ils sont capables de traiter des volumes d'information très importants en peu de temps grâce à leurs instructions spécialisées. Cette faculté, associée à d'autres caractéristiques matérielles des systèmes graphiques, améliore sensiblement leurs performances.

Le circuit d'Intel est le premier vrai coprocesseur graphique et certainement pas le dernier. Il intègre, le processeur graphique le processeur d'affichage, le controleur de DRAM et l'unité d'interface de bus. Offrant ainsi de hautes performances graphiques.

Deuxieme partie

D E S C R I P T O N D U G D C

- 1 - Introduction
- 2 - Architecture interne
- 3 - Jeu d'instructions

1 : INTRODUCTION

Le I82786 est un VLSI de technologie HC_MOS III, sa liaison avec le système-hôte peut se faire par un bus qui peut être configuré en 8 ou 16 bits par programme. Les caractéristiques de son bus sont cependant optimisées pour fonctionner en liaison avec le CPU I80286, son architecture interne est orientée I80286, il supporte les fonctions graphiques de base : fenêtrage, bit-bit (24 Mbits/s), cha-bit (25000 caractères/s), zoom, tracé de vecteurs, cercles, arcs, polygones, créations de masques, opérations logiques au niveau de bits. Il dispose de son propre jeu d'instructions, il peut adresser jusqu'à 4 Mega-octets de mémoire d'affichage et 8 plans mémoire pour la couleur, sa vitesse d'horloge est de 20 MHz max, sa résolution écran-type pilote 640x480x8 Bits ou 1400x1400x1 Bit.

2 : ARCHITECTURE INTERNE

L'architecture interne du CRTC se décompose en trois modules de base (fig 4) :

- Le Procésseur de visualisation (DP) ;
- Le Procésseur graphique (GP) ;
- L'unité d'interface de bus (BUI) avec le contrôleur de DRAM.

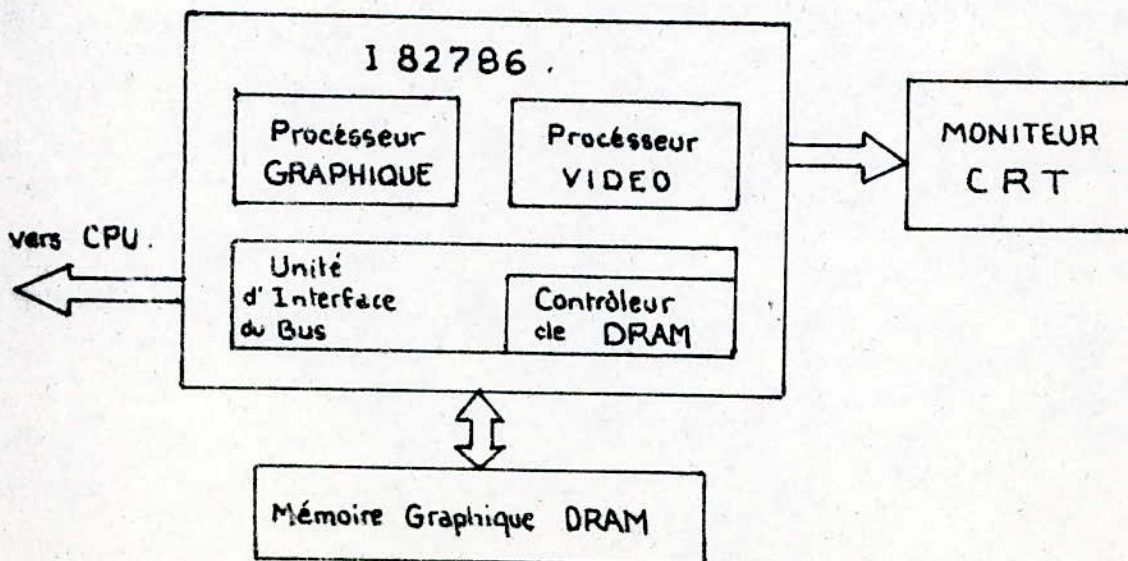


fig 4 : Sous système graphique

2.1 : LE PROCESSEUR GRAPHIQUE

2.1.1 : Introduction:

Le Processeur graphique est un processeur indépendant qui génère et met à jour toutes les objets graphiques, il exécute les commandes graphiques placées dans une liste en mémoire externe, supporte le tracé rapide de lignes élémentaires, arcs de cercles, cercles, rectangles, polygones, caractères et autres courbes, grâce à un algorithme micro-programmé. Il détermine en fonction des paramètres de masque et de transparence, les pixels à traiter.

2.1.2 : Registres du Processeur Graphique

ADRESSES	REGISTRES
BASE + 20H	O P C O D E
BASE + 22H	LINK ADRESS (LOWER WORD)
BASE + 24H	LINK ADRESS (UPPER WORD)
BASE + 26H	S T A T U S
BASE + 28H	INSTRUCTION POINTER (LOWER)
BASE + 2AH	INSTRUCTION POINTER (UPPER)

fig 5 : Registres du processeur graphique

Le processeur graphique (G.P) est constitué de différents registres de controles, ces registres sont au nombre de six , ils peuvent etre accedés directement par le CPU externe (fig 5).

1) Registre d'état :

Ce registre interne de 8 bits contient les indicateurs suivant :

	7	6	5	4	3	2	1	0
Adresse de BASE + 26H	GPOOL	GRCD	GINT	GPSC	GBCOV	GBMOV	GCTP	GIBMD

1. GPOLL - Poll state
Ce bit activée indique que le processeur graphique est en état d'attente .
2. GRCD - Reserved Command
Ce bit est activée si le processeur graphique détecte un opcode illegal .
3. GINT -Generate Interrupt
Ce bit est activé si la commande INT-GEN (génération d'interruption) est donné au processeur graphique par le CPU.
4. GPSC - pick successful
Ce bit est activé à la fin d'une commande Pick , dont le résultat est positif , c'est à dire que le tracé s'inscrit à l'interieur de la fenetre définie par la commande Define Clip-Rectangle.
5. GBCOV - bit-map overflow for BIT-BLT or CHAR-BLT
Ce bit est activé si une tentative d'exécution d'une commande CHAR ou BIT -BLT atteint une zone mémoire hors de la fenetre définie .
6. GBMOV - bit-map overflow for geometric commands
Ce bit est activée si le résultat de tracés de figure géométrique tombe hors de la fenetre définie .
7. GCTP -Character Trap
Ce bit est activé , si l'un des caractères donné comme paramètre de la commande CHAR , à son bit Trap , mis à 1.
8. GIBMD - Illegal Bit Map Define
Ce bit est activé , si la commande DEF-BIT-MAP est exécuté avec des paramètres incorrects .

2) Pointeur d'instruction :

Le pointeur d'instructions contient , l'adresse sur 22 bits, de l'instruction en cours d'exécution .

Il est constitué de deux registres .

		Adresse
GCIPL	Instruction Pointer Lower	BASE + 28H
GCI PH	IC Upper	BASE + 2AH

2.1.3 : Structure des commandes du processeur graphique

Le format standart des commandes est figuré ci-dessous :

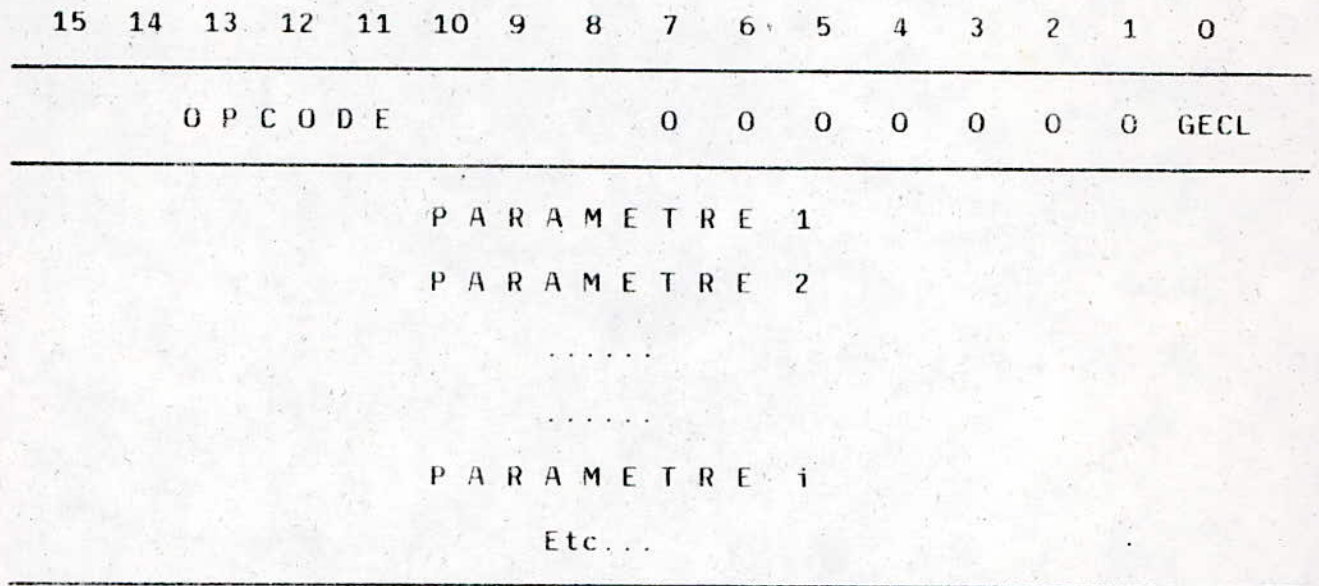


fig 7 : format des commandes

Chaque commande consiste en un code d'opération (Opcode), un bit de fin de liste de commande (GECL) et une liste de paramètres relatifs à la commande.

Chaque fois que le processeur graphique reçoit une commande, il teste le bit GECL. Si le bit GECL est à "0", la commande est exécutée et la commande suivante sera donc cherchée de la liste de commande. Si le bit GECL est à "1", la commande ne sera pas exécutée et le processeur graphique se mettra en état d'attente (POLL-STATE). Le processeur reste dans cet état jusqu'à réception d'une commande LINK dont le bit GECL est 0, la commande LINK indique au processeur graphique qu'une liste de commandes a été placée en mémoire pour exécution.

L'adresse mémoire où se trouve la liste en question est donnée par les deux paramètres de la commande LINK.

La commande LINK est placée directement dans les registres internes du processeur graphique. La structure d'une liste de commande est donnée par la figure 8 :

REGISTRE DE CONTROLE

LINK	0
LINK ADRESS LOWER	
LINK ADRESS UPPER	

MEMOIRE EXTERNE

	ECL
OPCCODE 1	0
PARAMETRE 1	
OPCODE 2	0
PARAMETRE 1	
PARAMETRE 2	
PARAMETRE 3	
OPCODE 3	0
POINTER	
ENTER MACRO	
OPCODE 7	0
OPCODE 8	0
PARAMETRE 1	
PARAMETRE 2	
OPCODE 9	0
STOP OPCODE 10	1
GRAPHICS SUBROUTINE	
OPCODE 4	0
PARAMETRE 1	
OPCODE 5	0
OPCODE 6	0
PARAMETRE 1	
PARAMETRE 2	
EXIT MACRO	
PARAMETRE 1	
PARAMETRE 2(LINK)	

fig 8 : Structure d'une liste de commande

2.1.4: CLIPPING RECTANGLE

Le clipping rectangle , définit la zone mémoire dans laquelle le processeur graphique peut dessiner , il doit être défini directement après la commande DEF-BIT-MAP .

Par défaut , le clipping rectangle occupe tout le bit-map.

2.1.5: LE MODE PICK

Le mode pick est un mode dans lequel le processeur graphique calcule tous les points à dessiner sans les mettre , dans le bit-map (mémoire affichée) , c'est un mode de tracé sans écriture dans la mémoire . Si un des points calculé s'inscrit dans la zone définie par le clipping rectangle , le processeur graphique met à 1 le bit GPSC de son registre d'état .

2.1.6: JEU DE CARACTERE

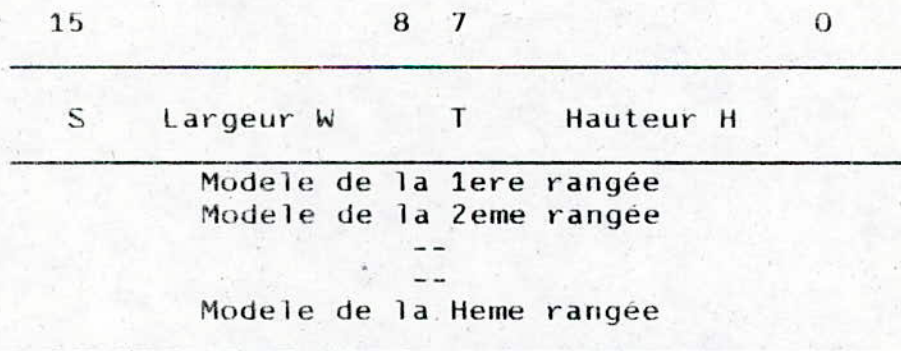
Les jeux de caracteres sont stockés dans la mémoire externe à partir d'une adresse paire.

L'information du caractère est stockée dans des mots consécutifs formant un bloc caractère.

Le jeu de caractère est sélectionné , par programmation de son adresse de base à travers la commande Def-Char-Set.

Le jeux peuvent être définis pour une codification de 8 bits ou 16 bits.

Le bloc de caractère a le format suivant:



S : bit indique si mis à 1 que le caractère sera superposé au précédent.

T : bit indique si mis à 1 , par le GDC , que le caractère dépasse la largeur allouée.

2.1.7 : OPERATIONS DE TRANSFERT DE BLOCS

L'exécution des opérations de transfert de blocs (BitBLT ops ou Raster ops), doit être aussi rapide que possible et totalement indépendante de l'organisation de la mémoire graphique.

Ces opérations de blocs permettent, par exemple, de définir une matrice $N \times N$ pixels dans la trame, puis d'exécuter une opération sur tous les pixels, et de transférer le résultat vers une matrice de même dimension située, ailleurs dans la mémoire.

2.2 : LE PROCESSEUR DE VISUALISATION

2.2.1: Introduction

Le processeur de visualisation est un processeur indépendant qui contrôle l'affichage du bit-map, il génère aussi les signaux de synchronisations et d'effacement pilotant le CRT.

Il gère les fenêtres, le curseur, les fonctions graphiques répétitives et la serialisation de pixels à afficher.

2.2.2: Registres du processeur d'affichage (ou visualisation)

Il existe deux différentes sortes de registres pour le processeur d'affichage :

1) Registres du processeur d'affichage

Ils sont au nombre de six, ils sont directement accessibles par le CPU externe (fig 9).

ADRESSE	REGISTRE
BASE + 40H	O P C O D E
BASE + 42H	MEMORY ADRESS (LOWER WORD)
BASE + 44H	MEMORY ADRESS (UPPER WORD)
BASE + 46H	REGISTRE - ID
BASE + 48H	S T A T U S
BASE + 4AH	DEFAULT VDATA

fig 9 : Registres du processeur de visualisation

Les quatre premiers registres sont utilisés pour envoyer des commandes au processeur d'affichage, le cinquième registre c'est le registre d'état, le dernier, c'est le registre qui contient la valeur par défaut de la video.

a) Registre d'état :

Ce registre interne de 8 bits contient les indicateurs suivants :

	7	6	5	4	3	2	1	0
Adresse de BASE + 48H	FRI	RCD	DOV	FMT	BLK	EVN	ODD	ECL

1. FRI- Frame interrupt.

Ce bit est activé toute les n trames ; où n est une valeur chargée dans le registre Frint (Registre Frame Interruption), variant entre 1 et 256 . Il est utilisé, par exemple, pour des applications d'animation, ou pour régler des vitesses de clignotement de caractères .

2. RCD-Reserved Command.

Ce bit est activé à la suite d'une commande illégale donnée au processeur d'affichage .

3. DOV-Descriptor Overrun.

Ce bit est activé si le processeur d'affichage n'a pas terminé le chargement de descripteurs de fenetre, lorsque l'effacement horizontal prend fin .

4. FMT-FIFO Empty.

Ce bit indique que la FIFO d'affichage est vide.

5. BLK-Blank.

Ce bit indique que le signal BLANK est activé.

6. EVN-Even Field.

En mode entrelacé durant le balayage de la trame paire, ce bit est mis à 1 .

7. ODD-Odd Field.

En mode entrelacé durant le balayage de la trame impaire, ce bit est mis à 1 .

8. ECL-End of Command List.

Ce bit est activé pour indiquer au CPU externe que l'exécution de la liste de commande est terminée.

b) Registres de controles d'affichage :

Les registres de controle d'affichage sont chargés sous controle du processeur graphique durant la période d'effacement vertical en réponse à une commande LOAD ALL (fig 10) .

Ils définissent pour le processeur d'affichage :

- Les caractéristiques des signaux de synchronisation à générer.
- La forme et la position du curseur.
- L'adresse des descripteurs de fenetre , etc .

00h : VSTAT : Video Status



bit C : génère le curseur s'il est mis à 1.

bit D : visualise le curseur s'il est mis à 1.

01h : IntMsk : Interrupt Mask Register



Les 8 bits de IntMask valident les interruptions du 182786, correspondantes respectivement, aux 8 bits du registre Display Processor Register Status (DP.Status) .
L'état 0 inhibe l'interruption .

02h : TripPt : Trip Point



Le GDC commence à visualiser les données en FIFO à partir du parametre TripPt .

03h : Frint : Frame Interrupt Register .



Frint : nombre de trames à visualiser .
(voir bit FRI du DP.Status)

04h : Registre RESERVE .

05h : CRT Mode

----- IL W S B AA -----

Bits IL : 00 ---> mode non entrelaçés
 01 ---> mode reserve
 10 ---> mode entrelaçés
 11 ---> mode entrelaçés-synchrone

Bit W : indique si les signaux de synchronisation sont générés ou non (si W=1), par le i82786.

Bit S : indique si les signaux de synchronisation Hsyn et Vsyn sont en sortie (S=1) ou en entrées (S=0)

Bit B : indique si le signal Blank est en sortie (B=1) ou en entrée (B=0) .

Bit AA : indique le mode d'accélération vidéo :

00 ---> mode normal (25 MHz)
10 ---> mode rapide (50 MHz)
01 ---> mode très rapide (100 MHz)
11 ---> mode ultra rapide (200 MHz)

Les registres suivants concernent les parametres de synchronisation des signaux video .

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
06h			--														HSynStp
07h			--														HFIdStrt
08h			--														HFIdStp
09h			--														LineLen
0Ah			--														VSynStp
0Bh			--														VFIdStrt
0Ch			--														VFIdStp
0Dh			--														FramLen

Les parametres à programmer dans ces registres sont donnés par la figure 10

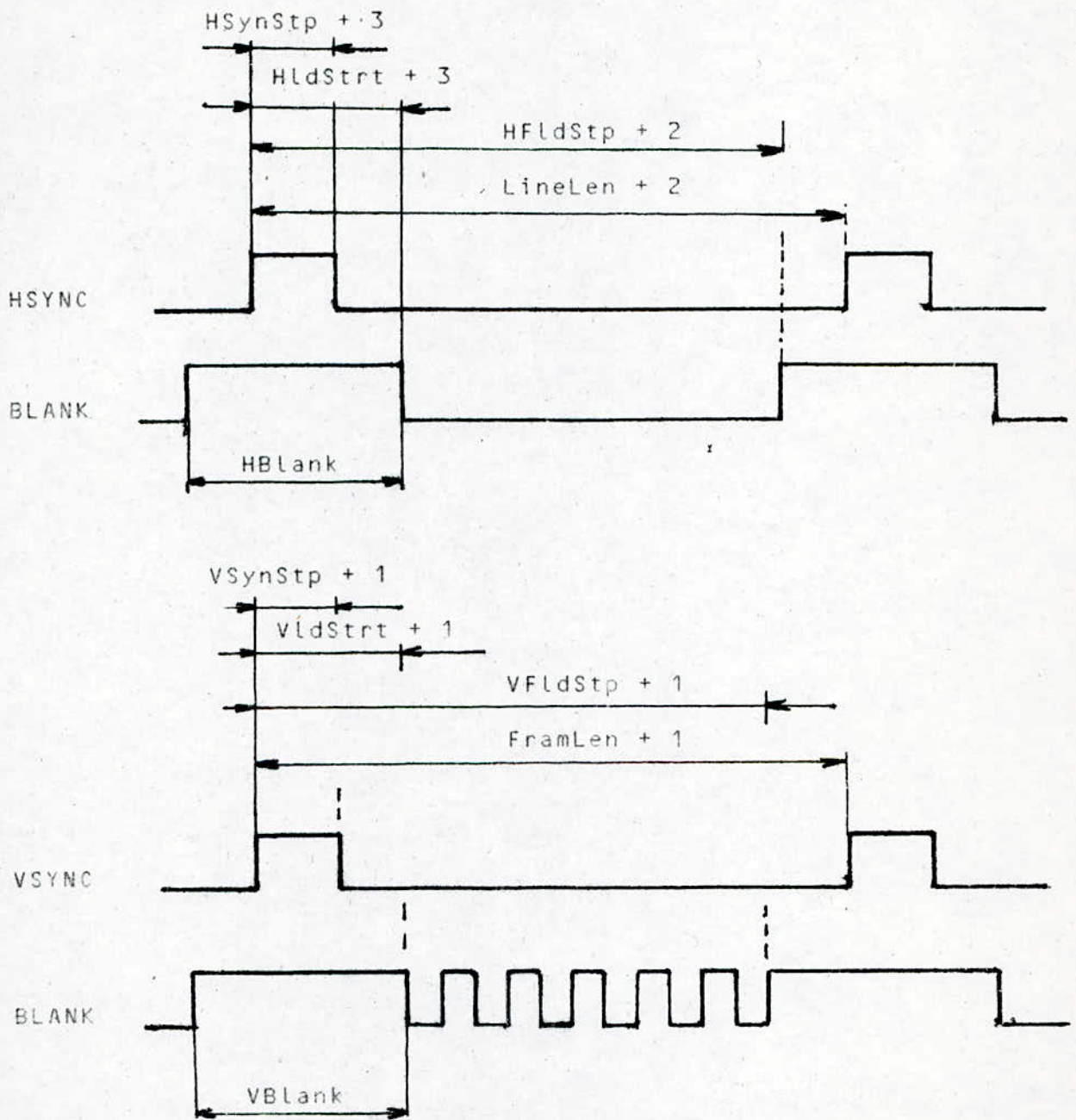


Fig 10

0Eh : DéSCRIPTOR Adress.Pointer (L)
 0Fh : DéSCRIPTOR Adress.Pointer (U)

Ces deux registres indiquent sur 32 bits , au i82786 , l'adresse en mémoire de la fenetre a afficher.

10h : Registre réservé

11h

XZoom	YZoom
-------	-------

Contient les facteurs de grossissement horizontal (XZoom) et vertical (YZoom).

12h :

----	FldColor
----	BdrColor

FldColor : définit, codé sur 8 bits, la couleur de fond (Background) du champs à être visualiser en absence de fenetres .

BdrColor : définit , codé sur 8 bits , la couleur du bord de la fenetre à afficher .

14h :

----	1BppPad
----	2BppPad
----	4BppPad

Ces registres fixent les bits non utilisés de la couleur , en mode 1 ,2, ou 4 bits par pixels.

17h : Csr Mode : cursor mode

S	X	T	CSt	CSc	-	CsrPad
---	---	---	-----	-----	---	--------

bit S : définit la taille de la matrice du curseur à generer (8 x 8 pixels pour S=0 et 16 x 16 pour S=1)

bit X : indique si le curseur est en mode normal (X=0) ou en mode croix (c'est à dire formé du croisement de deux lignes perpendiculaires, sur l'écran).

bit T : indique si le curseur est en mode opaque (T=1) ou transparent (T=0).

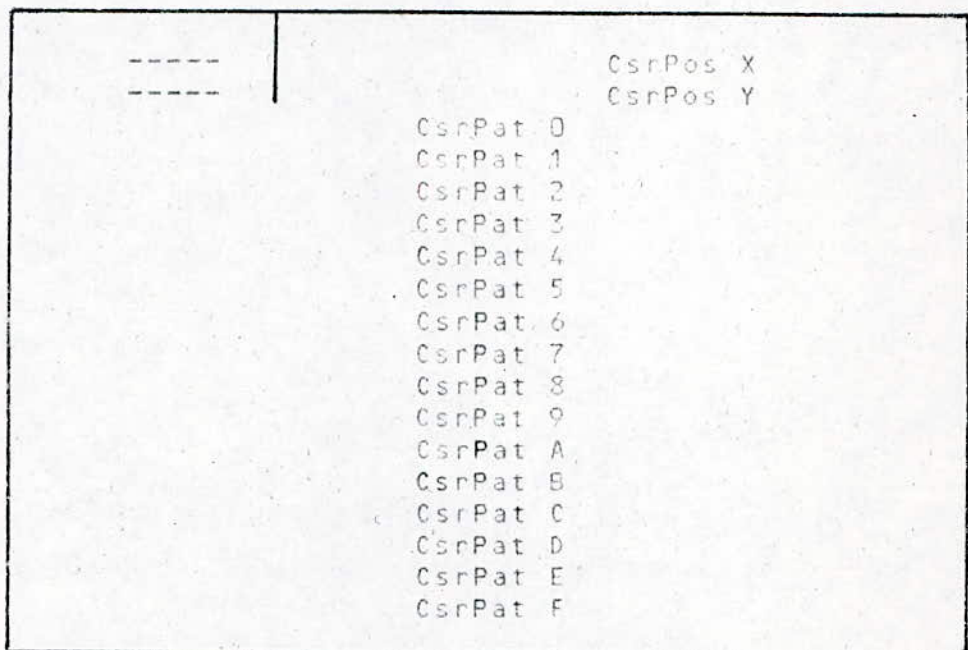
bits CSt : indique le status du curseur , pendant son affichage.

bits CSC : indique si le curseur sera visualisée en background (10) , en foreground (01) , en block (11) .

CsrPad : Ce registre fixe les bits non utilisés, à la couleur du curseur.

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

18h :
19h :
1Ah :
1Bh :
1Ch :
1Dh :
1Eh :
1Fh :
20h :
21h :
22h :
23h :
24h :
25h :
26h :
27h :
28h :
29h :



CsrPos X : contient l'abscisse X de la position du curseur

CsrPos Y : contient l'ordonnée Y de la position du curseur

CsrPat i : indiquent le modèle du curseur sur 16 mot de 16 bits, en mode 8 x 8 pixels , seuls les poids forts des huit premiers registres sont pris en compte.

2.2.3: ORGANISATION DU BIT-MAP

Le bit-map est la zone mémoire qui contient les informations affichables .

L'information affichée peut être une partie du bit-map. Un pixel dans le bit-map peut être défini par 1, 2, 4 ou 8 bits, ceci permet d'utiliser aussi bien le CRT monochrome que couleurs avec 256 niveaux de gris (ou couleurs) simultanément affichable .

2.2.4: FENETRAGES

La " fenêtre " matérielle (hardware window) est un mécanisme qui permet le changement dynamique de la trame affichée sans changer le contenu de la mémoire d'écran ; une fenêtre peut résider n'importe où dans la mémoire écran (bit-map).

La fenêtre matérielle peut avoir une taille arbitraire et peut être placée à la frontière d'un bit .

Pour éviter le problème du positionnement de la fenêtre matérielle, il faut que celle-ci soit alignée à la même frontière que les mots classiques dans la mémoire graphique (ici il s'agit de la frontière de 16 bits).

2.2.5: LE CURSEUR

Le processeur d'affichage supporte un curseur simple qui peut être représenté par 8 x 8 pixels ou 16 x 16 pixels, il peut être positionné n'importe où sur l'écran et peut être défini soit en mode transparent soit en mode opaque .

2.2.6: CONTROLE DU CRT

Les signaux de synchronisation Hsynch, Vsynch et le BLANK sont programmables . La période et le rapport en ligne de chacun, sont programmables séparément permettant ainsi l'utilisation de n'importe quel type de moniteurs .

2.2.7: GROSSISSEMENT D'UNE FIGURE : LE ZOOM

L'effet zoom (au sens d'agrandissement) est obtenu par multiplication des pixels (leur répétition) , jusqu'à 64 fois en horizontal et en vertical .

Chaque pixel affiché est répété dans les deux axes orthogonaux, le nombre de fois indiqué dans l'opération de zoom .

Par exemple pour un facteur de zoomX égale à 20, et zoomY égale à 30 chaque point de la figure sera représenté par 20 x 30 = 600 points sur l'écran . Pour ce faire il suffit d'insérer dans le registre de contrôle correspondant le paramètre choisi .

2.2.7: DESCRIPTEURS DE FENETRE

L'image affichée à l'écran provient d'un intervalle contigu de la mémoire graphique (bit-map).

Le processeur hôte divise virtuellement l'écran (la trame) en bandes horizontales, la largeur d'une bande (nombre de lignes la constituant) est arbitraire.

Une nouvelle bande peut être définie sur toute ligne en cours (ce qui ouvre un choix de la forme des fenêtres : circulaire, irrégulière, etc.), voir figure 10.

Une bande (strip) est composée de plusieurs rectangles (tiles) qui correspondent à des fenêtres séparées, voir figure 11.

BANDE 1	Rectangles 1 (tile 1)			
BANDE 2	Rect 1	Rect 2	Rect 3	
BANDE 3	Rect 1	Rect 2	Rect 3	Rect 4
BANDE 4	Rect 1		Rect 2	Rect 3
BANDE 5	Rectangle 1			

fig 10 : Approche < Mosaïque > du fenêtrage d'Intel.

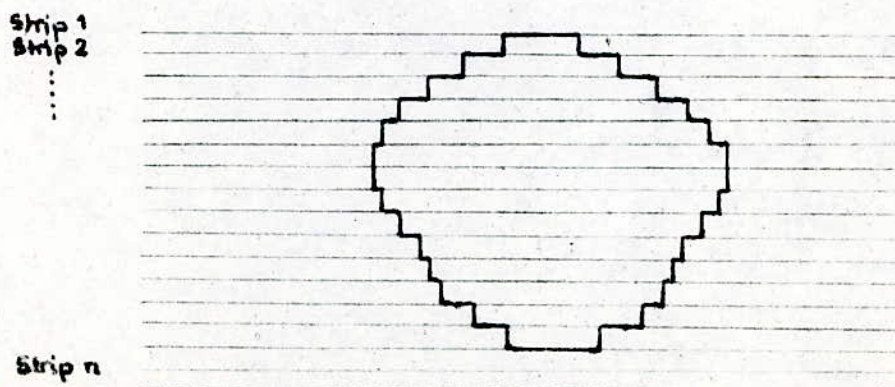


fig 11 : Fenêtre de forme quelconque

Le CPU envoie dans la mémoire graphique, l'information sur l'organisation de l'écran sous forme de descripteurs.

A partir des descripteurs, le processeur d'affichage accède aux données-pixels dans les différents bit-map et les prépare pour l'affichage en forme de fenêtres.

Durant la période d'effacement vertical, le processeur d'affichage accède à la première bande de la trame à partir de son adresse contenue dans un des registres de contrôle d'affichage (Display Control Register Block).

Le format des descripteurs de fenêtre est en figure 12.

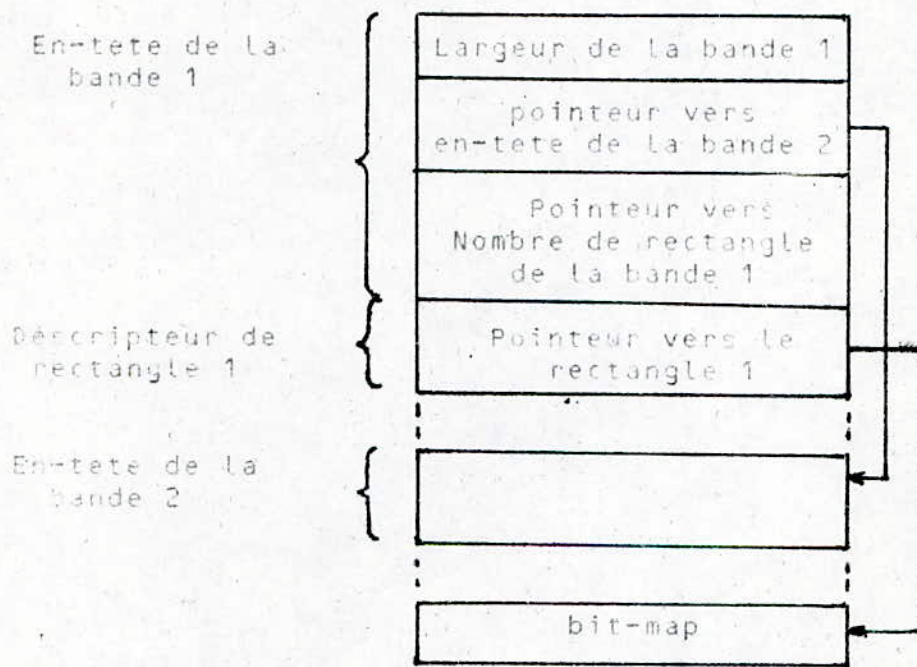


fig 12 : Descripteurs de fenetre

Ce descripteur contient l'en-tête de la bande numero 1 suivi des descripteurs de tous les rectangles de cette bande.

L'en-tête de la bande définit sa longueur, le pointeur vers le descripteur de la bande suivante et le nombre de rectangles dans la bande 1.

Tout descripteur de rectangle est réduit à un pointeur vers la zone de bit-map contenant effectivement l'information à afficher.

La modification de l'image est faite grâce au changement du descripteur de l'écran avec les valeurs actuelles, sans déplacer les blocs de pixels.

2.3 : L'UNITE D'INTERFACE DE BUS (BUI) AVEC LE CONTROLEUR DE DRAM

Le BUI controle tous les accès mémoires : accès CPU externe , accès processeur graphique , et accès processeur d'affichage.

Il controle aussi l'accès du CPU aux registres internes.

Le BUI comporte :

2.3.1: Internal Relocation Register

	15	1	0
Adresse de BASE + 0h :	----- Base Adress		MIO
	----- valeur par défaut (xxx0)		

L'adresse de base détermine la location des 128 octets du bloc des registres internes , elle doit etre écrite dans ce registre avant toute tentative d'accès aux autres registres .

Le bit MIO sélectionne pour les registres internes soit le mode mémoire soit le mode entrée/sortie .

MIO=0 mode entrée/sortie

MIO=1 mode mémoire

2.3.2: BUI control Register

	6	5	4	3	2	1	0
Adresse de BASE + 4h :	VR	WT	BCP	GI	DI	WP1	WP2
	----- valeur par défaut (010000)						

VR: Ce bit specifie la mémoire utilisee par le GCD

0-mémoire normale (DRAM)

1-mémoire à double port (VRAM)

WT: Ce bit détermine le nombre d'états d'attente

0-max 1(2) états d'attente

1-min 2(3) états d'attente

BCP: Ce bit determine la taille du bus externe.

0=bus de 8 bits

1= bus de 16 bits

GI: Ce bit est une copie de l'interruption genéré par le processeur graphique.

0= pas d'interruption

1= interruption présente

WP1: Ce bit est un bit de protection des registres BUI , sauf les bits WP1 et WP2.

WP2: Ce bit est un bit de protection des registres BUI et également des bits WP1 et WP2.

2.3.3: Refresh Control Register

Adresse de BASE + 6h :	5 4 3 2 1 0
	Refresh Scalar
	valeur par défaut (010010)

Le scalaire de rafraichissement de 6 bits détermine la fréquence du cycle de rafraichissement de la mémoire graphique.

2.3.4: Dram Control Register

Ce registre spécifie la configuration de la mémoire utilisée par le GDC.

Adresse de BASE + 8h :	6 5 4 3 2 1 0
	RW1 RWO DC1 DCO HT2 HT1 HTO
	valeur par défaut (1100101)

RW1 RWO (nombre de rangées configurée)

- 0 0 : 1 rangé
- 0 1 : 2 rangé
- 1 0 : 3 rangé
- 1 1 : 4 rangé

DC1 DCO (configuration de la DRAM)

- 0 0 : mode page , non entre-lacé
- 0 1 : mode page , entre-lacé
- 1 0 : mode page rapide , non entre-lacé
- 1 1 : mode page rapide , entre-lacé

HT2	HT1	HT0	(taille de la mémoire)
0	0	0	: 8K (1rangée, 8 colonnes)
0	0	1	: 16K (7rangées, 7 colonnes)
0	1	0	: 32K (7rangées, 8 colonnes)
0	1	1	: 64K (8rangées, 8 colonnes)
1	0	0	: 128K (8rangées, 9 colonnes)
1	0	1	: 256K (9rangées, 9 colonnes)
1	1	0	: 512K (9rangées, 10 colonnes)
1	1	1	: 1M (10rangées, 10 colonnes)

2.3.4: Registres de priorité :

1) Display Processor Priority

	6	5	4	3	2	1	0
Adresse de BASE + Ah :	-----						
			F P L			S P L	

	valeur par défaut (110011)						

2) Graphics Processor Priority

	6	5	4	3	2	1	0
Adresse de BASE + Bh :	-----						
			F P L			S P L	

	valeur par défaut (101010)						

3) External CPU Priority

	6	5	4	3	2	1	0
Adresse de BASE + Ch :	-----						
			F P L				

	valeur par défaut (111)						

Ces trois derniers registres séparés spécifie la priorité de la demande d'interruption pour chaque processeur.

FPL- First Priority Level

Autorise le processeur à prendre le bus le premier .

SPL- Subsequent Priority Level

Autorise le processeur à maintenir le bus le premier, lors d'un transfert de blocs .

3 : JEU D'INSTRUCTIONS

Chaque instruction du processeur graphique est constituée d'un opcode et d'une suite de paramètres.

Nous pouvons les classer en quatre catégories principales.

3.1 : COMMANDES NON GRAPHIQUES :

<u>COMMANDES</u>	<u>OPCODE</u>	<u>PARAMETRES</u>
NOP = No Operation	0300h	_____
LINK = Link to next command	0200h	Link Address Low Link Address High
INTR_GEN = Generate Interrupt	0E00h	_____
DUMP_REG = Dump Register	2900h	Dump Address Low Dump Address High
LOAD_REG = Load Register	3400h	Load Address Low Load Address High
ENTER-MACRO = Enter Macro	0F00h	Macro Address Low Macro Address High
EXIT-MACRO = Exit Macro	1700h	_____
HALT = Enter Pool State	xxx1h	_____

3.2 COMMANDES DE CONTROLE GRAPHIQUE :

<u>COMMANDES</u>	<u>OPCODE</u>	<u>PARAMETRES</u>
DEF-BIT-MAP = Define Bit-map	1A00h	Origin add Low Origin add High Xmax Ymax Bits/pixels
DEF-CLIP-RECT = Define Clip Rectangle	4600h	Xmin Ymin Xmax Ymax
DEF-COLORS = Define Colors	3D00h	Foreground Color Background Color
DEF-TEXTURE = Define Texture Opaque/Transparent	0600/ 0700h	Pattern
DEF-LOGICIAL-OP = Define Logic Operation	4100h	Color Bit Mask Function code
DEF-CHAR-SET = Define Characters Set (Word/Byte)	0A00/ 0B00h	Font Addr Low Font Addr High
DEF-CHAR-ORIENT = Define Char Orientation	4000h	Path /Rotation
DEF-CHAR-SPACE = Define Inter Char Space	4D00h	Inter Char Space
ABS-MOV = Move	4F00h	X Coordinate Y Coordinate
RELATIVE-MOV = Relative Move	5200h	dx dy
ENTER-PICK = Enter Pick Mode	4400h	_____
EXIT-PICK = Exit Pick Mode	4500h	_____

3.3 : COMMANDES GEOMETRIQUES:

<u>COMMANDES</u>	<u>OPCODE</u>	<u>PARAMETRES</u>
POINT = Draw Point	5300h	dx dy
INCR-POINT = Draw Incremental Points	8400h	Array Addr Low Array Addr High N (Nb of pts)
LINE = Draw Line	5400h	dx dy
CIRCLE = Draw Circle	8E00h	radius
RECT = Draw Rectangle	5800h	dx dy
POLYLINE = Draw Polyline	7400h	Array Addr Low Array Addr High N (Nb of pts)
POLYGON = Draw Polygon	7300h	Array Addr Low Array Addr High N (Nb of pts)
ARC = Draw Arc	6800h	dxmin dymin dxmax dymax radius
HORIZ-LINES = Draw Series of Horizontal Lines	BA00h	Array Addr Low Array Addr High N (Nb of pts)

3.4 : COMMANDES DE TRANSFERT :

<u>COMMANDES</u>	<u>OPCODE</u>	<u>PARAMETRES</u>
BIT-BLT = Bit Block Transfert within bit-map	6400h	Source x coord Source y coord dx dy
BIT-BIT-M = Bit Block Transfert across bit-map	AE00h	Source Addr Low Source addr high Source Xmax Source Ymax Source x coord Source y coord dx dy

troisieme partie

CONCEPTION DU MODULE GRAPHIQUE

- 1 - Introduction
- 2 - Etude de la carte
- 3 - Adaptabilité de la carte

1 : INTRODUCTION

La carte graphique comporte essentiellement trois parties (voir fig 13) :

- Le coprocesseur graphique (GDC) dont la description est donnée ailleurs.
- La mémoire d'écran
- Les circuits d'interface avec le CPU externe

Elle est commandée est adressée par un micro-contrôleur MCS8031 d'intel .

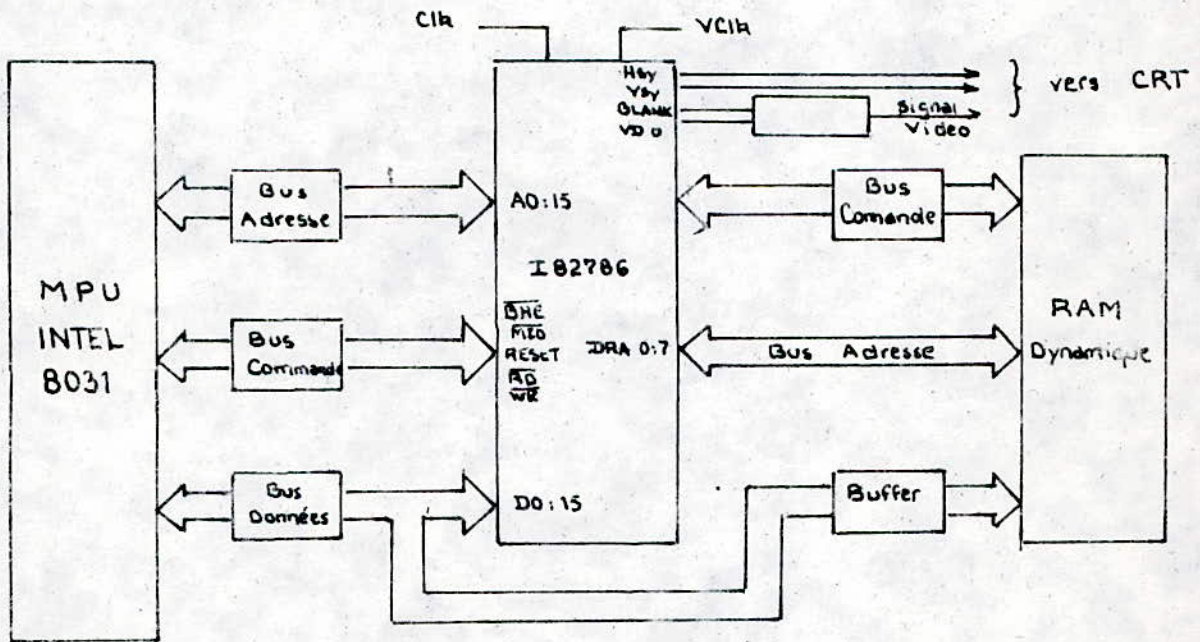


Fig 13 : Schéma synoptique de la carte graphique

2: ETUDE DE LA CARTE

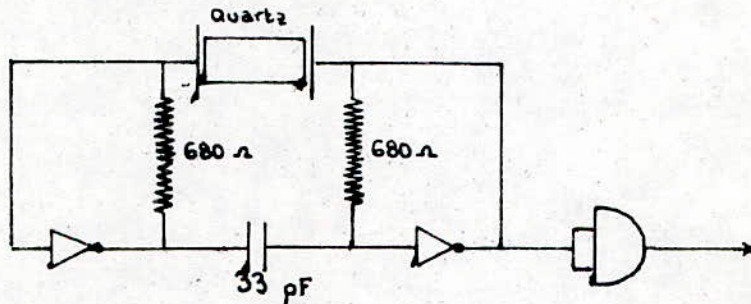
2.1: HORLOGES DU GDC:

Les horloges externes nécessaires au pilotage du GDC sont générées d'une manière classique à l'aide de deux oscillateurs à quartz de respectivement 11 MHz et 20 MHz pour l'horloge système clk et l'horloge vidéo vclk.

2.1.1: Horloge système Clk:

L'horloge système synchronise la communication du GDC au CPU externe, sa fréquence étant assurée par un quartz de 11 MHz connecté comme le montre le schéma ci-dessus.

La porte logique "Nand" placée en fin du circuit constitue la mise en forme du signal d'horloge.



2.1.2: Horloge vidéo Vclk:

L'horloge vidéo synchronise les registres de sérialisation vers le CRT.

Sa fréquence étant assurée par un quartz de 20 MHz connectée dans un circuit identique au précédent.

Notons que la fréquence maximale de l'horloge vidéo acceptée par le GDC est de 25 MHz.

2.2: BUS D'ADRESSE

Le bus d'adresse du GDC est constitué de 22 bits dont seulement 16 sont utilisés. Les 6 autres bits sont forcés à 0.

Les 16 bits de poids faible sont reliés au bus d'adresse du CPU par l'intermédiaire d'un latch 16 bits.

2.2.1: Décodage de la carte graphique:

Le circuit de décodage d'adresse de la carte graphique utilise l'espace adressable entre 34 et 64 K.

La carte est alors vue par l'U.C comme étant adressable à partir de l'adresse de base (8800H) à l'adresse (FFFFH).

2.2.2: Réalisation du circuit d'adressage:

Le bus d'adresse est verrouillé par un latch validé par un signal ALE (Addr Latch Enable) généré par la carte unité centrale, et le signal de décodage d'adresse (fig 14)

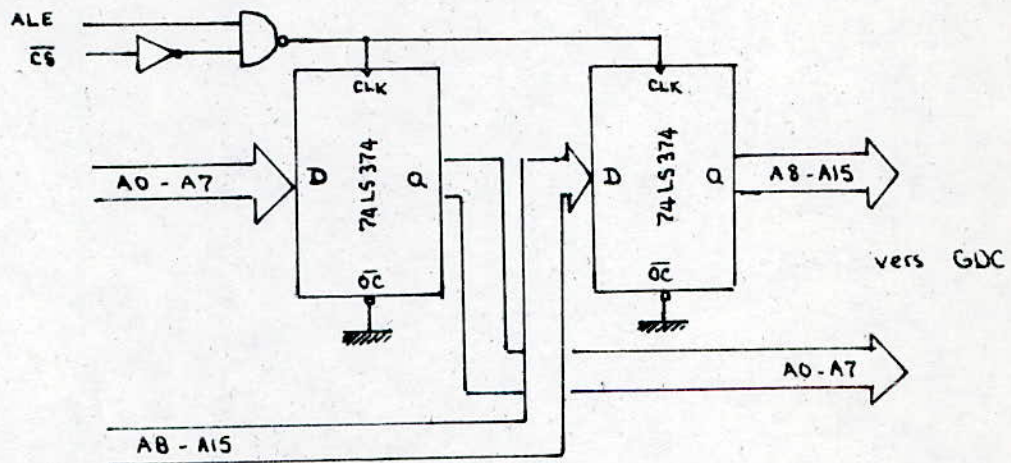


fig 14 : Circuit d'adressage

2.3: SIGNAUX DE CONTROLES

2.3.1: Le signal RESET:

Le signal RESET (Broche en entrée du GDC) actif au niveau haut, stoppe toute activité du processeur graphique , il doit être synchronisé sur le front descendant de l'horloge et avoir une durée minimale de dix cycles d'horloge.

Pour cela nous le générons par le circuit suivant :

La broche 3 du port 1 du CPU 8051 permettra de simuler un signal par software synchronisé par un flip flop avec le front descendant de l'horloge (fig 15).

La durée minimale du signal RESET est alors assurée par un simple programme .

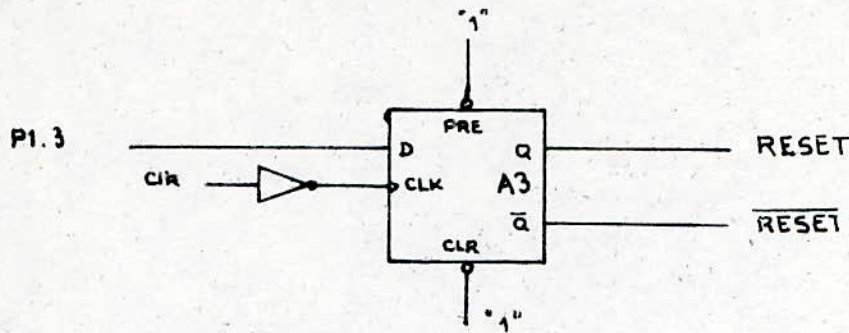


fig 15 : Génération du signal RESET

2.3.2: Le signal BHE:

Le signal BHE nous permet, durant le reset de choisir le type d'interface (synchrone ou asynchrone), et en fonctionnement normal de choisir le poids fort ou faible du mot adressé.

Pour cela nous le générons par le circuit suivant:

La broche 4 du port 1 provenant du CPU est mis à l'état bas pendant que le signal reset est actif pour le choix d'un interface asynchrone. Ce signal est remis à l'état haut pendant les opérations normales et BHE reçoit alors le bit d'adresse A0 qui différencie entre le poids fort et le poids faible du mot adressé (fig 16).

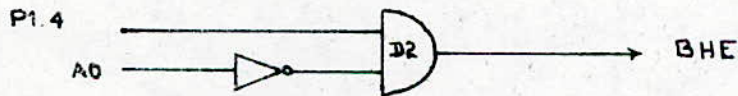


fig 16 : Génération du signal BHE

2.3.3: Le signal MIO:

Le signal MIO permet de choisir entre les deux modes d'adressage possibles, le mode mémoire ou le mode entrée/sortie.

Un niveau haut sur ce signal sélectionnera le mode mémoire, un niveau bas sélectionnera le mode entrée/sortie.

Ce signal est activée directement par programme.

2.3.4: Le signal SEN:

Le signal SEN, lorsqu'il est mis à l'état haut, indique au processeur hôte que le GDC a pris en compte la demande d'accès aux registres ou à la mémoire et que le cycle peut prendre fin.

- En mode écriture l'état haut de SEN indique que le GDC a pris en compte les données
- En mode lecture il indique que les données sont prêtes sur le bus.

2.4: ANALYSE DES ECHANGES CPU - GDC

2.4.1: Analyse des opérations d'écriture/lecture du GDC

2.4.1.1: Signaux de contrôle:

Les signaux de contrôle sur l'interface GDC sont illustrés par les chronogrammes de la figure 15.a

Le GDC ainsi sollicité se met alors en mode esclave (slave mode) et activera, avec un temps de retard minimal TS24, le signal SEN aux registres internes ou à la mémoire graphique.

Le temps de retard TS24 est tel que:

$$4T_c + 20 < TS24 < - \quad \text{où } T_c = \text{cycle d'horloge}$$

La durée active du signal (RD/WR) TS14 est telle que:

$$2T_c + 30 < TS14 < -$$

Le temps T16 de maintien du bus d'adresse et des signaux CS, BHE, MIO est tel que:

$$2T_c + 30 < TS16 < -$$

Pour une horloge de 11 MHz, le cycle d'horloge T_c est égal à 90,9 ns, on trouve alors:

$$TS16 > 2(90) + 30 = 211,8 \text{ ns}$$

$$TS24 > 4(90,9) + 20 = 383,6 \text{ ns}$$

$$TS14 > 2(90,9) + 30 = 211,8 \text{ ns}$$

2.4.1.2: bus de données:

Le transfert des données entre le GDC et le CPU s'établit suivant le chronogramme de la figure 15.b

En effet, dès l'apparition du signal SEN, durant le cycle d'écriture, le GDC est alors disponible à recevoir les données.

La durée active du signal SEN est de TS25 avec:

$$4 T_c - 25 < TS25 < 4T_c + 35 \quad \text{où } T_c = \text{cycle d'horloge}$$

Le temps de maintien de données est donné par TS20 avec:

$$4 T_c < TS20 < - \quad ; \quad \text{pour une écriture interne}$$

$$3 T_c + T_{dh} + 30 < TS20 < - \quad ; \quad \text{pour une écriture mémoire}$$

ou T_{dh} représente le temps de maintien de données lors d'une écriture en mémoire, il dépend des mémoires utilisées. Pour les mémoires TMS, T_{dh} est inférieur à 30 ns.

De ce fait, il faut assurer d'une part, que le temps de maintien de données est tel que:

$$TS20 > \max ((4T_c) , (3T_c + T_{dh} + 30))$$

d'autre part, que les données restent sur le bus après la disparition (retour à l'état bas) du signal SEN, c'est à dire:

$$TS20 > \max(TS25) \quad \text{soit} \quad TS20 > 4T_c + 35$$

$$\text{en définitive : } TS20 > \max ((4T_c), (4T_c + 35), (3T_c + T_{dh} + 30))$$

Pour une horloge de 11 MHz, le cycle d'horloge T_c est égal à 90,9 ns et en prenant $T_{dh} = 30$ ns, on obtient alors pour TS20:

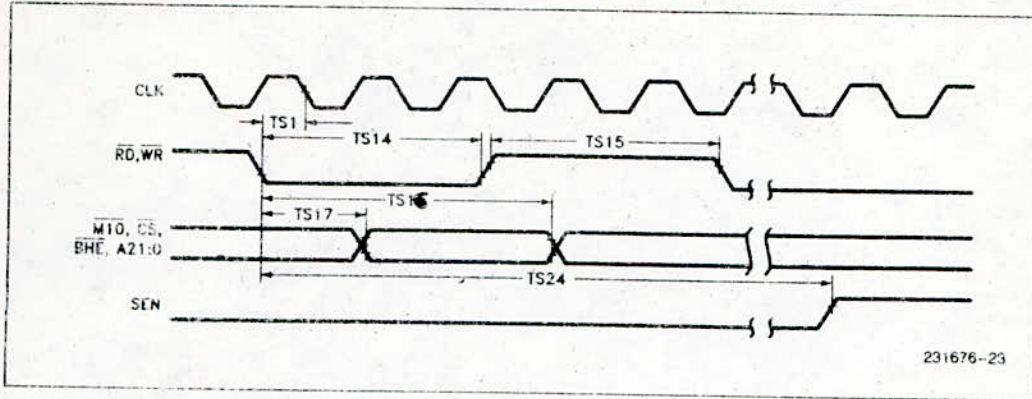
$$TS20 > \max ((4 \times 90,9 + 35) , (3 \times 90,9 + 60))$$

$$TS20 > \max (398,6 \text{ ns} , 332,7 \text{ ns})$$

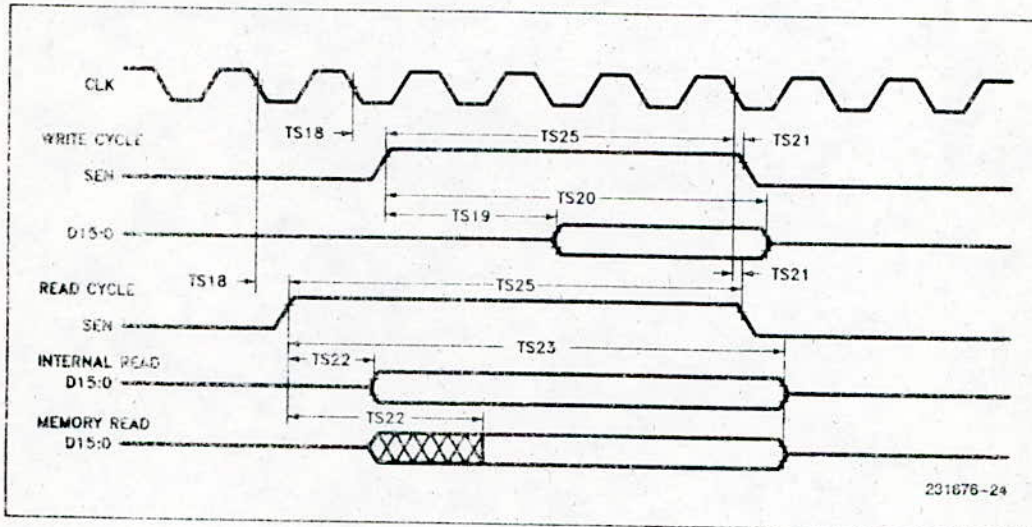
$$\text{soit: } TS20 > 398,6 \text{ ns}$$

Cette condition est réalisée en retardant le signal SEN d'une horloge (90 ns) et en utilisant ce signal pour ouvrir les buffers D1 et D2 de données en écriture. Ce retard est suffisant pour assurer la bonne prise en compte des données par le GDC.

ASYNCHRONOUS SLAVE INTERFACE



SEN/DATA—SLAVE INTERFACE



Pendant le cycle de lecture, les données issues du GDC apparaissent sur le bus avec un retard TS22 par rapport au front montant de SEN.

Ce délai est tel que: $TS22 < Tc + 40$

pour $Tc = 90,9 \text{ ns}$: $TS22 < 130,9 \text{ ns}$

Cette condition est prise en compte en retardant le signal SEN de 3 périodes d'horloge (270 ns) et en utilisant ce signal comme horloge, le front montant servant alors à mémoriser les données issue du GDC, dans deux latches E1, F1.

2.4.2: Analyse des opérations d'écriture/lecture du CPU

2.4.2.1: Cycle de lecture

La fig 16 montre comment le 8031 effectue un cycle de lecture. Le cycle de lecture commence sur le front montant du signal ALE. Le 8031 place une adresse sur les lignes d'adresse, et met à niveau bas le signal RD, pour indiquer qu'il s'agit d'une lecture. Les données issues de la mémoire externe apparaissent sur le bus avec un retard TRd.

Ce délai est tel que : $- < TRd < (5Tc - 170) \text{ ns}$

Le temps de maintien des données sur le bus après la disparition du signal RD est TRh tel que : $0 < TRh < -$

La durée active du signal RD est tel que : $6Tc - 100 < TRD < -$

Pour une horloge de 11 MHz ; $Tc = 90,9 \text{ ns}$, ce qui donne :

$- < TRd < 284,5 \text{ ns}$

$0 < TRh < - \text{ ns}$

$445,4 < TRD < - \text{ ns}$

2.4.2.2: Cycle d'écriture

La fig 17 montre comment le 8031 effectue un cycle d'écriture. Le cycle d'écriture commence aussi sur le front montant du signal ALE.

Le 8031, après avoir placée l'adresse sur le bus d'adresse, active au niveau bas le signal WR, indiquant le cycle d'opération d'écriture.

Le temps de stabilisation des données pour une écriture T_{Ws} est tel que :

$$7T_c - 180 < T_{Ws} < - \text{ ns}$$

La durée active du signal d'écriture WR est tel que :

$$6T_c - 100 < T_{WR} < - \text{ ns}$$

Le temps de maintien de données sur le bus T_{Wh} après la disparition du signal WR est tel que :

$$2T_c - 90 < T_{Wh} < - \text{ ns}$$

Pour une horloge de 11 MHz ; $T_c = 90,9 \text{ ns}$, ce qui donne :

$$456,3 < T_{Ws} < - \text{ ns}$$

$$91,8 < T_{Wh} < - \text{ ns}$$

$$445,4 < T_{WR} < - \text{ ns}$$

2.4.3: CONCLUSION

L'interface CPU - GDC étant entièrement asynchrone , et le CPU n'ayant pas le signal de synchronisation des accès .

On est alors amené à commander par programme l'organisation des accès au GDC, c'est à dire réaliser la synchronisation par logiciel.

Ceci revient tout simplement à générer des signaux de lecture et d'écriture compatible d'un coté avec le CPU et d'autre coté avec le GDC.

Ces conditions sont réalisés en mémorisant les données et les adresses dans des bascules attribués.

De ce fait le signal \overline{CS} du GDC, reçoit le signal issu de la combinaison des signaux WR_{status} , RD_{status} et du chip-select généré par le CPU, à travers des deux portes logiques E3 .

Le signal MIO serait généré par soft via la broche 2 du port 1 du CPU (P1.2), le signal BHE reproduit le bit d'adresse $A0$ à travers la porte D2, les adresses seront mémorisées dans les latches E2, F2 et les données, dans les latches D1 et C1 pour l'écriture , E1 et F1 pour la lecture .

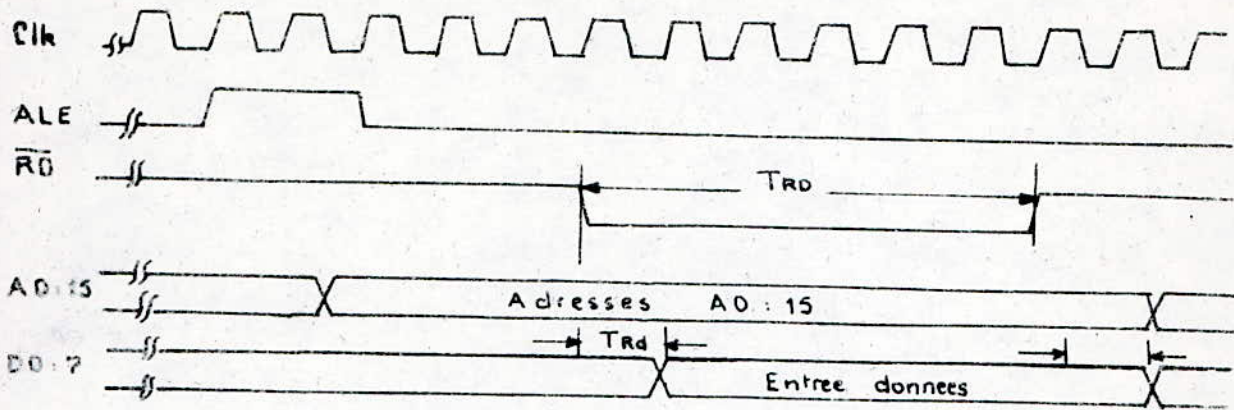


Fig 16 : cycle de lecture

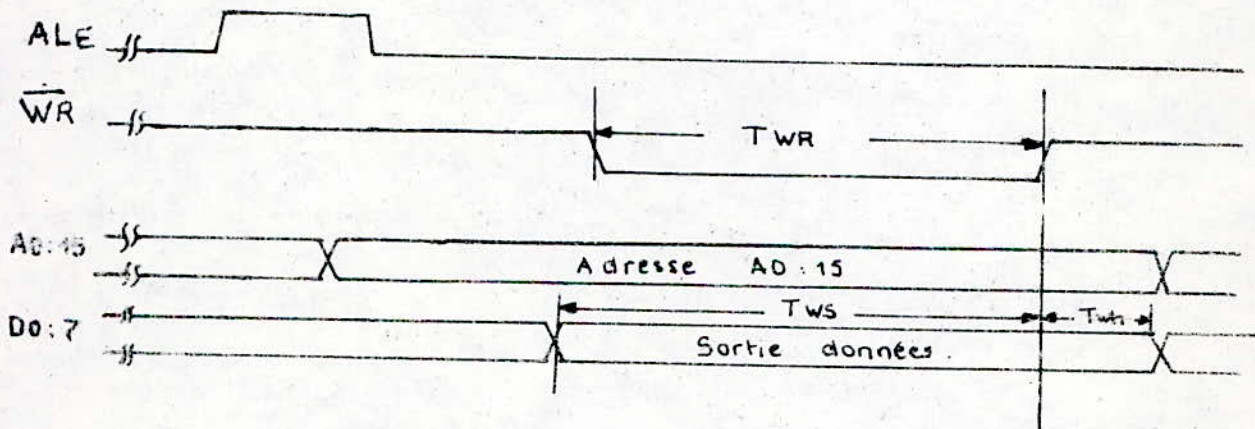


Fig 17 : Cycle d'écriture.

2.5: CONCEPTION ET REALISATION DES CIRCUITS ECRITURE/LECTURE :

2.5.1: Circuit d'écriture

Le bus de données entre GDC (16 bits) et le CPU (8 bits) n'étant pas compatible, l'écriture doit se faire en 2 cycles de bus successifs, en commençant obligatoirement par le poids faible. Le GDC mémorise le poids faible et attend que le poids fort soit envoyé avant d'écrire effectivement dans le registre adressé.

Ceci nous conduit à la réalisation du circuit d'écriture donné en figure 18

Les données transitent par un buffer bidirectionnel dont le sens de transfert est commandée par le signal WR et validée par le signal WR.

Les deux latches D1 et C1 mémorisent le poids faible et fort du mot provenant du CPU. La validation de l'un ou de l'autre des latch se fera par la combinaison de WR et de A.

Le signal WRstatus (sortie de la bascule A3) est simulé afin d'indiquer un fin de cycle (fig 18)

2.5.2: Ecriture vue du coté CPU :

Le CPU envoie un ordre d'écriture sur l'interface avec une adresse et des données, le signal WRstatus passe alors à l'état bas et il est renvoyé vers le GDC qui reste en attente jusqu'au passage de WRstatus à l'état haut.

Le CPU peut éventuellement envoyer un autre ordre d'écriture et ainsi de suite (voir chronogramme fig 20)

2.5.3: Circuit de lecture :

De la même manière, la lecture se fera sur trois cycles de bus successifs (dans le cas d'un mot de 16 bits); et en deux cycles (dans le cas d'une lecture d'un octet).

Le premier cycle sert à déclencher un cycle de lecture; lecture poids faible ou poids fort (dans le cas ou l'un des buffers mémorise déjà des données par défaut).

Ceci nous conduit à la réalisation du circuit de lecture de la fig 21.

Les deux latches sont validée altrenativement par la combinaison de RD et A.

Le signal RDstatus a été simulé afin de nous indiquer la fin du cycle.

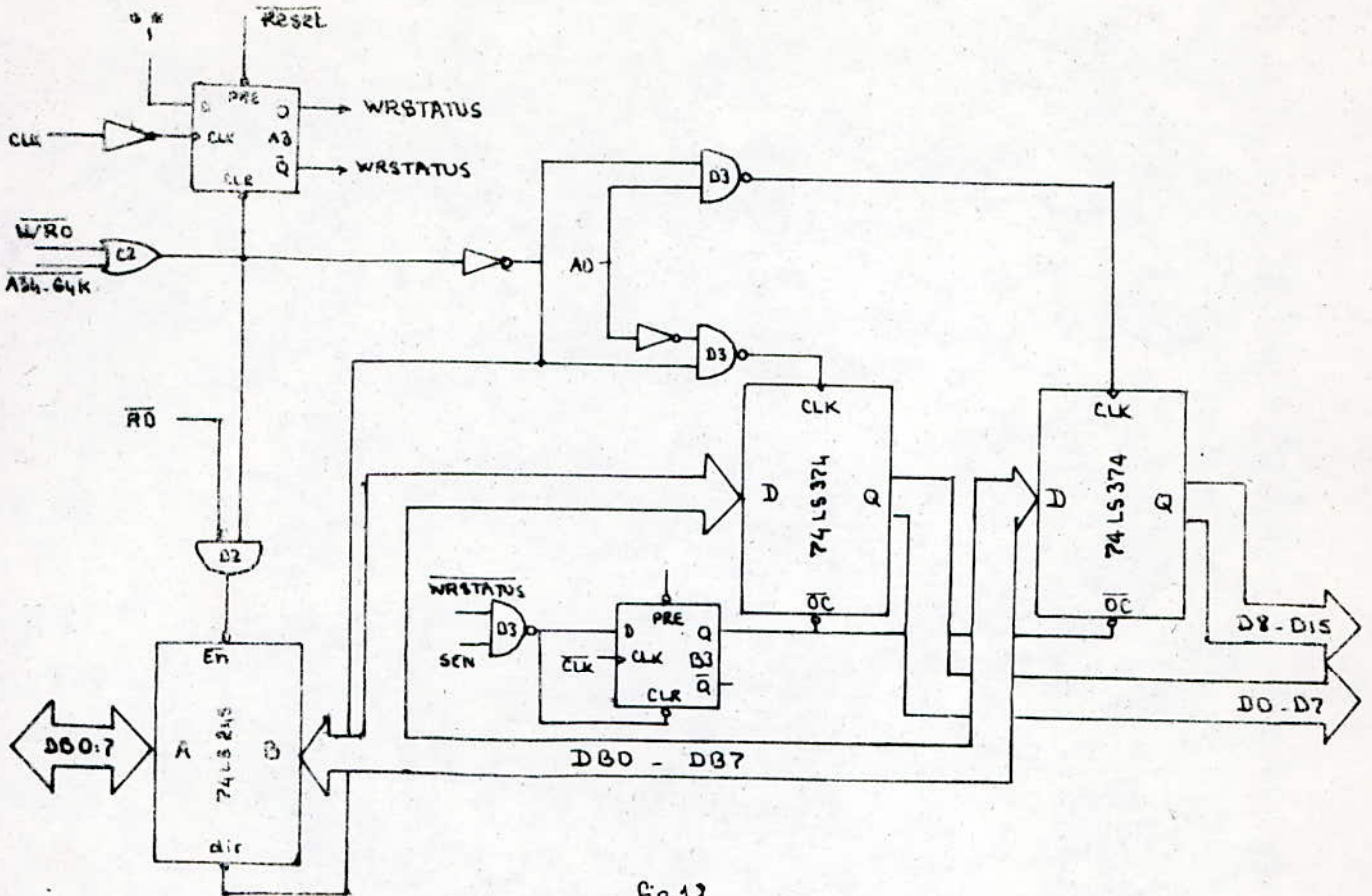


Fig 17

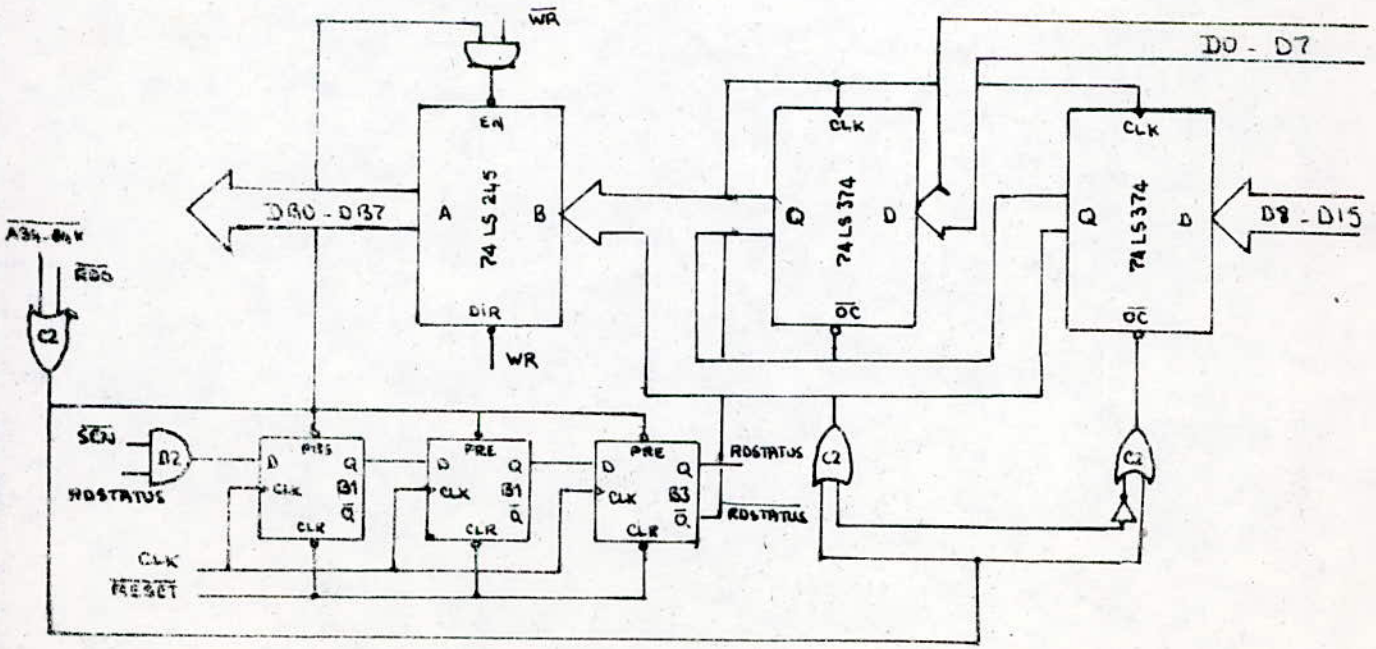


Fig 21

2.5.4: Lecture vue du côté CPU

Le CPU envoie un ordre de lecture sur l'interface avec une adresse, le signal RDstatus passe à l'état haut et son complémentaire RDstatus est alors renvoyé vers le GDC qui reste en attente jusqu'au passage de RDstatus à l'état bas.

Le CPU peut éventuellement envoyer un autre ordre de lecture et ainsi de suite (voir chronogramme fig 23).

2.6: MEMOIRES

La configuration de la mémoire choisie est organisée en 8 rangées de 8 colonnes.

L'interfaçage du GDC et la mémoire sera assuré par le bus d'adresse multiplexée (DRAO-7) de 8 lignes permettant d'adresser 64K mots de 16 bits. Cette mémoire est composée de 16 boîtiers de 64Kx1 bits chacun.

Ces mémoires sont dynamiques, elles présentent les avantages d'un faible encombrement et d'un faible coût dû à leur taux d'intégration élevé.

2.6.1: Mémoire d'écran

A chaque bit de la mémoire d'écran correspond un point sur l'écran pour un moniteur monochrome.

Vu la technologie actuelle des tubes cathodiques, les hautes résolutions étant relativement trop coûteuses et dans la plupart des applications, l'espace d'affichage de 800x380 points s'avère suffisant, ce qui correspond à une taille mémoire de 19K mots de 16 bits, le reste de la mémoire servant à la mémoire système.

2.6.2: Rafraichissement de la DRAM:

Le cycle de rafraichissement des RAM dynamiques est assurée par le processeur graphique, sa fréquence étant programmable dans le registre interne Refresh - control.

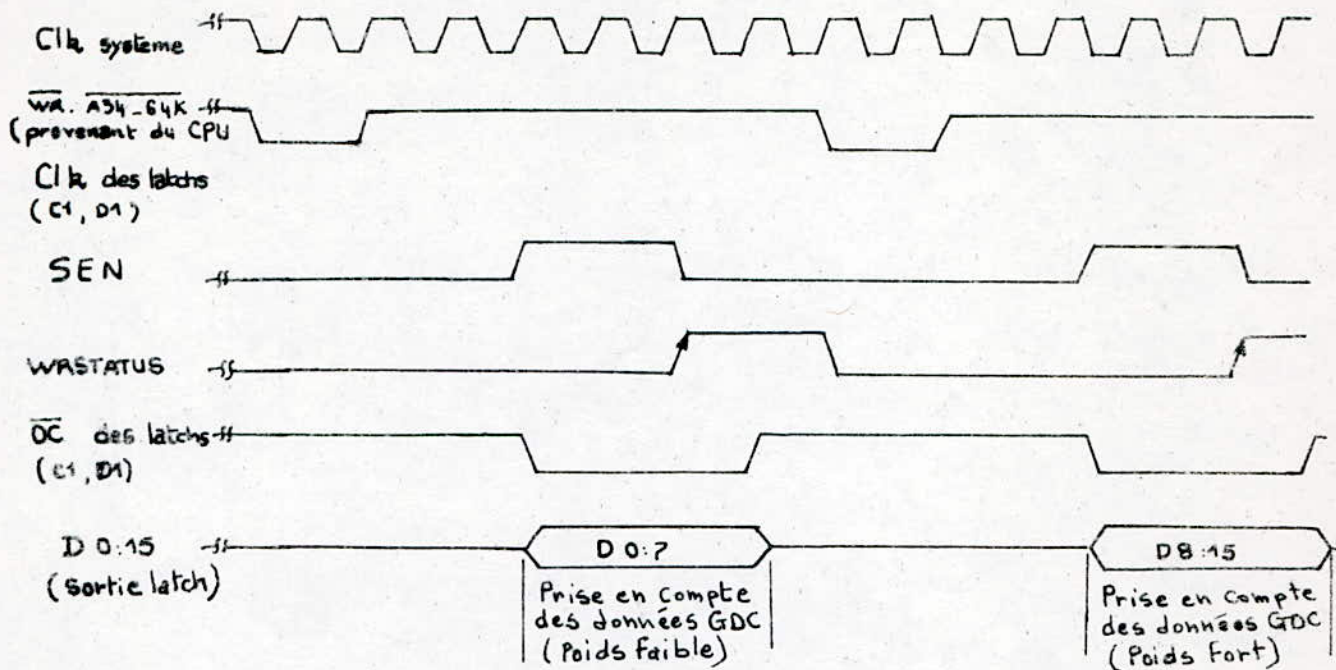


Fig 19 : Chronogrammes du cycle d'écriture.

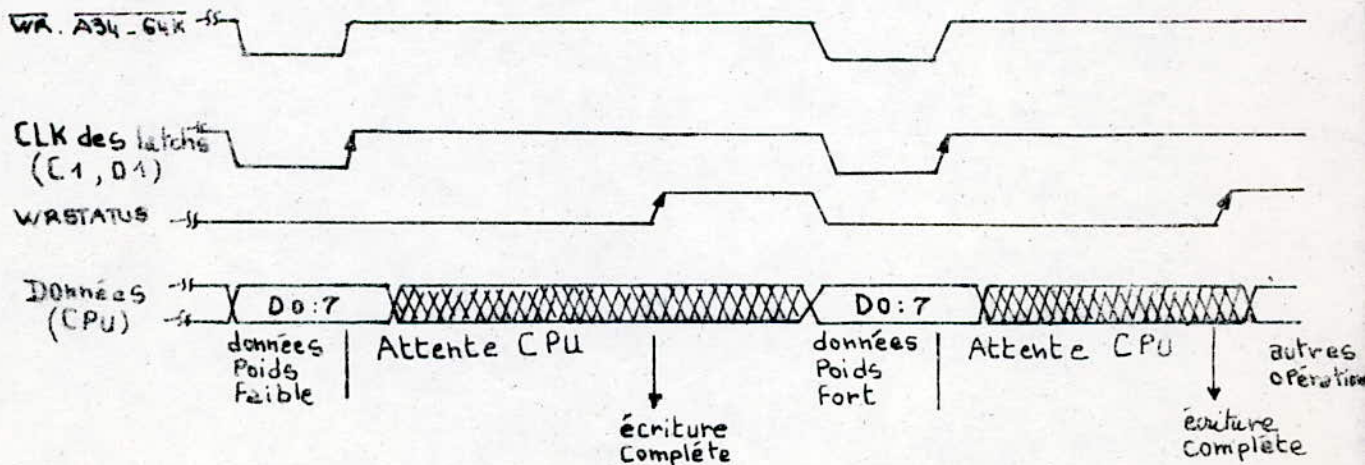


Fig 20 : Écriture vue du côté CPU.

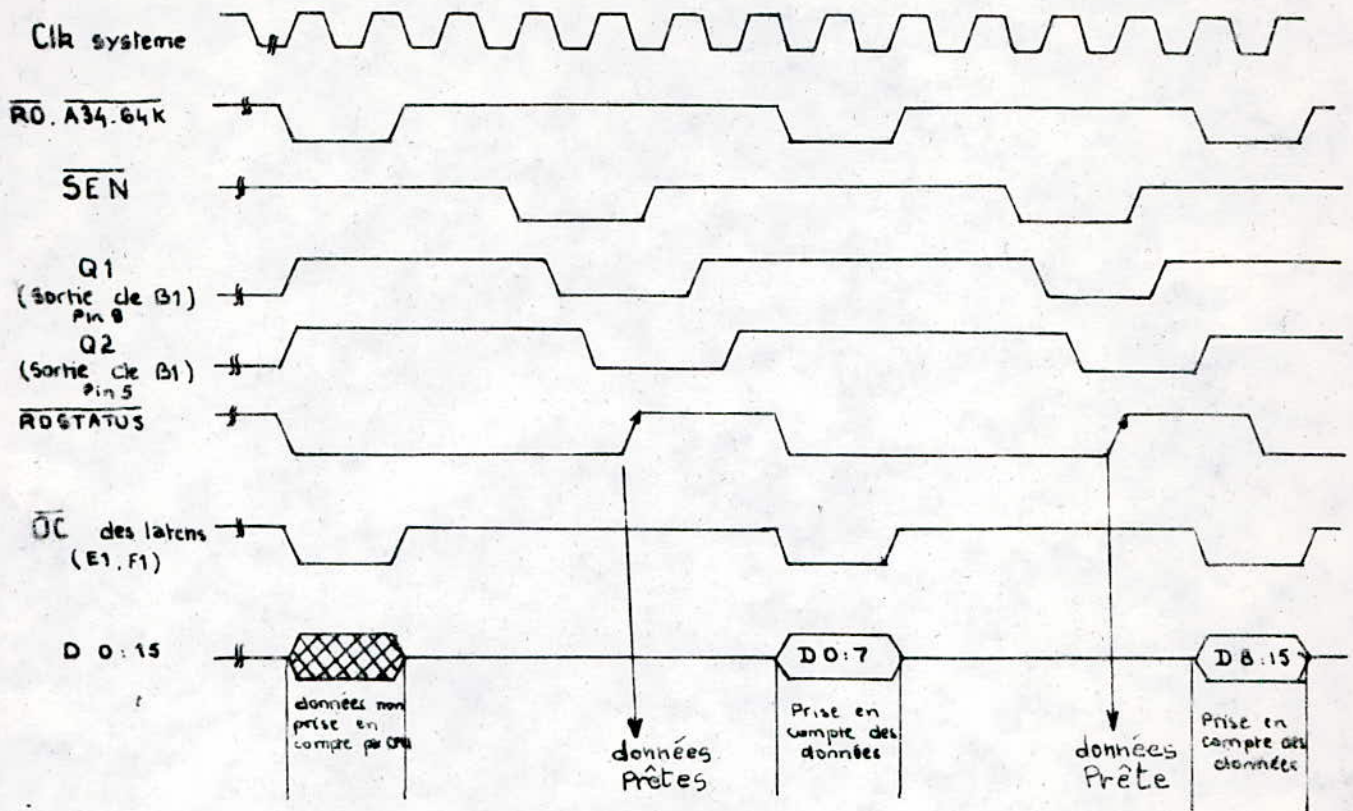


Fig 22 : Chronogrammes du cycle de lecture.

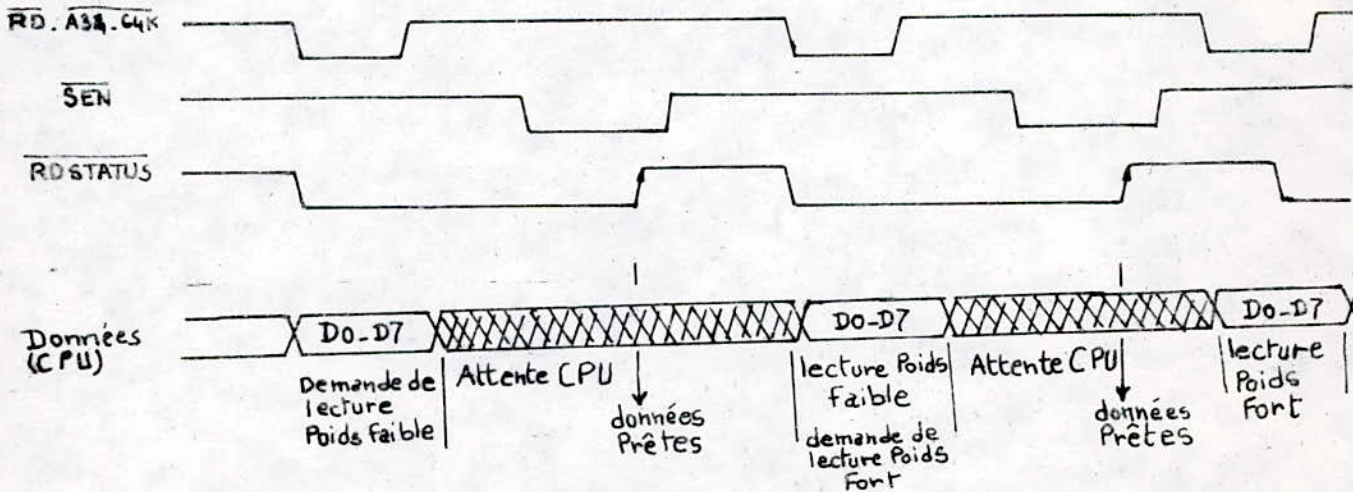


Fig 23 : Lecture vue du côté CPU.

3 - ADAPTABILITE DE LA CARTE :

La carte graphique " vierge " (sans la programmation) ainsi conçue, pourra s'adapter à n'importe quel micro-processeur 8 bits moyennant des modifications mineures dans les circuits d'adressage .

D'une manière plus générale, l'implantation de cette carte dans un système utilisant un bus de donnée nécessitera une adaptation qui se réduit à l'implantation d'un interface de bus soit :

- Des buffers pour les signaux de contrôle.
- Un circuit de décodage d'adresse.
- Un circuit de génération des ordres RD et WR pour les bus ne comportant qu'un signal unique R/W.

Quatrieme partie :

REALISATION ET
MISE AU POINT DE LA CARTE

- 1 - Systemes de développement
- 2 - Initialisation de la carte
- 3 - Le moniteur graphique et sa structure

1: Systeme de développement

Pour développer une application, on fait appel à des programmes spécifiques tels que l'éditeur, l'assembleur, l'émulateur, etc.

Une application donnée exige donc la mise au point, à la fois de son matériel et de son logiciel.

Le système de développement de la fig(25) a été utilisé pour la mise au point de la carte, il comprend:

- Un micro-ordinateur PDP11.03 avec un package cross-assembleur pour le microprocesseur 8031 d'intel.

- Une carte émulateur 8031, reliée au PDP11 par un canal serial RS232C.

1.1: Le PDP11 :

Le package cross assembleur du 8031 sur PDP11 comporte :

- Un cross-assembleur, un éditeur de liens, un gestionnaire de bibliothèques.

- Un convertisseur du format objet généré par le cross-assembleur en format intel (OBJCNV).

- Un module chargeur (LOAD) conçu et réalisé par l'équipe du laboratoire péri-informatique, permettant de charger le programme objet dans la mémoire, de la carte émulateur, en vue de son exécution.

- Un éditeur qui sert à éditer des programmes d'application des textes ou toutes séquences étudiées par l'utilisateur.

1.2: L'émulateur :

L'émulateur à base du microprocesseur 8031 d'intel a aussi été réalisé par l'équipe du laboratoire.

C'est un module qui s'ajoute au système de développement, il dispose d'un moniteur assez développé et d'une mémoire utilisateur de 64 Koctets.

Cependant, pour mettre au point un programme, on le chargera en mémoire et on le lancera. On pourra alors afficher les contenus des registres, les positions mémoires, l'état des circuits d'E/S ; on pourra introduire des points d'arrêts dans le programme afin d'examiner ce qui se passe à ce moment dans la machine ; ou encore on lancera l'exécution du programme en pas-à-pas, afin d'observer, après chaque instruction ou groupe d'instructions spécifiées, ce qui se passe, on modifiera certaines données ; etc.

L'émulateur est relié au PDP11 par un canal serial RS232C. Il permet le chargement d'un programme exécutable à partir d'une disquette qui se trouve dans le PDP11.

4.2 : INTRODUCTION :

Les coprocesseurs graphiques déchargent l'unité centrale des tâches de gestion de l'écran et de la mémoire d'image.

Ils ont aussi été conçus pour simplifier le travail des concepteurs, le libérant ainsi des tâches de traitement élémentaire et de l'écriture des algorithmes de génération de figures types.

Grâce à l'intégration à grande échelle, ces algorithmes sont câblés à l'intérieur du circuit et la génération d'un certain nombre de ces figures devient ainsi automatique.

Le travail du concepteur se réduira alors à la programmation des registres internes de ces coprocesseurs graphiques, pour l'écriture des primitives graphiques nécessaires aux logiciels graphiques.

4.3 : INITIALISATION DE LA CARTE :

Cette phase est effectuée uniquement une fois, à la mise sous tension de la carte graphique.

4.3.1 : REGISTRES DE CONTROLES :

1) Internal Relocation : (0260h)

Nous nous fixons de loger en mémoire, les registres internes du GDC, ainsi que la mémoire de travail à partir de l'adresse 9800 Hexa.

Afin de déterminer, le paramètre à programmer dans ce registre, pour avoir cette adresse comme BASE, il suffit de décaler de 6 bits vers la gauche le nombre binaire 9800H nous obtiendrons 0260H. Tous les registres seront adressable à l'adresse 9800H + Offset.

2) BIU Control: (0020h)

3) Refresh Control : (0009h)

Le paramètre de rafraichissement des RAM Dynamiques est calculé selon la relation suivante :

$$\text{Refresh Control} = E \left[\frac{2 \cdot T_{\text{ref}} \cdot F_{\text{clk}}}{16 \cdot \text{Refresh-row}} \right] - 1 = 9$$

Avec : Tref : temps d'accès à la DRAM : 2 ms
Fclk : fréquence d'horloge : 20MHz
Refresh-row : nombre de rangée de la DRAM : 256

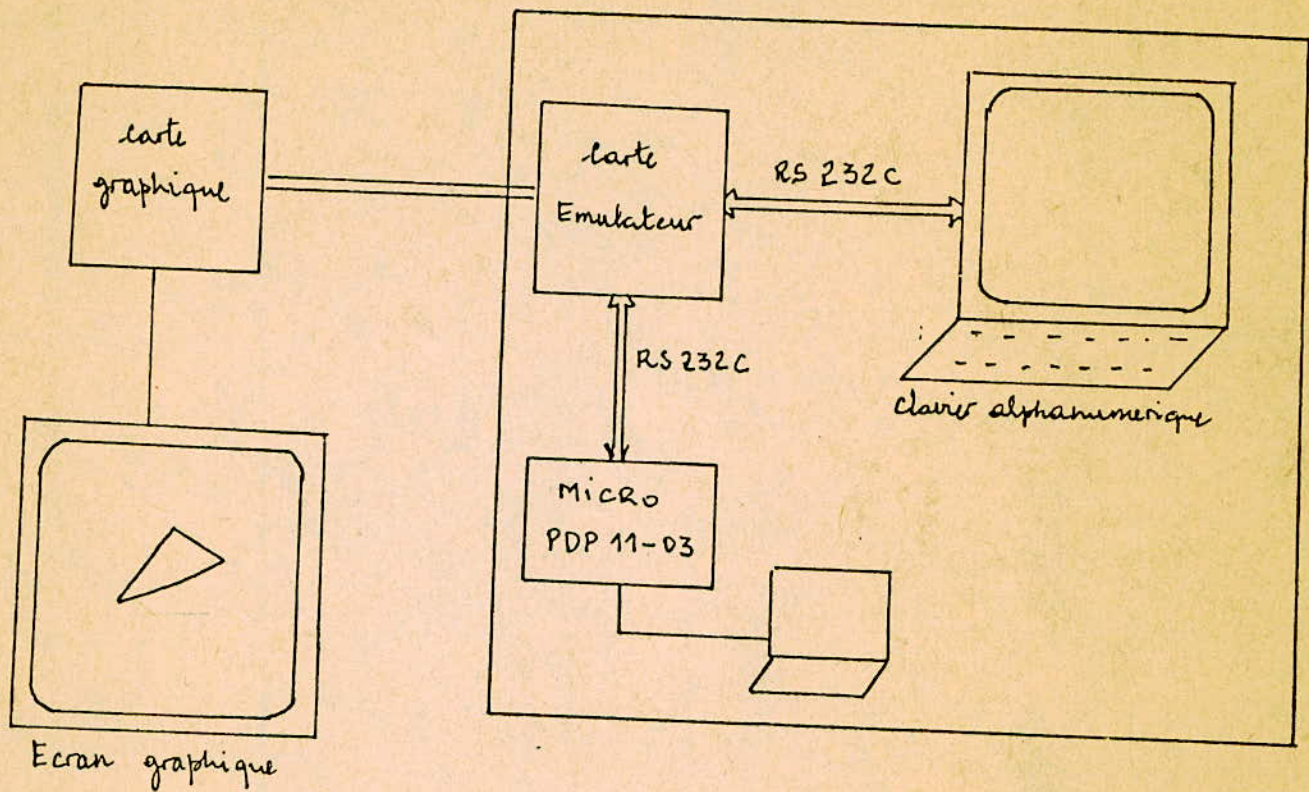


fig 25

SYSTEME DE DEVELOPPEMENT

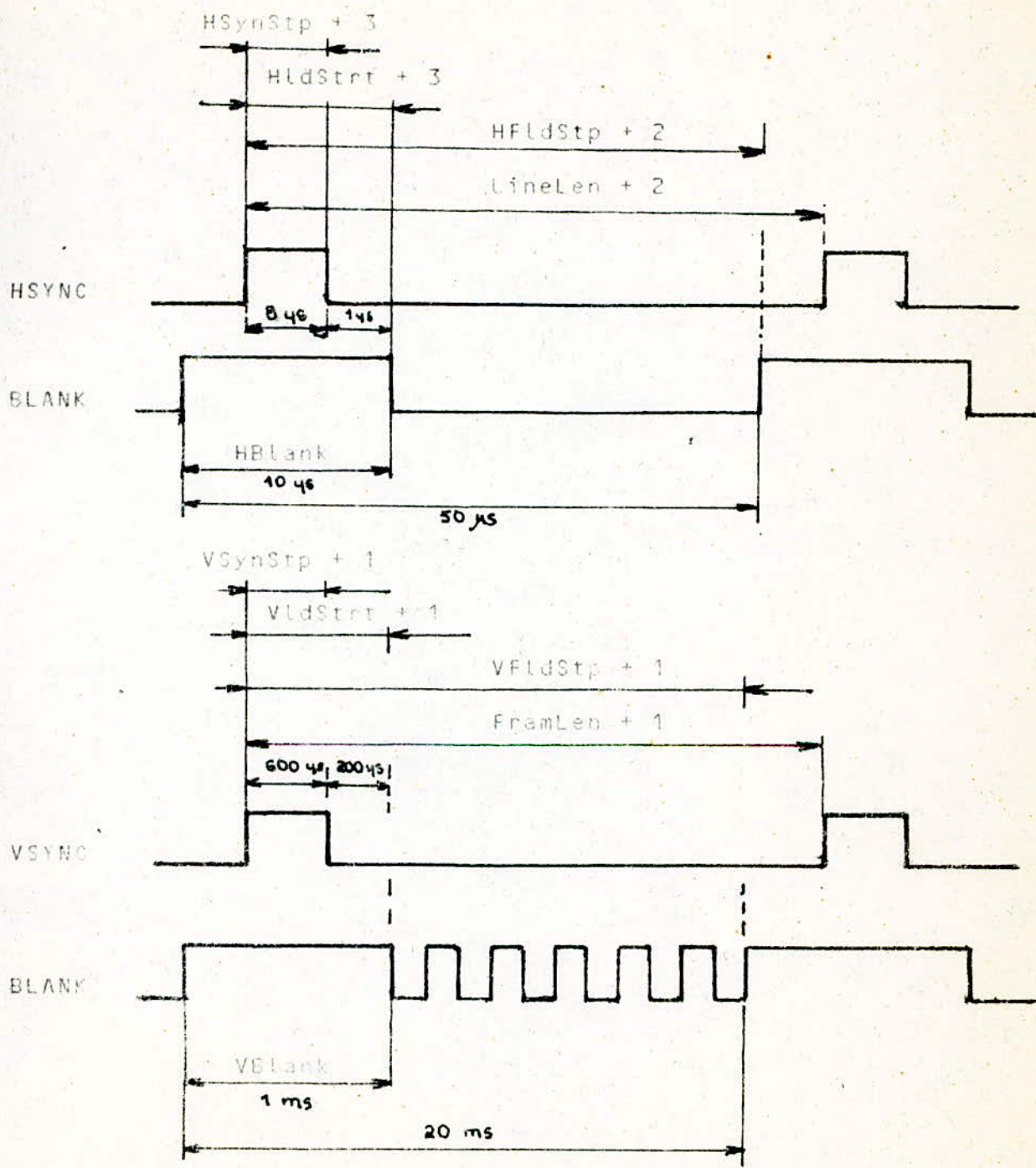


fig 24

Tous les paramètres doivent être exprimés en nombre de pixels (sachant qu'à 20 MHz, celui-ci représente 50 ns).

Le temps de retour horizontale :

$$Hsynclks = E (Vclk \times THsyn) = E (20 \cdot 10^6 \times 8 \cdot 10^{-6}) = 160$$

$$Hbackclks = E (Vclk \times THback) = E (20 \cdot 10^6 \times 10^{-6}) = 20$$

$$Hblankclks = E (Vclk \times THblank) = E (20 \cdot 10^6 \times 10^{-5}) = 200$$

d'où :

$$HsynStp = Hsynclks - 3 = 160 - 3 = 157 \quad \text{soit } 9D \text{ H}$$

$$HFldStrt = HsynStp + Hbackclks = 157 + 20 = 177 \quad \text{soit } B1 \text{ H}$$

$$HFldStp = HFldStrt + Hres = 157 + 800 = 957 \quad \text{soit } 03D1 \text{ H}$$

$$LineLen = Hblankclks + Hres - 3 = 200 + 800 - 3 = 997 \quad \text{soit } 03E5 \text{ H}$$

Et de même manière, nous obtiendrons les paramètres de synchronisation verticale :

$$Vsynlines = E (Fclk \times TVsyn) = E (20 \cdot 10^3 \times 6 \cdot 10^{-4}) = 12$$

$$Vbacklines = E (Fclk \times TVback) = E (20 \cdot 10^3 \times 2 \cdot 10^{-4}) = 4$$

$$VFieldlines = E (Fclk / FVsyn) = E (20 \cdot 10^3 / 50) = 400$$

$$VsynStp = Vsynclks - 1 = 12 - 1 = 11 \quad \text{soit } 0B \text{ H}$$

$$VFldStrt = VsynStp + Vbacklines = 11 + 4 = 15 \quad \text{soit } 0F \text{ H}$$

$$VFldStp = VFldStrt + Vres = 15 + 380 = 395 \quad \text{soit } 018B \text{ H}$$

$$FramLen = Vfieldlines - 1 = 399 \quad \text{soit } 018F \text{ H}$$

2) Séquences d'initialisation :

D'après les calculs effectués dans le paragraphe précédent la séquence d'initialisation du processeur de visualisation est la suivante :

VStat	:	0003 H
IntMask	:	00FF H
TripPt	:	0018 H
Frint	:	0000 H
CRT Mode	:	0000 H
HSynStp	:	009D H
HFldStrt	:	00B1 H
HFldStp	:	03D1 H
LineLen	:	03E5 H
VSynStp	:	000B H
VFldStrt	:	000F H
VFldStp	:	018B H
FramLen	:	018F H
Desc. Adr. Low	:	00D4 H
Desc. Adr. Upp	:	0000 H
Zoom	:	0000 H
FldColor	:	0000 H
BdrColor	:	00FF H
1Bpp Pad	:	0000 H
2Bpp Pad	:	0000 H
4Bpp Pad	:	0000 H
CsrMode	:	A002 H
CsrPosX	:	01FE H
CsrPosY	:	0064 H
CsrPat0	:	0180 H
CsrPat1	:	03C0 H
CsrPat2	:	07E0 H
CsrPat3	:	0FF0 H
CsrPat4	:	1FF8 H
CsrPat5	:	3FFC H
CsrPat6	:	7FFE H
CsrPat7	:	FFFF H
CsrPat8	:	07E0 H
CsrPat9	:	07E0 H
CsrPatA	:	07E0 H
CsrPatB	:	07E0 H
CsrPatC	:	07E0 H
CsrPatD	:	07E0 H
CsrPatE	:	07E0 H
CsrPatF	:	07E0 H
CsrPat1	:	07E0 H

4.4 : LE MONITEUR GRAPHIQUE ET SA STRUCTURE

Le software à été réalisé uniquement dans le cadre de la démonstration et de la vérification du fonctionnement correct de la carte graphique.

Toutes les séquences décrites, tout au long de ce rapport ont été programmées en langage assembleur INTEL 8031.

Le software écrit pour cette carte est organisé sous forme de moniteur graphique.

Il se compose de plusieurs parties :

1) Initialisation globales :

Cette partie permet de faire toutes l'initialisation nécessaires au GDC en lui envoyant tout les parametres.

Elle permet aussi d'initialiser les différentes variables utilisées par le moniteur, durant l'exécution des différentes commandes graphiques.

2) Attente d'une commande :

Après l'initialisation le processeur graphique se met en attente d'une commande.

La commande tapée au clavier ainsi que ses parametres est sauvegardée dans une mémoire FIFO en vue de son exécution ulterieure. Cette routine permet de gérer les interruptions clavier et de traiter les touches DEL et RETURN du clavier qui permettent d'effacer un caractere ou d'envoyer la commande.

3) Analyse de la commande:

Cette partie du moniteur permet d'analyser la commande et d'en déterminer sa nature. Lorsqu'une commande est erronnée, le moniteur fera apparaitre des messages d'erreurs.

4) Controle des données :

Cette phase effectue un controle sur les données introduites par l'utilisateur (les nombres doivent etre formés de chiffres décimaux, et etre inferieurs à 999).

5) Exécution des commandes graphiques :

Après l'analyse de la commande tous les parametres sont fournis au GDC pour l'exécution. Si une erreur est détectée par le GDC , un message d'erreur est affiché par le moniteur graphique.

CONCLUSION GENERALE:

La carte graphique et son moniteur de gestion sont actuellement opérationnels .

En fait , le but de ce projet étant la conception et la réalisation d'une carte d'évaluation à base du nouveau produit , le Intel 82786 , fonctionnelle sur le plan hardware .

Le moniteur de gestion dont il est question n'est rien moins qu'un moyen de dialogue entre l'environnement hôte et la carte proprement dite. Par là même, l'utilisateur éventuel aura nécessairement besoin de réécrire son propre logiciel, d'une manière beaucoup élaborée en rapport avec l'environnement hôte.

Cette élaboration consistera en l'exploitation de toutes les possibilités offertes par le GDC.

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

- Principles of interactive Computer Graphics
William M. NEWMANN et R.F. SPROULL
- Images et ordinateur: Introduction à l'infographie
Pierre MORVAN et Michel LUCAS
- Microprocesseurs et micro-ordinateur
R LYON-CAEN et J.M. CROZET
- Data Book Texas Instrument (notes d'applications)
- MCS-51 Family of Single chip Microcomputers User's
Manual
INTEL juillet 1981
- PC TECH JOURNAL JULY 1987
Intelligent Graphics Processors. VOL. 5 NO. 7
- INTEL Application Note (AP-270)
82786 Hardware Configuration Mai 1986
- INTEL Application Note (Advanced Information)
82786 CMOS Graphics Coprocessor Mai 1986

ANNEXE

SCHEMA DE LA CARTE
GRAPHIQUE

