



Fonds National
des Thèses

Reproduction & Diffusion interdite

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

**MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE**

**UNIVERSITE DE BLIDA
INSTITUT D'ELECTRONIQUE**

THESE

Pour obtenir le grade de

MAGISTER EN ELECTRONIQUE

Option Communication

Présentée par

Mountassar MAAMOUN

THEME

**CONCEPTION ET REALISATION D'UN SYSTEME
MIXTE LOGICIEL/MATERIEL
POUR L'AFFICHAGE DES IMAGES PC
SUR UN MONITEUR TV**

Devant le jury composé de :

**M. A.GUESSOUM
M. A.NAMANE M.
H.SALHI
M. M.BENSEBTI
M. A.BEN BELKACEM**

*Professeur JE Blida
Chargé de Cours JE Blida
Maître de Conférence JE Blida
Maître de Conférence JE Blida
Chargé de Cours ENS Kouba*

**Président
Rapporteur
Examineur
Examineur
Examineur**

Blida, Algérie, 2001



REMERCIEMENTS

Je tiens à remercier :

Tout d'abord, Monsieur A.NAMANE, de m'avoir accordé sa confiance et qui m'a fait le plaisir d'encadrer ma thèse et de m'avoir soutenu tout le long de ma recherche avec ses conseils et ses suggestions.

Monsieur A.GUESSOUM, Professeur à l'Université de Blida, pour son aide et qui m'a fait l'honneur de présider le jury de cette thèse.

Monsieur KAZED et Monsieur AIDJA, qui ont accepté la lourde tâche d'être les deux membres du comité de lecture pour mon travail.

Monsieur M.BENSEBTI Directeur de la PGRS, qui m'a encouragé, pour son aide et pour avoir accepté de participer au jury.

Monsieur SALHI, Maître de Conférence à IE de Blida et Monsieur BEN BELKACEM de l'Ecole Normale Supérieure de Kouba, qui m'ont fait le plaisir de participer au jury.

Une place particulière dans mes remerciements est réservée à ma famille, ma mère, ainsi qu'à ma sœur SAIDA et mon frère MOUHAMED.

Un grand merci à ma femme GHANIA, pour son aide et son soutien moral tout le long de mon travail ; Merci GHANIA.

Un merci chaleureux pour ma deuxième famille, ma tante KHEIRA, mon oncle AISSA, mes sœurs SAFIA, SAMIA, NAZIHA, WIDAD, MERIEM, NORA et mes frères ALI, AMIN, MOURAD, BOUKHARI, DJAMEL et les autres.

Mes amis, Khaled, Mehdi, Mohamed, Salem, Yacine, Sofien, Djahid et deux mercis pour le charmant boualem.

A tous ceux qui ont participé de près ou de loin, à la contribution de ma thèse.

INTRODUCTION

Le monde de l'informatique et des micro-ordinateurs a connu, durant la dernière décennie, un essor important. Car, depuis la création du premier ordinateur, lent en vitesse et faible en mémoire et capacités de calcul, personne n'a pu prédire ce que l'informatique et les ordinateurs sont devenus aujourd'hui.

Cet essor a été réellement touché par le grand public surtout, après que IBM avait lancé, au début des années 80 [36], le premier micro-ordinateur personnel (PC de Personal Computer). Depuis, des perfectionnements et améliorations, matériels et logiciels, n'ont pas cessé de voir le jour.

Du côté logiciel, nous assistons à une véritable révolution. Que ce soit pour les systèmes d'exploitation (DOS, Unix, Windows), les jeux, les langages de programmation, les utilitaires, etc., une multitude de logiciels leur a été dédiée.

DOS (Disk Operating System), le premier système d'exploitation pour PC, a connu, depuis sa première version très limitée, de nombreuses mises-à -jour, jusqu'à arriver à la version 7.10A, mais éclipsé par l'apparition de son grand frère Windows. Ce dernier, n'était qu'une simple interface graphique, permettant la gestion du PC de manière plus interactive et plus conviviale que le DOS; s'est évolué jusqu'à devenir le système d'exploitation, Windows.

Les langages de programmation, de leur côté, ont eu leur part dans cette révolution. Depuis le langage de programmation BASIC (Beginners Allpurpose Symbolic Instruction Code), un grand nombre d'autres langages, plus performants, ont été réalisés. Parmi ceux-ci, nous citons Fortran, Pascal et C.

L'avancée qu'ont connu les logiciels s'est réalisée grâce au développement révolutionnaire de la micro-électronique (le hard). Car, depuis le premier microprocesseur d'Intel 8080, plusieurs autres microprocesseurs, de plus en plus performants, ont été réalisés. Les perfectionnements effectués concernent le nombre de bits des mots que le microprocesseur peut traiter, la vitesse d'exécution des instructions (vitesse d'horloge), les techniques de traitements des données ainsi que l'architecture interne des microprocesseurs [44](le i486 a un pipeline et le Pentium en a deux).

Ces perfectionnements matériels n'ont pas concerné seulement les microprocesseurs et donc la carte mère, mais ils ont touché les autres modules de l'unité système, tels que la carte graphique et les cartes contrôleurs ainsi que les autres cartes, dites d'extension, à qui les constructeurs des PC ont réservé des slots où elles doivent être insérées [20]. Parmi ces cartes d'extension, nous citons la carte son, la carte tuner et la carte scanner. Nous notons que les slots d'extension ne sont pas seulement réservés aux cartes mises au marché par certains constructeurs, mais ils peuvent être occupés par des cartes personnelles dédiées à des fonctions particulières, telles que la transmission de données haut débit, le traitement d'images, le contrôle des robots, etc.

En 1981 l'ordinateur personnel du géant de l'informatique américaine, l'IBM PC avait un écran monochrome¹ et ne possédait pas de mode graphique proprement dit. Avec 80 colonnes sur 25 lignes, l'utilisateur devait pouvoir subvenir à l'ensemble de ses besoins professionnels. Aujourd'hui, le PC offre plus de 16 millions de couleurs et pour acquérir un PC ou un compatible, l'une des premières questions est le choix de la carte graphique. L'affichage est un domaine qui n'a pas cessé de subir des perfectionnements et améliorations prodigieuses.

¹ Deux couleurs, noir et blanc.

Si nous voulons confier à un PC une gestion de tâches complexes d'acquisition et de traitement, et en même temps l'affichage permanent d'une image ou une séquence d'images occupant la totalité de l'écran, un deuxième système d'affichage avec un maximum d'autonomie par rapport au processeur du PC est indispensable, afin de réserver la carte graphique et l'écran du PC pour l'affichage habituel.

L'objet de nos recherches, qui s'inscrit dans le cadre de l'équipe de recherche **LATSI**¹, vise deux objectifs : La conception et la réalisation d'un système mixte logiciel/matériel pour PC-IBM et compatibles, qui a pour objet la réalisation de la fonction inverse de la carte tuner, soit, la visualisation d'une image ou une séquence d'images traitées ou stockées sur ordinateurs, sur un écran TV. Elle assure, donc, la conversion des fichiers images (image numérique) en un signal vidéo composite (image analogique). Le deuxième objectif est d'assurer le transfert de données du PC vers l'extérieur avec un système performant qui offre la gestion d'un espace mémoire externe relativement grand et donne la possibilité d'une commande simultanée d'un grand nombre de cartes prototypes.

Si en théorie, les étapes d'une conception matérielle paraissent presque simples, elles sont beaucoup plus complexes dans la pratique [44], car la conception d'une carte d'extension met le concepteur en contact direct avec les signaux du bus. La connaissance de la synchronisation de ces signaux est très importante pour les concepteurs de cartes d'extensions qui doivent respecter les protocoles sous peine de voir défaillir l'ordinateur, mais IBM n'a jamais publié ce protocole, ce qui oblige le concepteur à tester en profondeur le fonctionnement matériel des cartes mères et des slots d'extensions pour pouvoir réussir.

La présente étude est répartie en quatre chapitres :

Le premier chapitre présente les principales formes de contributions des ordinateurs dans l'industrie et les techniques architecturales des PC, pour ensuite entamer les principaux outils de conception mixte logicielle/matérielle, langage de programmation, technologie matérielle et le canal de lien entre les deux.

¹ Laboratoire du traitement d'image et du signal de l'institut d'électronique de l'université de BLIDA.

Le deuxième chapitre présente la théorie des images analogiques et les principales formes des images numériques.

Le troisième chapitre Contribution et Réalisation logicielle.

Dans ce chapitre, nous présentons premièrement : le système et les avantages de l'adressage proposé (Adressage Physique Étendu) pour contribuer à l'adressage des cartes prototypes ; deuxièmement : la structure générale du programme réalisé et les étapes du traitement logiciel.

Le quatrième chapitre explique le fonctionnement matériel de la carte et les principaux modules et leurs circuits de base.

La Thèse sera terminée par une conclusion générale où nous commentons le travail effectué et nous proposons des améliorations et perspectives.

CHAPITRE 1

GENERALITES SUR LES PC ET LA CONCEPTION MIXTE LOGICIELLE/MATERIELLE

PRINCIPES GENERAUX SUR LES IMAGES ANALOGIQUES ET LES IMAGES NUMERIQUES

2.1. INTRODUCTION

La puissance et le développement qu'ont connu les PC ont poussé plusieurs industriels et chercheurs à mettre les moyens nécessaires pour exploiter les performances des PC standards (PC-IBM et compatibles) au lieu d'utiliser des ordinateurs très coûteux spécialisés dans des applications professionnelles.

Dans notre cas nous allons exploiter les performances des PC pour réaliser la conversion des images traitées ou stockées en un signal vidéo composite (image analogique). Ce chapitre présente premièrement des généralités sur les images analogiques en expliquant le principe de reproduction des images et la forme du signal vidéo. Deuxièmement nous présentons les images numériques avec leurs formes principales.

2.2. IMAGE ANALOGIQUE

2.2.1. Principe de reproduction des images

Les caractéristiques de l'œil ont rendu la reconstruction des images animées très possible. En effet, l'inertie du système optique permet de donner l'impression d'images animées par une succession d'images fixes de caractéristiques très proches. Ce principe est utilisé par le cinéma¹. En télévision chaque image (photo) est en outre décomposée en un certain nombre de points élémentaires. On transmet 25 images² par seconde pour reconstituer le mouvement (nombre d'images supérieur à la valeur de persistance rétinienne qui est de 25 images par seconde). Lors de l'analyse séquentielle des points d'une ligne et grâce à la persistance rétinienne, l'œil intègre les sensations visuelles de chaque point ; il analyse simultanément tous les points, et perçoit une image complète [5][17].

2.2.2. Principe du balayage entrelacé

On va procéder avec une certaine variante; il faut en effet s'imaginer que le tube cathodique est en réalité assimilable à une juxtaposition de petites lampes réparties sur tout l'écran. Ces lampes seront allumées à tour de rôle, mais une seule à la fois est l'intensité avec laquelle chaque lampe va briller donnera notre échelle de gris. La lampe va s'allumer puis s'éteindre avec une certaine inertie, et il faut qu'elle garde la lumière le plus longtemps possible, mais si elle est éclairée au maximum dans une image (blanc) elle doit pouvoir être éteinte dans l'image suivante (noir). Ce qui veut dire que nous sommes obligé d'avoir une rémanence décrite. nous aurons le haut du tube cathodique qui va s'éteindre quand on analyse le bas de l'écran. et la zone d'ombre va ainsi se déplacer sur tout l'écran au rythme de l'analyse de l'image.

Pour éviter ce phénomène, on va analyser l'image d'abord en s'occupant des lignes impaires; lorsque l'image est terminée, on se préoccupe des lignes paires et de cette manière le scintillement ne sera pratiquement pas perceptible à l'œil humain. On est conduit à explorer

¹ 24 images/s.

² Deux systèmes sont encore en application, 25 images/s (Europe et autres) et 30 images/s (USA et Japon).

l'image par trames entrelacées. On transmet successivement deux trames par image[5][17]. On explore d'abord toutes les lignes impaires ensuite toutes les lignes paires de l'image. On transmet ainsi 50 trames par seconde (i.e. 25 images) : la fréquence de balayage est doublée pour devenir égale à la fréquence du secteur d'alimentation 50 Hz.

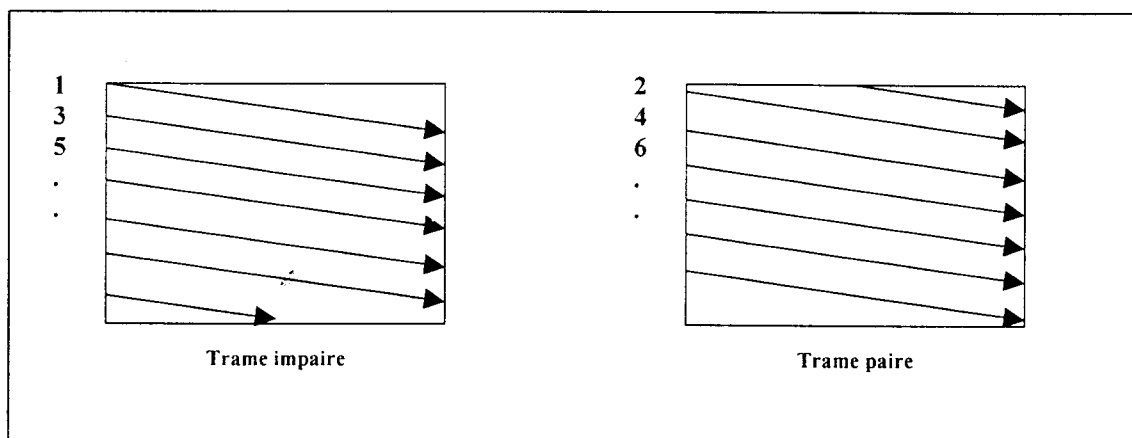


Figure.2.1. *Le balayage entrelacé*

2.2.3. Définition de l'image

Le choix du nombre de lignes d'analyse est un compromis entre la satisfaction visuelle (distance optimale de vision, dimensions de l'image) et la complication des circuits. Pour le confort maximal d'observation, la distance normale d'observation doit être égale à 4 fois la plus grande dimension de l'objet (la diagonale).

Le format rectangulaire ($3/4$ ou $9/16$) de l'écran répond le mieux aux soucis d'esthétique. Pour le format $3/4$ nous avons

$$\frac{H}{L} = \frac{3}{4}$$

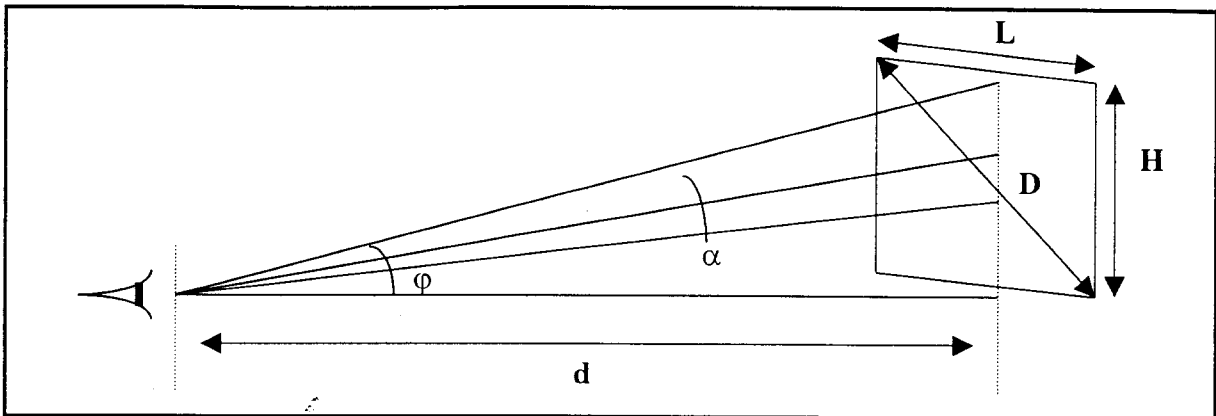


Figure. 2.2. Distance optimale de vision

$$D = \sqrt{H^2 + L^2} = \sqrt{H^2 + (4H/3)^2} = \frac{5H}{3}$$

$$d = 4D = \frac{20H}{3}$$

$$\operatorname{tg} \frac{\varphi}{2} = \frac{H/2}{d} = \frac{H}{2} \frac{3}{20H} = 0.075 \Rightarrow \varphi = 8^\circ 40' = 520'$$

Le pouvoir de résolution de l'œil est de : une minute ($\alpha=1'$). Le nombre minimal de lignes correspond au nombre maximal d'éléments que l'œil pourra discerner dans la hauteur H, d'où :

$$n = \frac{\varphi}{\alpha} = 520$$

Il faut donc au moins 520 lignes d'analyse [5][17].

2.2.4. Le choix des 625¹ lignes pour la norme B

Pour faire le choix de nombre de lignes dans une image T.V., il faut respecter plusieurs critères. Il faut avoir un nombre impair de lignes, pour réaliser un entrelacement rigoureux des deux trames; il faut 20 lignes pour transmettre les tops de synchronisation trame (suppression trame = 18 à 22 H \pm 12 μ s); les fréquences lignes et trames doivent être

¹ 625 lignes avec 50 trames/s en Europe.

rigoureusement asservies; elles sont obtenues, à l'émission, par divisions successives de la fréquence d'un oscillateur pilote (31.25 KHz).

2.2.5. BANDE PASSANTE

Ce mot est apparu lorsqu'on a abordé le problème de la rapidité de transmission des informations, nous allons voir que le problème est plus complexe qu'un choix de qualité. Le raisonnement est le suivant :

Si on a opté à un confort visuel avec 625 lignes dans le sens vertical, il faudrait penser à obtenir la même définition, dans le sens horizontal.

Nous avons vu qu'il y a en réalité, (environ) 560 lignes. Le rapport largeur/hauteur est de 4/3 donc il faut :

$$560 \times 4/3 = 746 \text{ points.}$$

Ces 746 points sont en réalité une suite de lignes blanches et noires. Il faut donc considérer qu'il s'agit d'un signal carré qui passe du blanc au noir successivement pendant 52 μ s (64 μ s moins le retour du spot en fin de ligne).

Il y aura donc $746/2 = 373$ ensembles N & B, pendant 52 μ s. Soit une période de $52\mu\text{s}/373=0.13941 \mu\text{s}$. D'où une fréquence de 7.17 Mhz. Le CCIR utilise un coefficient de définition horizontale de 0.97 ce qui nous amène à 5 MHz.

2.2.6. Le signal vidéo composite

Ce signal [5][17] que nous avons déjà abordé est en effet, le signal le plus important de toute l'étude de la télévision. Bien comprendre le « pourquoi » et le « comment » de ce signal et tout ce qui découle devient « presque » simple. Voyons les étapes et données successives qui permettent de construire ce signal.

Le tube cathodique va produire une lumière en fonction de la valeur de la tension entre deux de ses électrodes. Plus la tension sera grande, et plus la lumière sera forte. En l'absence de signal, le TRC (Tube à Rayons Cathodiques) devra rester éteint. Il faudra donc fixer deux

limites : Celle du blanc maximum et celle de la coupure du tube (on appelle le point d'extinction : « cut off »). Nous appellerons ces deux niveaux, tout simplement : niveau du blanc et niveau du noir. En fait, le niveau du noir se situera juste au-dessus du cut off car le noir doit être visible en fonction du réglage du niveau de luminosité du téléviseur.

Le top de synchronisation ligne est la période, pendant laquelle le TRC est au cut off, est appelé « blanking »; il signifie période de blocage du tube. Le top de synchronisation trame doit être différent des tops lignes également placés dans la zone de cut off. Cette condition est encore plus impérative que pour les tops lignes, car le retour trame dure plusieurs lignes.

2.2.6.1. Détails du signal vidéo composite

Le signal vidéo composite est composé de deux parties (Figure 2.5.), 70% information vidéo et 30% information de synchronisation. Les 70% du signal vidéo composite seront considérés comme la partie positive du signal, et la partie inférieure (les 30%) comme la partie négative. Cette partie négative représente les tops de synchronisation ligne et trame qui vont servir à synchroniser les oscillateurs.

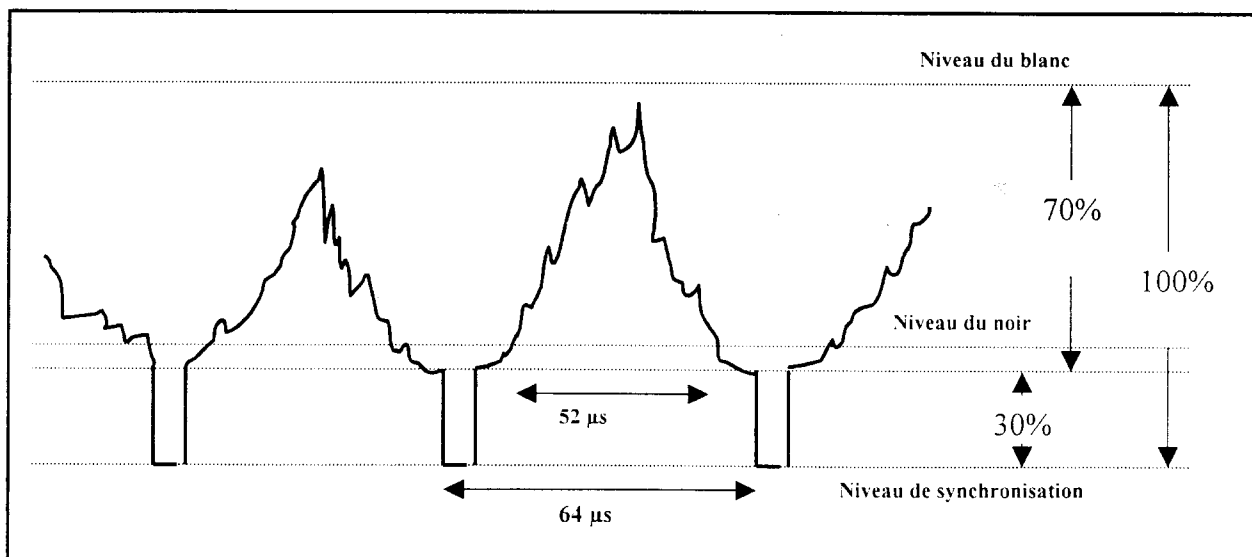


Figure. 2.3. Signal vidéo composite

2.2.6.2. Top de synchronisation ligne

Le top de synchronisation ligne (Figure 2.4.) est précédé par un palier avant de 1.3 à 1.8 μs , qui assure un passage par le blocage du tube. Le palier arrière est très important car il prendra un rôle vital dans les émissions en couleur. Sa durée de vie est de 5.4 μs .

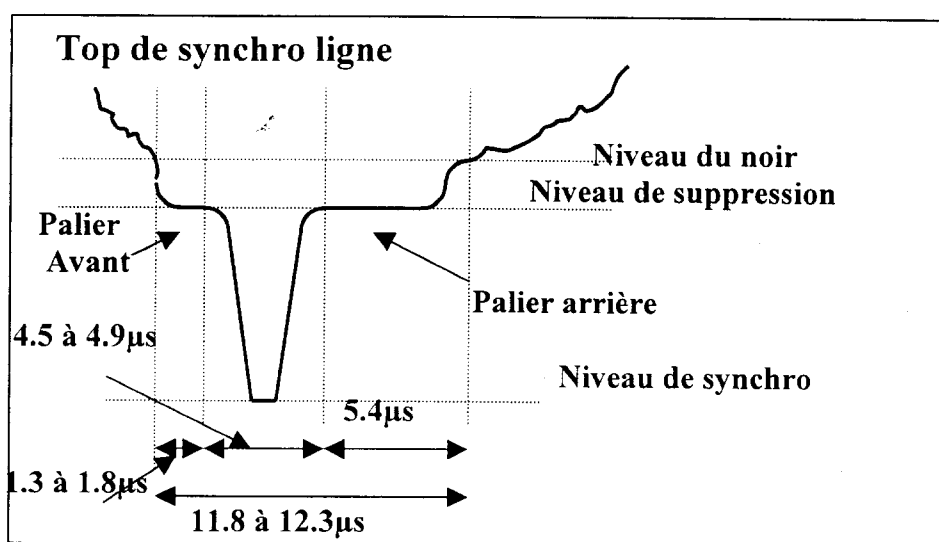


Figure. 2.4. Top de synchronisation ligne

2.2.6.3. Top de synchronisation trame

L'impulsion verticale est décomposée en un train de 5 impulsions de synchronisation verticale, larges de 27 μs chacune. Le signal de synchronisation trame complet (Figure 2.5.) (impulsions préégalisation verticale et postégalisation incluses) est appliqué après différenciation à la B.T.H. Le balayage ligne ne sera pas interrompu durant le retour trame.

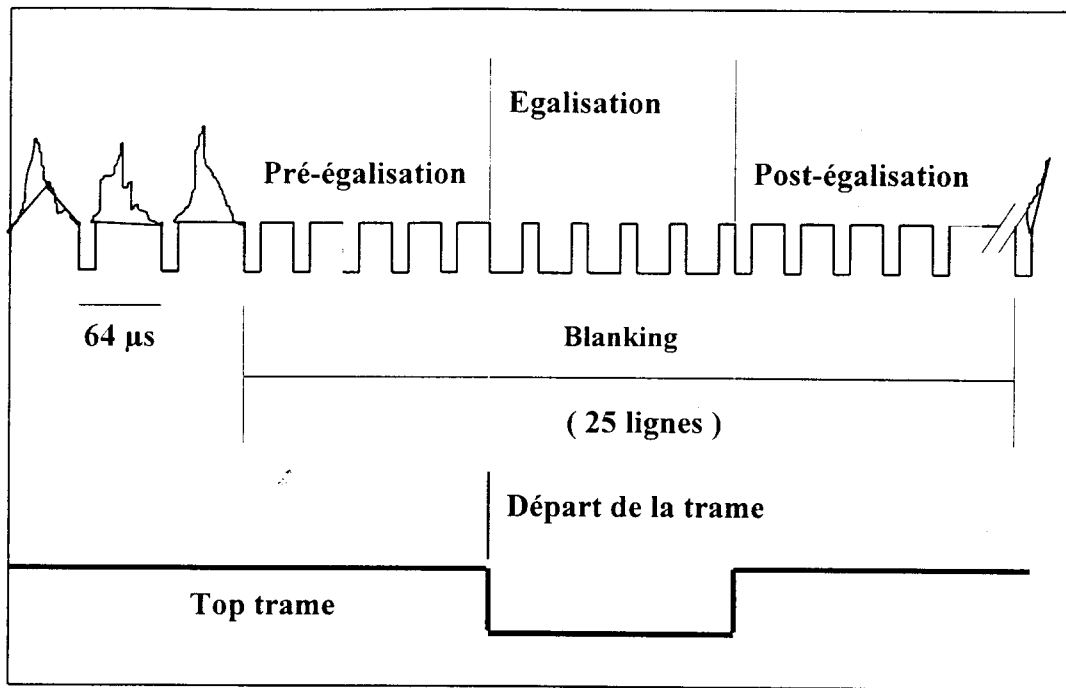


Figure. 2.5. Top de synchronisation trame

2.3. IMAGE NUMÉRIQUE

2.3.1. Un modèle d'image

Très souvent les images physiques visibles qu'on rencontre en traitement d'image sont des images bidimensionnelles monochromes ou monospectrales, par exemple une photo dite « en noir et blanc » ou « en niveaux de gris ». La Figure 2.6. donne un exemple d'une image de ce type qui est représentée par une fonction $f(x, y)$ traduisant l'intensité lumineuse du point de coordonnées x et y .

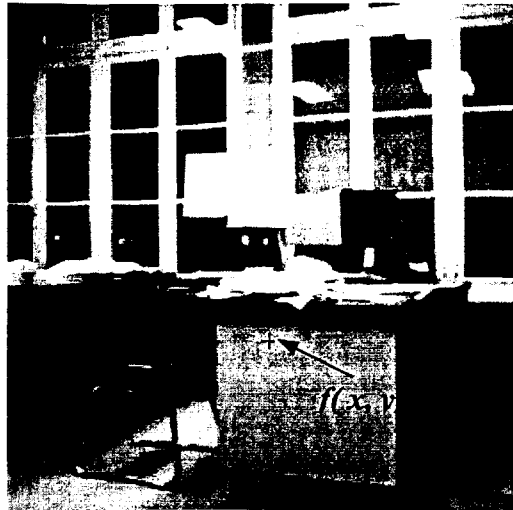


Figure 2.6. Une image monochrome d'une scène d'intérieur.

Cette intensité est appelée **niveau de gris** (voir Figure 2.7.), on distingue dans les images de qualité jusqu'à 256 niveaux de gris (de 0 pour le noir jusqu'à 255 pour le blanc).

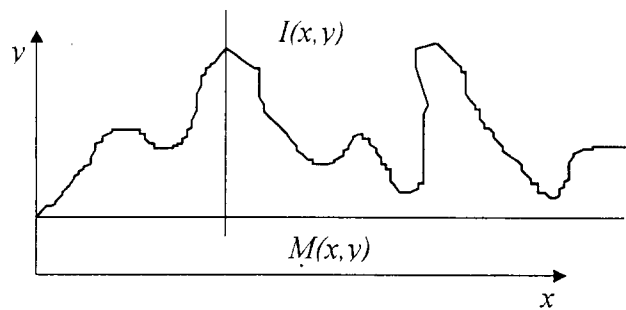


Figure 2.7. Amplitude de la fonction $f(x, y)$ ou intensité lumineuse.
 $I = f(x, y)$ représente l'intensité au point M .

La modélisation de l'image analogique à partir de son support matériel s'obtient par une numérisation en espace et en fréquence.

- Discrétiser les coordonnées (x, y) , c'est échantillonner l'image.
- Discrétiser l'amplitude d'un point, c'est quantifier les niveaux de gris.

L'image analogique continue, $I = f(x, y)$, est alors approximée par des échantillons $\delta(x, y)$ (signaux de DIRAC) que l'on écrit dans un tableau de $N \times N$ cases, sous la forme de valeurs discrètes $f(N, N)$ (Figure 2.8.).

$$I = \begin{pmatrix} f(0,0) & f(0,1) & \dots & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & & & f(1,N-1) \\ \dots & \dots & & & \dots \\ \dots & \dots & & & \dots \\ f(N-1,0) & \dots & & & f(N-1,N-1) \end{pmatrix}$$

Figure 2.8. Modèle de l'image numérisée

Le tableau comporte N^2 valeurs; chacune d'elle est appelée **pixel**¹, néologisme anglais qui vient de la contraction de l'expression « picture element ». La mémoire utile pour stocker un pixel peut varier de 1 bit (cas des images monochromes) à 24 bits (images en 16 millions de couleurs). Généralement, on utilise des images à 256 couleurs qui nécessitent 8 bits par pixel [14][19].

Il existe plusieurs formats d'images informatiques stockées sous formes de fichiers. Le stockage peut être avec une méthode de compression ou sans compression, ces fichiers se différencient en première vue par leur extension (*.ima, *.img, *.pcx, *.cpt, *.bmp, *.gif, *.jff, *.jpg, *.jtf, etc.....). Nous allons aborder dans les paragraphes qui suivent deux exemples, les fichiers images non compressés (format brut et format BMP) et les fichiers compressés (utilisant la norme JPEG) [12].

2.3.2. Les fichiers images non compressés

2.3.2.1. Le format brut

Les fichiers images de formats bruts sont généralement les fichiers d'extension *.ima ou *.img. Si on a l'image numérique, $I = f(i, j)$, qui est écrite dans un tableau de (*hauteur***largeur*) cases, sous la forme de la figure 2. 9.

¹ Pour plus de détails, voir [14].

$$I = \begin{pmatrix} f(0,0) & f(0,1) & \dots & \dots & f(0,L-1) \\ f(1,0) & f(1,1) & & & f(1,L-1) \\ \dots & \dots & & & \dots \\ f(H-1,0) & \dots & & & f(H-1,L-1) \end{pmatrix}$$

$$0 \leq i \leq (L-1) ; 0 \leq j \leq (H-1)$$

Figure 2. 9. Modèle de l'image numérique

- H est la hauteur de l'image et L la largeur de l'image.

Le fichier, $F = d(x)$, qui représente le stockage de l'image I , sera écrit dans un tableau, unidimensionnel sous la forme de la figure 2. 10.

$$F = \begin{pmatrix} d(0) \\ d(1) \\ \dots \\ d(T-2) \\ d(T-1) \end{pmatrix} \quad 0 \leq x \leq T-1$$

Figure 2. 10. Modèle du fichier d'image brut

- T représente la taille du fichier F .

Les valeurs $d(x)$ du fichier F remplissent la condition suivante :

$$d(i + L \cdot j) = f(i, j)$$

2.3.2.2. Le format BMP¹

Les fichiers BMP sont les types les plus utilisés pour les stockages des images informatiques non compressées couleurs ou noir et blanc. La structure du fichier BMP est donnée dans le paragraphe suivant.

¹ Le format BMP de Bitmap Windows.

Structure des fichiers BMP

Un fichier BMP comporte quatre parties distinctes qui sont :

- La première appelée « **BITMAPFILEHEADER** », fournit des informations générales sur le fichier ;
- La deuxième appelée « **BITMAPINFOHEADER** », contenant des informations sur l'image proprement dite ;
- La troisième représente la table des couleurs. Elle est constituée de plusieurs structures « **RGBQUAD** ».
- Et en fin, une dernière rassemblant le codage complet de chacun des pixels de l'image.

Les trois premières parties sont appelées le **HEADER**

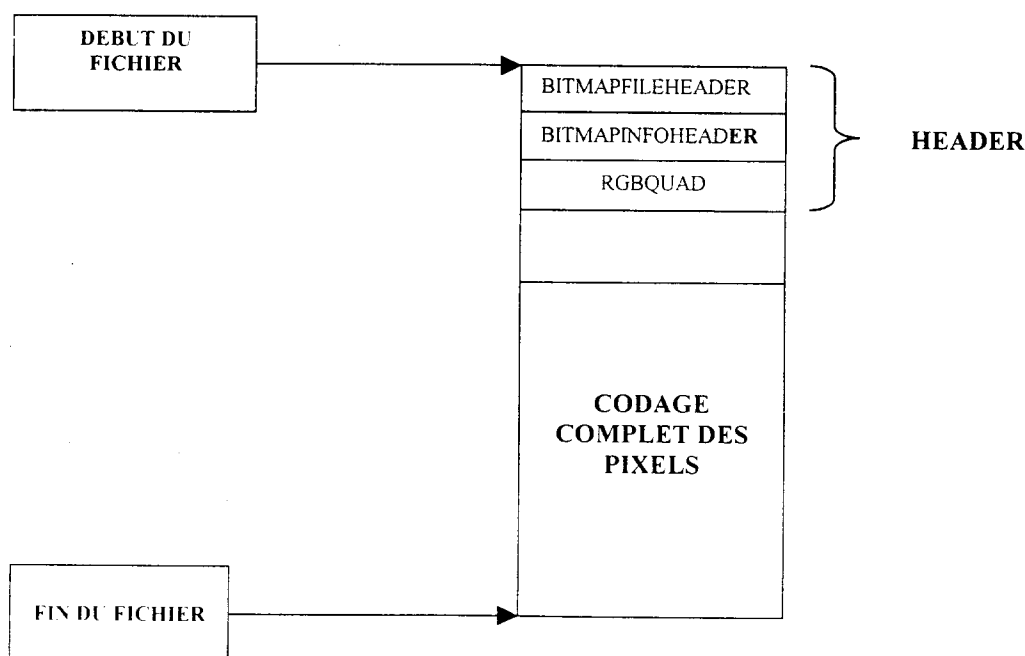


Figure 2.11. Fichier BMP

Si on garde la même image numérique, $I = f(i, j)$, qui est écrite dans un tableau de (*hauteur*largeur*) cases, sous la forme de la figure 06. Le fichier, $F_{bmp} = d(x)$, qui représente le stockage de l'image I avec le format BMP, sera écrit dans un tableau, unidimensionnel sous la forme de la figure 2. 12.

$$Fbmp = \begin{pmatrix} d(0) \\ d(1) \\ \dots \\ \dots \\ \dots \\ d(K-1) \\ \dots \\ \dots \\ \dots \\ d(R) \\ \dots \\ \dots \\ \dots \\ \dots \\ d(T-1) \end{pmatrix} \quad 0 \leq x \leq T-1$$

Figure 2. 12. Modèle du fichier image BMP

- T représente la taille du fichier $Fbmp$.
- $(K-1)$ = taille de l'entête du fichier BMP

Pour le chargement de l'image I avec le format BMP, les valeurs $d(x)$ du fichier $Fbmp$ doivent remplir la condition suivante :

$$d((T-1) - L \cdot (j+1) + i) = f(i, j)$$

avec :

- $(T-1) - R = H \cdot L$

et

- H représente la hauteur de l'image et L la largeur de l'image.

Remarques:

- Chaque ligne de l'image contient un nombre d'octets multiple de quatre. Des octets nuls sont éventuellement ajoutés à la fin de la ligne.
- Un fichier BMP est chargé de telle façon que la première information de la zone de données (qui contient les valeurs des pixels) correspond au coin inférieur gauche de l'image.

2.3.3. La norme JPEG

2.3.3.1. Introduction

Le groupe JPEG (*Joint Photographic Experts Group*), créé au début des années 90 regroupe des industriels et universitaires concernés par la compression d'images. Son but est d'établir des spécifications portant sur la compression conservative et non-conservative d'images. Nous nous intéresserons ici uniquement à la compression non-conservative [12].

2.3.3.2. Principes de la norme du groupe JPEG

Ces spécifications se basent sur le fait que les informations pertinentes d'une image, caractérisée par son signal bidimensionnel $Img(x,y)$, sont uniquement les composantes de basse fréquence du signal. Le signal est transformé en un ensemble bidimensionnel de valeurs de fréquences grâce à la *transformée de cosinus discrète* (DCT) [12]. La précision des entiers représentant les valeurs de fréquences est alors diminuée pour rendre nulles les valeurs de hautes fréquences qui ne sont pas pertinentes. Cette étape, appelée "quantification" représente la partie non conservative de la compression. Les résultats sont alors comprimés de manière conservative en codant efficacement toutes les fréquences devenues nulles (voir résumé, figure 2.13.).

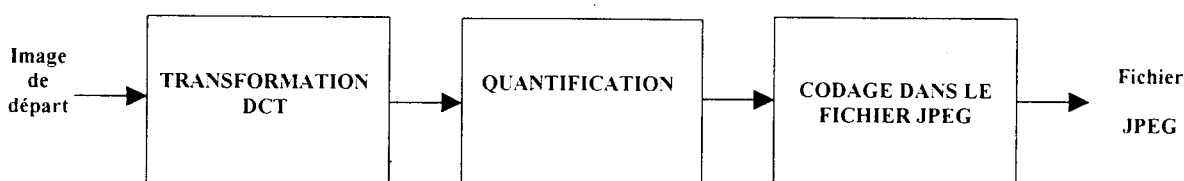


Figure. 2.13. Principe général de la compression JPEG

La décompression effectue les opérations inverses de celles de la compression. Elle commence par décompresser le résultat de la troisième étape de la compression. Elle obtient alors un tableau de valeurs de fréquences quantifiées dont beaucoup sont nulles. La quantification inverse est appliquée pour retrouver le tableau de valeurs de fréquences. Il s'agit d'une approximation du tableau résultat de la DCT dans la compression puisque les hautes fréquences non pertinentes auront 0 comme valeur. Il suffit alors d'appliquer la *transformée de cosinus discrète inverse* pour obtenir l'approximation du signal de l'image qui avait été comprimée.

La méthode de quantification détermine la qualité de l'image résultat. Elle ne fait pas partie de la spécification JPEG.

2.3.3.3. La transformée de cosinus discrète (DCT)

Pour faciliter la compréhension de cette transformée, nous allons décrire la transformée équivalente, la transformée de Fourier rapide (FFT, *Fast Fourier Transform*), travaillant sur un signal discret unidimensionnel.

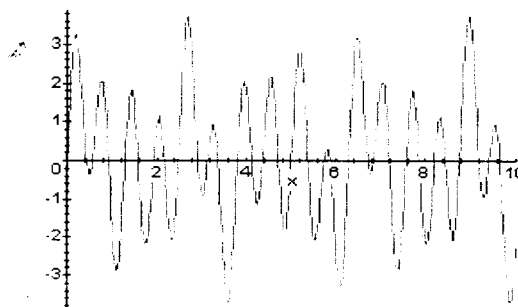


Figure. 2.14. *Signal avant FFT*

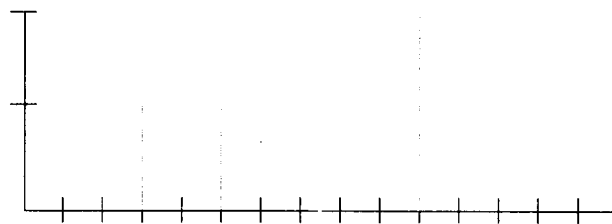


Figure. 2.15. *Signal traité par FFT*

Comme on le voit des figures 2.14. et 2.15., le gain de place est très important : au lieu de stocker la totalité de l'information, on ne conserve que la couple (fréquence-amplitude). Pour obtenir un gain de place optimal, on ne conserve que les composantes les plus significatives.

2.3.3.4. Mise en oeuvre de la méthode JPEG

Le calcul de la DCT ne peut pas se faire sur une image entière d'une part parce que cela générerait trop de calculs et d'autre part parce que le signal de l'image doit absolument être représenté par une matrice carrée. Dès lors, le groupe JPEG impose la décomposition de

l'image en blocs de 8 pixels sur 8 pixels. La méthode de compression sera donc appliquée indépendamment sur chacun des blocs. Les plus petits blocs en bordure devront être traités par une autre méthode.

Des recherches expérimentales poussées ont amené les concepteurs du JPEG à proposer la matrice de quantification définie de la façon suivante :

$$quantum(i,j) = 1 + ((1+i+j).qualité) / (i,j) \in [0,7]^2$$

Le codage du reste de la matrice DCT quantifiée va se faire en parcourant les éléments dans l'ordre imposé par une séquence particulière appelée *séquence zigzag* (voir figure 2.16.). Cette séquence a la propriété de parcourir les éléments en commençant par les basses fréquences et de traiter les fréquences de plus en plus hautes.

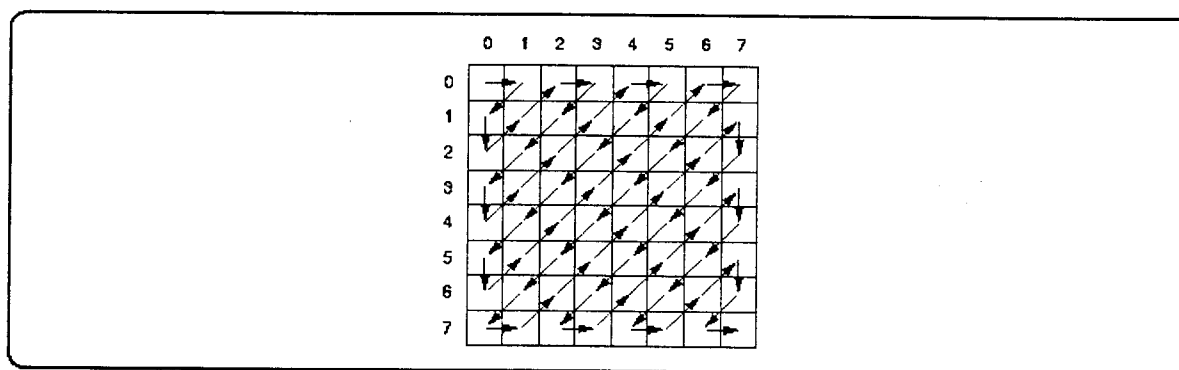


Figure 2.16. *Ordre de la séquence zigzag*

Il est très surprenant de voir la qualité de l'approximation des valeurs des pixels obtenue par décompression alors que la matrice DCT quantifiée (celle qui est réellement codée) ne semble plus contenir beaucoup d'informations. A l'heure actuelle la méthode de compression JPEG est parmi les plus utilisées parce qu'elle atteint des taux de compression très élevés sans que les modifications de l'image ne puissent être décelées par l'œil humain. De plus, beaucoup d'implémentations permettent de choisir la qualité de l'image comprimée grâce à l'utilisation de matrices de quantification paramétrables.

2.4. CONCLUSION

L'objectif ciblé est la conversion des images BMP 256 niveaux de gris (images numériques) pour les afficher sur un moniteur TV (la sortie du système est un signal vidéo composite avec la norme 625 lignes).

Pour un premier objectif nous avons :

- Utilisé des images BMP de 340*256 pixels et tronquer les images de tailles supérieures.
- Limité la bande passante du signal vidéo à 0.85 MHz (contraintes du au caractéristiques des RAM et du convertisseur D/A utilisées).

CHAPITRE 2

PRINCIPES GENERAUX

SUR

LES IMAGES ANALOGIQUES

ET

LES IMAGES NUMERIQUES

PRINCIPES GENERAUX SUR LES IMAGES ANALOGIQUES ET LES IMAGES NUMERIQUES

2.1. INTRODUCTION

La puissance et le développement qu'ont connu les PC ont poussé plusieurs industriels et chercheurs à mettre les moyens nécessaires pour exploiter les performances des PC standards (PC-IBM et compatibles) au lieu d'utiliser des ordinateurs très coûteux spécialisés dans des applications professionnelles.

Dans notre cas nous allons exploiter les performances des PC pour réaliser la conversion des images traitées ou stockées en un signal vidéo composite (image analogique). Ce chapitre présente premièrement des généralités sur les images analogiques en expliquant le principe de reproduction des images et la forme du signal vidéo. Deuxièmement nous présentons les images numériques avec leurs formes principales.

2.2. IMAGE ANALOGIQUE

2.2.1. Principe de reproduction des images

Les caractéristiques de l'œil ont rendu la reconstruction des images animées très possible. En effet, l'inertie du système optique permet de donner l'impression d'images animées par une succession d'images fixes de caractéristiques très proches. Ce principe est utilisé par le cinéma¹. En télévision chaque image (photo) est en outre décomposée en un certain nombre de points élémentaires. On transmet 25 images² par seconde pour reconstituer le mouvement (nombre d'images supérieur à la valeur de persistance rétinienne qui est de 25 images par seconde). Lors de l'analyse séquentielle des points d'une ligne et grâce à la persistance rétinienne, l'œil intègre les sensations visuelles de chaque point ; il analyse simultanément tous les points, et perçoit une image complète [5][17].

2.2.2. Principe du balayage entrelacé

On va procéder avec une certaine variante; il faut en effet s'imaginer que le tube cathodique est en réalité assimilable à une juxtaposition de petites lampes réparties sur tout l'écran. Ces lampes seront allumées à tour de rôle, mais une seule à la fois est l'intensité avec laquelle chaque lampe va briller donnera notre échelle de gris. La lampe va s'allumer puis s'éteindre avec une certaine inertie, et il faut qu'elle garde la lumière le plus longtemps possible, mais si elle est éclairée au maximum dans une image (blanc) elle doit pouvoir être éteinte dans l'image suivante (noir). Ce qui veut dire que nous sommes obligé d'avoir une rémanence décrite. nous aurons le haut du tube cathodique qui va s'éteindre quand on analyse le bas de l'écran, et la zone d'ombre va ainsi se déplacer sur tout l'écran au rythme de l'analyse de l'image.

Pour éviter ce phénomène, on va analyser l'image d'abord en s'occupant des lignes impaires; lorsque l'image est terminée, on se préoccupe des lignes paires et de cette manière le scintillement ne sera pratiquement pas perceptible à l'œil humain. On est conduit à explorer

¹ 24 images/s.

² Deux systèmes sont encore en application, 25 images/s (Europe et autres) et 30 images/s (USA et Japon).

l'image par trames entrelacées. On transmet successivement deux trames par image[5][17]. On explore d'abord toutes les lignes impaires ensuite toutes les lignes paires de l'image. On transmet ainsi 50 trames par seconde (i.e. 25 images) : la fréquence de balayage est doublée pour devenir égale à la fréquence du secteur d'alimentation 50 Hz.

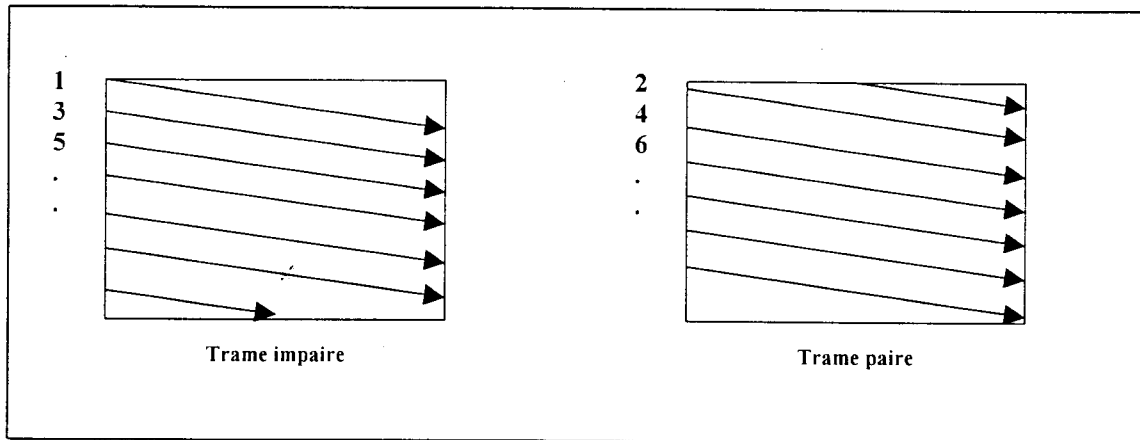


Figure.2.1. *Le balayage entrelacé*

2.2.3. Définition de l'image

Le choix du nombre de lignes d'analyse est un compromis entre la satisfaction visuelle (distance optimale de vision, dimensions de l'image) et la complication des circuits. Pour le confort maximal d'observation, la distance normale d'observation doit être égale à 4 fois la plus grande dimension de l'objet (la diagonale).

Le format rectangulaire (3/4 ou 9/16) de l'écran répond le mieux aux soucis d'esthétique. Pour le format 3/4 nous avons

$$\frac{H}{L} = \frac{3}{4}$$

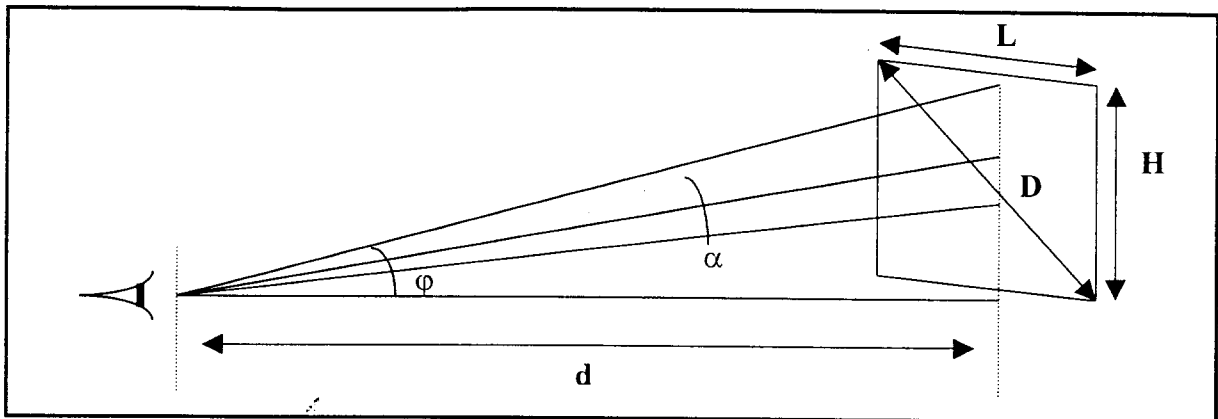


Figure. 2.2. Distance optimale de vision

$$D = \sqrt{H^2 + L^2} = \sqrt{H^2 + (4H/3)^2} = \frac{5H}{3}$$

$$d = 4D = \frac{20H}{3}$$

$$\operatorname{tg} \frac{\varphi}{2} = \frac{H/2}{d} = \frac{H}{2} \frac{3}{20H} = 0.075 \Rightarrow \varphi = 8^\circ 40' = 520'$$

Le pouvoir de résolution de l'œil est de : une minute ($\alpha=1'$). Le nombre minimal de lignes correspond au nombre maximal d'éléments que l'œil pourra discerner dans la hauteur H, d'où :

$$n = \frac{\varphi}{\alpha} = 520$$

Il faut donc au moins 520 lignes d'analyse [5][17].

2.2.4. Le choix des 625' lignes pour la norme B

Pour faire le choix de nombre de lignes dans une image T.V., il faut respecter plusieurs critères. Il faut avoir un nombre impair de lignes, pour réaliser un entrelacement rigoureux des deux trames; il faut 20 lignes pour transmettre les tops de synchronisation trame (suppression trame = 18 à 22 H \pm 12 μ s); les fréquences lignes et trames doivent être

¹ 625 lignes avec 50 trames/s en Europe.

rigoureusement asservies; elles sont obtenues, à l'émission, par divisions successives de la fréquence d'un oscillateur pilote (31.25 KHz).

2.2.5. BANDE PASSANTE

Ce mot est apparu lorsqu'on a abordé le problème de la rapidité de transmission des informations, nous allons voir que le problème est plus complexe qu'un choix de qualité. Le raisonnement est le suivant :

Si on a opté à un confort visuel avec 625 lignes dans le sens vertical, il faudrait penser à obtenir la même définition dans le sens horizontal.

Nous avons vu qu'il y a en réalité, (environ) 560 lignes. Le rapport largeur/hauteur est de 4/3 donc il faut :

$$560 \times 4/3 = 746 \text{ points.}$$

Ces 746 points sont en réalité une suite de lignes blanches et noires. Il faut donc considérer qu'il s'agit d'un signal carré qui passe du blanc au noir successivement pendant 52 μ s (64 μ s moins le retour du spot en fin de ligne).

Il y aura donc $746/2 = 373$ ensembles N & B, pendant 52 μ s. Soit une période de $52\mu\text{s}/373=0.13941 \mu\text{s}$. D'où une fréquence de 7.17 Mhz. Le CCIR utilise un coefficient de définition horizontale de 0.97 ce qui nous amène à 5 MHz.

2.2.6. Le signal vidéo composite

Ce signal [5][17] que nous avons déjà abordé est en effet, le signal le plus important de toute l'étude de la télévision. Bien comprendre le « pourquoi » et le « comment » de ce signal et tout ce qui découle devient « presque » simple. Voyons les étapes et données successives qui permettent de construire ce signal.

Le tube cathodique va produire une lumière en fonction de la valeur de la tension entre deux de ses électrodes. Plus la tension sera grande, et plus la lumière sera forte. En l'absence de signal, le TRC (Tube à Rayons Cathodiques) devra rester éteint. Il faudra donc fixer deux

limites : Celle du blanc maximum et celle de la coupure du tube (on appelle le point d'extinction : « cut off »). Nous appellerons ces deux niveaux, tout simplement : niveau du blanc et niveau du noir. En fait, le niveau du noir se situera juste au-dessus du cut off car le noir doit être visible en fonction du réglage du niveau de luminosité du téléviseur.

Le top de synchronisation ligne est la période, pendant laquelle le TRC est au cut off, est appelé « blanking »; il signifie période de blocage du tube. Le top de synchronisation trame doit être différent des tops lignes également placés dans la zone de cut off. Cette condition est encore plus impérative que pour les tops lignes, car le retour trame dure plusieurs lignes.

2.2.6.1. Détails du signal vidéo composite

Le signal vidéo composite est composé de deux parties (Figure 2.5.), 70% information vidéo et 30% information de synchronisation. Les 70% du signal vidéo composite seront considérés comme la partie positive du signal, et la partie inférieure (les 30%) comme la partie négative. Cette partie négative représente les tops de synchronisation ligne et trame qui vont servir à synchroniser les oscillateurs.

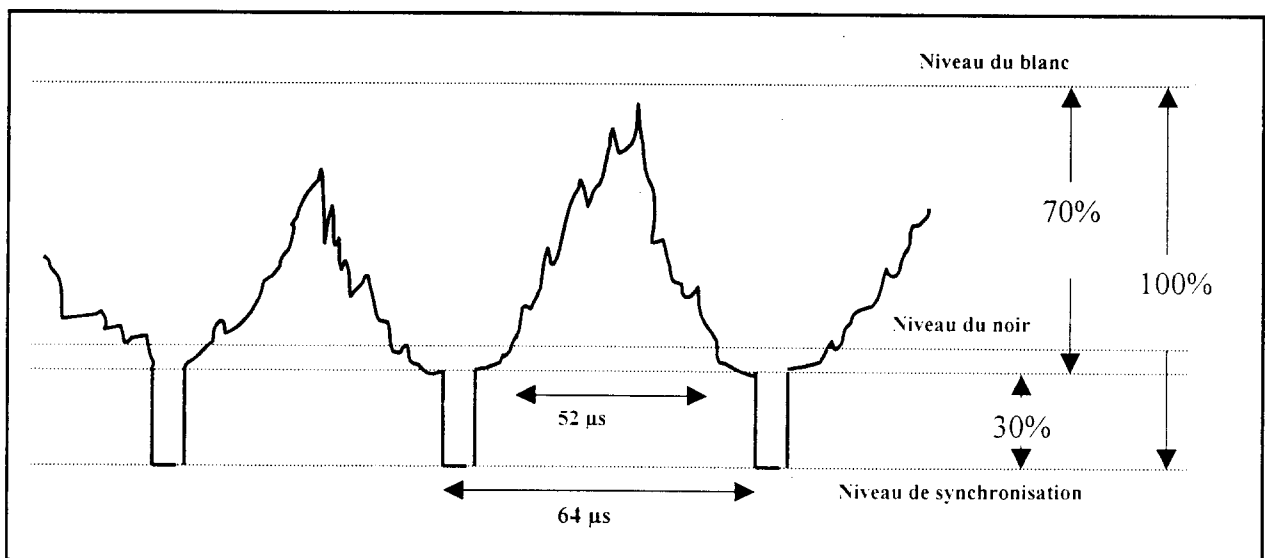


Figure. 2.3. Signal vidéo composite

2.2.6.2. Top de synchronisation ligne

Le top de synchronisation ligne (Figure 2.4.) est précédé par un palier avant de 1.3 à 1.8 μs , qui assure un passage par le blocage du tube. Le palier arrière est très important car il prendra un rôle vital dans les émissions en couleur. Sa durée de vie est de 5.4 μs .

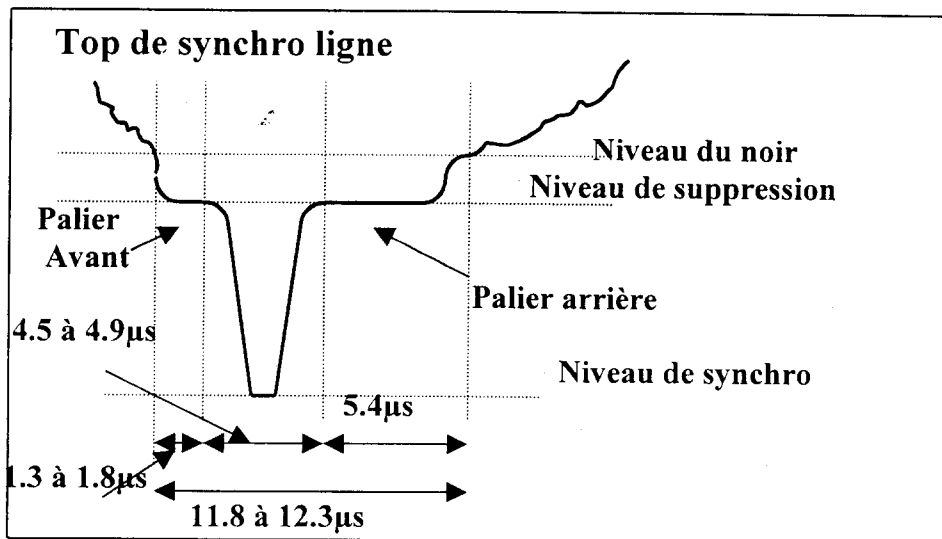


Figure. 2.4. Top de synchronisation ligne

2.2.6.3. Top de synchronisation trame

L'impulsion verticale est décomposée en un train de 5 impulsions de synchronisation verticale, larges de 27 μs chacune. Le signal de synchronisation trame complet (Figure 2.5.) (impulsions préégalisation verticale et postégalisation incluses) est appliqué après différenciation à la B.T.H. Le balayage ligne ne sera pas interrompu durant le retour trame.

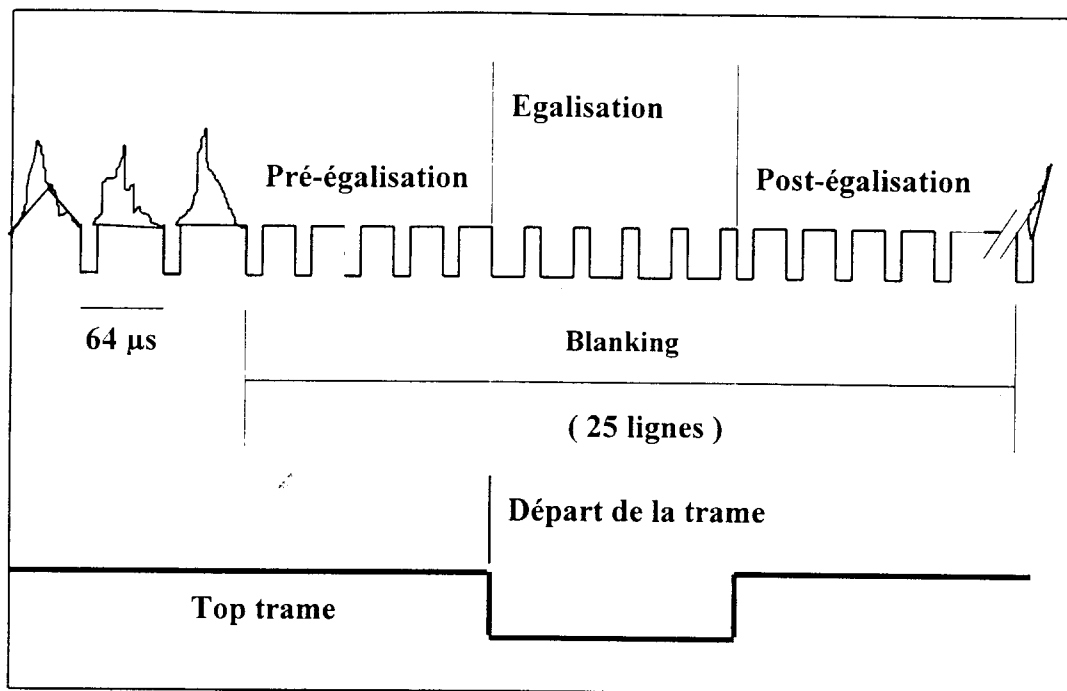


Figure. 2.5. Top de synchronisation trame

2.3. IMAGE NUMÉRIQUE

2.3.1. Un modèle d'image

Très souvent les images physiques visibles qu'on rencontre en traitement d'image sont des images bidimensionnelles monochromes ou monospectrales, par exemple une photo dite « en noir et blanc » ou « en niveaux de gris ». La Figure 2.6. donne un exemple d'une image de ce type qui est représentée par une fonction $f(x, y)$ traduisant l'intensité lumineuse du point de coordonnées x et y .



Figure 2.6. Une image monochrome d'une scène d'intérieur.

Cette intensité est appelée **niveau de gris** (voir Figure 2.7.), on distingue dans les images de qualité jusqu'à 256 niveaux de gris (de 0 pour le noir jusqu'à 255 pour le blanc).

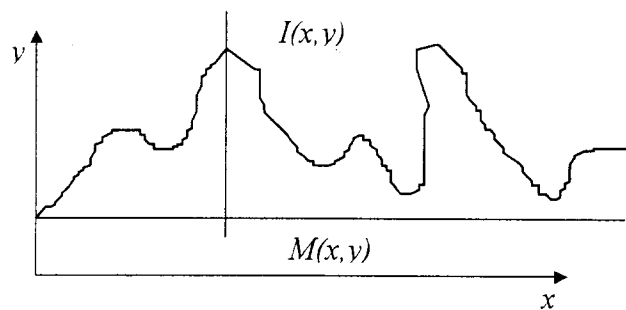


Figure 2.7. Amplitude de la fonction $f(x, y)$ ou intensité lumineuse.
 $I = f(x, y)$ représente l'intensité au point M .

La modélisation de l'image analogique à partir de son support matériel s'obtient par une numérisation en espace et en fréquence.

- Discrétiser les coordonnées (x, y) , c'est échantillonner l'image.
- Discrétiser l'amplitude d'un point, c'est quantifier les niveaux de gris.

L'image analogique continue, $I = f(x, y)$, est alors approximée par des échantillons $\delta(x, y)$ (signaux de DIRAC) que l'on écrit dans un tableau de $N*N$ cases, sous la forme de valeurs discrètes $f(N, N)$ (Figure 2.8.).

$$I = \begin{pmatrix} f(0,0) & f(0,1) & \dots & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & & & f(1,N-1) \\ \dots & \dots & & & \dots \\ \dots & \dots & & & \dots \\ f(N-1,0) & \dots & & & f(N-1,N-1) \end{pmatrix}$$

Figure 2.8. Modèle de l'image numérisée

Le tableau comporte N^2 valeurs; chacune d'elle est appelée **pixel**¹, néologisme anglais qui vient de la contraction de l'expression « picture element ». La mémoire utile pour stocker un pixel peut varier de 1 bit (cas des images monochromes) à 24 bits (images en 16 millions de couleurs). Généralement, on utilise des images à 256 couleurs qui nécessitent 8 bits par pixel [14][19].

Il existe plusieurs formats d'images informatiques stockées sous formes de fichiers. Le stockage peut être avec une méthode de compression ou sans compression, ces fichiers se différencient en première vue par leur extension (*.ima, *.img, *.pcx, *.cpt, *.bmp, *.gif, *.jff, *.jpg, *.jtf, etc.....). Nous allons aborder dans les paragraphes qui suivent deux exemples, les fichiers images non compressés (format brut et format BMP) et les fichiers compressés (utilisant la norme JPEG) [12].

2.3.2. Les fichiers images non compressés

2.3.2.1. Le format brut

Les fichiers images de formats bruts sont généralement les fichiers d'extension *.ima ou *.img. Si on a l'image numérique, $I = f(i, j)$, qui est écrite dans un tableau de (*hauteur***largeur*) cases, sous la forme de la figure 2.9.

¹ Pour plus de détails, voir [14].

$$I = \begin{pmatrix} f(0,0) & f(0,1) & \dots & \dots & f(0,L-1) \\ f(1,0) & f(1,1) & & & f(1,L-1) \\ \vdots & \vdots & & & \vdots \\ f(H-1,0) & \vdots & & & f(H-1,L-1) \end{pmatrix}$$

$$0 \leq i \leq (L-1) ; 0 \leq j \leq (H-1)$$

Figure 2. 9. Modèle de l'image numérique

- H est la hauteur de l'image et L la largeur de l'image.

Le fichier, $F = d(x)$, qui représente le stockage de l'image I , sera écrit dans un tableau, unidimensionnel sous la forme de la figure 2. 10.

$$F = \begin{pmatrix} d(0) \\ d(1) \\ \vdots \\ d(T-2) \\ d(T-1) \end{pmatrix} \quad 0 \leq x \leq T-1$$

Figure 2. 10. Modèle du fichier d'image brut

- T représente la taille du fichier F .

Les valeurs $d(x)$ du fichier F remplissent la condition suivante :

$$d(i + L \cdot j) = f(i, j)$$

2.3.2.2. Le format BMP¹

Les fichiers BMP sont les types les plus utilisés pour les stockages des images informatiques non compressées couleurs ou noir et blanc. La structure du fichier BMP est donnée dans le paragraphe suivant.

¹ Le format BMP de Bitmap Windows.

Structure des fichiers BMP

Un fichier BMP comporte quatre parties distinctes qui sont :

- La première appelée « **BITMAPFILEHEADER** », fournit des informations générales sur le fichier ;
- La deuxième appelée « **BITMAPINFOHEADER** », contenant des informations sur l'image proprement dite ;
- La troisième représente la table des couleurs. Elle est constituée de plusieurs structures « **RGBQUAD** ».
- Et en fin, une dernière rassemblant le codage complet de chacun des pixels de l'image.

Les trois premières parties sont appelées le **HEADER**

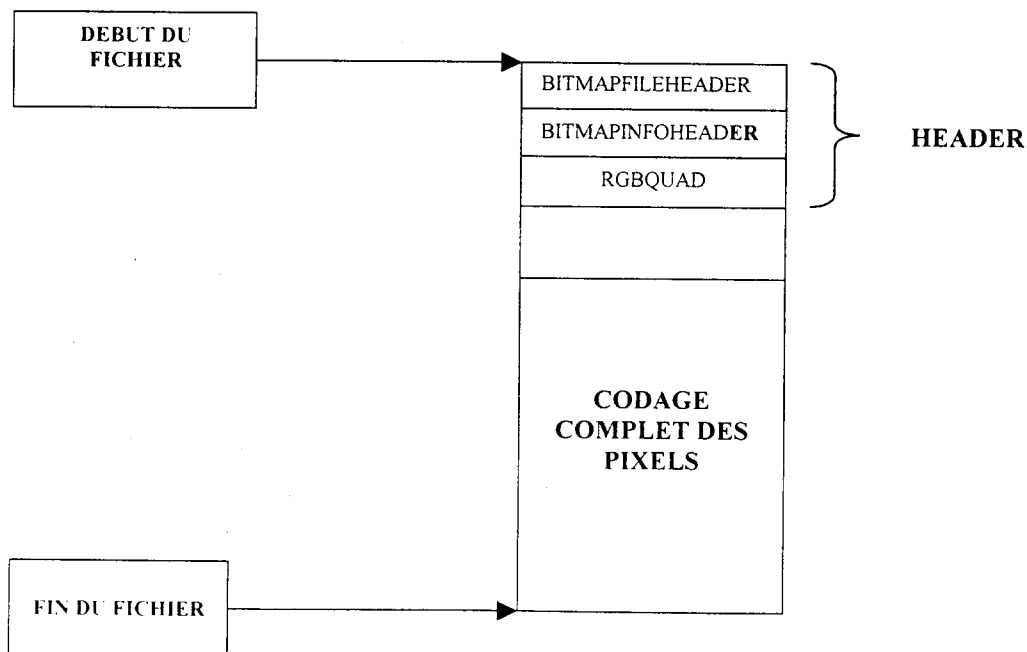


Figure 2.11. Fichier BMP

Si on garde la même image numérique, $I = f(i, j)$, qui est écrite dans un tableau de (*hauteur***largeur*) cases, sous la forme de la figure 06. Le fichier, $F_{bmp} = d(x)$, qui représente le stockage de l'image I avec le format BMP, sera écrit dans un tableau unidimensionnel sous la forme de la figure 2. 12.

$$Fbmp = \begin{pmatrix} d(0) \\ d(1) \\ \dots \\ \dots \\ d(K-1) \\ \dots \\ \dots \\ d(R) \\ \dots \\ \dots \\ d(T-1) \end{pmatrix} \quad 0 \leq x \leq T-1$$

Figure 2. 12. Modèle du fichier image BMP

- T représente la taille du fichier $Fbmp$.
- $(K-1)$ = taille de l'entête du fichier BMP

Pour le chargement de l'image I avec le format BMP, les valeurs $d(x)$ du fichier $Fbmp$ doivent remplir la condition suivante :

$$d((T-1) - L \cdot (j+1) + i) = f(i, j)$$

avec :

- $(T-1) - R = H \cdot L$

et

- H représente la hauteur de l'image et L la largeur de l'image.

Remarques:

- Chaque ligne de l'image contient un nombre d'octets multiple de quatre. Des octets nuls sont éventuellement ajoutés à la fin de la ligne.
- Un fichier BMP est chargé de telle façon que la première information de la zone de données (qui contient les valeurs des pixels) correspond au coin inférieur gauche de l'image.

2.3.3. La norme JPEG

2.3.3.1. Introduction

Le groupe JPEG (*Joint Photographic Experts Group*), créé au début des années 90 regroupe des industriels et universitaires concernés par la compression d'images. Son but est d'établir des spécifications portant sur la compression conservative et non-conservative d'images. Nous nous intéresserons ici uniquement à la compression non-conservative [12].

2.3.3.2. Principes de la norme du groupe JPEG

Ces spécifications se basent sur le fait que les informations pertinentes d'une image, caractérisée par son signal bidimensionnel $Img(x,y)$, sont uniquement les composantes de basse fréquence du signal. Le signal est transformé en un ensemble bidimensionnel de valeurs de fréquences grâce à la *transformée de cosinus discrète* (DCT) [12]. La précision des entiers représentant les valeurs de fréquences est alors diminuée pour rendre nulles les valeurs de hautes fréquences qui ne sont pas pertinentes. Cette étape, appelée "quantification" représente la partie non conservative de la compression. Les résultats sont alors comprimés de manière conservative en codant efficacement toutes les fréquences devenues nulles (voir résumé, figure 2.13.).

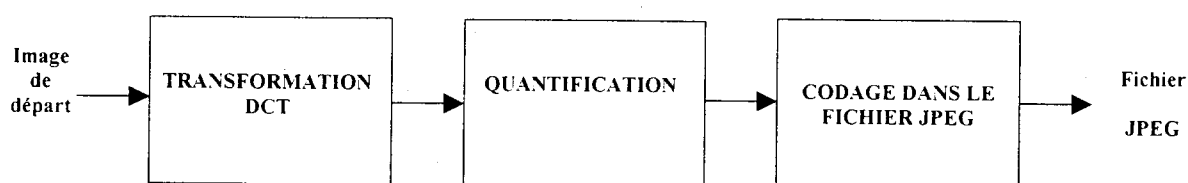


Figure. 2.13. Principe général de la compression JPEG

La décompression effectue les opérations inverses de celles de la compression. Elle commence par décompresser le résultat de la troisième étape de la compression. Elle obtient alors un tableau de valeurs de fréquences quantifiées dont beaucoup sont nulles. La quantification inverse est appliquée pour retrouver le tableau de valeurs de fréquences. Il s'agit d'une approximation du tableau résultat de la DCT dans la compression puisque les hautes fréquences non pertinentes auront 0 comme valeur. Il suffit alors d'appliquer la *transformée de cosinus discrète inverse* pour obtenir l'approximation du signal de l'image qui avait été comprimée.

La méthode de quantification détermine la qualité de l'image résultat. Elle ne fait pas partie de la spécification JPEG.

2.3.3.3. La transformée de cosinus discrète (DCT)

Pour faciliter la compréhension de cette transformée, nous allons décrire la transformée équivalente, la transformée de Fourier rapide (FFT, *Fast Fourier Transform*), travaillant sur un signal discret unidimensionnel.

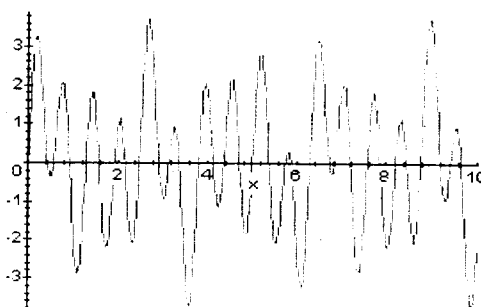


Figure. 2.14. *Signal avant FFT*

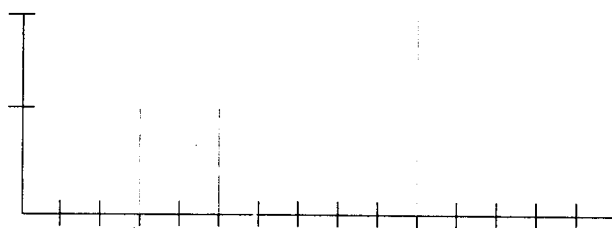


Figure. 2.15. *Signal traité par FFT*

Comme on le voit des figures 2.14. et 2.15., le gain de place est très important : au lieu de stocker la totalité de l'information, on ne conserve que la couple (fréquence-amplitude). Pour obtenir un gain de place optimal, on ne conserve que les composantes les plus significatives.

2.3.3.4. Mise en oeuvre de la méthode JPEG

Le calcul de la DCT ne peut pas se faire sur une image entière d'une part parce que cela générerait trop de calculs et d'autre part parce que le signal de l'image doit absolument être représenté par une matrice carrée. Dès lors, le groupe JPEG impose la décomposition de

l'image en blocs de 8 pixels sur 8 pixels. La méthode de compression sera donc appliquée indépendamment sur chacun des blocs. Les plus petits blocs en bordure devront être traités par une autre méthode.

Des recherches expérimentales poussées ont amené les concepteurs du JPEG à proposer la matrice de quantification définie de la façon suivante :

$$quantum(i,j) = 1 + ((1+i+j).qualité) / (i,j) \in [0,7]^2$$

Le codage du reste de la matrice DCT quantifiée va se faire en parcourant les éléments dans l'ordre imposé par une séquence particulière appelée *séquence zigzag* (voir figure 2.16.). Cette séquence a la propriété de parcourir les éléments en commençant par les basses fréquences et de traiter les fréquences de plus en plus hautes.

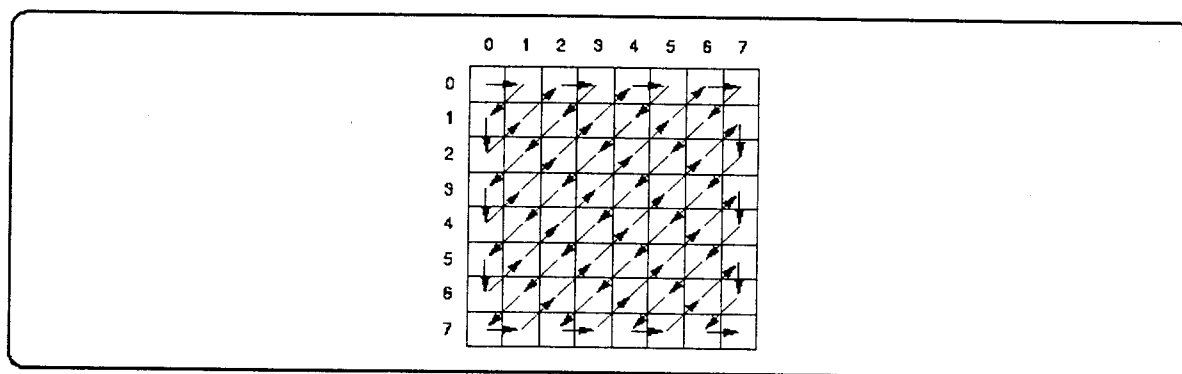


Figure 2.16. *Ordre de la séquence zigzag*

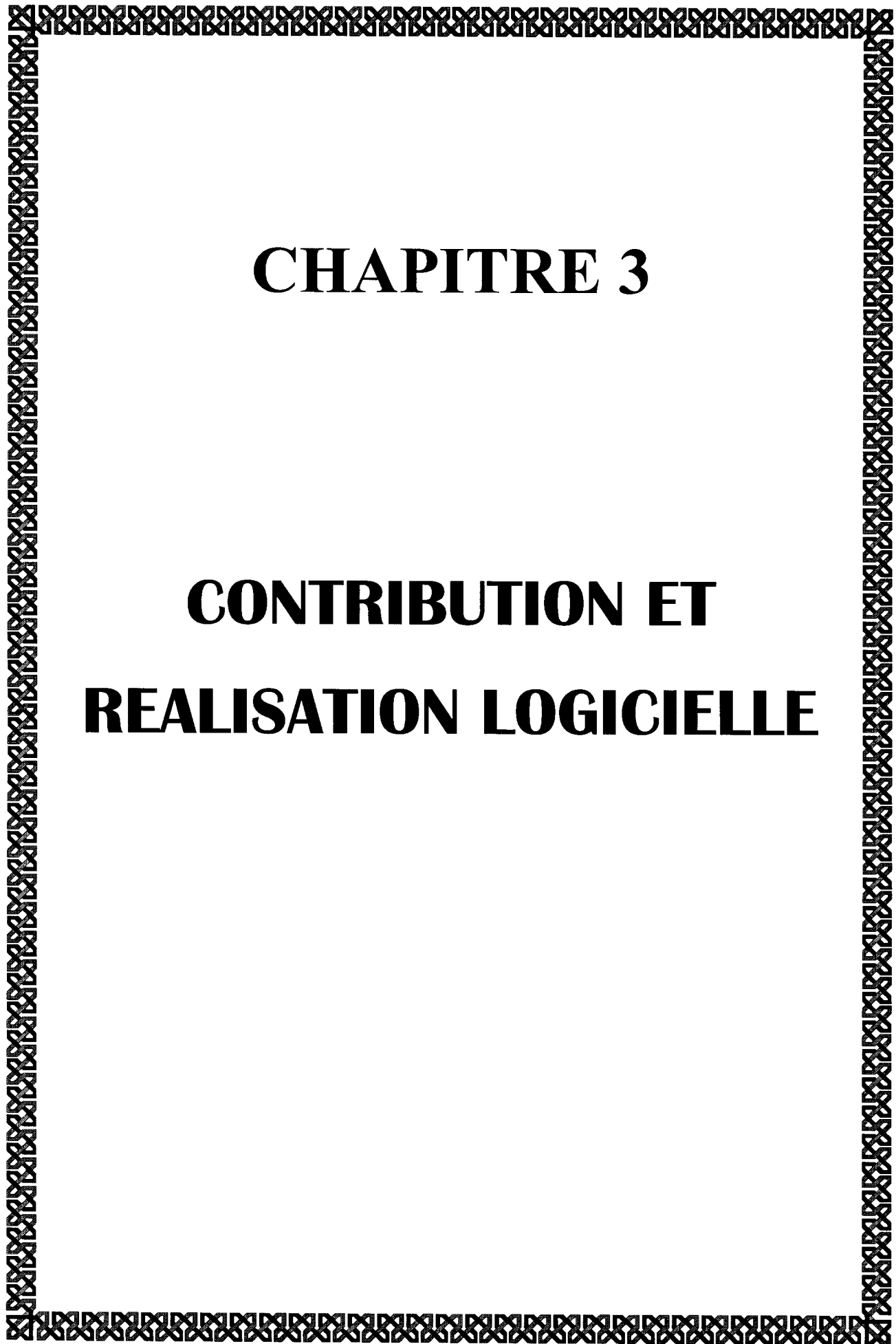
Il est très surprenant de voir la qualité de l'approximation des valeurs des pixels obtenue par décompression alors que la matrice DCT quantifiée (celle qui est réellement codée) ne semble plus contenir beaucoup d'informations. A l'heure actuelle la méthode de compression JPEG est parmi les plus utilisées parce qu'elle atteint des taux de compression très élevés sans que les modifications de l'image ne puissent être décelées par l'œil humain. De plus, beaucoup d'implémentations permettent de choisir la qualité de l'image comprimée grâce à l'utilisation de matrices de quantification paramétrables.

2.4. CONCLUSION

L'objectif ciblé est la conversion des images BMP 256 niveaux de gris (images numériques) pour les afficher sur un moniteur TV (la sortie du système est un signal vidéo composite avec la norme 625 lignes).

Pour un premier objectif nous avons :

- Utilisé des images BMP de 340*256 pixels et tronquer les images de tailles supérieures.
- Limité la bande passante du signal vidéo à 0.85 MHz (contraintes du au caractéristiques des RAM et du convertisseur D/A utilisées).



CHAPITRE 3

CONTRIBUTION ET REALISATION LOGICIELLE

CONTRIBUTION ET REALISATION LOGICIELLE

3.1. INTRODUCTION

Dans la grande panoplie des logiciels existants, les applications tiennent la plus importante place. Car un ordinateur n'est rien sans application. C'est pour cette raison qu'aucun système d'exploitation n'est livré sans au moins un petit logiciel de traitement de texte et de dessin. Le système d'exploitation (ou OS, pour "Operating System") est le composant logiciel le plus important de l'ordinateur, son rôle est d'assurer une interface entre le matériel, la machine, et l'humain. Il est constitué d'un ensemble d'instructions, de programmes préconçus et compréhensibles par le matériel, dont l'utilisateur pourra se servir pour exécuter certaines tâches avec l'ordinateur : lancer un programme, lire un document, gérer ses fichiers, etc. Il est aussi chargé de gérer une partie du matériel.

Le système d'exploitation **Windows** est un environnement graphique multitâche où s'exécutent des programmes conçus pour lui. Les applications MS-DOS fonctionnent avec Windows, mais le cas inverse est impossible. Pour cela nous avons réalisé la première version de notre programme sous DOS et la deuxième sous Windows.

Il est remarquable de dire que, les techniques matérielles développées ne font pas toujours l'objet des publications car, ces techniques entrent dans les secrets commerciaux et industriels, et pour grouper tous les détails d'un sujet hard ça nécessite un effort personnel considérable et plusieurs tests matériels.

Nous avons cité dans le premier chapitre que la fixation du décodage des périphériques¹ à 12 bits et de la zone des cartes prototypes (300-31F) [1] n'offre que 32 adresses seulement pour réaliser les tâches des cartes prototypes. Les paragraphes qui vont suivre présentent notre contribution à l'élargissement de la capacité d'adressage et la partie logicielle réalisée.

3.2. CONTRIBUTION A LA MULTIPLICATION D'ADRESSES

Pour remédier à l'inconvénient de la limite de la zone d'adressages des cartes prototypes, nous avons proposé une architecture mixte logicielle/matérielle qui réalise un adressage physique étendu². La figure 3.1. montre le synoptique de base de la partie matérielle. La figure 3.2. représente le principe de la procédure logicielle. Dans ce système le bus de données du bus d'extension assure la canalisation des données proprement dites, et la canalisation des adresses destinées à être validées sur le nouveau bus.

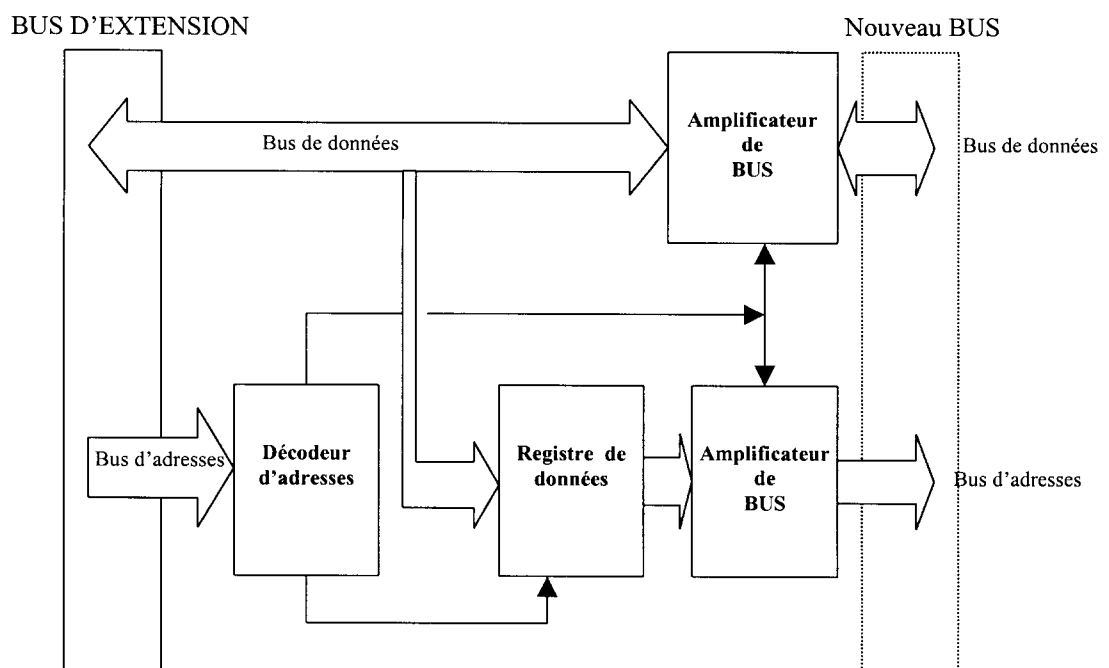


Figure 3.1. Synoptique de la multiplication d'adresses

¹ 16 bits avec le bus PCI.

² C'est une technique d'adressage matériel que nous avons développée.

Le mécanisme de cette technique passe par deux phases. Dans la première phase le logiciel assure la présence des valeurs des adresses destinées à être présentes sur le nouveau bus, comme des données validées sur le bus système à l'adresse **adr1** - cette adresse doit être comprise entre 300H et 31FH -

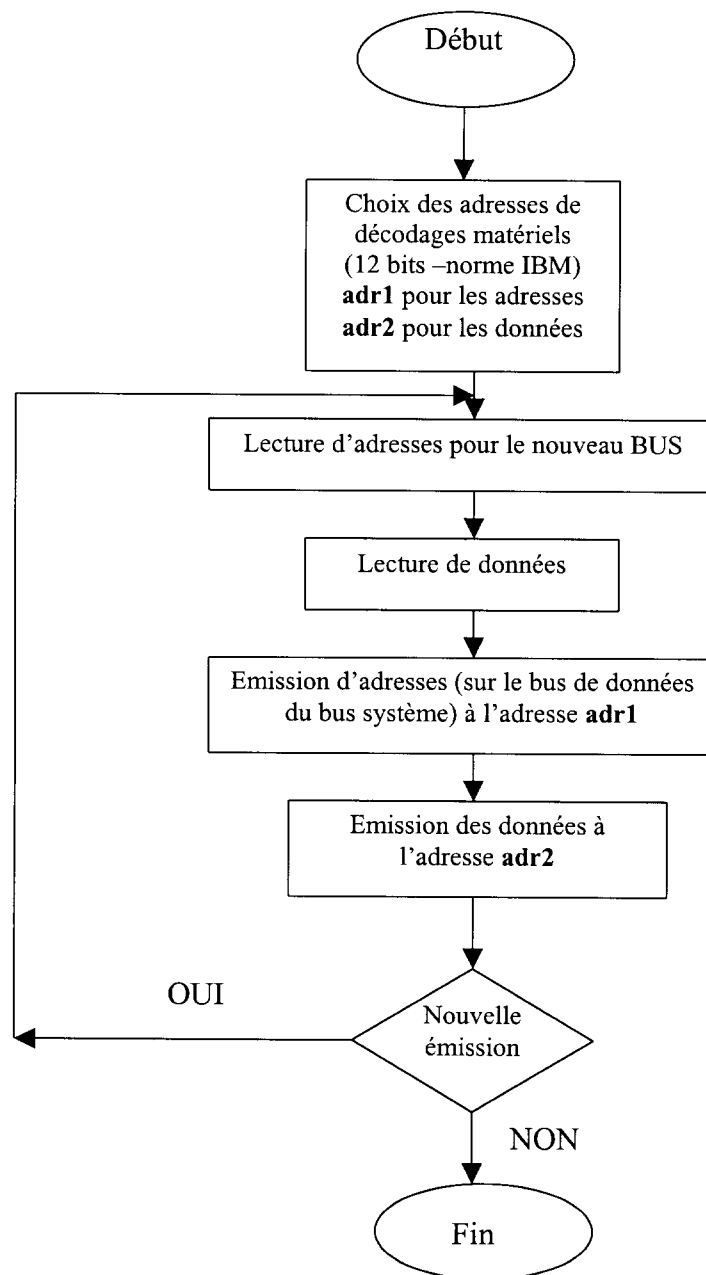


Figure 3.2. Principe de la procédure logicielle. .

Le décodeur d'adresses est conçu pour réaliser deux tâches la première est d'actionner le circuit registre afin de mémoriser le contenu du bus de données et de le faire

véhiculer à l'entrée de l'amplificateur de bus jusqu'à la prochaine validation de **adr1**. Durant cette phase le bus d'adresse du nouveau bus est à l'état haute impédance car l'amplificateur de bus n'est pas validé, cette validation sera faite dans la deuxième phase par l'autre commande du circuit de décodage.

La deuxième phase débute par la présence des valeurs des données destinées à être présentes sur le nouveau bus, assurées par le logiciel. Ces données sont validées sur le bus système à l'adresse **adr2** (adresse doit être comprise entre 300H et 31FH mais **adr1** # **adr2**). La deuxième tâche réalisée par le décodeur consiste à valider les deux amplificateurs de bus si l'adresse **adr2** est validée sur le bus d'extension. Dans cette deuxième phase les valeurs des deux bus de données et d'adresses sont valides sur le nouveau bus.

Les avantages de cette technique sont très intéressants, nous citons parmi ces avantages :

- La capacité de codage matériel est très grande sur le nouveau bus vu que la largeur du deuxième bus d'adresse résulte du nombre de bits des données utilisées sur le bus d'extension. Si le deuxième bus dispose d'un adressage sur 16 bits donc, le codage d'adresses peut varier de 0 à 2^{16} et le PC peut commander 2^{16} cartes.
- Avec cette méthode le processeur du PC peut accéder à l'espace mémoire d'un processeur auxiliaire, si ce dernier utilise l'espace adressable du nouveau bus dans l'architecture des multiprocesseurs.
- Cette méthode offre une vitesse de transmission de données très grande.

L'application proposée dans notre recherche est d'utiliser cette méthode pour réaliser la visualisation des images PC sur un moniteur TV. L'application est basée sur le deuxième avantage¹.

3.3. LE LOGICIEL

Le logiciel que nous avons réalisé TVSEND est un logiciel qui assure la transmission des images BMP 256 niveaux de gris vers notre carte connectée au slot

¹ Nous avons adressé tout l'espace mémoire de notre carte -pendant l'écriture- par la technique proposée.

d'extension ISA (8 bits) et il est développé sous un environnement C. Le choix de l'environnement C est dû aux plusieurs caractéristiques et avantages qu'il présente.

Malgré ses avantages et dans l'absence de l'environnement C¹ sur le PC, une application générée par le compilateur C risque de présenter quelques mauvais fonctionnements. Si le programme désire appeler des fonctions spécifiques au mode graphique, il faut que les fichiers qui décrivent la carte graphique soient présents. Dans notre cas les fichiers sont : le fichier EGAVGA.bgi et les fichiers *.chr (nous avons utilisé trois fichiers *.chr). Pour remédier à ce problème, nous avons intégré ces fichiers dans le corps exécutable de l'application et nous avons conçu cette dernière de telle manière qu'elle assure l'extraction de ces fichiers. Les étapes suivies pour cette solution sont :

- Génération de l'exécutable avec une compilation C.
- Création d'un nouveau fichier en copiant toutes les données du fichier exécutable.
- Intégration des fichiers dans le nouveau fichier.
- Modifications des paramètres de l'en-tête du nouveau fichier.

Le matériel sur lequel nous avons testé notre réalisation est:

- Un PC 486DX2- 66 M Hz; 4 Mo de RAM; 340 Mo de DD; carte graphique SVGA; écran 14'' SVGA.
- Un PC **586** 133 M Hz ; 12 Mo de RAM ; 640 Mo de DD ; carte graphique SVGA.
- Un PC **Pentium MMX** 166 M Hz ; 16 Mo de RAM ; 2.1 Go de DD ; carte graphique SVGA.
- Un PC **M II** 300 M Hz ; 32 Mo de RAM ; 3.2 Go de DD ; carte graphique SVGA.
- Un PC **Pentium II** 300 M Hz ; 32 Mo de RAM ; 4.3 Go de DD ; carte graphique AGP.
- Un PC **Pentium II** (Celeron) 433 M Hz ; 32 Mo de RAM ; 8.4 Go de DD ; carte graphique AGP.
- Un PC **Pentium III** 450 M Hz ; 64 Mo de RAM ; 8.4 Go de DD ; carte graphique AGP.

¹ C ou C++ sous DOS (pour une application non Windows).

3.4. LES FICHIERS D'EXTENSION .EXE

Les programmes portant l'extension **.EXE** sont des programmes qui sont relogeables en mémoire. Le système d'exploitation peut charger un programme **EXE** à n'importe quelle adresse de segment [44]. Un programme EXE ne peut être exécuté que si toutes les données d'adresses absolues contenues dans le programme ont été au préalable converties à l'adresse de début de segment de programme. Le système assure cette tâche automatiquement. L'en-tête se trouve avant le programme du module EXE et il contient les informations nécessaires à la translation d'un programme EXE. Le tableau suivant décrit le début de la structure de l'en-tête EXE¹.

Déplacement	Type de données	Signification
00h	2 octets	Doivent contenir les deux octets 4Dh et 5Ah.
02h	Mot	Nombre d'octets dans la dernière page du fichier, la longueur d'une page étant de 512 octets.
04h	Mot	Taille totale du fichier (en-tête compris) en pages.
06h	Mot	Nombre d'entrées dans la table de translation.
08h	Mot	Taille de l'en-tête en paragraphes de 16 octets. Le dernier paragraphe est pris en compte, même s'il n'est pas complètement occupé. La taille de l'en-tête permet de définir le début du programme.
0Ah	Mot	Nombre minimum de paragraphes de 16 octets, qui doivent rester disponibles au-dessus du programme.
0Ch	Mot	Nombre maximal de paragraphes de 16 octets qui peuvent être occupés au-dessus du programme.

Figure 3.3. Structure de l'en-tête .EXE

¹ Exécutable DOS.

3.5. PROGRAMME REALISE

3.5.1. Structure générale du programme

La structure générale du fonctionnement du programme est présentée sur l'organigramme¹ de la figure 3.4.

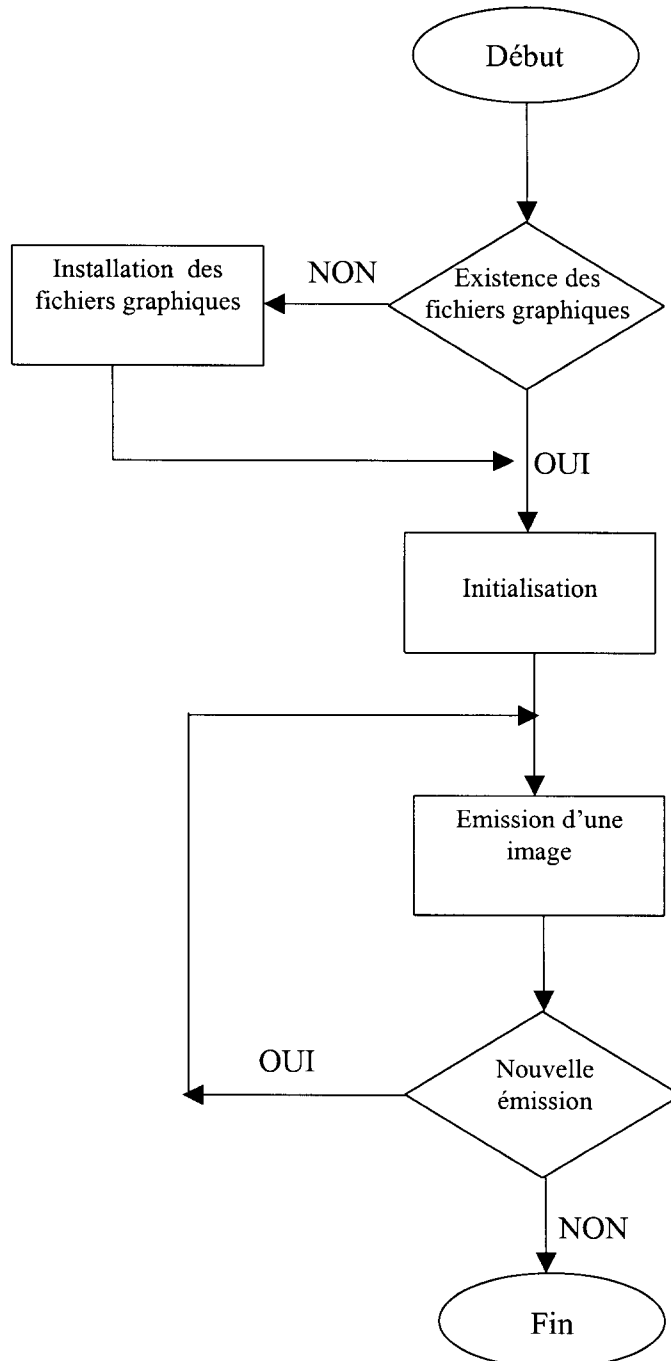


Figure 3.4. Structure générale du programme

¹ La version Windows du programme ne fait pas d'installation graphique.

Avant l'initialisation, le logiciel teste l'existence du fichier EGAVGA.bgi qui assure la configuration de la carte graphique en mode graphique et les fichiers *.chr pour les polices d'écriture. Si un de ces fichiers est inexistant, le logiciel l'installe sur le même chemin de l'application même si l'exécution se fait à partir d'une disquette.

Après l'initialisation, le programme ouvre une fenêtre en mode texte pour la spécification du chemin du fichier BMP et transmet l'image stockée dans ce fichier à notre carte (si la localisation de ce dernier est terminée avec réussite). Dans les deux cas le logiciel donne la possibilité d'émettre une nouvelle image avec la touche « o » du clavier suite au message affiché sur l'écran « **Voulez-vous émettre une nouvelle image (O/N)** ». Si la touche « n » du clavier est sélectionnée le message suivant s'affiche « **Frapper une touche pour sortir du programme** ».

3.5.2. Installation des fichiers graphiques

Les étapes de l'installation des fichiers graphiques sont présentées sur la figure 3.5.

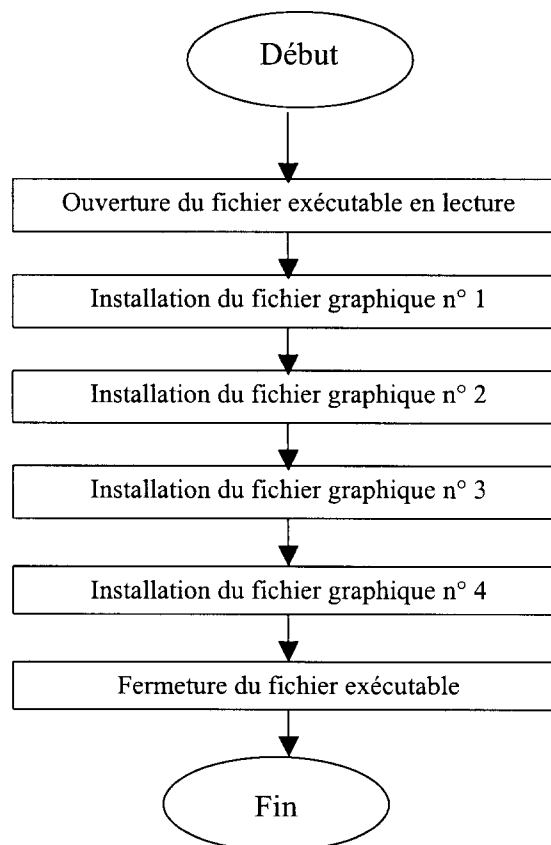


Figure 3.5. Installation des fichiers graphiques

La figure 3.6 illustre l'organigramme de l'installation d'un fichier graphique à partir du fichier exécutable.

L'installation consiste à copier les octets du fichier exécutable de la zone réservée au fichier n°x dans un nouveau fichier sur le lecteur de travail.

Le fichier n°1 : **EGAVGA.BGI**

Les fichiers n°2 à 4 : ***.CHR**

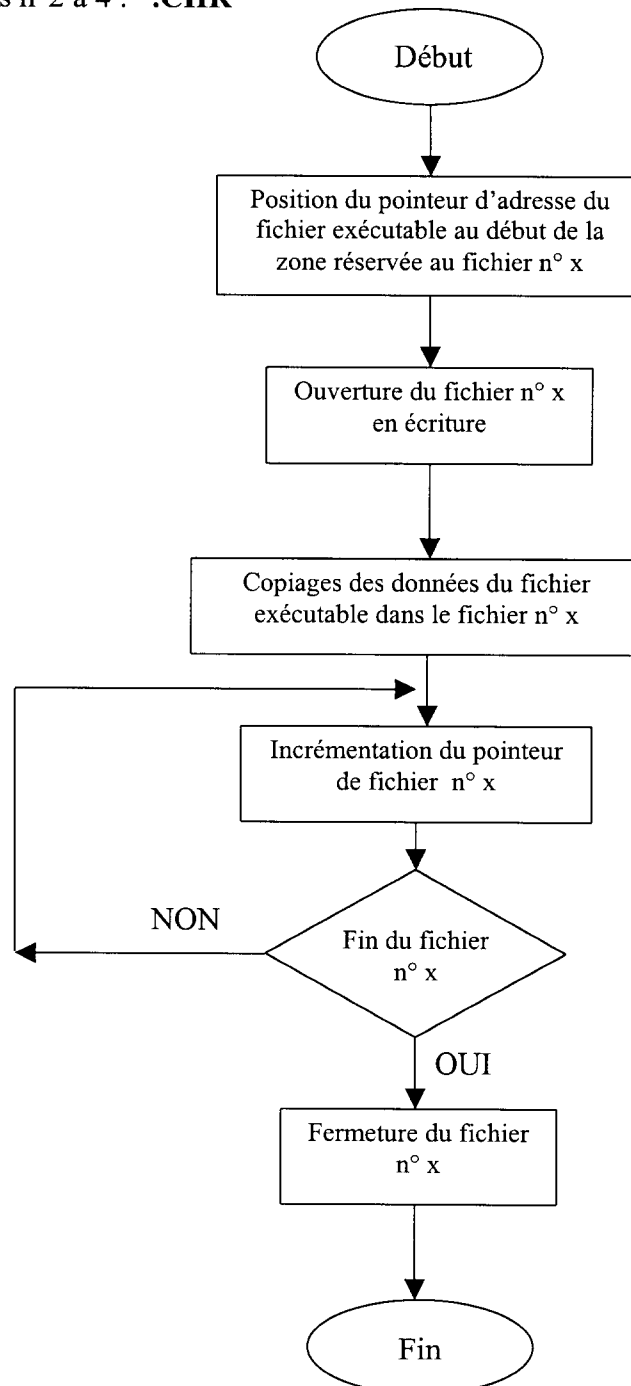


Figure 3.6. Installation d'un fichier à partir du fichier exécutable

3.5.3. Emission d' image

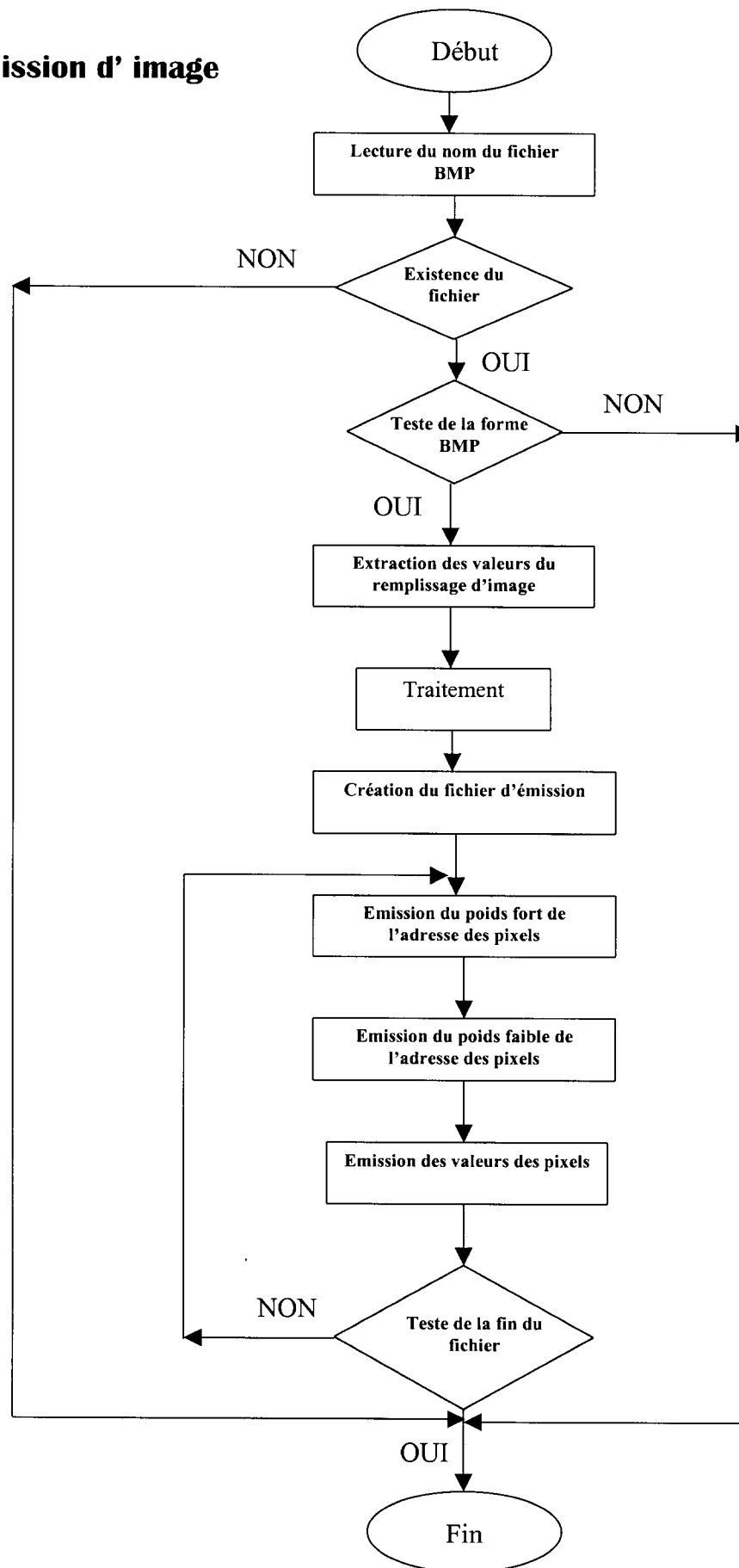


Figure 3.7. Synoptique de l'émission des fichiers BMP

Dès la validation du fichier BMP tapé sur l'écran, le programme teste l'existence et la forme de ce fichier, ensuite récupère la dimension et la position des valeurs des pixels qui forment l'image.

Les valeurs récupérées seront traitées afin de construire le fichier qui sera émis à notre carte. Avant d'émettre chaque octet à la carte, le logiciel envoie les deux octets du poids fort et faible de l'adresse d'écriture sur la mémoire de notre carte. A la fin de l'envoi de tous les octets du fichier d'émission le programme affiche le message suivant « **Le travail est accompli avec succès** ».

3.6. TRAITEMENT REALISE

3.6.1. Requantification

Nous avons vu au chapitre 2 la différence entre les images analogiques [5] (image TV) et les images numériques [14][19] et nous pouvons déduire qu'une conversion analogique-numérique ne suffit pas pour produire des images TV à partir des données numériques stockées dans les fichiers image.

La première différence des deux images est que l'une est une image entrelacée et la deuxième est progressive (non-entrelacé). La deuxième différence est : si par exemple l'image numérique est codée sur 8 bits (256 niveaux de gris), la valeur « 0 » représente le niveau noir et la valeur 255 représente le niveau blanc. Pour les images analogiques l'information est divisée en deux parties, la partie supérieure représente le niveau de gris et occupe 70 % et la partie inférieure représente les informations de synchronisation et occupe les 30 %.

Pour cela le programme accomplit des procédures de traitement pour que le convertisseur A/D de la carte génère à sa sortie, un signal vidéo composite.

L'image numérique, $I = f(i, j)$, est écrite dans un tableau de « hauteur*largeur » cases, sous la forme de la figure 3.8.

- $0 \leq i \leq (\text{largeur} - 1)$
- $0 \leq j \leq (\text{hauteur} - 1)$

$$I = \begin{pmatrix} f(0,0) & f(0,1) & \dots & \dots & f(0,L-1) \\ f(1,0) & f(1,1) & & & f(1,L-1) \\ \dots & \dots & & & \dots \\ \dots & \dots & & & \dots \\ f(H-1,0) & \dots & & & f(H-1,L-1) \end{pmatrix}$$

Figure 3.8. Modèle de l'image numérique

La première opération de traitement est la multiplication de la matrice I par un facteur A pour avoir une matrice I' [14][19].

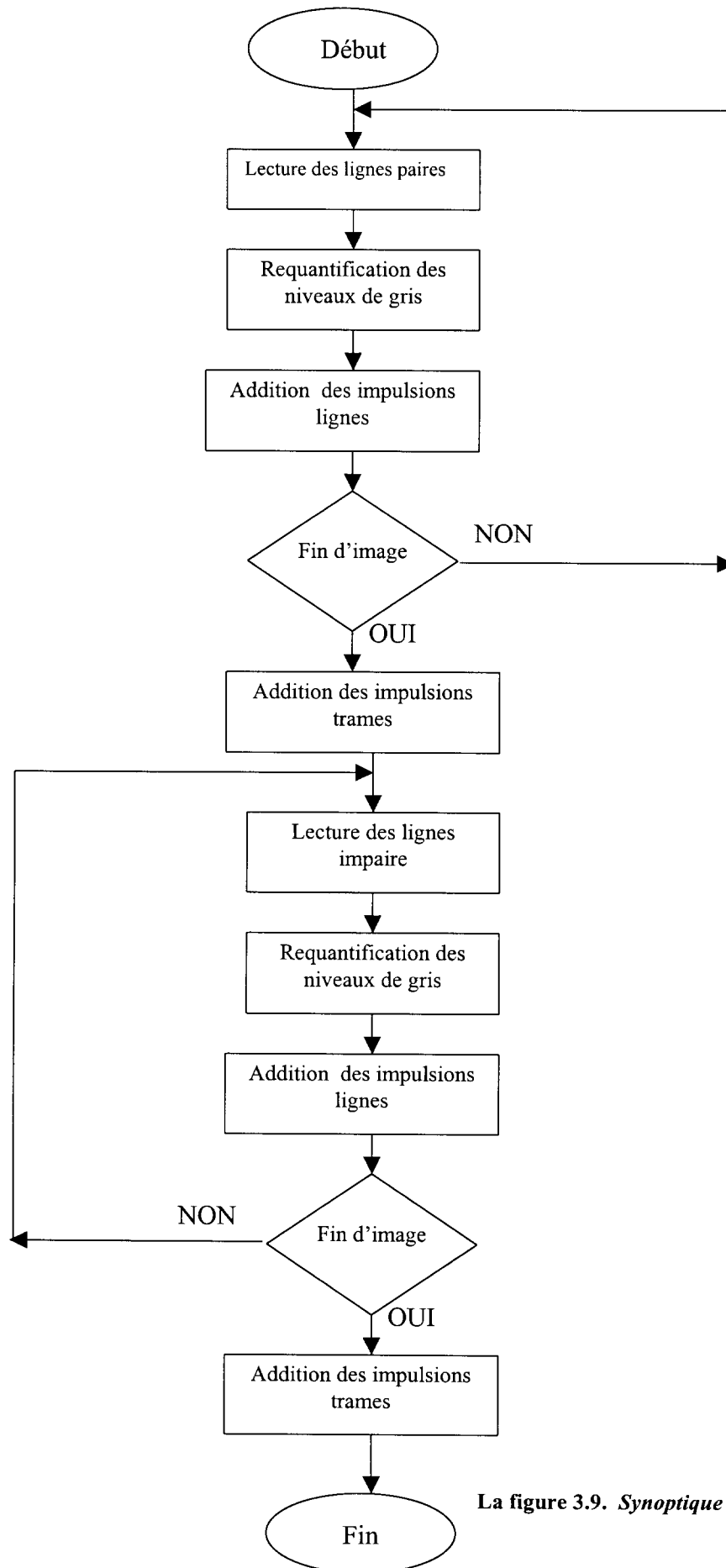
$$I' = A \times \begin{pmatrix} f(0,0) & f(0,1) & \dots & \dots & f(0,L-1) \\ f(1,0) & f(1,1) & & & f(1,L-1) \\ \dots & \dots & & & \dots \\ \dots & \dots & & & \dots \\ f(H-1,0) & \dots & & & f(H-1,L-1) \end{pmatrix}$$

- $A = \frac{7}{10}$

La deuxième opération de traitement est l'addition de la matrice I' avec la matrice B pour avoir une matrice I'' .

$$I'' = I' + \begin{pmatrix} b(0,0) & \dots & \dots & \dots & b(0,L-1) \\ \dots & \dots & & & \dots \\ \dots & \dots & & & \dots \\ \dots & \dots & & & \dots \\ b(H-1,0) & \dots & \dots & \dots & b(H-1,L-1) \end{pmatrix}$$

- $b(i, j) = 77$ (représente les 30 %)
- $i \in [0, \text{hauteur}]$
- $j \in [0, \text{largeur}]$
- $L = \text{largeur}$
- $H = \text{hauteur}$



La figure 3.9. Synoptique du traitement

La troisième opération de traitement est l'addition des impulsions lignes et des impulsions trames. La figure 3.9 représente le synoptique du traitement d'image effectué par le programme.

3.6.2. Changement de la définition horizontale

La bande passante de notre signal vidéo est limitée¹ à 0.85 M Hz à cause du temps de réponse des RAM utilisées ce qui donne un nombre de 104 points par ligne (information de synchronisation y compris). Dans ces 104 points nous avons seulement 85 points qui représentent l'information vidéo, ce qui ramène à diminuer la définition horizontale de 340 points à 85 points (une division par 4)[23][10].

La figure 3.10. représente le principe de ce traitement.

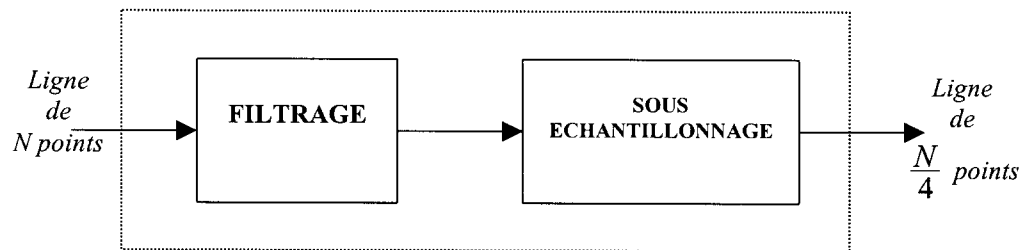


Figure 3.10. Principe du changement de la définition horizontale

3.6.3. Addition des impulsions lignes

Dans les images analogiques une ligne est précédée d'une impulsion de synchronisation (chapitre 2) qui représente 17 % à 18 % de la période de la ligne, ce qui impose l'addition des données représentatives des impulsions lignes aux fichiers images.

Si l_N (figure 3.11.) représente la ligne dans le fichier image de départ, l_A représente la ligne d'émission pour la ligne analogique.

¹ Pour générer un signal vidéo de bande passante supérieur, il faut utiliser des RAM et des convertisseurs plus rapides.

$$l_N = \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ \vdots \\ \vdots \\ d_{84} \\ d_{85} \end{pmatrix}$$

Figure 3.11. *Modèle de la ligne numérique*

$$l_A = l_{N'} + T_l$$

$$l_{N'} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ \dot{0} \\ d_0 \\ d_1 \\ \vdots \\ \vdots \\ d_{83} \\ d_{84} \end{pmatrix} \quad T_l = \begin{pmatrix} Top_0 \\ Top_1 \\ \vdots \\ \vdots \\ \vdots \\ Top_{18} \\ 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{pmatrix}$$

Figure 3.12. *Modèle de la ligne avec impulsion ligne*

Avec :

Top_i représentent la valeur de l'impulsion ligne

Le même principe est appliqué pour les impulsions trames, mais dans ce cas l'addition des impulsions trames est réalisée sur des images complète.

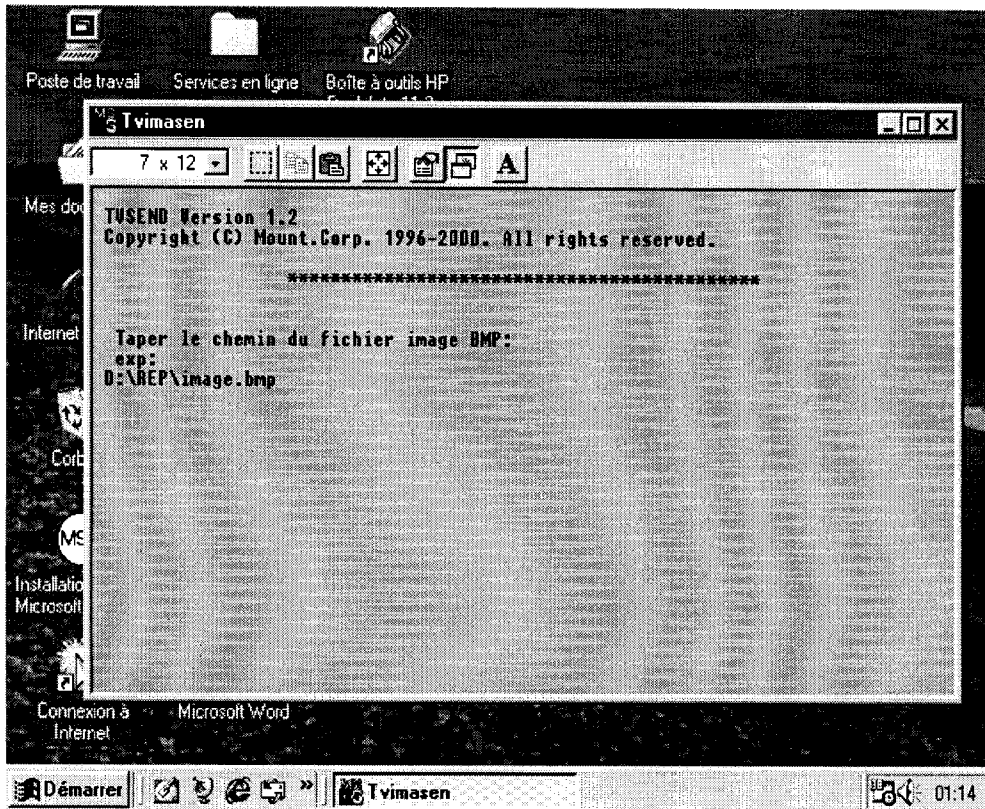


Figure 3.13. La fenêtre de dialogue.

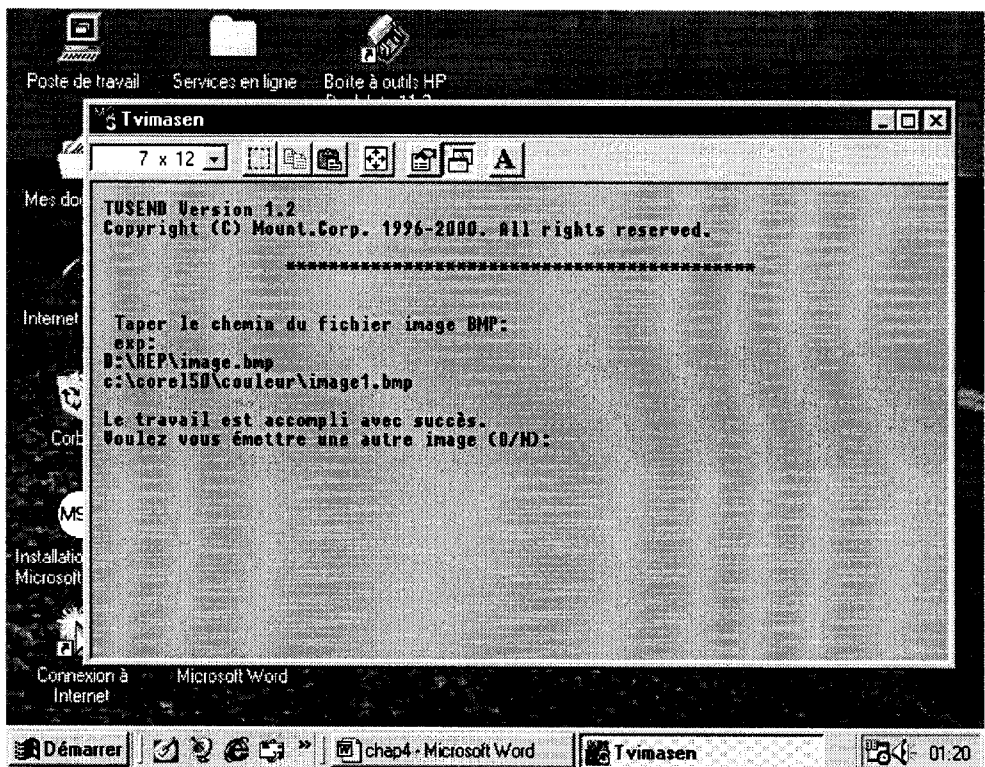


Figure 3.14. La fenêtre de dialogue après localisation et émission du fichier.

3.7. LE LOGICIEL SOUS WINDOWS

3.7.1. Introduction

Le logiciel que nous avons réalisé sous Windows est développé aussi sous un environnement C. La version Windows du logiciel assure le même traitement¹ que la version DOS, mais elle est basée sur le concept de la Programmation Orientée Objet (POO). Le programme réalisé avec POO n'est pas exécuté de façon séquentielle comme dans un environnement classique. Il s'agit de programmation « événementielle », des séquences de programme sont exécutées, suite à des actions de l'utilisateur (clique, touche enfoncée etc...), détectées par Windows.

Le menu de **TVsend** sous Windows présente plus de commandes que celui réalisé sous DOS. Il est plus convivial et plus interactif que ce dernier.

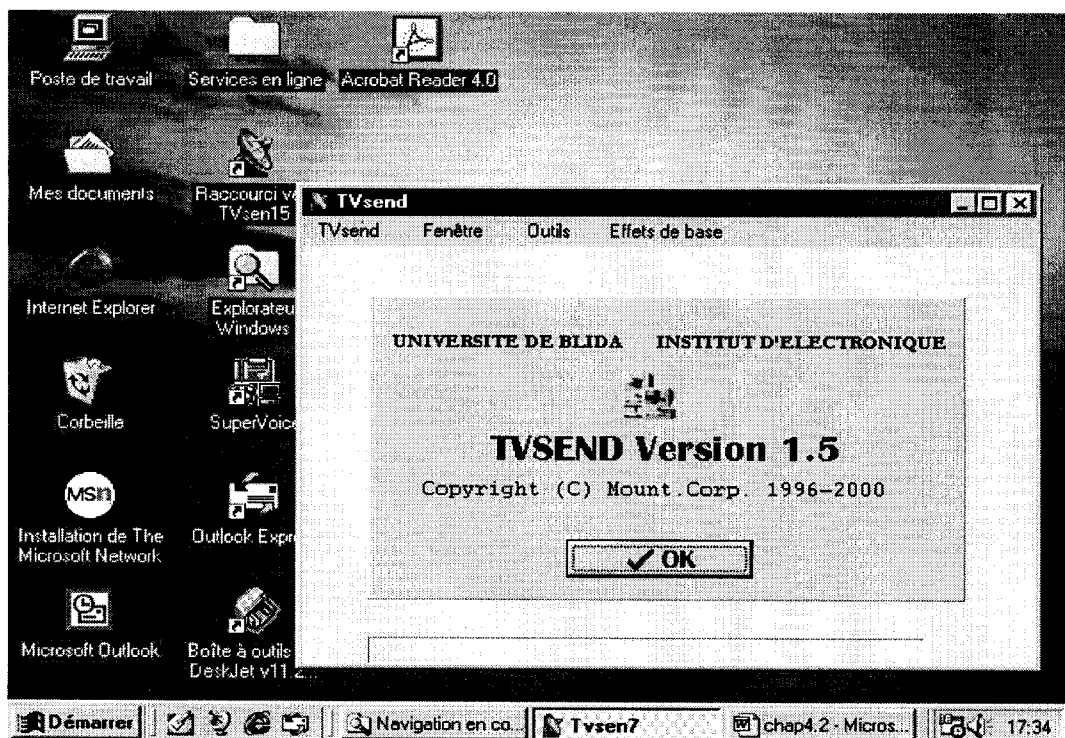


Figure 3.15. Fenêtre Principale

¹ La version Windows n'installe pas des fichier graphiques, car elle s'exécute sous une interface graphique.

3.7.2. Fonctionnement

L'exécution du logiciel fait apparaître la fenêtre de menu principal (Figure 3.15.). Dans ce menu, on trouve la fonction principale *TVsend* (Figure 3.16.) qui assure les tâches suivantes :

- Emission d'une image visible (Affichée par l'ouverture d'image).
- Emission d'une image sans passer par l'étape de la visualisation.
- Emission immédiate¹ de la dernière image.
- Fermeture de la fenêtre principale (Quitter).

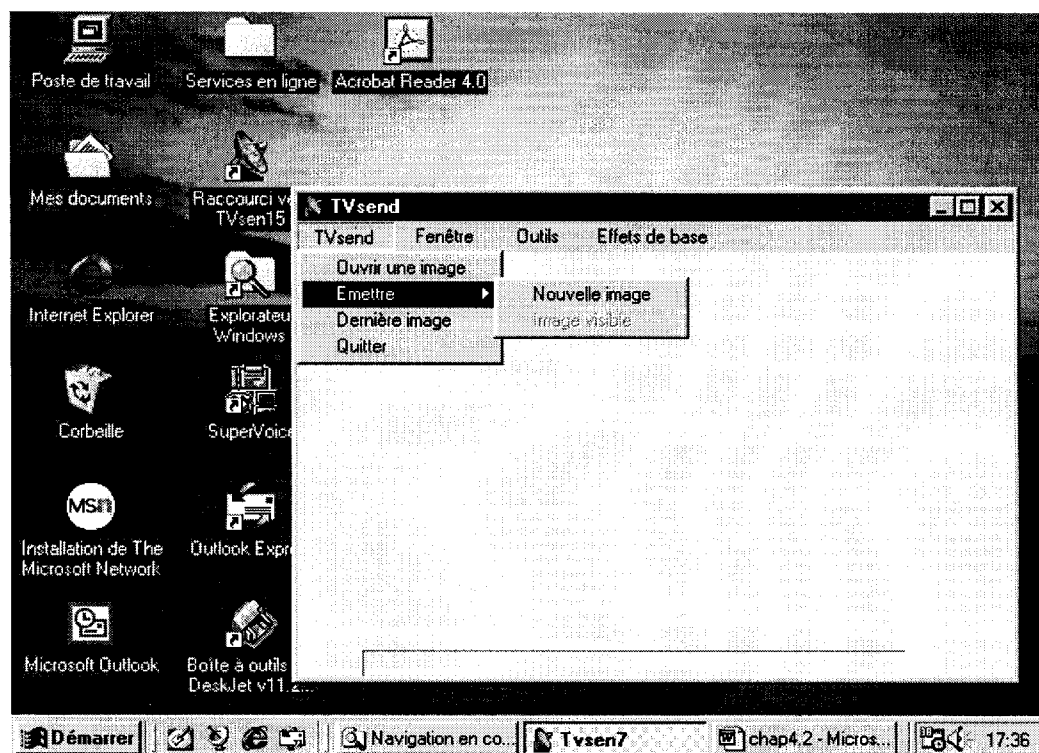


Figure 3.16. Fonction principale « TVsend »

Ouvrir une image :

Assure l'ouverture d'une boîte de dialogue pour la sélection du fichier image BMP (Figure 3.17.), qui va être affiché sur la fenêtre du menu principal.

¹ Sans la répétition du traitement.



Figure 3.17. Boite de dialogue pour ouverture d'image

La figure 3.18. présente la visualisation d'une image (image bureau) sur la fenêtre du menu principal.



Figure 3.18. Visualisation d'image sure la fenêtre principale

Emettre > Nouvelle image :

Assure l'ouverture d'une boîte de dialogue (Figure 3.19.) pour la sélection du fichier qui va être transmis sans passer par la visualisation sur la fenêtre du menu principal.

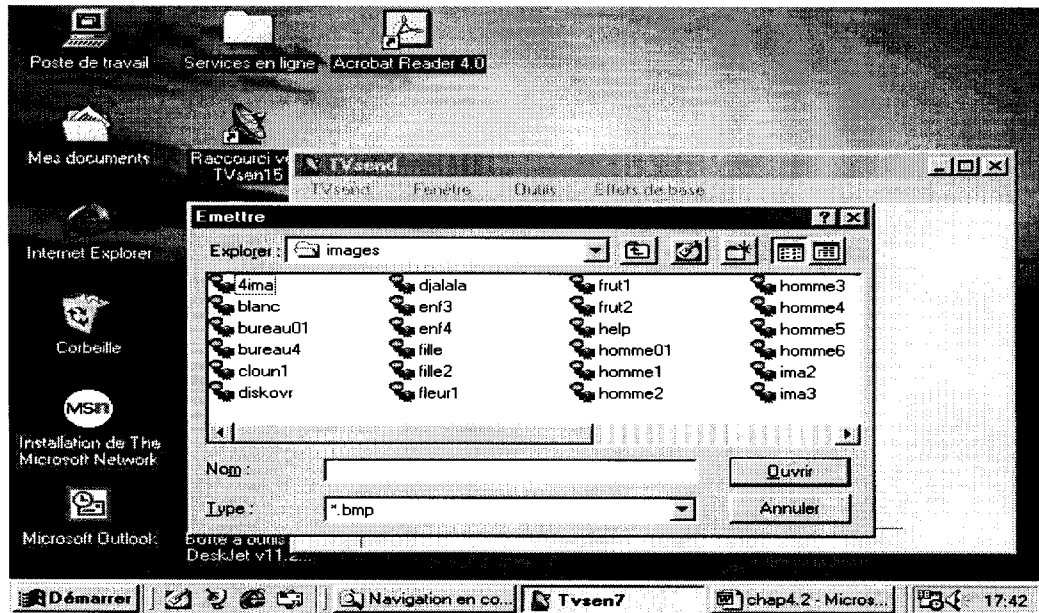


Figure 3.19. Boîte de dialogue pour une émission d'image sans passer par l'affichage

Emettre > Image visible :

Assure l'émission d'une image affichée sur la fenêtre du menu principal, voir figure 3.20.

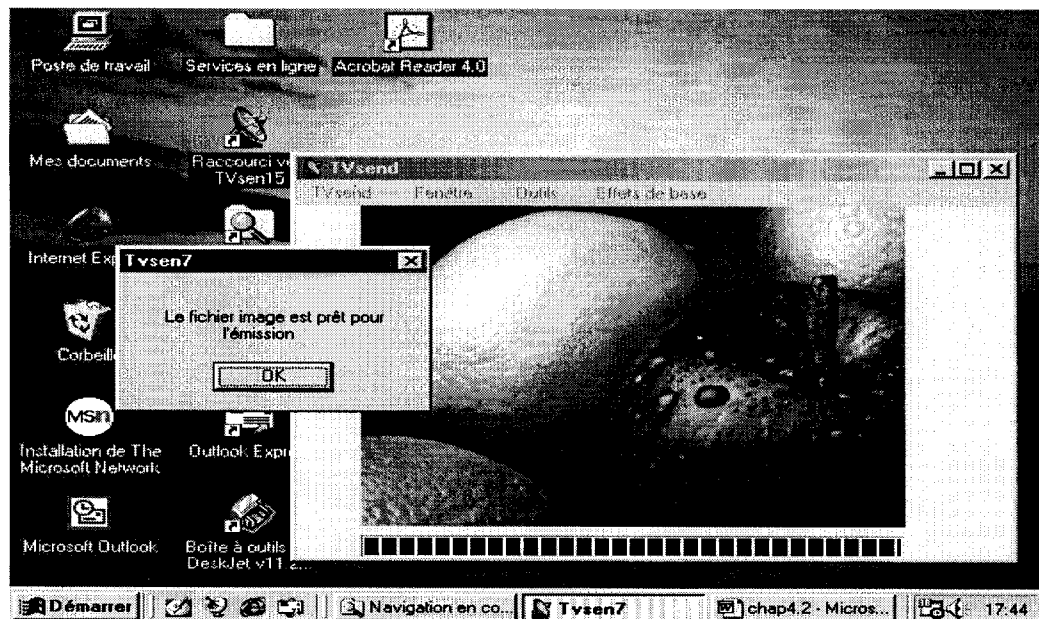


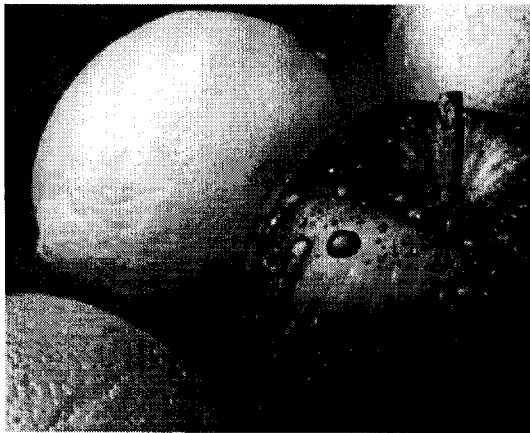
Figure 3.20. Emission d'une image visible

Effets de base :

L'exécution de cette option sur le menu principal assure les traitements d'images de base suivants :

- Variation de luminosité
- Variation de contraste
- Binarisation

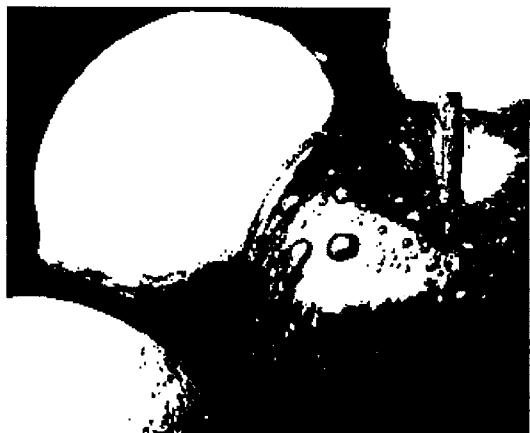
Binarisation



(a)



(b)



(c)

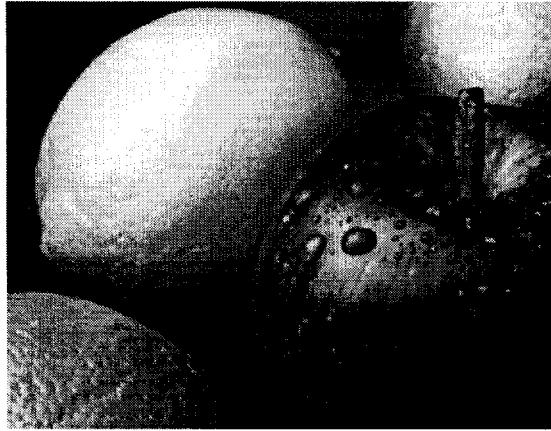


(d)

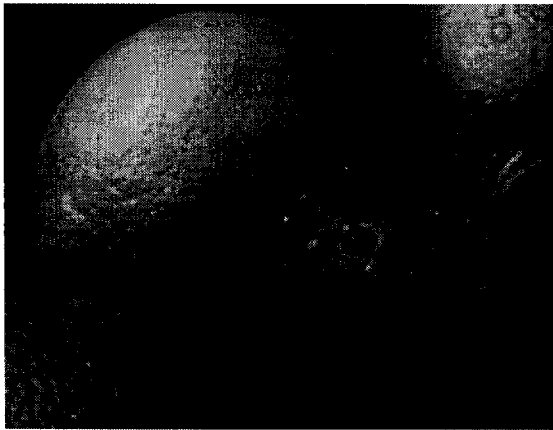
Figure 3. 21. Résultat de la binarisation. Figure (a) et (b) images 256 niveaux de gris. Figure (c) et (d) images binarisées.

Variation de luminosité

L'exécution de cette fonction, assure la variation de la luminosité.



(a)



(b)

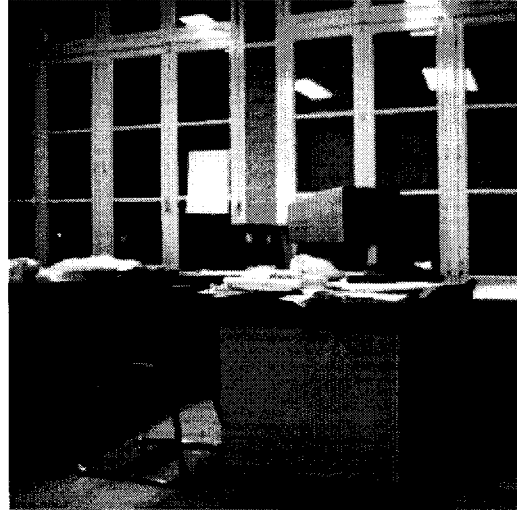


(c)

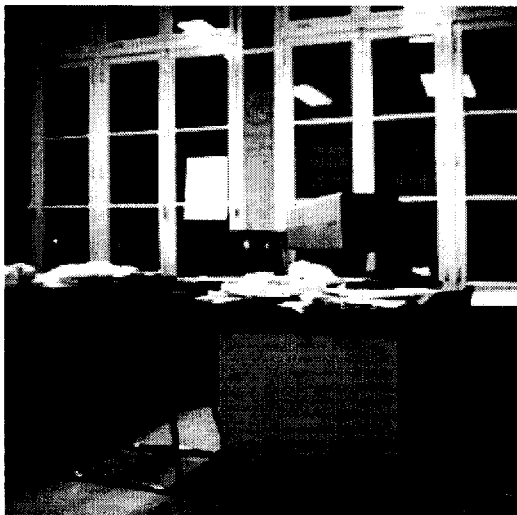
Figure 3. 22. Résultat de variation de luminosité. Figure (a) image de base. Figure (b) image avec diminution de luminosité. Figure (c) image avec augmentation de luminosité.

Variation de contraste

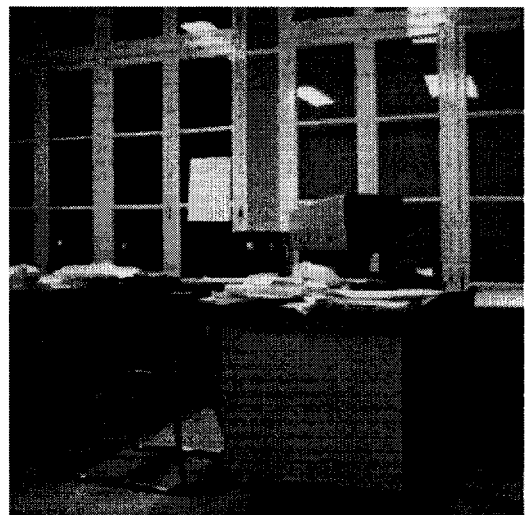
L'exécution de cette fonction, assure la variation du contraste.



(a)



(b)



(c)

Figure 3. 23. Figure (a) image de départ. Figure (b) image avec plus de contraste. Figure (c) image avec réduction de contraste.

3.8. CONCLUSION

Dans la partie logicielle de notre travail, nous avons réalisé un programme (une version sous DOS et une version sous Windows) qui assure :

- La recherche et le test des fichiers BMP tapés sur la fenêtre de dialogue.
- Le traitement de ces fichiers, pour les adapter à l'émission vers notre carte sur les adresses 300H, 301H et 303H.
- L'émission der fichiers.

Pendant les tests d'émissions des fichiers images (la taille des fichiers est 65000 octets) sur le bus ISA 8 bits, nous sommes arrivés à une vitesse d'émission supérieure à 64 K octets/s, ce qui donne une perspective à une émission d'images dynamiques, avec l'utilisation d'un codage inspiré de la norme MPEG. Le principe est d'émettre la totalité de la première image dans la première phase, et dans la deuxième phase, on ne transmet que les points qui changent de valeurs.

CHAPITRE 4

REALISATION MATERIELLE

REALISATION MATERIELLE

4.1. INTRODUCTION

L'interfaçage des PC avec les périphériques d'entrées-sorties est un sujet très important de la micro informatique. Le but de notre réalisation est de toucher deux points dans ce domaine : le premier et d'ajouter un périphérique d'affichage ; un moniteur TV classique. Le deuxième but de cette réalisation est de concevoir un système de communication qui a pour rôle, le perfectionnement de l'interfaçage avec les cartes à microprocesseur et de donner aux PC-IBM et compatibles un outil¹, pour commander plusieurs cartes d'interfaçages par un seul PC.

Nous présentons dans ce chapitre la partie matérielle de notre conception qui réalise l'affichage des images de 256 niveaux de gris sur un TV. La carte réalisée assure une gestion autonome de la mémoire d'affichage, après la transmission des données.

Cette carte est composée de trois parties :

- Module d'interface.
- Module de stockage et conversion A/D²
- Module de lecture d'image

Le module d'interface est basé sur la technique exposée au début du chapitre 3, et utilise un bus de données à 8 bits pour construire une adresse de 16 bits du nouveau bus. La mémoire utilisée dans le circuit de stockage est une RAM³ de 64 Ko et la capacité du convertisseur numérique analogique est 8 bits.

¹ Le système d'adressage proposé (voir chapitre 3).

² Analogique numérique.

³ RAM statique.

4.2. SCHEMA SYNOPTIQUE DE LA CHAÎNE DE VISUALISATION

La chaîne de visualisation que nous proposons est décrite par la figure 4.1. Elle est constituée de deux écrans de visualisation, une unité système ayant une carte graphique et notre carte d'extension et, bien sûr, un clavier.

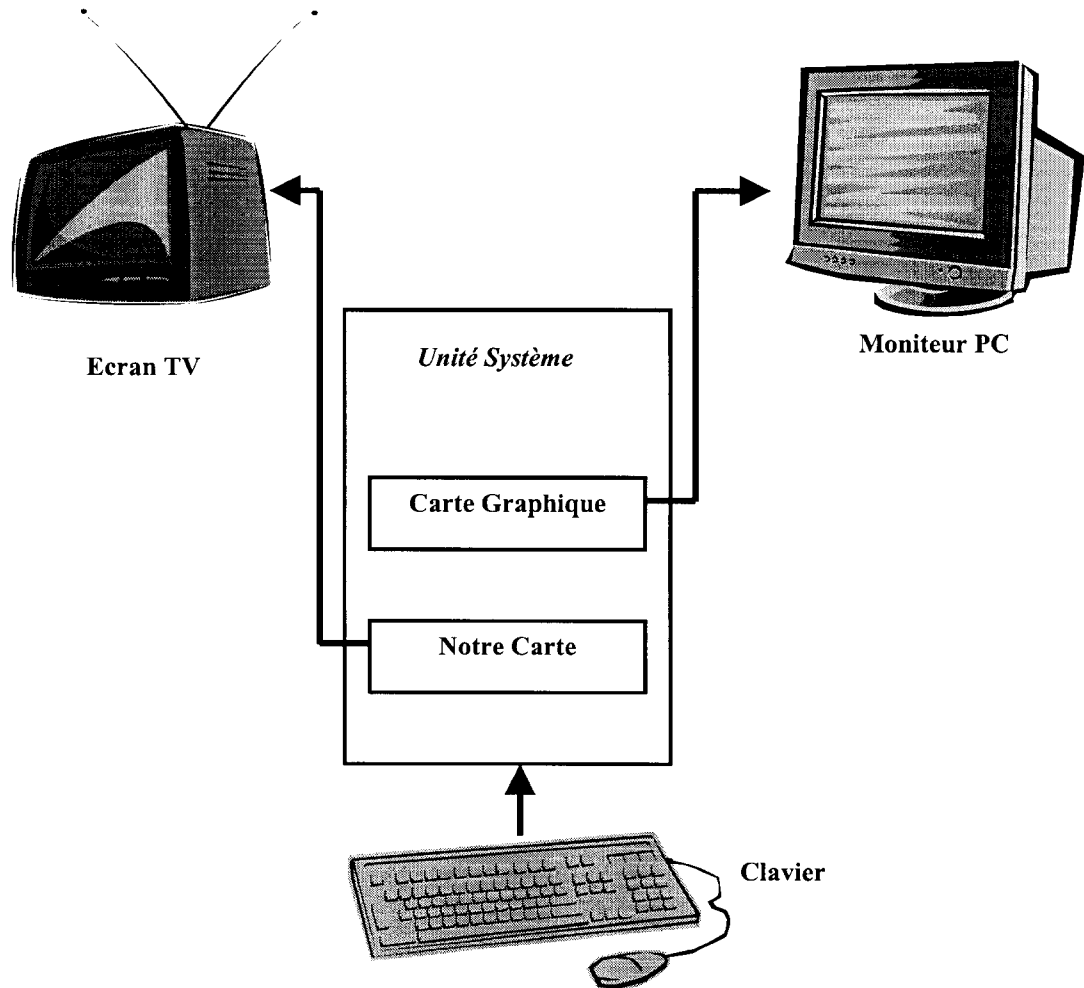


Figure 4.1. La chaîne de visualisation

Le premier écran qui est le moniteur classique du PC nous permet l'accomplissement des tâches habituelles sur micro-ordinateur ; il reçoit ses signaux à partir de la carte graphique. Quant au deuxième écran TV, il nous permet la visualisation des images traitées et converties du numérique à l'analogique¹. Cet écran est commandé par notre carte.

¹ Signal vidéo composite adapté à 75 Ω .

4.3. PRINCIPE DE L'ARCHITECTURE DE LA CARTE

Le synoptique globale de la carte proposée qui réalise l'affichage sur un moniteur TV est illustré par la figure 4.2.

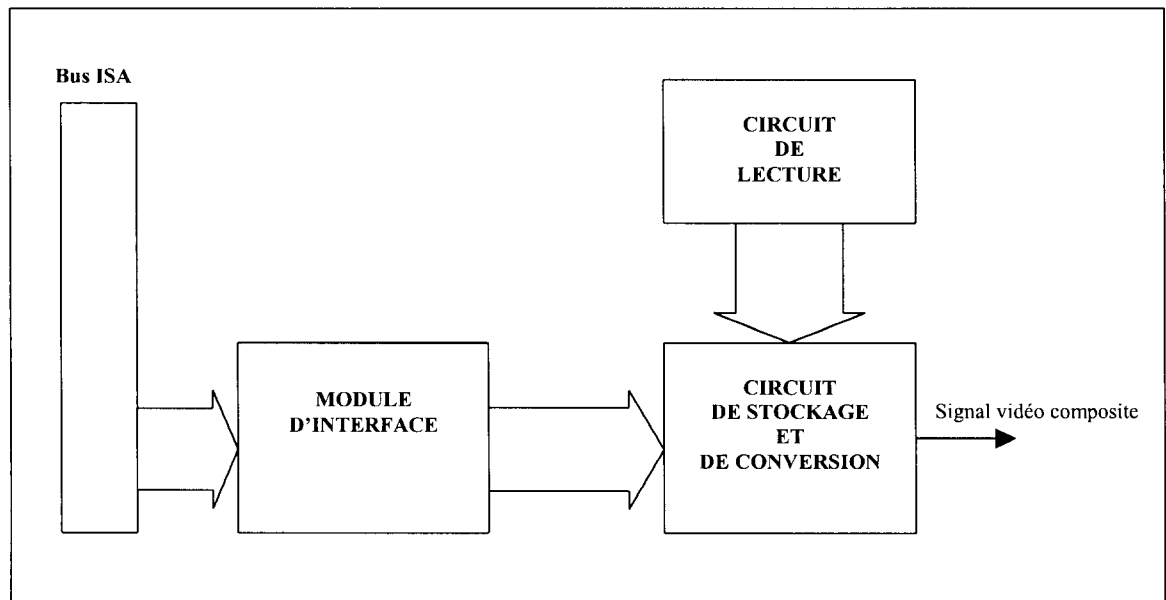


Figure 4. 2. *Synoptique globale de la carte*

Le module de l'interface est relié directement au bus d'extension ISA de l'IBM PC [20] (8bits pour avoir une compatibilité avec tous les PC IBM et compatibles) et assure l'écriture sur la mémoire¹ vidéo du circuit de stockage et conversion.

Le module de lecture est basé sur l'utilisation d'un compteur d'adresses 16 bits avec une base de temps² qui assure la génération d'un signal vidéo composite avec la norme 625 lignes.

Le module de stockage et conversion assure la mémorisation et la conversion analogique à fin de produire le signal vidéo composite.

¹ RAM statique de notre carte.

² 1.625 MHz.

4.4. MODULE D'INTERFACE

Dans le module d'interface, la structure adoptée pour adresser la mémoire de notre carte d'extension est illustrée sur la figure 4.3.

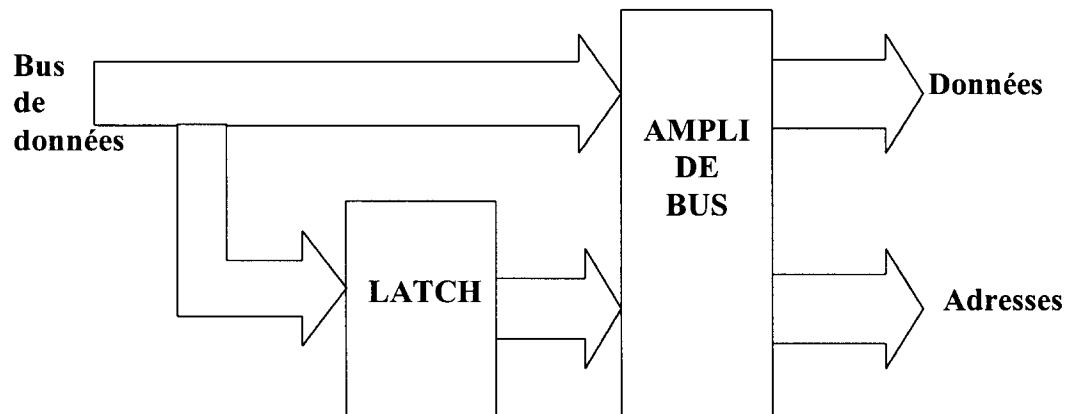


Figure 4. 3. Principe d'adressage de la RAM en lecture

Le système d'adressage est composé d'un 'latch' et d'un amplificateur de bus. Le bus de données du bus d'extension constitue l'entrée du système. La sortie est composée de bus de données et celui des adresses. Les deux bus de sortie commande la RAM de notre carte pendant le cycle d'écriture.

Le principe de fonctionnement de l'interface est le même que celui de l'Adressage Physique Étendu. La transmission des données du PC vers la carte d'extension passe par deux étapes. Premièrement et avant d'émettre chaque donnée du fichier image, on transmet la position (adresse) de cette donnée (sur le plan mémoire de notre carte) par le bus de données. Deuxièmement, on transmet les données du fichier, et à ce moment l'amplificateur de bus sera activé pour valider les deux bus de sorties.

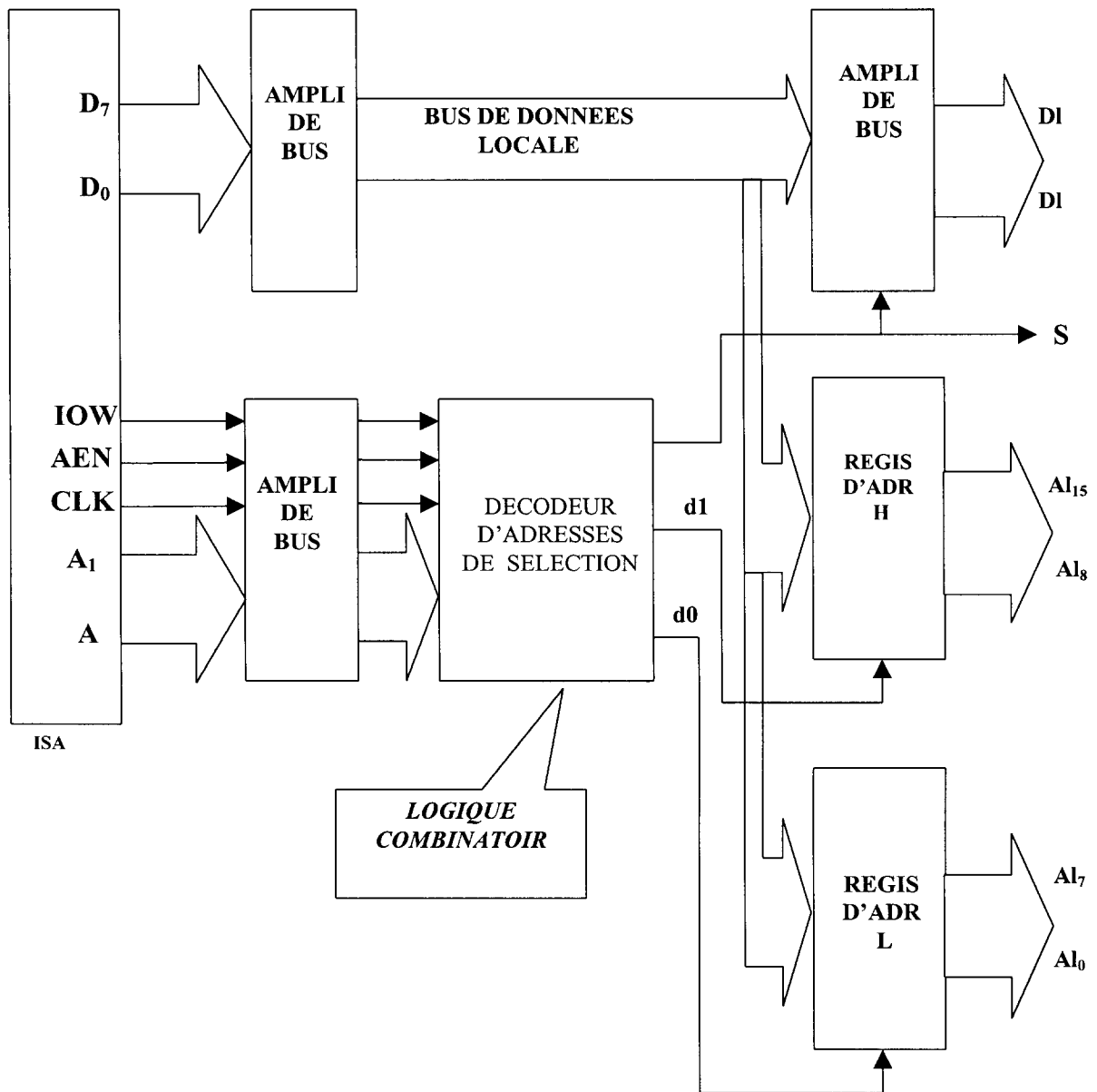


Figure 4. 4. Synoptique du module de l'interface

Le rôle du module est l'interfaçage comme son nom l'indique, et la canalisation des données récupérées sur le bus ISA¹ vers la mémoire RAM de la carte. La figure 4.4 illustre le synoptique du module.

¹ Bus ISA 8 bits.

Les amplificateurs de bus reliés aux lignes (D0-D7, IOW, AEN, CLK et A11-A0) sont des circuits d'isolements. Le décodage des adresses de la carte est effectué avec le décodeur d'adresses conçu avec la logique combinatoire. Les lignes d1 et d0 valident le chargement des deux registres H et L qui gardent leurs contenus jusqu'à l'arrivée des valeurs des données sur le bus de données. Le signal de la ligne S valide l'amplificateur de bus (D17-D10) et assure l'écriture sur la mémoire de la carte.

4.5. MODULE DE STOCKAGE ET CONVERSION

Le module de stockage et de conversion sert, d'une part au stockage sur la RAM des données binaires qui représentent l'intensité numérique du signal vidéo. D'autre part, assure la conversion analogique à l'aide d'un convertisseur numérique analogique. La figure 4.5 représente les éléments essentiels qui constituent l'unité de stockage et de conversion.

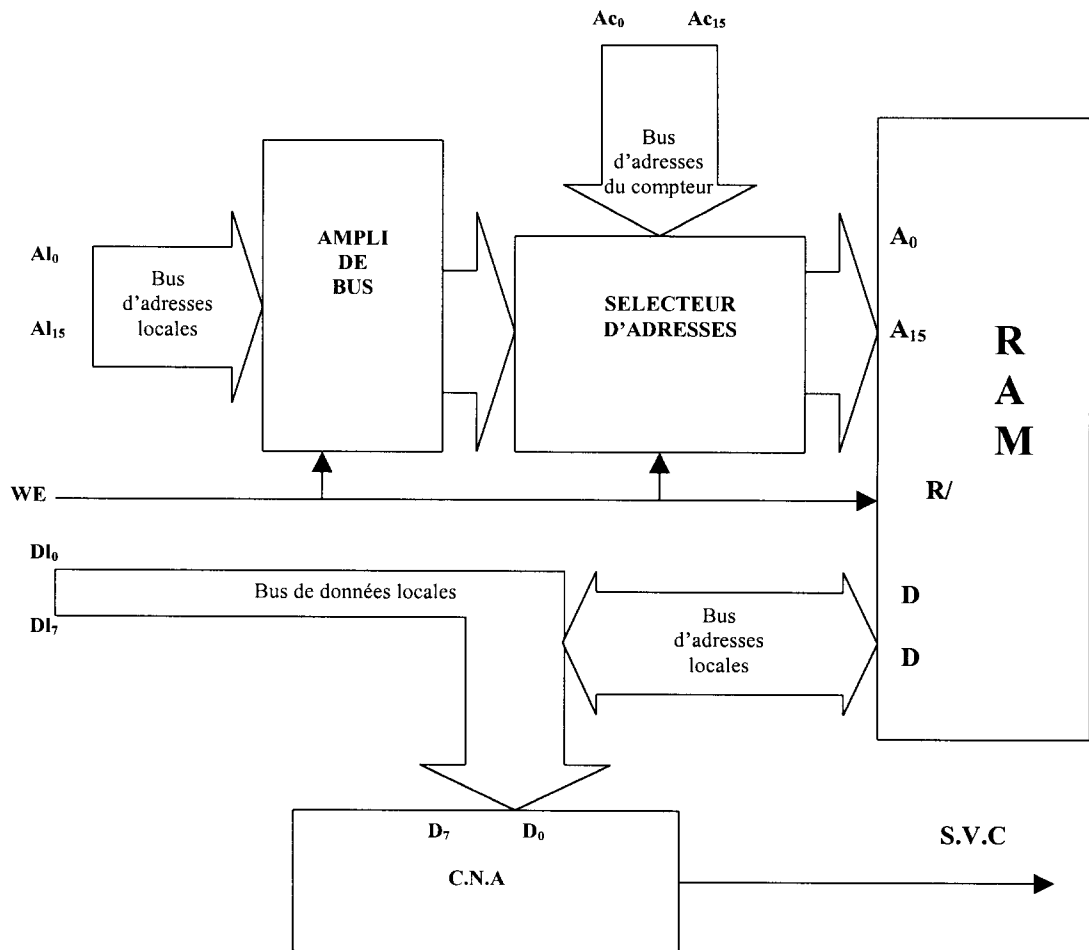


Figure 4. 5. Unité de stockage et de conversion

Le sélecteur d'adresses assure la commutation entre les adresses (Ali) et les adresses (Aci) en fonction du signal S. Si le signal S est actif, le sélecteur d'adresses commute sur les lignes A10-A115, et la mémoire¹ réalise un cycle d'écriture. Dans le cas contraire le sélecteur commute sur les lignes Ac0-Ac15, et la mémoire est validé en lecture.

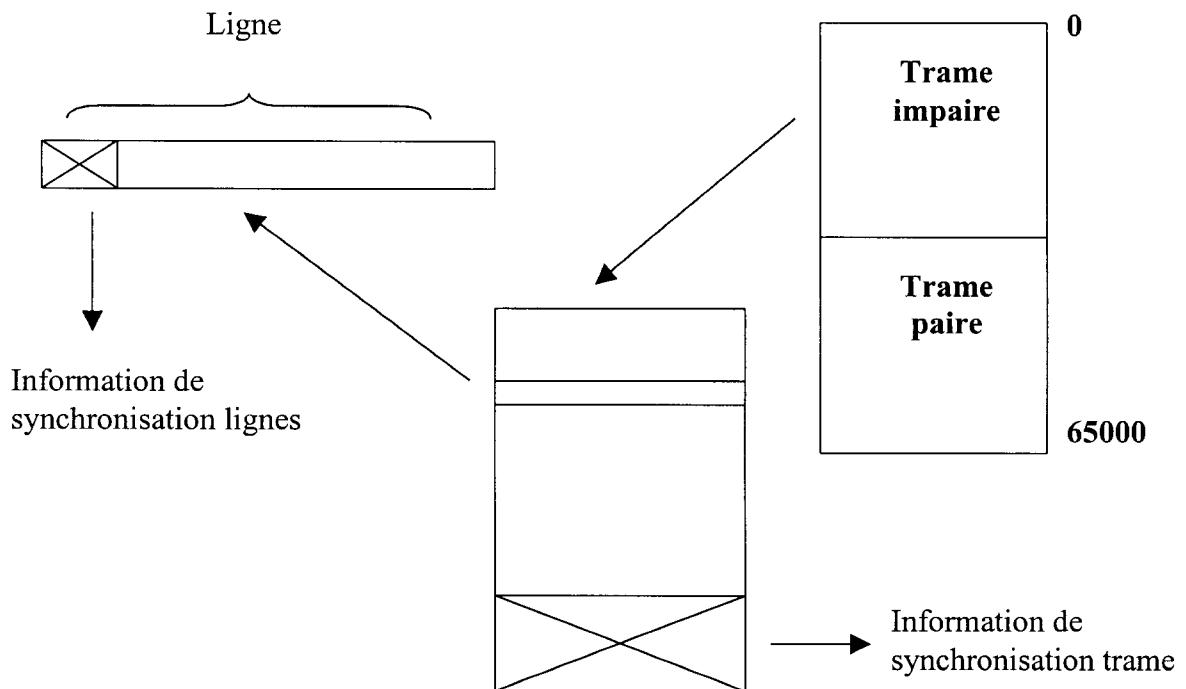


Figure 4. 6. *Structure de stockage du fichier image*

Le signal vidéo composite généré est le résultat d'une conversion numérique-analogique du fichier image stocké dans la RAM statique de notre carte. L'image numérique stockée dans la RAM est composée des deux trames impaire et paire, et chaque trame est composée d'une partie supérieure qui représente les lignes visibles de la trame et d'une partie inférieure non visible qui représente l'information de synchronisation trame. Les lignes visibles sont composées d'une information de synchronisation ligne sur 19 points et d'une information vidéo sur 85 points.

¹ La mémoire de notre carte.

4.6. MODULE DE LECTURE

Pour pouvoir adresser la mémoire d'image en lecture, nous avons besoin d'un générateur d'adresses. La taille des images que nous stockons est de 65000 octets, ce qui nécessite 16 bits d'adresses. Si nous utilisons des compteurs 8 bits, nous avons besoin de deux circuits. Il faut ajouter à cela des circuits pour la remise à zéro des compteurs à chaque arrivée à l'adresse 65000.

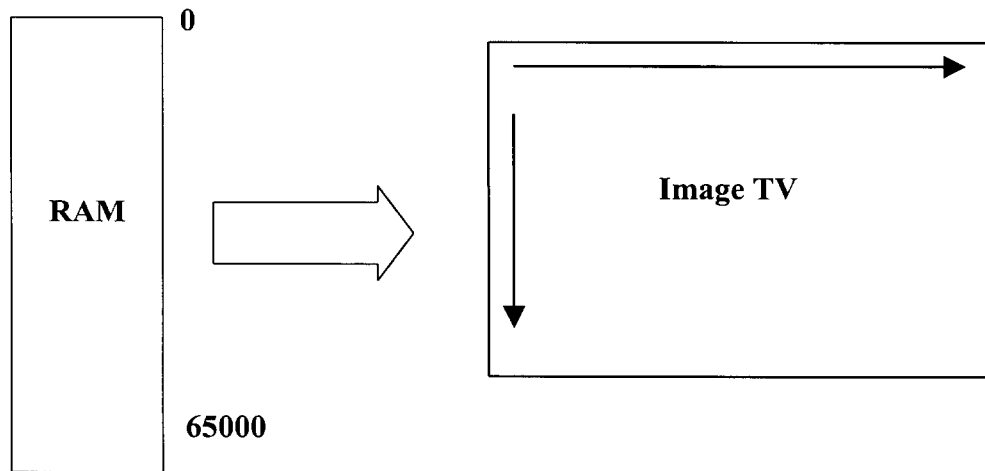


Figure 4. 7. *Variation de lecture et balayage d'image*

Le changement d'adresse de la valeur 0 à la valeur 65000 assure un balayage d'image de gauche vers la droite et de haut vers le bas.

L'unité de lecture est composée de trois parties essentielles :

- Un compteur 16 bits
- Un décodeur d'adresse de fin lecture
- Base de temps

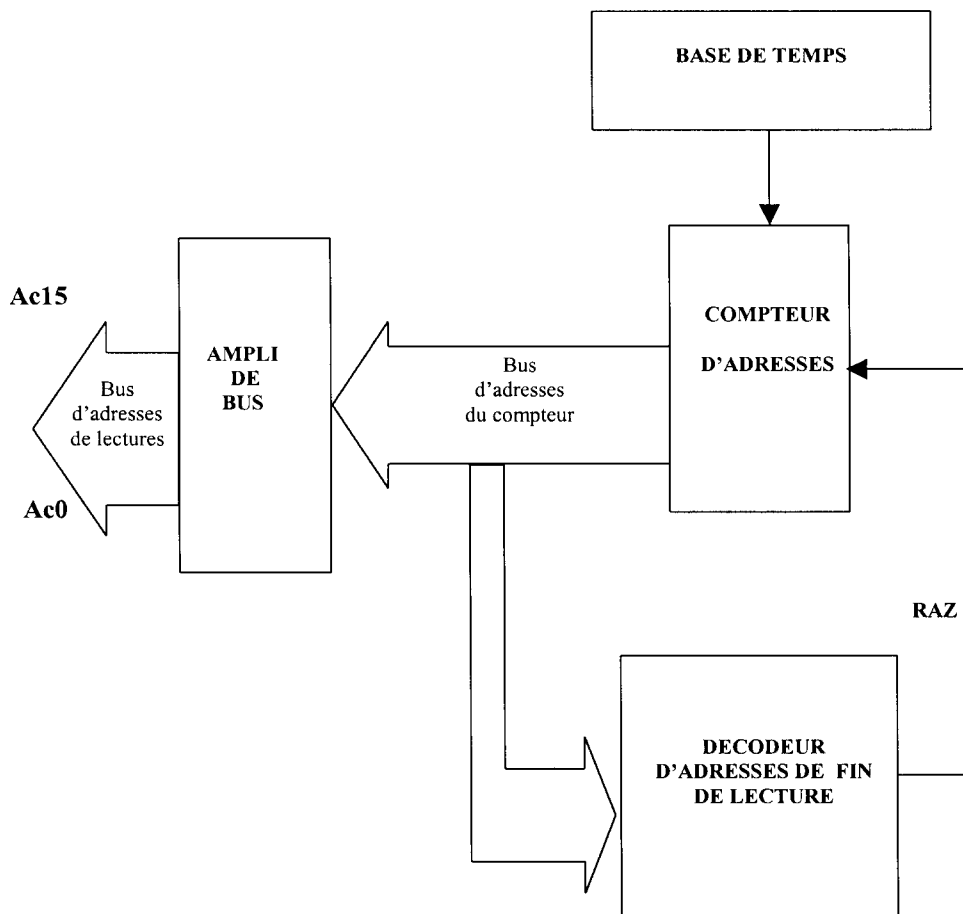


Figure 4. 8. Synoptique du circuit de lecture

Remarque : Le module de lecture est validé pendant la phase de lecture et ses sorties sont en haute impédance pendant la phase d'écriture.

La base de temps alimente l'entrée d'horloge du compteur d'adresses. Les lignes d'adresses générées par le compteur, sont connectées au décodeur de fin de lecture, qui assure une remise à zéro du compteur chaque arrivée à la fin de la zone mémoire réservée à l'image.

Les lignes d'adresses sont amplifiées pour commander les lignes d'adresses de la RAM, si le sélecteur ne reçoit pas un signal de la ligne S.



4.7. REALISATION ET FONCTIONNEMENT

4.7.1. Décodage de la carte

Voire figure 4.9 qui représente le schéma électrique du module de l'interface.

Le circuit de décodage dans le module d'interface est relié au bus d'adresses et au bus de commande par les deux circuits amplificateurs de bus 74LS244 pour éviter les problèmes de surcharges du bus¹ IBM-PC, même raison pour l'utilisation des amplificateurs 74LS244 pour le bus données.

Nous avons utilisé trois adresses pour le décodage 300H, 301H et 303H des 32 adresses autorisées par les constructeurs des PC IBM et compatibles (300H à 31FH). Le circuit décodeur utilise comme variables d'entrées les lignes d'adresses A11 à A0, AEN, CLK et IOW.

Pour les trois adresses on a plusieurs conditions communes, la lignes de demande d'écriture, la validation de l'adresse à travers la ligne AEN et la présence d'une adresse 30XH sur le bus d'adresses. Pour les raisons précédentes, une partie du circuit de décodage sera commune pour les trois fonctions de sorties S, d0 et d1.

4.7.2. Chargement des registres H et L

Les deux registres H et L sont composés de deux circuits 'latch' de type D, 74LS373 qui assurent la sauvegarde des valeurs de la nouvelle adresse jusqu'à la présence des données du fichier de transmission et l'adresse 300H sur le ISA d'IBM PC.

¹ Bus ISA.

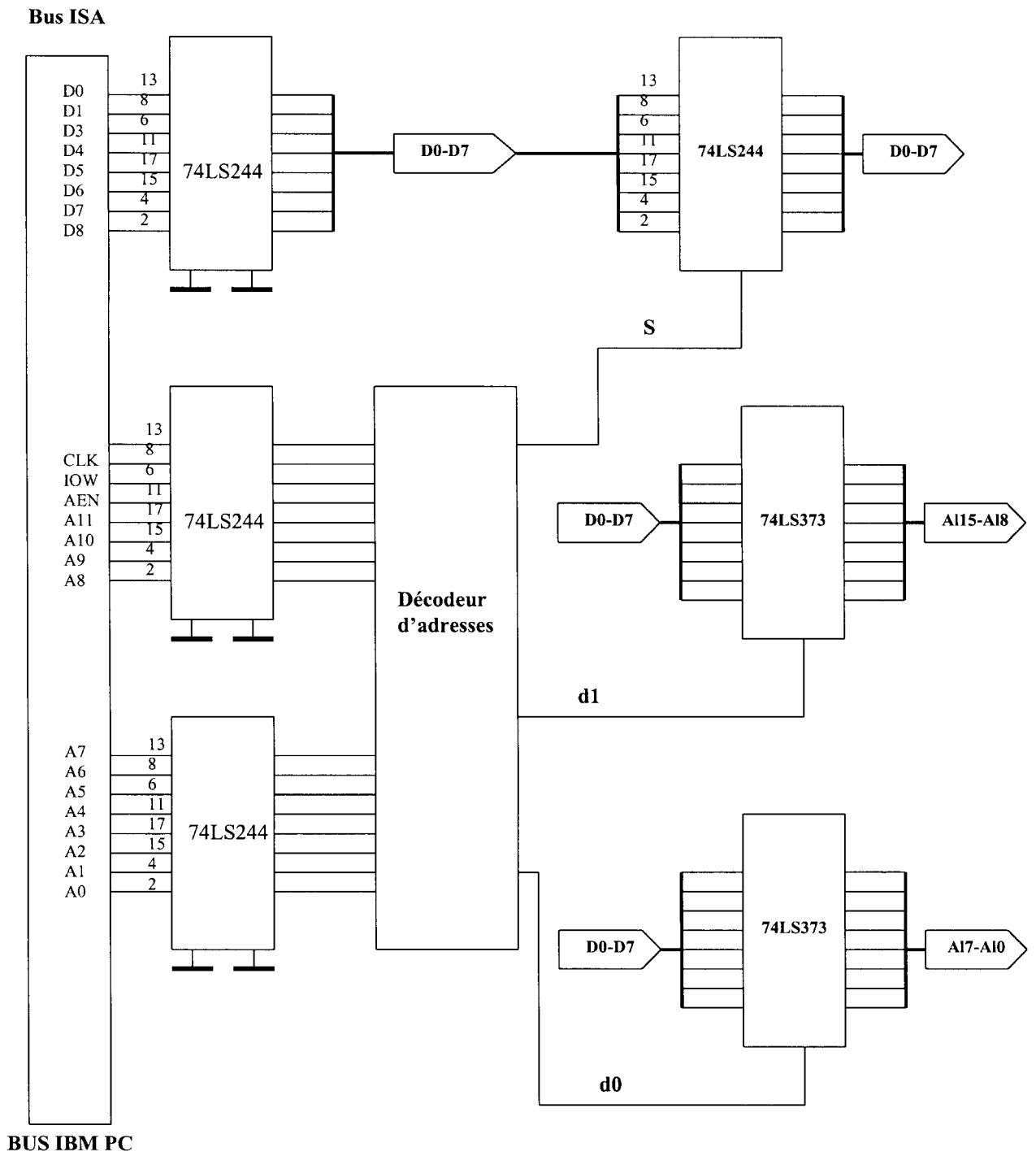


Figure 4. 9. Module d'interface

4.7.3. Générateur des signaux de contrôles

Le générateur des signaux de contrôles assure la commande du bus de contrôle des circuits RAM de notre carte et réalise l'écriture des données issus du bus d'extension ISA en temps réel¹. Le circuit est réalisé avec la logique câblée figure 4.10. Les signaux d'entrées de ce circuit sont S, d0 et d1 (signaux de sorties du décodeur d'adresse du module d'interface). Les signaux de sorties sont CS, WE et OE.

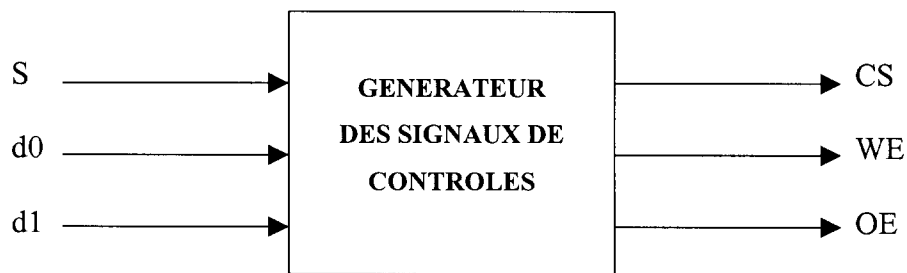


Figure 4. 10. Générateur des signaux de contrôles

La figure 4.11. présente le chronogramme des signaux de contrôle pour chaque écriture de donnée sur la mémoire de notre carte.

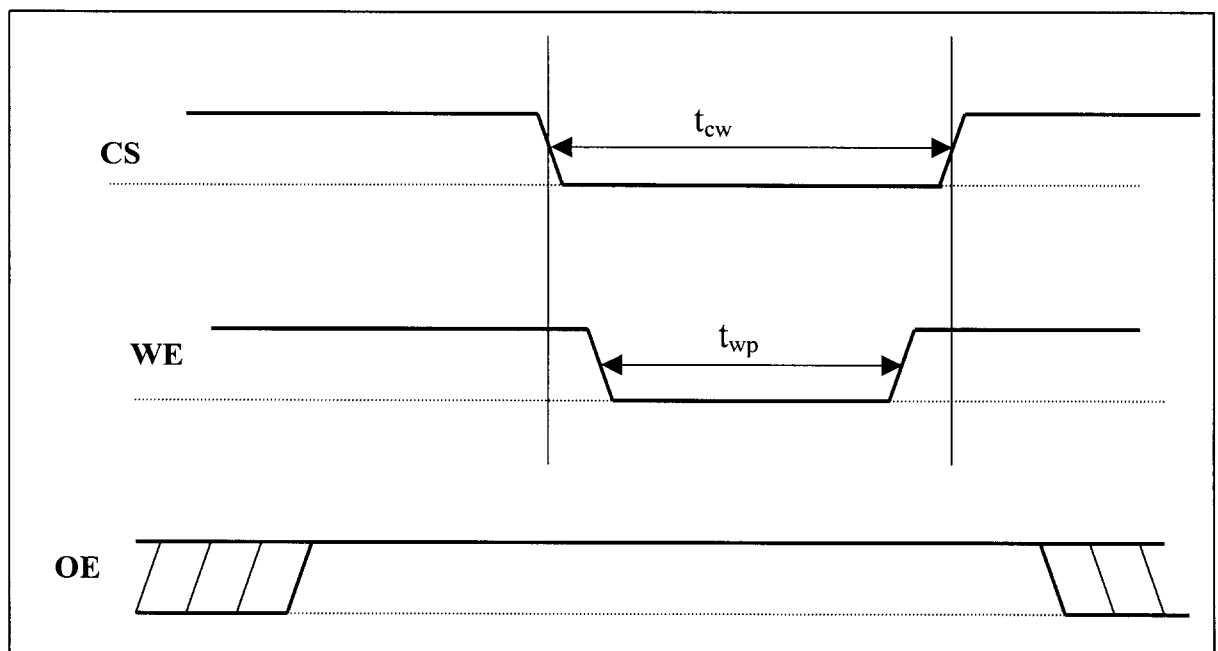


Figure 4. 11. Chronogramme des signaux de contrôles d'écritures

¹ Sans l'utilisation d'un micro contrôleur.

- $t_{cw} = 500 \text{ ns}$
- $t_{wp} = 350 \text{ ns}$

Le signal OE est toujours à l'état bas durant la lecture.

4.7.4. Base de temps

La fréquence¹ de la base de temps est calculée de la manière suivante :

- Nombres de lignes d'une image TV est 625.
- Nous avons fixé le nombre de points sur une ligne à 104 points
- Nombre d'images par seconde est 25.

Alors la fréquence sera :

$$\begin{aligned} F_{bt} &= 625 * 104 * 25 \\ &= 1,625 \text{ MHz} \end{aligned}$$

Le circuit utilisé pour réaliser la base temps est un Astable.

4.7.5. Le compteur d'adresses (LECTURE)

Le compteur d'adresses est basé sur deux circuits compteurs 8 bits 74LS 393 montés en cascade pour réaliser un compteur 16 bits. La base de temps alimente l'entrée d'horloge du premier circuit et l'horloge du deuxième circuit sera alimentée par la huitième ligne du poids fort du premier 74 LS 393.

Les lignes d'adresses du compteur 16 bits sont connectées au décodeur de fin de lecture, qui assure la remise à zéro du compteur par l'action des deux entrées de remise à zéro des deux compteurs.

Les lignes d'adresses sont amplifiées par les deux amplificateurs de bus 74LS244 pour commander les lignes d'adresses de la RAM en lecture. Les deux circuits amplificateurs seront validés si le signal de la ligne S est inactif.

Voir figure 4.12.

¹ Pour ne pas dépasser le temps de réponse des circuits utilisés.

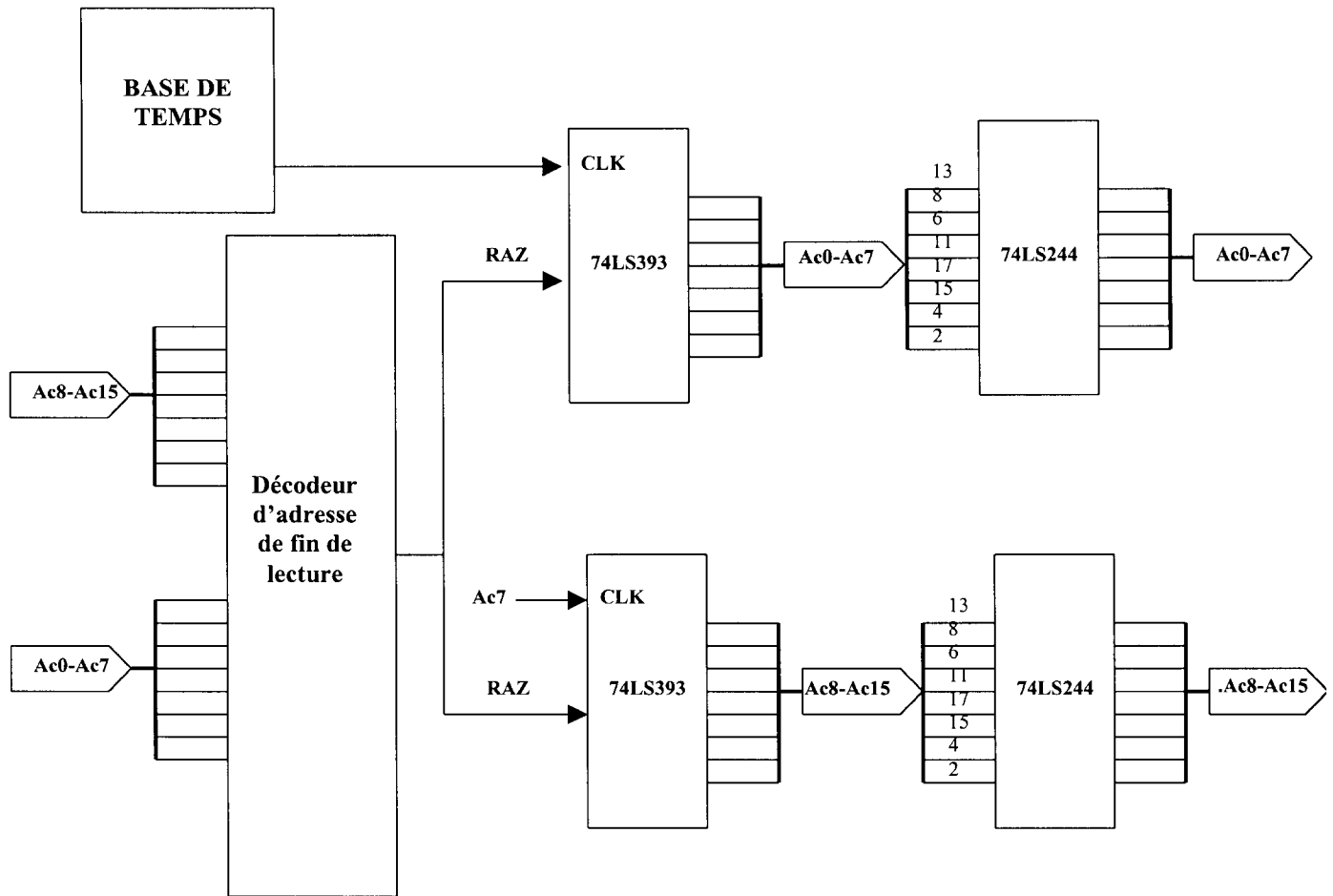


Figure 4. 12. *Module de lecture.*

4.7.6. Mémoire

Cette partie de la carte est réalisée avec deux circuits de mémoires RAM statiques 62256¹ (32 Ko) et en utilisant la ligne A15 pour la commutation de la validation entre ces deux circuits pour réaliser un stockage sur 64 Ko.

Voir figure 4.13.

¹ Nous représentons par 62256 tous les RAM de même caractéristiques (KM62256, HM62256, μ PD43256, GM76C256, M628032, etc.).

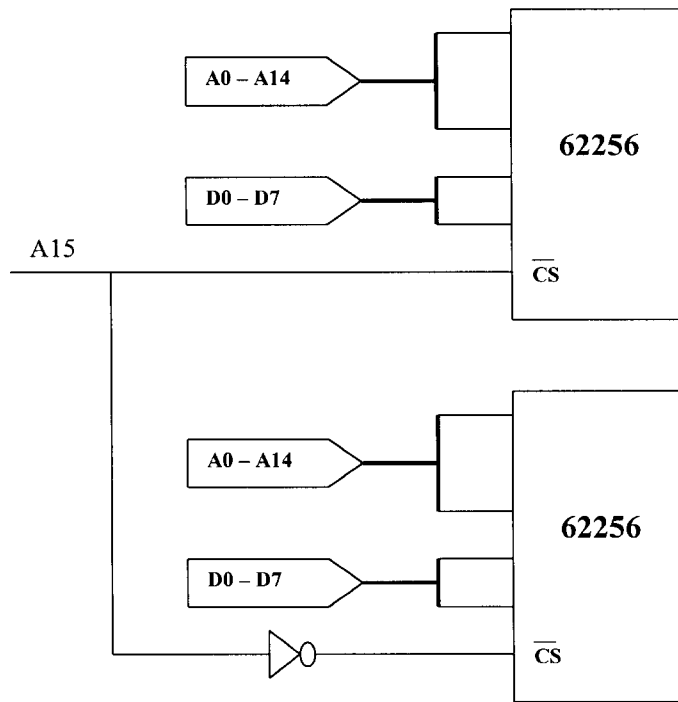


Figure 4. 13. Mémoire 64 Ko

Pour pouvoir écrire sur les RAM statiques il faut que les signaux de contrôles remplissent les conditions d'écritures du constructeur de la RAM statique [16][34][40].

Nous avons réalisé notre carte pour fonctionner avec un grand nombre de RAM statiques de même taille [27] [18] [42] [16]] [40] [34] [43] [2].

Notre chronogramme d'écriture proposé est présenté sur la figure 4.14. avec la ligne OE égale à « 1 » durant la phase d'écriture.

- $t_{wc} = 4 \mu s$
- $t_{cw} = 500 ns$
- $t_{wp} = 350 ns$

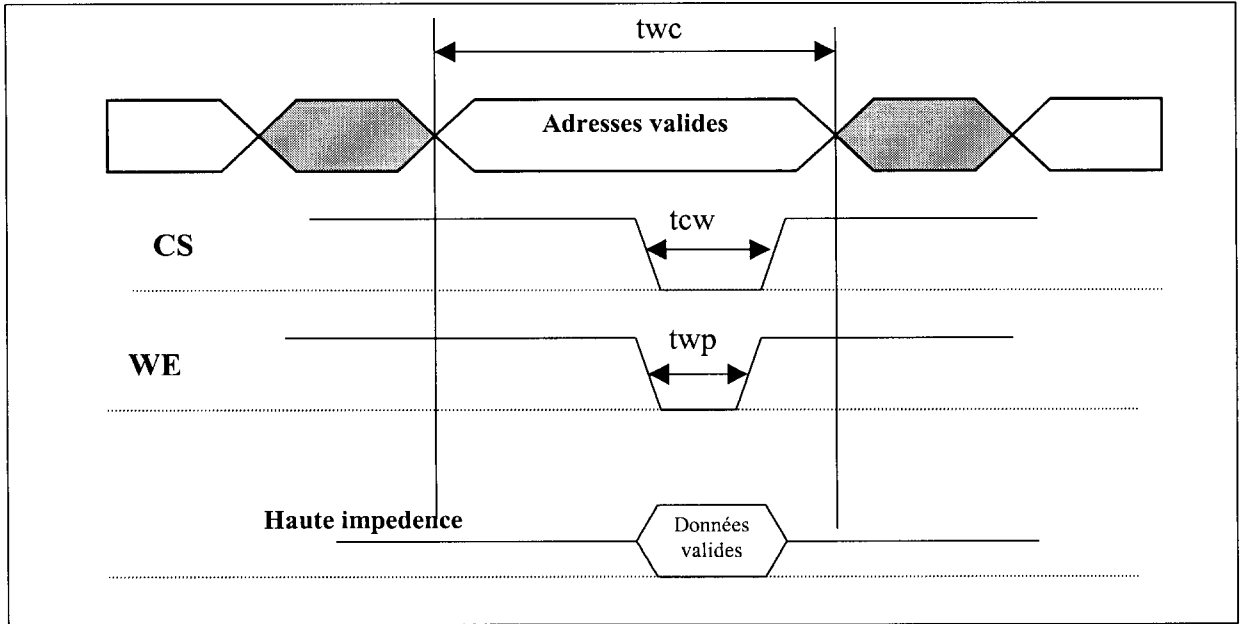


Figure 5. 14. Chronogramme d'écriture

Le chronogramme de lecture est présenté sur la figure 5.12. avec CS = 0, WE = 1 et OE = 0.

- $trc^1 = 615 \text{ ns}$

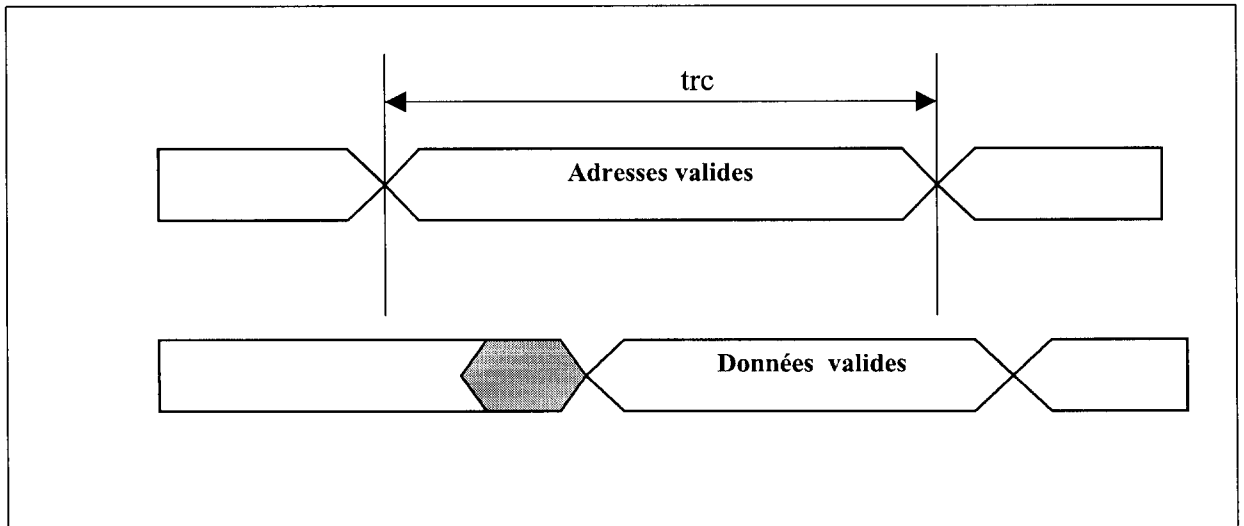


Figure 4. 15. Chronogramme de lecture

¹ Base de temps 1.625 MHz.

4.7.7. Conversion numérique analogique

Le circuit¹ DAC 08 [35] est un circuit monolithique convertisseur D/A 8 bits est qui offre un temps de conversion inférieur à 150 ns et un temps de propagation des bits maximum de 60 ns. Le DAC08 possède deux sorties complémentaires de courant permettant 20 v crête à crête de sorties.

La tension de sortie est donné par la formule suivante :

$$V_O = \frac{V_{ref}}{R_{ref}} \left(\frac{A_1}{2} + \frac{A_2}{4} + \dots + \frac{A_8}{256} \right) R_0$$

Si $R_0 = R_{ref}$

On a :

$$V_O = V_{ref} \cdot A_i$$

Avec A_i l'état des bits.

Pour avoir une tension max, $A_i = 255/256$.

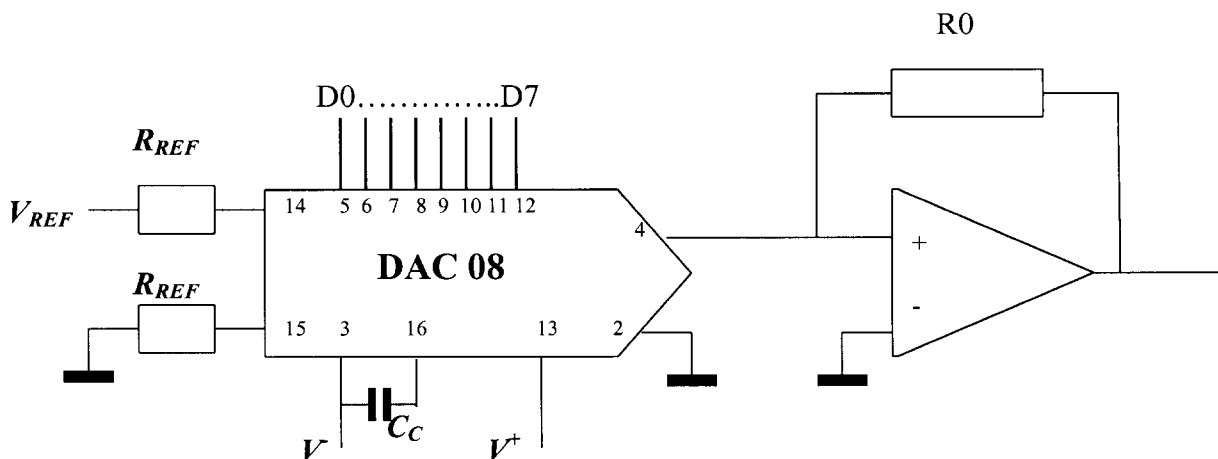


Figure 4. 16. Convertisseur numérique analogique 8 bits²

¹ Nous avons choisi ce circuit uniquement pour vérifier le principe de notre architecture.

² Pour plus de détails, voir [35].

4.8. CONCLUSION

Notre réalisation nous a permis de dévoiler quelques secrets de la microinformatique et des techniques hard, de proposer et réaliser notre propre solution avec des composants standards. L'architecture proposée assure une transmission de données (entre le bus ISA et la mémoire de notre carte) en temps réel (sans l'utilisation d'un microcontrôleur).

Nous avons réalisé les trois modules de notre carte et nous avons fait les derniers tests sur un PC doté d'une carte mère *Xcel2000* avec processeur *Intel Celeron* 433 MHz.

Caractéristiques techniques de la carte :

Composants utilisés :

- SRAM μ PD43256 de NEC [34]
GM76C256 de Goldstar [27]
- Latch 74LS373
- Buffers unidirectionnels 74LS244
- Compteurs 8 bits 74LS393
- Convertisseur D/A 8 bits DAC08 [35]
- AmpliOp LF351 [35]
- Portes logiques

Bus :

- Bus de données 8 bits
- Bus d'adresses¹ 12 bits

¹ 12 bits pour réaliser le décodage des adresses.

Ports d'entrée/sortie :

- 300Hex.
- 301Hex.
- 303Hex.

Signal généré :

- Signal vidéo¹ composite d'amplitude de 1,5 v.

¹ Signal adapté à la ligne 75 Ω .

CONCLUSION GENERALE

Dans ce travail, nous avons réalisé un système qui affiche des images PC (niveau de gris) sur un écran TV et nous nous sommes intéressés à la concrétisation de notre architecture qui repose principalement sur l'approche de la conception avec les fonctions logiques et non pas sur l'utilisation des microcontrôleurs ou l'utilisation des processeurs graphiques. Nous avons réussi à réaliser un transfert de données avec l'adressage proposé et nous avons effectué un affichage plein écran des images en niveaux de gris. La carte est réalisée en logique câblée et génère un signal vidéo composite normalisé (norme 625 lignes) ce qui donne une grande stabilité d'affichage sur un TV classique. Le système proposé peut produire d'autres standards avec un minimum de modifications, car l'information de synchronisation est modifiable par logiciel. Il reste à varier la fréquence d'horloge du module de lecture.

Le système que nous avons réalisé remplit les objectifs pour lesquels il a été conçu, ceci principalement grâce à l'utilisation de notre conception qui s'adapte bien à l'implantation matérielle. Sur le plan électronique les techniques utilisées permettent d'obtenir un système réalisable avec la logique programmable ce qui va donner une structure très compacte.

Notre système peut être utilisé dans beaucoup d'autres applications, parmi les quelles nous citons :

- L'utilisation dans les systèmes de contrôles par PC qui nécessitent plusieurs périphériques d'affichages.

- Transmission des images PC par des ondes hertziennes par une simple modulation AM classique.
- Enregistrement des images ou séquences d'images sur des bandes magnétiques (par exemple les cassettes vidéo).
- L'utilisation de plusieurs cartes pour former un grand panneau d'affichage composé de plusieurs écrans TV commandés par PC.
- La production des images à effets spéciaux pour la télévision.

Perspectives

Malgré que notre carte réalisée est opérationnelle, elle reste un prototype de laboratoire qui a servi à démontrer la faisabilité d'un tel système. En se basant sur le même principe et la même architecture, une carte plus performante réalisant un affichage couleur à haute définition, peut être réalisée. Plusieurs solutions peuvent être appliquées. Nous proposons deux solutions.

Pour une première solution, deux modifications principales seront nécessaires pour produire un signal vidéo composite de 5 Mhz. Premièrement, il faut utiliser une RAM statique d'une capacité supérieure ou égale à 400 Ko et avec un temps d'accès inférieur à 50 ns. Deuxièmement, il faut utiliser un convertisseur numérique analogique qui travaille à des fréquences vidéo.

Pour une deuxième solution, on peut produire les trois signaux de couleurs, rouge, vert et bleu par les modifications suivantes. Premièrement, il faut utiliser trois plans mémoires pour les trois couleurs. Deuxièmement, il faut générer les signaux de synchronisations et synchroniser la lecture sur les trois plans mémoires.

BIBLIOGRAPHIE

- [1] M.Althaus. « *LE LIVRE D'OR PC* » Sybex, France, Mais 1992.
- [2] « *ARIZONA MICROCHIP TECHNOLOGY DATABOOK* » USA, 1995.
- [3] M. Aumiaux « *LES SYSTEMES A MICROPROCESSEUR* » Edition MASSON, France, 1982.
- [4] S.Benedetto, E.Biglieri, V.Castellani. « *DIGITALE TRANSMISSION THEORY* » Prentice-Hall International Editions, USA, 1987.
- [5] K.B.Benson. « *TELEVISION ENGINEERING HANDBOOK* » McGRAW-HILL, USA, 1986.
- [6] H.Bennassar. « *COURS DE MICROPROCESSEURS 16 BITS* » Office des Publications Universitaires, Algérie, 1993.
- [7] G. Bjontegaard, K. O. Lillevold, and R. Danielsen. « *A COMPARISON OF DIFFERENT CODING FORMATS FOR DIGITAL CODING OF VIDEO USING MPEG-2* » IEEE TRANSACTION ON IMAGE PROCESSING, VOL. 5, NO. 8, AUGUST 1996.
- [8] M. Bomanns. « *LES CARTES GRAPHIQUES SUR PC* » EDITIONS MICRO APPLICATION, France, 1989.
- [9] J.C. Chauveau, G. Chevalier et B. Chevalier. « *MEMOTECH ELECTRONIQUE COMPOSANTS* » Editions CASTELLA, France, 1989.
- [10] A.Chehikian. « *ALGORITHMES OPTIMAUX POUR LA GENERATION DE PYRAMIDES D'IMAGE PASSE-BAS ET LA LAPLACIENNES* » Recherches, Traitement du signal, Volume 9 – n° 4, France, 1992.

-
- [11] R. Delsol. « *ELECTRONIQUE NUMERIQUE* » BERTI Editions, Algérie, 1992.
- [12] O.Delgrange. « *LA COMPRESSION INFORMATIQUE* » Computer Based Learning Unit, University of Leeds, 1996.
- [13] M. Effros. « *UNIVERSAL MULTI RESOLUTION SOURCE CODING* » IT Worskhop on Detection, Estimation, Classification and Imaging, Santa Fe, NM, USA, Feb. 1999.
- [14] R.C.Gonzalez and P.Wintz. « *DIGITAL IMAGE PROCESSING* » ADDISON-WESLEY PUBLISHING COMPANY, USA, 1989.
- [15] J.Han « *CONCEPTION ET REALISATION D'UNE MEMOIRE PARTAGEE REPARTIE* » Thèse, Institut National Polytechnique de Grenoble, France, novembre 1996.
- [16] « *HITACHI ELECTRONIC COMPONENTS DATABOOK* » Japan,1996.
- [17] J. Herben. « *LA TELEVISION EN COULEURS* » EDITIONS RADIO DUNOD, France, 1991.
- [18] « *HYUNDAI SEMICONDUCTORS DATABOOK* » Korea, 1992.
- [19] Z.Hussain. « *DIGITAL IMAGE PROCESSING* » ELLIS HORWOOD LIMITED, Englend 1991.
- [20] <http://www.ibm.com/>
- [21] T. Ben Ismail. « *SYNTHESE AU NIVEAU SYSTEME ET CONCEPTION DE SYSTEMES MIXTES LOGICIELS/MATERIELS* » Thèse, Institut National Polytechnique de Grenoble, France, 1996.
- [22] S. Jo and Y. H. Lee. « *A STATISTICAL FEEDFORWARD/FEEDBACK BUFFER CONTROL FOR THE TRANSMISSION OF DIGITAL VIDEO SIGNALS COMPRESSED BY DCT-BASED INTRAFIELD CODING* » IEEE TRANSACTION ON IMAGE PROCESSING, VOL. 5, NO. 3, MARCH 1996.
- [23] R.Koudsieh. « *CONSTRUCTION DE PYRAMIDE D'IMAGE EN TEMPS REEL VIDEO* » Thèse, Institut National Polytechnique de Grenoble, France, 1992.

- [24] G. Leblanc. « *TURBO C* » EYROLLES, France, 1988.
- [25] S. Leibson. « *MANUEL DES INTERFACES* » McGRAW-HILL, France, 1984.
- [26] A. Levine. « *TURBO C DES OUTILS* » BERTI Editions, Algérie, 1993.
- [27] « *GOLDSTAR SEMICONDUCTORS DATABOOK* » Korea, 1993.
- [28] H. Lilen. « *80386 MODES DE FONCTIONNEMENT* » EDITIONS RADIO, France, 1989.
- [29] J. Maillefert. « *LANGAGE VHDL* » IUT. CACHAN, 1997.
- [30] G. F. Marchioro. « *DECOUPAGE TRANSFORMATIONNEL POUR LA CONCEPTION DE SYSTEMES MIXTES LOGICIEL/MATERIEL* » Thèse, Institut National Polytechnique de Grenoble, France, novembre 1998.
- [31] F. Milsant. « *COURS D'ELECTRONIQUE* » Editions CHIHAB-EYROLLES, Algérie, 1994.
- [32] D. Martin et F. Piette « *CLEFS POUR PC-AT ET COMPATIBLES* » Editions PSI, France, 1987.
- [33] <http://www.necel.com/>
- [34] « *NEC SEMICONDUCTORS DATABOOK* » Japan, décembre 1997.
- [35] « *NATIONAL SEMICONDUCTORS DATABOOK* » USA, 1996.
- [36] « *L'ORDINATEUR INDIVIDUEL* »
<http://www.ordinateur-individuel.com/>
- [37] « *PHILIPS SEMICONDUCTORS FOR TELEVISION AND VIDEO SYSTEMS DATA HANDBOOK* » 1997.
- [38] K. Rose and S. L. Regunathan, « *TOWARDS OPTIMAL SCALABILITY IN PREDICTIVE VIDEO CODING*, » *Proceedings of ICIP98*, Chicago, Oct. 1998.

-
- [39] K. Rose, P. Wu and S. L. Regunathan, « *EFFICIENT SNR SCALABILITY IN PREDICTIVE VIDEO CODING*, » *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 1998.
- [40] « *SAMSUNG ELECTRONICS DATABOOK* » korea, 1997.
- [41] M.Said, M.Mohamed et M.Abderahmene « *ELECTRONIQUE DIGITALE* » Les Editions Infor-Z, Algérie, 1990.
- [42] « *SGS-THOMSON MICROELECTRONICS DATABOOK* » 1995.
- [43] « *TEXAS INSTRUMENTS DATABOOK* » USA, 1998.
- [44] M.Tischer. « *LA BIBLE PC* » EDITIONS MICRO APPLICATION, France, Novembre 1994.
- [45] TISSERANT. « *COMPRESSION MPEG* » Nancy, 1997.

ANNEXES

ANNEXE A

DEVELOPPEMENT D'APPLICATION AVEC LE VHDL

Pour développer une application de logique programmable, il faudra suivre la démarche de la figure 2.2.

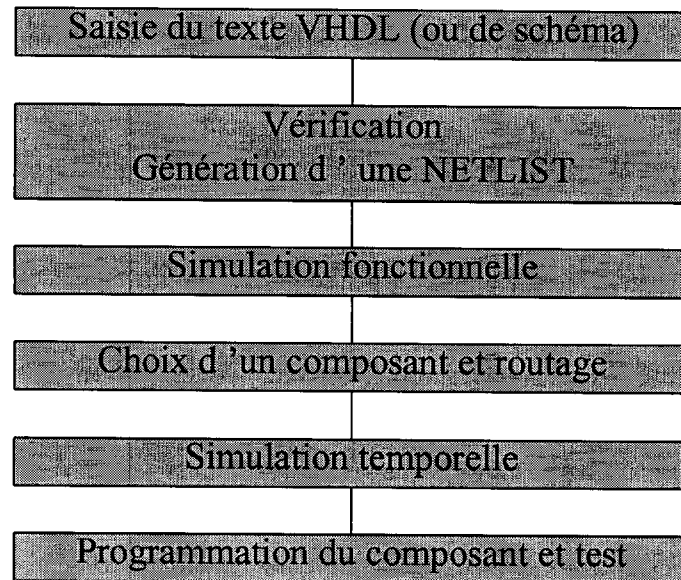


Figure 2. 2. Les étapes de Développement une application avec le VHDL

a. Analyse du problème

Cette étape permet de répondre aux questions suivantes qui vont guider le concepteur dans les étapes suivantes.

Quelle fonction doit remplir mon circuit ?

Quels sont les signaux d'entrée et de sortie ?

Peut-on scinder le circuit en blocs ?

b. Description du problème en V.H.D.L

Le circuit a été divisé en blocs possédant des entrées, des sorties et réalisant une fonction bien définie. Chaque bloc peut alors être vu comme une boîte noire. C'est la première étape dans la rédaction du code.

L'étape de vérification permet de valider la syntaxe du programme. La « netlist » est un fichier contenant la description de l'application sous forme d'équations logiques.

c. Simulation fonctionnelle

Lors de l'étape de simulation fonctionnelle, on valide l'application, indépendamment de l'architecture et des temps de propagation du futur circuit cible.

Un point important lors de la programmation en VHDL est de conserver à l'esprit que ce que l'on programme sera placé dans un circuit hardware.

d. Choix d'un composant et routage

L'étape de routage génère les informations permettant d'intégrer l'application dans le circuit choisi. Une « rétro annotation » est effectuée dans la « netlist » : les temps de propagation du composant cible y sont pris en compte.

e. Simulation et Programmation du circuit

Lors de la simulation temporelle, on peut évaluer les performances de l'application générée.

La simulation du circuit se base sur le fichier de programmation. C'est une étape importante dans le développement d'un circuit car elle permet d'éviter la programmation du circuit alors que la fonction désirée n'est pas réalisée.

La programmation s'effectue au moyen d'un programmeur spécifique au circuit utilisé.

ANNEXE B

LES PROCESSEURS INFERIEUR A 32 BITS

En premier on a, le **8086** qui possède un bus d'adresses sur 20 bits et un bus de données sur 16 bits. Le **80286** possède un bus d'adresses sur 24 bits et le bus de données sur 16 bits. Le **80386-SX** possède un bus d'adresses sur 24 bits et le bus de données sur 16 bits.

LES PROCESSEURS A 32 BITS

En premier on a, le **80386** possède un bus d'adresses sur 32 bits et le bus de données sur 32 bits. Le **80486** possède un bus d'adresses sur 32 bits et un bus de données sur 32 bits et réunit sur une puce les fonctionnalités d'un **80386** associé à un coprocesseur **80387**. Un **80486** peut être programmé comme un **80386** auquel serait connecté un coprocesseur **80387**.

Pentium apparu en 1993, il est aujourd'hui largement dépassé, ses fréquences de fonctionnement allaient de 60 à 200 MHz, avec une vitesse de bus à 66 MHz. Ce processeur est à l'origine du formidable essor de l'informatique familiale dans la mesure où il offrait un niveau de performance jusqu'alors jamais atteint, chose très bénéfique aux applications multimédias. Sa principale différence avec les microprocesseurs précédents est sa capacité à exécuter deux instructions simultanément, alors que les puces de la génération **80486** ne pouvaient en exécuter qu'une seule à la fois. Cela améliore considérablement les performances, comme on peut le deviner.

Pentium MMX apparue en 1997 avec des fréquences allant de 166 à 233 MHz. La particularité de cette puce est bien évidemment la technologie **MMX** (MultiMedia eXtension), jeu de 57 instructions spéciales intégrées au microcode du processeur, et destinées à accélérer le traitement des applications multimédia. Certains disent que cette technologie n'était qu'une opération commerciale d'Intel pour rester sur le devant de la scène des nouveautés technologiques. Malgré tout de nombreuses applications - principalement des jeux - en tirèrent parti.

Pentium II Les premiers Pentium II utilisaient la vitesse de bus à 66 MHz. Ce sont les modèles cadencés à 233, 266, 300 et 333 MHz. Ceux qui viennent après, utilisent une vitesse de bus de 100 MHz et sont cadencés à 350, 400 et 450 MHz. Outre les différences de vitesses avec les processeurs Pentium des générations précédentes, le Pentium II se distingue par sa mémoire cache L2, directement montée sur le support. Cette technique est inspirée de la technologie du Pentium Pro, cependant la mémoire cache L2 n'est pas ici cadencée à la même vitesse que le processeur, mais à la moitié de cette vitesse.

Pentium III le Pentium III n'est en fait qu'un simple Pentium II auquel on a ajouté 71 nouvelles instructions de type MMX, le but est identique : optimiser le traitement des données multimédia (images 3D, sons, vidéo, etc...). La fréquence de départ de la puce est de 450 MHz, mais la version à 500 et 550 MHz est disponible et une version de 800 MHz commence à être commercialisée.

ANNEXE C

FICHER BMP DE BITMAP WINDOWS

a. BITMAPFILEHEADER

La structure du BITMAPFILEHEADER est composée de :

1. Identification du fichier codée sur 2 octets (BM pour "BITMAP").
2. Longueur totale du fichier codée sur 4 octets.
3. Zone réservée et mise à zéro codée sur 4 octets
4. Offset de données BITMAP codée sur 4 octets.

b. BITMAPINFOHEADER

La structure du BITMAPINFOHEADER est composée de :

1. Taille occupée par le BITMAPINFOHEADER Codée sur 4 octets
2. Largeur de l'image en pixels codée sur 4 octets.
3. Hauteur de l'image en pixels codée sur 4 octets.
4. Nombre de surfaces de l'image (toujours égale à 1) codée sur 2 octets.
5. Nombre de bits de couleur (1, 4, 8 ou 24) codée sur 2 octets.
6. Méthode de compression (0 = non compressé) codée sur 4 octets.
7. Taille de l'image en octets codée sur 4 octets.
8. Résolution horizontale en pixels par mètre codée sur 4 octets.
9. Résolution verticale en pixels par mètre codée sur 4 octets.
10. Nombre de couleurs utilisées (0=toutes) codée sur 4 octets.
11. Nombre de couleurs importantes (0=toutes) codée sur 4 octets.

RGBQUAD:

La structure du RGBQUAD est composée de :

1. Intensité de bleu codée sur 1 octet.
2. Intensité de vert codée sur 1 octet.
3. Intensité de rouge codée sur 1 octet.

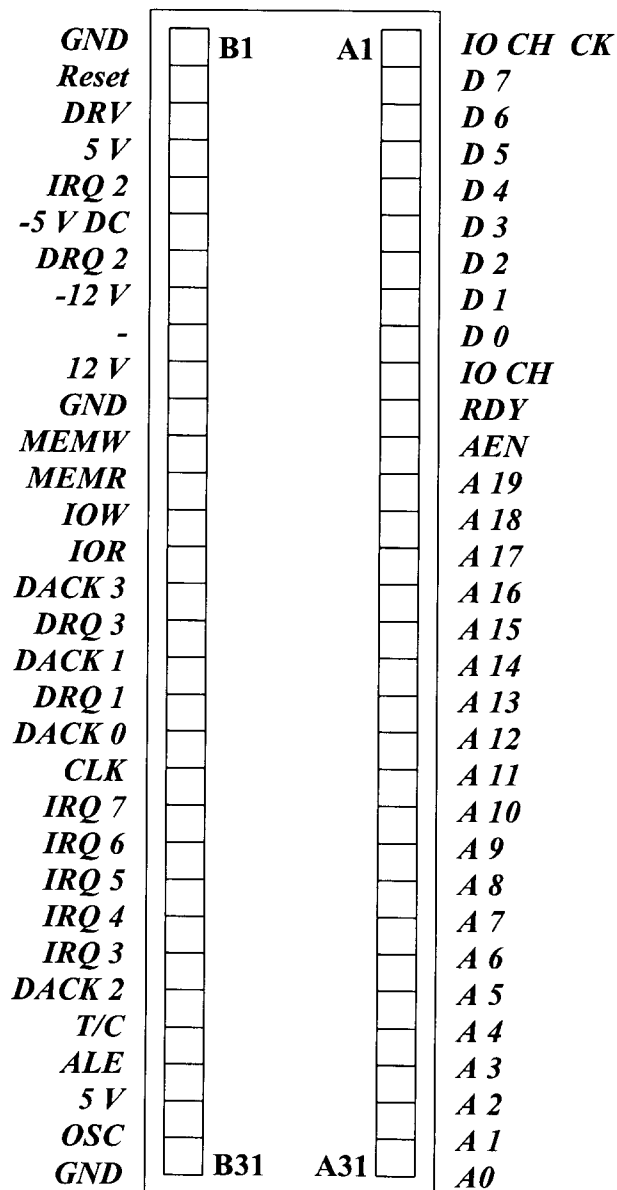
Remarques:

- Chaque ligne de l'image contient un nombre d'octets multiple de quatre. Des octets nuls sont éventuellement ajoutés à la fin de la ligne.
- Un fichier BMP est chargé de telle façon que la première information de la zone de données (qui contient les valeurs des pixels) correspond au coin inférieur gauche de l'image.

ANNEXE D

BROCHAGES DU BUS ISA

Le bus ISA d'IBM PC comporte 62 broches. La figure suivante représente le brochage de ce bus.



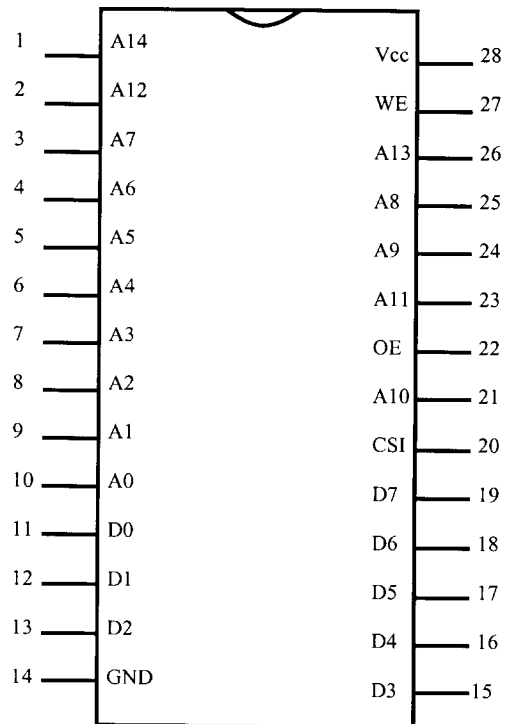
Brochages du bus ISA.

- I/O CHCK :** Contrôle des entrées/sorties. Un signal 0 indique au processeur une erreur de parité ou de mémoire sur la carte enfichable.
- IO CK RDY :** Vérification de la disponibilité des entrées/sorties. Cette ligne permet de ralentir le flux de données entre l'ordinateur et la carte.
- DRV :** Ce signal permet de réinitialiser l'ordinateur.
- MEMW :** Mémoire : écriture de données (*memory write*). Ce signal avertit la mémoire de travail qu'elle doit enregistrer les données qui se trouvent sur le bus de données.
- MEMR :** Mémoire : lecture de données (*memory read*). Ce signal avertit la mémoire de travail qu'elle doit lire les données qui se trouvent sur le bus de données. Le signal est produit par le processeur ou par l'unité DMA.
- DRQ :** Demande d'accès direct à la mémoire.
- IRQ :** Demande d'interruption. Ce signal permet à la carte de déclencher une interruption de niveau 2.
- DACK :** Accusé de réception DMA. Cette ligne accuse réception d'un signal DRQ.
- T/C :** Compteur des accès DMA.
- ALE :** Verrouillage des adresses.

ANNEXE E

BROCHAGES DES SRAM 32 KO

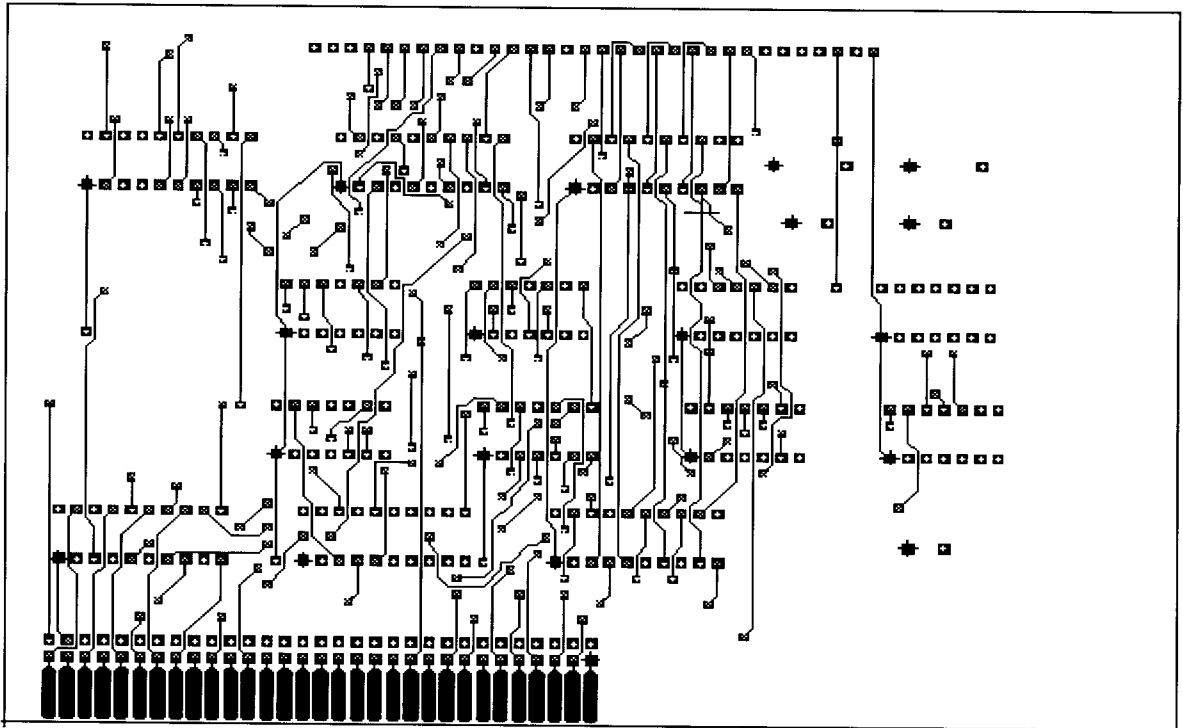
CMOS - 256 K – 32 K x 8



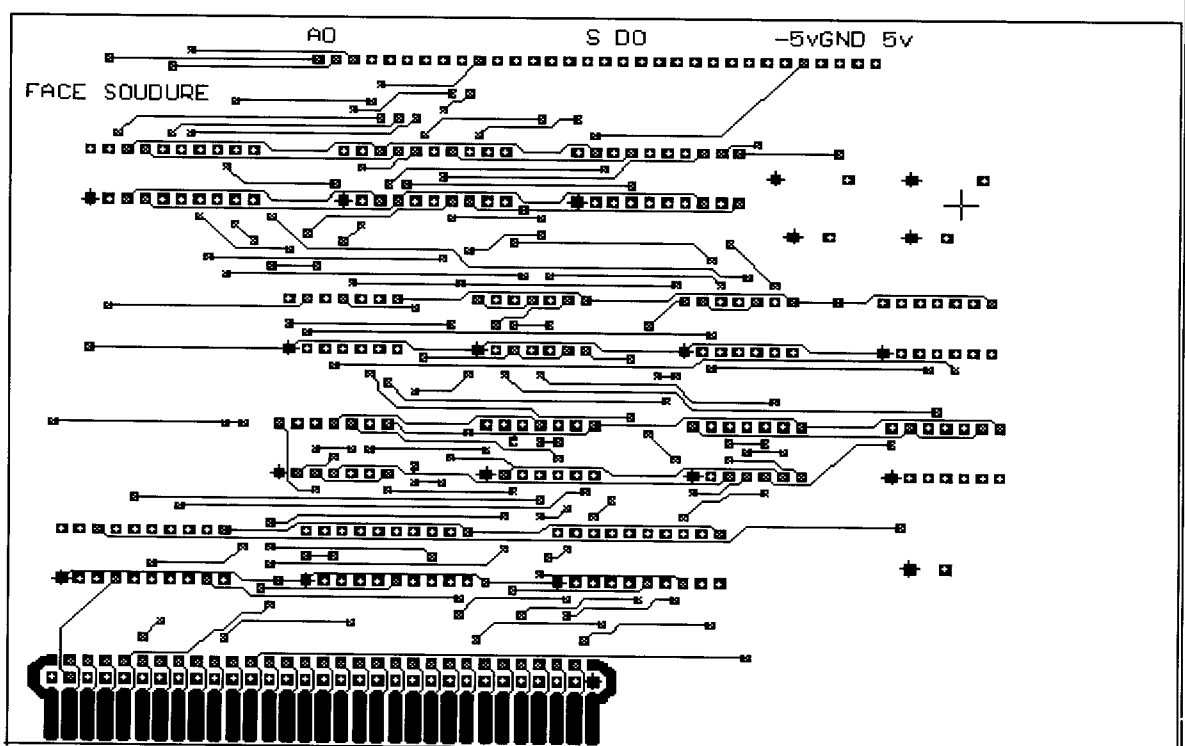
Nom des broches	Fonction
A ₀ ~A ₁₄	Entrées des adresses
WE	Validation d'écriture
CS	Validation du circuit
OE	Validation des sorties
D ₀ ~D ₇	Entrées-sorties des données
Vcc	Alim. +5 V
GND	Masse

ANNEXE F

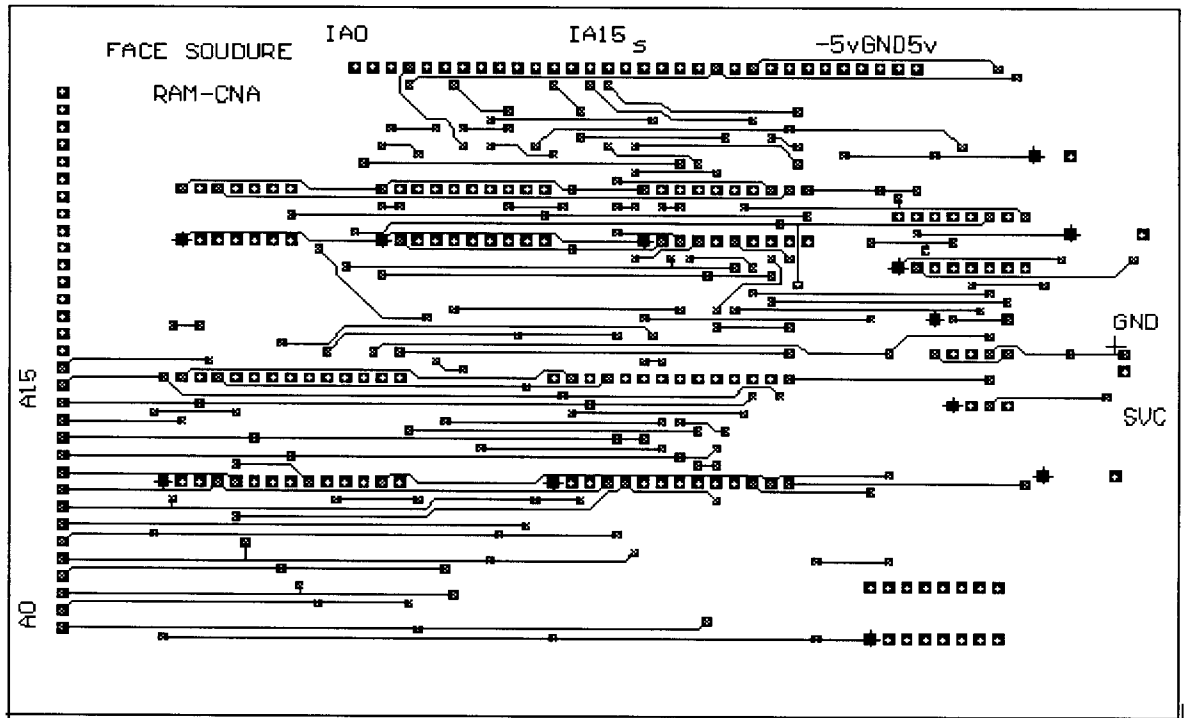
CIRCUITS IMPRIMES DES MODULES



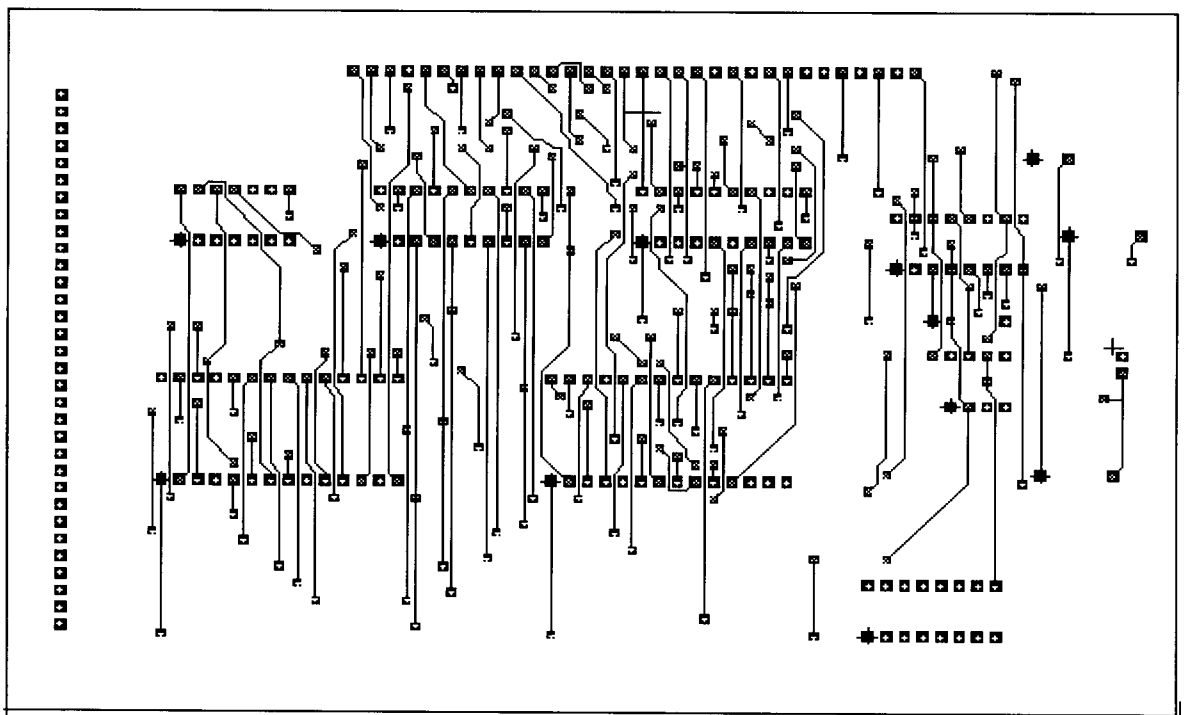
Dessin du circuit imprimé du module d'interface (face composants)



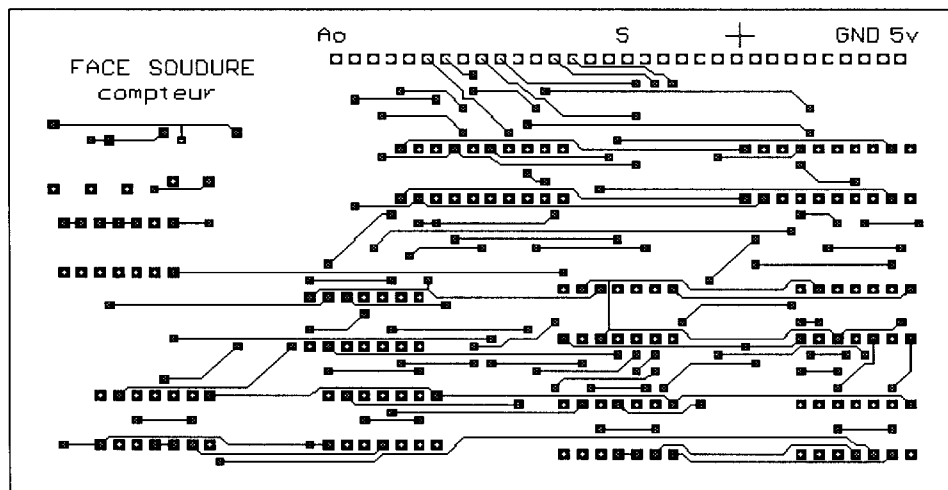
Dessin du circuit imprimé du module d'interface (face soudures)



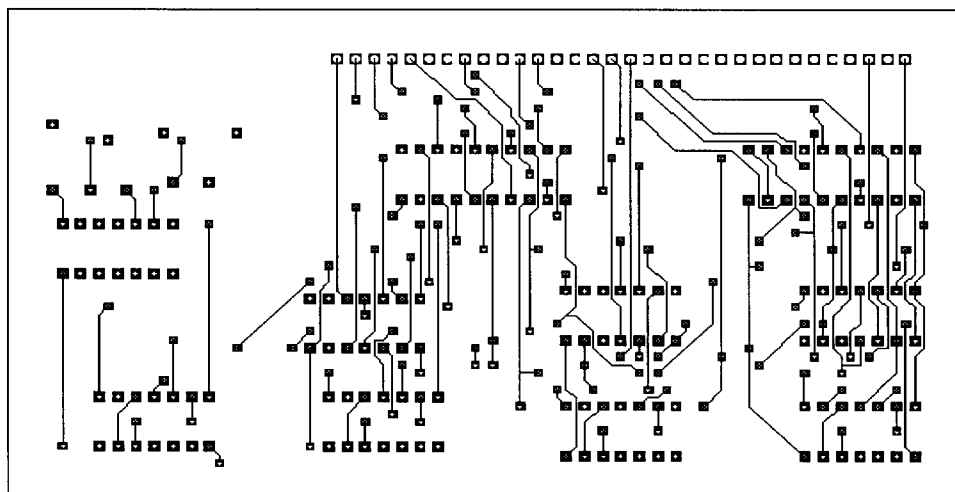
Dessin du circuit imprimé du module de stockage et conversion (face soudures)



Dessin du circuit imprimé du module de stockage et conversion (face composants)



Dessin du circuit imprimé du module de lecture (face soudures)



Dessin du circuit imprimé du module de lecture (face composants)