

34/84

ECOLE NATIONALE POLYTECHNIQUE

200

DEPARTEMENT : ELECTRONIQUE .

PROJET DE FIN D'ETUDES

SUJET

Conception et Realisation  
d'un Systeme d'Amelioration  
d'Images

Proposé par :

M. BESSALAH H.

Etudié par :

Melle KRIBES MEBARKA  
Melle ATTAB HACIBA

Dirigé par :

M. BESSALAH H.



PROMOTION : Janvier 1984

الجمهورية الجزائرية الديمقراطية الشعبية  
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

—»O«—

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

—»O«—

المدرسة الوطنية للعلوم الهندسية  
ECOLE NATIONALE POLYTECHNIQUE D'ALGER

—»O«—

DEPARTEMENT ELECTRONIQUE

—»O«—

CENTRE DE DEVELOPPEMENT DES TECHNIQUES AVANCEES  
LABORATOIRE ARCHITECTURE DES SYSTEMES

PROJET DE FIN D'ETUDES  
INGENIORAT EN ELECTRONIQUE

**THEME**

**Conception et Réalisation  
d'un Système d'Amélioration  
d'Images**

Proposé et suivi par :  
M. BESSALAH H.

Présenté par :  
Melle ATTAB Haciba  
Melle KRIBES Mebarka

Promotion Janvier 1984

D E D I C A C E S

A mon père et ma mère,  
A ma grand'mère  
A la mémoire de ma soeur Hlima  
A mes frères et sœurs  
A Salim, Mahrez, Naïma et Naouel,  
A toutes mes amies,

Mébarka,

A mes parents,

Haviba.

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

Nos plus vifs remerciements à M<sup>o</sup> BESSALAH H., Responsable du Laboratoire Architecture des systèmes dont l'aide et les conseils nous ont été particulièrement précieux, et pour les connaissances que nous avons pu acquérir au cours de notre projet.

Nous présentons l'expression de notre profonde gratitude à M<sup>o</sup> ABDELAOUI A., Directeur du Centre de Développement des Techniques Avancées, pour avoir bien voulu nous accueillir au C.D.T.A, et à Mme RASNAAMA Faïza, Secrétaire du C.D.T.A, pour avoir réglé tous nos problèmes administratifs.

Nous remercions vivement Melle HELIFAAA., M<sup>o</sup> MAHDI A., M<sup>o</sup> BELHADJ pour leur amicale collaboration.

Nous tenons à remercier M<sup>o</sup> BOUZID M., pour son amicale collaboration.

Nous tenons particulièrement à remercier Mme HAKEM R., pour le soin apporté à la dactylographie de cette thèse.

Nous remercions vivement le personnel technique pour leur aide compétente et cordiale.

Que tous ceux qui ont contribué à l'accomplissement de ce travail trouvent en ces lignes l'expression de notre reconnaissance.



# S O M M A I R E

INTRODUCTION .....	1
I - GENERALITES SUR LE TRAITEMENT DES IMAGES ET LEUR AMELIORATION .....	
1.1 - INTRODUCTION .....	4
1.2 - SEQUENCE DE TRAITEMENT .....	5
1.3 - OPERATIONS DE TRAITEMENT .....	5
1.3.1. MANIPULATION D'INTENSITE .....	5
1.3.2. OPERATIONS LOCALISEES .....	7
1.3.3. OPERATIONS GLOBALES .....	8
1.4 - OPERATIONS D'AMELIORATION D'IMAGES .....	8
1.4.1. AUGMENTATION DU PAS DE QUANTIFICATION .....	9
1.4.2. EXPANSION DE CONTRASTE .....	9
1.4.3. FILTRAGE SPATIAL PASSE-BAS .....	10
1.4.4. FILTRAGE SPATIAL PASSE-HAUT .....	12
1.4.5. DIFFERENTIATION STATISTIQUE .....	13
1.5 - CONCLUSION .....	15
2 - FILTRES NUMERIQUES BIDIMENSIONNELS	
2.1 - INTRODUCTION .....	16
2.2 - CHOIX DU TYPE DE FILTRE ET DES DIFFERENTES METHODES DE SYNTHESE .....	18
2.3 - SYNTHESE DES FILTRES NON RECURSIFS .....	20
2.4 - DETERMINATION DES COEFFICIENTS DU FILTRE .....	23
2.5 - METHODES DES FENETRES .....	26
2.6 - TRANSFORMATION DE MC CLELLAN POUR DES FILTRES DE DIMENSIONS $2 N \times 2 N$ .....	29
2.7 - CONCLUSION .....	31

3 - ETUDE DES METHODES ALGORITHMIQUES ET STRUCTURELLES DE REALISATION DE L'ALGORITHME DE BASE.	
3.1 - INTRODUCTION .....	32
3.2 - GENERALITES SUR LA REPRESENTATION DE MICROALGORITHMES.....	33
3.3 - METHODES CONVENTIONNELLES DE MULTIPLICATION ET LEURS MICROALGORITHMES .....	34
3.4 - METHODES ALGORITHMIQUES D'ACCELERATION DE LA MULTIPLICATION.	
3.5 - ASPECTS STRUCTURELS D'AMELIORATION DE LA VITESSE D'EXECUTION DE LA MULTIPLICATION .....	46
3.5.1. STRUCTURE TABULAIRE .....	46
3.5.2. STRUCTURE MATRICIELLE .....	49
3.5.3. STRUCTURE PIPELINE .....	52
3.6 - CONCLUSION .....	58
4 - ANALYSE COMPARATIVE ET SOLUTION RETENUE.	
4.1 - CONFIGURATION DU SYSTEME DE TRAITEMENT DU SIGNAL .....	59
4.2 - ANALYSE COMPARATIVE DES DIFFERENTES STRUCTURES DE REALISATION DE L'OPERATION DE BASE .....	61
4.3 - PRESENTATION DE LA SOLUTION RETENUE - SYSTEME PIPELINE-TABULAIRE D'AMELIORATION D'IMAGES .....	65
4.4 - CONCLUSION .....	68
5 - SYSTEME D'AMELIORATION D'IMAGES.	
5.1 - INTRODUCTION .....	69
5.2 - ARCHITECTURE DU SYSTEME D'AMELIORATION D'IMAGES .....	79
5.3 - LE SEQUENCEUR .....	74
5.4 - L'UNITE DE COMMANDE .....	80
5.5 - L'UNITE DE TRAITEMENT .....	83
5.6 - REALISATION D'UN SYSTEME D'AMELIORATION D'IMAGES .....	101
CONCLUSION .....	109

ANNEXE

## I N T R O D U C T I O N

Le traitement numérique du signal a pris depuis quelques années une ampleur particulière.

A l'origine, les méthodes de traitement numérique de signaux ont été développées dans le but de pouvoir simuler sur ordinateur numérique les systèmes analogiques de traitement de signaux, ce qui permettrait de vérifier rapidement les propriétés de ces systèmes et d'en optimiser les paramètres avant de les réaliser.

Actuellement, le traitement numérique du signal est couramment utilisé en télé-communication pour l'analyse et la synthèse de la parole, en traitement d'images, etc...

Le développement du traitement numérique du signal est dû aux nombreuses possibilités qu'il offre par rapport à un traitement analogique, et aussi au développement de la technologie des circuits intégrés digitaux qui a permis la réalisation d'équipements digitaux ayant une absence de dérive parfaite.

Parallèlement à cet effort technologique, un grand effort dans le domaine de la théorie des signaux et des systèmes a dû être fourni pour trouver des structures particulièrement bien adaptées aux exigences spécifiques de ce type de traitement.

L'introduction d'algorithmes de calcul rapides, tel que la transformée de Fourier rapide (F.F.T: Fast Fourier Transform) a permis d'effectuer des traitements très rapides jusque là longs et coûteux.

A l'origine, le traitement d'images était réservé à des images dont la valeur justifiait amplement un traitement coûteux.

.../...

Le traitement d'images a largement bénéficié de l'augmentation des performances et de la diminution de prise des équipements digitaux et est actuellement utilisé dans de nombreux domaines : photos aériennes, images, radiologiques, scintigraphie, thermographie, etc ... Et ceci, parce que l'acquisition de l'information dans ces domaines fait de plus en plus appel à l'emploi des images.

La procédure d'amélioration constitue l'une des opérations fondamentales du traitement d'images.

Cette procédure requiert l'exécution d'un très grand nombre d'opérateurs à une vitesse qui se chiffre par des millions d'opérations par seconde.

L'utilisation des gros ordinateurs s'est avérée non efficace de plusieurs points de vue : système d'exploitation non rentable, universalité non compatible avec le problème traité, encombrement, coût élevé, etc...

Ceci explique la tendance, actuelle de réalisation de systèmes spécialisés, orientés vers la résolution de problèmes concrets, par exemple les systèmes à architecture systolique.

C'est dans cet esprit que nous avons entamé notre projet qui consiste en la conception et la réalisation d'un système orienté vers la résolution des problèmes liés à l'amélioration d'images.

Notre projet s'articule sur cinq chapitres où sont reflétées les différentes étapes d'étude, de conception et de réalisation.

Le premier chapitre est consacré à une étude sur les généralités de traitement des images et leur amélioration.

Dans le deuxième chapitre est effectuée une étude détaillée sur les filtres numériques bidimensionnels.

Le troisième chapitre est consacré à l'étude des méthodes algorithmiques et structurelles de réalisation de l'opération de base.

Dans le quatrième chapitre, une étude comparative des différentes structures et algorithmes s'est faite dans le but de retenir une solution appropriée à notre réalisation.

Le cinquième chapitre est consacré à la conception et à la réalisation d'un système d'amélioration d'images.

CHAPITRE I CHAPITRE I

GENERALITES SUR LE TRAITEMENT DES IMAGES  
ET LEUR AMELIORATION

1 - INTRODUCTION

La procédure d'amélioration des images consiste en l'exécution d'un ensemble d'opérations dont le but est : soit, l'amélioration de la perception visuelle de l'image (exemple : cas de la télévision) soit, sa transformation sous une forme qui facilitera d'autres traitements numériques.

Les traitements d'amélioration tirent profit des connaissances accumulées par les physiologistes sur la vision humaine. Le nombre des opérations d'amélioration d'images susceptibles d'être effectuées est élevé.

Au début, nous donnerons la séquence de traitement, un aperçu sur les différentes catégories d'opérations existantes dans le traitement d'images, ensuite, nous donnerons quelques opérations d'améliorations d'images.



## 2 - Séquence des traitements. (1)

Le déroulement de la séquence de traitement peut être schématisé comme l'indique la figure 1.1.

Un tel découpage est artificiel dans la mesure où il peut y avoir recouvrement entre les différentes étapes. Certains traitements simples ne font appel qu'à une ou à quelques unes de ces étapes.

## 3 - Opérations de traitement (1)

Les opérations que l'on rencontre le plus fréquemment dans la séquence des traitements relèvent des méthodes traditionnelles du traitement du signal à savoir :

- les manipulations d'intensité ou opérations ponctuelles,
- les opérations localisées,
- les opérations globales.

### 3.1) Manipulation d'intensité

Le principe de traitement est le suivant : la valeur du point d'image résultat "Y" dépend de l'intensité "X" du point original de mêmes coordonnées. La même fonction "f (X)" est appliquée sur tous les points de l'image. Cette opération s'effectue en remplaçant l'intensité du point par la fonction de transfert.

$$Y (m,n) \longrightarrow f (X)$$

.../...

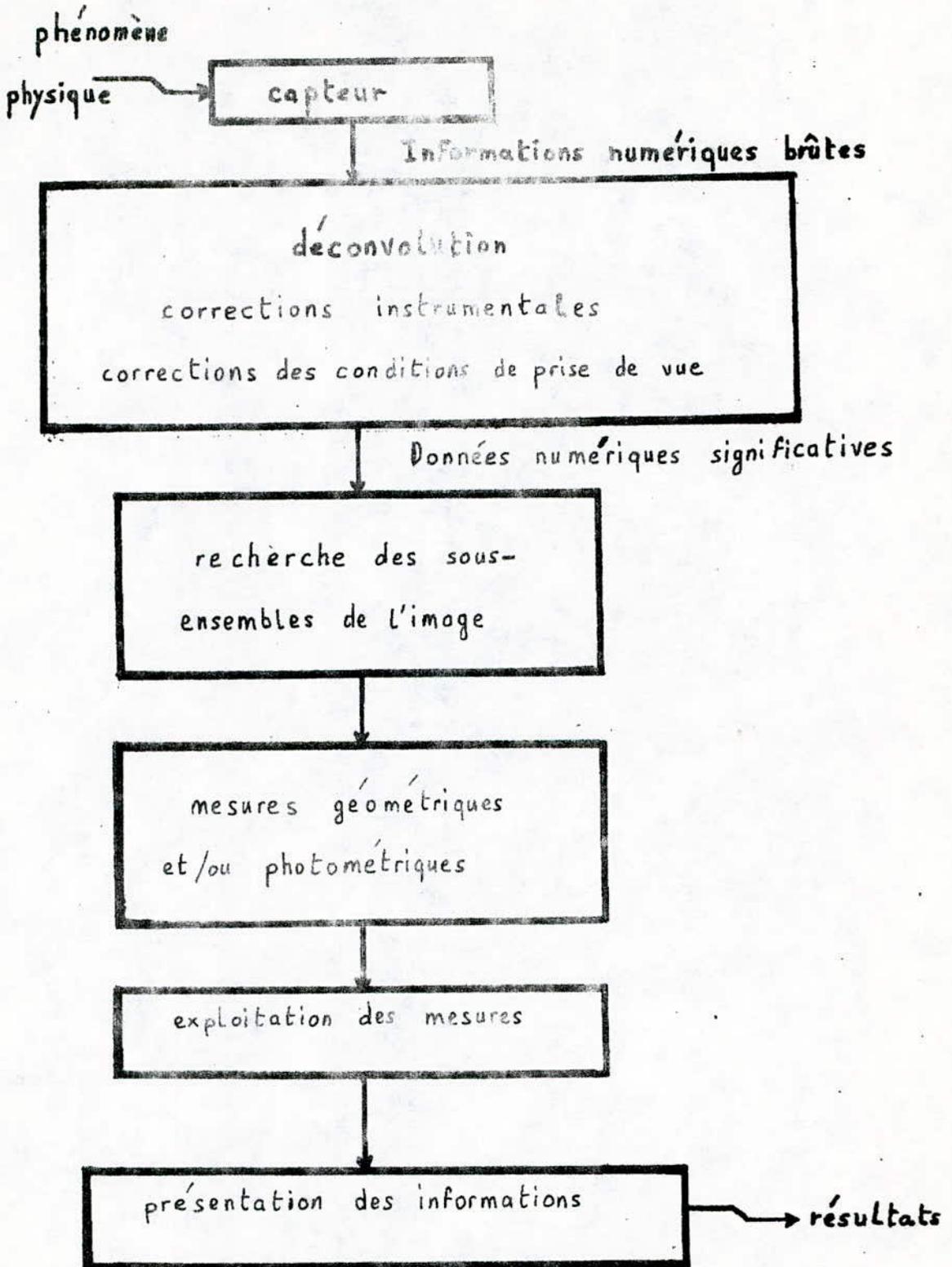


Fig. 1 - 1

Ce type de traitement est généralement réalisé par des tables de correction; il couvre une plage très étendue d'applications exemple correction de non linéarité des capteurs, la fixation d'un seuil, la réalisation de fenêtre, l'augmentation de contraste, etc ...

### 3.2) Opérations localisées

La détermination d'un point de l'image traitée de coordonnées (m, n) est fonction des points voisins du point correspondant de l'image originale. Les points voisins sont contenus dans une fenêtre rectangulaire, de dimensions 2b x 2a éléments. L'opérateur est un masque de mêmes dimensions. Le choix de la taille de la fenêtre est fonction de la résolution spatiale des informations recherchées.

Le masque est appliqué à tous les points de l'image.

Cette manipulation est une opération de convolution discrète.

Elle a la forme suivante :

$$Y(m,n) = \sum_{k=-b}^{b-1} \sum_{l=-a}^{a-1} X(m-k, n-l) c(k,l)$$

Les utilisations sont très variées suivant la matrice des coefficients C. Exemple : intégration, différenciation, manipulations de fréquence spatiales, etc ...

.../...

### 3.3) Opérations globales

Chaque élément de l'image résultat dépend de la totalité des éléments de l'image originale. La transformée de Fourier en constitue un exemple.

Le temps nécessaire pour exécuter ce type de traitement sur un ordinateur conventionnel est souvent très élevé. Afin de le réduire, des algorithmes rapides ont été mis au point tels que la transformée de Fourier rapide (FFT). Ces algorithmes sont, parfois adaptés aux processeurs parallèles ou effectués par des processeurs spécialisés.

### 4 - Opérations d'amélioration (1) - (7)

Dans les systèmes d'amélioration des images l'objectif n'est pas la restitution de l'image sous une forme proche d'un original idéal car un tel problème est résolu dans des cas liés à la restauration d'images. Lors du traitement numérique l'amélioration de l'image est liée au problème de l'extraction de l'information. Nous pouvons citer comme exemple un système d'amélioration d'images qui effectue l'accentuation des contours de l'image traitée par un filtrage en haute fréquence, l'image ainsi traitée est introduite dans un ordinateur qui effectue un lissage, et détermine sa forme et ses dimensions. Dans l'exemple précédent, le système d'amélioration est utilisé dans le but de souligner des indices remarquables de l'image originale, donc dans le but de faciliter l'extraction de l'information.

Dans ce paragraphe, nous examinerons quelques opérations qui permettent d'améliorer les conditions d'observation d'images et de faciliter leur traitement.

.../...

#### 4.1) Augmentation du pas de quantification

Cette opération a pour but d'obtenir moins de niveaux discrets en vue d'améliorer la perception visuelle de l'image, car le système visuel d'un observateur ne distingue qu'un nombre limité de niveaux de gris dans une image.

Le nombre de pas de quantification peut être modifié en appliquant la fonction suivante :

$$F(X) = \text{partie entière de } \left( \frac{X}{I \text{ pas}} \right) \cdot I \text{ pas}$$

cù  $I \text{ pas} = N_e / N_s$

$N_e$  : étant le nombre de niveaux de gris dans l'image originale

$N_s$  : étant le nombre de niveaux de gris voulu dans l'image résultat.

#### 4.2) Expansion de contraste

Le faible contraste constitue le défaut le plus répandu au sein des images photographiques et vidéo. La correction de telles images peut être réalisée en utilisant des méthodes photographiques seulement dans ce cas, la réalisation d'une fonction de transfert de niveau à grande précision est difficile. Dans le cas d'une image numérisée l'obtention est relativement facile, seulement dans le cas d'un opérateur non linéaire, il faudra tenir compte de l'erreur de l'échantillonnage.

##### 4.2.1 - Expansion avec saturation.

Lorsqu'une image, ou une zone de cette image est faiblement contrastée, les valeurs possibles de l'intensité n'occupent qu'une faible portion de la plage dynamique de digitalisation. Cela pro-

vient soit des conditions de prise de vue, soit de la nature de l'image. Le contraste est augmenté à l'aide de la fonction suivante:

$$f ( X ) = ( X - X_{\min} ) / F_c$$

$$\text{où } F_c = ( X_{\max} - X_{\min} ) / M$$

$X_{\max}$  et  $X_{\min}$  étant respectivement le maximum et le minimum d'intensité dans l'image.

$M$  : étant le nombre de niveaux de quantification.

#### 4.2.2 - Expansion avec dépassement d'échelle.

Cette opération est destinée à renforcer le contraste d'une image dont les intensités peuvent couvrir la totalité de la plage dynamique. Cette opération fait appel à la fonction suivante :

$$f ( X ) = ( X \cdot F_c ) \text{ modulo } 2^n$$

où  $F_c$  : est un facteur numérique

$n$  : est le nombre de bits de quantification de l'image.

Cette opération doit être utilisée avec précaution en raison de la génération possible de faux contours due au dépassement d'échelle.

#### 4.3) Filtrage spatial passe-bas.

L'influence des bruits de perturbation de natures différentes peut être minimisée en utilisant des méthodes classiques de filtrage. Une autre approche est fondée sur l'utilisation heuristique des méthodes de traitement. Le fait, qu'un élément déformé diffère de ses voisins, fût fondamental dans la mise au point de plusieurs algorithmes de suppression des bruits.

Exemple :

$O_1$	$O_2$	$O_3$
$O_8$	X	$O_4$
$O_7$	$O_6$	$O_5$

$$\text{si } ( X = \frac{1}{8} \sum_{i=1}^8 O_i )$$

$$\text{alors } X = \frac{1}{8} \sum_{i=1}^8 O_i$$

Vu que l'image a été déconvoluée, son spectre renferme des fréquences hautes (dus aux bruits).

En conséquence, un simple filtrage spatial à basse fréquence peut être utilisé comme moyen efficace de suppression des bruits. Le filtrage spatial passe bas revient à effectuer un lissage sur les données bidimensionnelles de l'image. Cette opération est effectuée, en général, lorsque le bruit est aléatoire et cohérent avec l'image. Cette méthode présente l'inconvénient de réduire la résolution spatiale de l'image.

A la base des méthodes de l'amélioration, la matrice de l'image traitée peut être obtenue à l'aide d'une convolution discrète de la matrice de l'image originale avec une matrice de filtre.

Les opérateurs de Laplace qui sont un exemple de ce type de filtrage utilisent les masques suivants :

$$\text{masque 1} \quad C = \frac{1}{9} \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

.../...

masque 2

$$C = \frac{1}{10}$$

$$\begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

masque 3

$$C = \frac{1}{16}$$

$$\begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}$$

Afin que la procédure de suppression du bruit n'entraîne pas un décalage de la luminance moyenne de l'image.

4.4) Filtrage spatial passe haut.

Ce type de filtrage a pour rôle d'augmenter les fréquences spatiales élevées de l'image.

Les expériences psychophysiques ont montré que les images photographiques ou vidéo ayant des contours accentués sont mieux perçues qu'une reproduction photographique idéale. Les méthodes d'accentuation des contours peuvent être réalisées de différentes façons.

L'accentuation des contours peut être exécutée en utilisant un filtrage numérique suivant la relation.

$$Y(m,n) = \sum_{k=-b}^{b-1} \sum_{l=-a}^{a-1} X(m-k, n-l) C(k,l)$$

.../...

Les masques suivants sont utilisés dans la réalisation d'un filtre H.F.

$$\begin{array}{l} \text{masque 1} \\ \text{masque 2} \\ \text{masque 3} \end{array} \quad C = \begin{array}{c} \left| \begin{array}{ccc} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{array} \right| \\ \left| \begin{array}{ccc} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{array} \right| \\ \left| \begin{array}{ccc} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{array} \right| \end{array}$$

Ces masques ont la particularité suivante : La somme de leurs éléments est égale à 1 (UN)

#### 4.5) Différenciation statistique

C'est une autre méthode d'accentuation des contours. La valeur de la luminance de chaque élément est divisée par l'écart de la moyenne quadratique  $\sigma(m, n)$

Chaque élément de la nouvelle image est donné sous cette forme..

$$Y(m, n) = X(m, n) \cdot \sigma(m, n)$$

où l'écart est égal à :

$$\sigma^2(m, n) = \sum_m \sum_n (X(m, n) - \bar{X}(m, n))^2$$

.../...

Cet écart est calculé dans le voisinage d'un point et a pour coordonnées (m,n)

La fonction  $\bar{X}$  correspond à la valeur moyenne de la luminance de l'image initiale dans un point ayant pour coordonnées (m,n)

L'image améliorée diffère de l'image initiale par le fait que la luminance est plus accentuée dans les contours, et elle est moindre dans les autres points de l'image.

Il faudra souligner que l'accentuation des contours entraîne l'accroissement du bruit.

L'opérateur proposé par Wallis R.H. (2) donne une image améliorée qui possède des moments du 1er et 2ème ordre. Cet opérateur est déterminé par l'expression suivante :

$$Y(m,n) = \left\{ \begin{array}{l} X(m,n) - \bar{X}(m,n) \\ \left( \frac{A \sigma_d}{A \sigma(m,n) + \sigma_d} \right) \end{array} \right\} + \left\{ \begin{array}{l} \alpha m_d + (1 - \alpha) \bar{X}(m,n) \end{array} \right\}$$

$m_d$  : étant la valeur moyenne

$\sigma_d$  : étant l'écart quadratique

$A$  : étant le coefficient d'amplification, introduit dans le but d'éliminer les grandes valeurs de luminance de l'image traitée.

$\alpha$  : étant le coefficient qui détermine la relation entre la luminance des contours et celle du fond de l'image améliorée.

.../...

5 - Conclusion.

Les résultats numériques de ces traitements d'amélioration sont, soit utilisés tels quels en vue d'autres traitements soit reproduits sous forme d'image visible sur un écran.

La plupart des opérations utilisées dans l'amélioration sont des opérations localisées et elles font appel à l'algorithme suivant :

$$Y(m,n) = \sum_{k=-b}^{b-1} \dots \sum_{l=-a}^{a-1} X(m-k, n-l) C(k,l)$$

Les traitements numériques envisagés influent sur le choix concernant la conception d'un système.

Le système d'amélioration d'image qui sera étudié et réalisé dans le cadre de ce travail utilise l'algorithme précédent. Les utilisations de cet algorithme varient suivant la matrice des coefficients  $C$  (filtre) dont l'étude fait l'objet du chapitre suivant (deux).

## CHAPITRE 2

### FILTRES NUMERIQUES BIDIMENSIONNELS

#### 2.1 - INTRODUCTION

Au cours de leur transmission et de leur acquisition les signaux peuvent être altérés par l'influence de phénomènes fluctuants dont l'évolution ne peut pas être prévue exactement.

La catégorie de traitement qui permet l'élimination, l'atténuation des informations inutiles en faveur des informations utiles est la fonction filtrage. Dans le cas de l'analyse et de l'amélioration d'images, cette catégorie de traitement permet d'exploiter au mieux l'information contenue dans une image (signal bidimensionnel).

L'avènement des réalisations numériques des filtres est dû à l'essor des techniques d'intégration à grande échelle et aux nombreuses possibilités qu'elles offrent par rapport aux réalisations équivalentes analogiques. Les réalisations numériques ont une absence de dérives parfaite, et sont souvent d'une conception plus simple que leurs équivalentes analogiques.

Un filtre numérique bidimensionnel T est défini par un algorithme de calcul, par lequel une séquence de nombres  $\{x(m,n)\}$ , dite séquence d'entrée, est transformée en une autre séquence de nombre  $\{y(m,n)\}$ , dite séquence de sortie. Le filtre effectue la transformation :

$$\{y(m,n)\} = T \left[ \{x(m,n)\} \right]$$

.../...

D'une manière générale, l'algorithme, de calcul d'un filtre numérique bidimensionnel est la relation de récurrence du type :

$$y(m,n) = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} x(k,l) \times h(m-k, n-l)$$

où  $h(m,n)$ , sont les coefficients du filtre.

La séquence  $\{h(m,n)\}$  est la séquence réponse impulsionnelle. C'est la réponse à une séquence de la forme.

$$d(m,n) = \begin{cases} 1 & (m,n) = (0,0) \\ 0 & (m,n) \neq (0,0) \end{cases}$$

Il existe deux types de filtres numériques : les filtres non réursifs (FIR, Finite impulse response) et les filtres réursifs (IIR, infinite impulse response).

Le calcul de ces filtres se fait par généralisation des méthodes appliquées au calcul des filtres unidimensionnels.

Dans le cas des FIR un élément de la séquence filtrée ( $y(m,n)$ ) ne dépend que de la séquence à filtrer.

Les coefficients du filtre sont issus soit de la troncature d'une réponse impulsionnelle infinie, soit d'un nombre fini d'échantillons de la réponse en fréquence du filtre  $A(w_1, w_2)$

En ce qui concerne les IIR, les éléments de la séquence filtrée dépendent d'éléments récemment calculés et de conditions initiales.

.../...

L'étude théorique des systèmes linéaires et invariants bénéficie des résultats considérables établis en algèbre linéaire, c'est pourquoi la plupart des filtres numériques utilisés actuellement vérifient ces propriétés :

- linéarité : le système qui est caractérisé par l'opérateur  $T[\cdot]$  est linéaire si pour deux entrées arbitraires  $x_1(m,n)$  et  $x_2(m,n)$ , et une constante complexe arbitraire  $c$ , on a :

$$T[x_1(m,n) + c x_2(m,n)] = T[x_1(m,n)] + c T[x_2(m,n)]$$

- invariance temporelle : si

$$y(m,n) = T[x(m,n)]$$

alors le système caractérisé par  $T[\cdot]$  est dit invariant si et seulement si

$$y(m-m_0, n-n_0) = T[x(m-m_0, n-n_0)]$$

pour tout  $x$  et pour tous les entiers arbitraires  $m_0$  et  $n_0$ .

## 2.2 - CHOIX DU TYPE DE FILTRES ET DES DIFFERENTES METHODES DE SYNTHESE.

### 2.2.1 - CHOIX DU TYPE DE FILTRE

Le système d'amélioration d'images qui sera développé dans la partie réalisation est destiné essentiellement pour l'application de filtres non récursifs.

.../...

Les avantages de ces filtres sont :

- grande stabilité,
- linéarité en phase,
- la causalité est peu importante, et la sommation est souvent effectuée d'une manière symétrique

Vu les avantages présentés par les RIF, cités ci-dessus, et le choix du type de système à réaliser, ils seront les seuls étudiés dans ce chapitre.

### 2.2.2 - METHODES DE SYNTHESE

Plusieurs méthodes de synthèse des filtres ont été proposées. Ces méthodes se divisent en deux catégories :

- les méthodes non itératives basées sur l'utilisation de la transformée de Fourier. Elles s'appliquent dans le domaine fréquentiel par la DFT (Discret Fourier Transform).
- les méthodes itératives appliquées directement dans le domaine spatial

La détermination des coefficients d'un filtre est un problème d'approximation. Deux critères sont utilisés dans la pratique, le critère de tchebycheff et le critère des moindres carrés.

A cause de certaines difficultés (par exemple la recherche des extrêmes sur une grille bidimensionnelle est beaucoup plus longue que sur une grille unidimensionnelle, cette recherche prend une grande partie du temps de calcul) ces techniques sont limitées à des filtres de petites dimensions, et des techniques suboptimales sont utilisées.

Parmi les méthodes suboptimales seules deux seront étudiées dans ce chapitre ; la méthode dite "méthode des fenêtres" et la méthode qui utilise la transformation de Mc Clellan.

Le critère des moindres carrés sera utilisé dans le cas où la réponse en fréquence n'est pas connue sous une forme analytique.

- La méthode des fenêtres : elle utilise une fenêtre de troncature sur le développement en série de Fourier, se généralise facilement à deux dimensions.

- La transformation de Mc Clellan : le filtre bidimensionnel est obtenu à partir d'un filtre unidimensionnel plus facile à calculer

### 2.3 - SYNTHÈSE DES FILTRES NON RECURSIFS

#### 2.3.1 - INTERPRETATION DANS LE DOMAINE TEMPOREL

Par définition, la séquence réponse impulsionnelle

$\{h(m, n)\}$  est la réponse à la séquence  $\{\delta(m, n)\}$

où  $\delta(m, n) = \begin{cases} 1 & (m, n) = (0, 0) \\ 0 & (m, n) \neq (0, 0) \end{cases}$

soit  $\{x(m, n)\}$  la séquence d'entrée

$$\{x(m, n)\} = \sum_k \sum_l x(k, l) \{\delta(m-k, n-l)\}$$

La réponse du filtre à la séquence  $\{\delta(m-k, n-l)\}$   
 est la séquence  $\{h(m-k, n-l)\}$  (invariance temporelle).

La réponse à la séquence  $x(k, l) \{\delta(m-k, n-l)\}$   
 est donc

$$x(k, l) \cdot \{h(m-k, n-l)\}$$

La séquence  $\{y(m, n)\}$  est la réponse à la séquence

$$\{x(m, n)\}$$

soit

$$\{y(m, n)\} = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} x(k, l) \{h(m-k, n-l)\}$$

soit  $\forall m, n$

$$y(m, n) = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} x(k, l) \cdot h(m-k, n-l)$$

En faisant un changement de variables  $\begin{cases} k = m - k \\ l = n - l \end{cases}$

$y(m, n)$  s'écrira

$$y(m, n) = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} h(k, l) x(m-k, n-l)$$

Pour un filtre causal

$$h(k, l) = 0 \quad \text{pour } k < 0 \text{ et } l < 0$$

et  $y(m, n)$  s'écrira

$$y(m, n) = \sum_{k=0}^{+\infty} \sum_{l=0}^{+\infty} h(k, l) x(m-k, n-l)$$

Si le filtre est à réponse impulsionnelle finie alors

$$h(m, n) = 0 \quad \text{pour } (m, n) \notin I_{MN}$$

$$I_{MN} = \left\{ (m, n) / 0 \leq m \leq M-1, \quad 0 \leq n \leq N-1 \right\}$$

N = nombre de colonnes

M = nombre de lignes

d'où l'équation qui constitue l'algorithme de calcul des filtres RIF :

$$y(m, n) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h(k, l) x(m-k, n-l)$$

2.3.2 - REPONSE EN FREQUENCE A UNE SEQUENCE DE LA FORME  
 $e^{j(\omega_1 m + \omega_2 n)}$

Les exponentielles complexes numérisées, de la forme :

$$x(m, n) = W^m z^n$$

sont fonctions propres des filtres numériques

$$\begin{aligned} x(m-k, n-l) &= W^{m-k} z^{n-l} \\ &= x(m, n) W^{-k} z^{-l} \end{aligned}$$

$$y(m, n) = x(m, n) \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h(k, l) W^{-k} z^{-l}$$

où  $\sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h(k, l) W^{-k} z^{-l}$  est la valeur

propre associée

$$\text{cas où } \begin{cases} w = e^{j\omega_1} \\ z = e^{j\omega_2} \end{cases}$$

La réponse en fréquence à une séquence de la forme  $e^{j(\omega_1 m + \omega_2 n)}$ , pour un filtre non récursif est par définition :

$$A(\omega_1, \omega_2) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h(k, l) e^{-j(\omega_1 k + \omega_2 l)}$$

## 2.4 - DETERMINATION DES COEFFICIENTS DU FILTRE

### 2.4.1 - DECOMPOSITION EN SERIE DE FOURIER

La réponse d'un filtre bidimensionnel est périodisée par l'échantillonnage, la réponse souhaitée  $D(\omega_1, \omega_2)$  doit également être périodique, elle peut donc être développée en série de Fourier comme toute fonction périodique.

En tronquant la série, on obtient une approximation de la réponse en fréquence.

La réponse en fréquence d'un filtre numérique étant analogue à un développement en série de Fourier tronquée, il suffit de prendre des coefficients du développement comme coefficients du filtre .

- cas des filtres unidimensionnels

$$D(\omega) = \sum_{n=-\infty}^{+\infty} C(n) e^{jn\omega}$$

avec

$$C(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} D(\omega) e^{-jn\omega} d\omega$$

- cas des filtres bidimensionnels

$$D(\omega_1, \omega_2) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} C(m, n) e^{j(m\omega_1 + n\omega_2)}$$

avec

$$C(m, n) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} D(\omega_1, \omega_2) e^{-j(m\omega_1 + n\omega_2)} d\omega_1 d\omega_2$$

La réponse aux autres fréquences étant alors fixée par repliement.

#### 2.4.2 - DECOMPOSITION EN SERIE DE FOURIER PAR DFT (9)

Quand la fonction à développer en série de Fourier n'est pas connue sous une forme analytique, le calcul des séries se fera par DFT (Discret Fourier Transform). La sommation continue dans le domaine des fréquences sera approximée par une sommation sur une grille discrète. Si la grille est suffisamment dense, l'approximation au sens des moindres carrés par un polynôme trigonométrique sur une grille discrète sera très voisine de celle obtenue sur le domaine continu.

Les coefficients peuvent alors être déterminés par DFT.

La réponse du filtre est donné par

$$A(\omega_1, \omega_2) = \sum_n \sum_m h(m, n) e^{-j(m\omega_1 + n\omega_2)} \\ \approx D(\omega_1, \omega_2)$$

Soit  $\omega_1 = \frac{2\pi k}{K}$  ,  $\omega_2 = \frac{2\pi l}{L}$

Les points de la grille discrète

avec  $\begin{cases} 0 \leq k < K \\ 0 \leq l < L \end{cases}$

Sur ces points la réponse du filtre est donnée par ;

$$D\left(\frac{2\pi k}{K}, \frac{2\pi l}{L}\right) \approx \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} h(m, n) e^{-j2\pi\left(\frac{km}{M} + \frac{ln}{N}\right)} \\ + \sum_{n=L-N}^{L-1} \sum_{m=K-M}^{K-1} h(m-K, n-L) e^{-j2\pi\left(\frac{km}{K} + \frac{ln}{L}\right)}$$

Les  $h(m, n)$  peuvent être calculés par DFT des échantillons de  $D$ . Sur les points de la grille, la DFT complète n'introduit pas d'erreur (interpolation). Les fonctions de base de la DFT sont orthogonales donc les éléments de la série tronquée (le membre de droite de l'équation précédente) représentent la meilleure approximation des échantillons de  $D$  au sens des moindres carrés. Ces considérations conduisent à l'algorithme suivant :

- 1 i-1 - Interpoler la réponse en fréquence souhaitée sur  $K \times L$  échantillons.

$$G(k, l) = D \left( \frac{2\pi k}{K}, \frac{2\pi l}{L} \right) \quad \text{avec } \begin{cases} 0 \leq k < K \\ 0 \leq l < L \end{cases}$$

i-2 Calculer la DFT inverse de la séquence  $G(K, l)$ ,

soit  $g(m, n)$  la séquence obtenue.

i-3 Tronquer la séquence  $g(m, n)$  ; les coefficients du filtre sont donnés par :

$$h(m, n) = g(m, n) W(m, n) \quad \text{avec } \begin{cases} 0 \leq m \leq M \\ 0 \leq n \leq N \end{cases}$$

$$h(m, n) = g[L-n, K-m] W(m, n) \quad \text{avec } \begin{cases} -N \leq n \leq 0 \\ -M \leq m < 0 \end{cases}$$

$W(m, n)$  étant la fenêtre de troncature.

i-4 Tester la courbe de réponse obtenue par DFT de la séquence  $g(m, n)$ .

## 2.5 - METHODES DES FENETRES

### 2.5.1 - METHODES DES FENETRES

Elle est fondée sur la propriété de périodicité de la réponse en fréquence du filtre numérique.

Le développement en série de Fourier mène à une réponse impulsionnelle infinie. On se ramènera à un nombre fini de termes de la séquence réponse, impulsionnelle en opérant une troncature de la séquence à l'aide d'une fenêtre de troncature.

.../...

Les fenêtres de troncature bidimensionnelles  $W(m,n)$  peuvent être obtenues à partir des fenêtres unidimensionnelles.

Les fenêtres de troncature sont données sous deux formes :

1ère forme :  $W(m,n) = W(m) \cdot W(n)$

2ème forme :  $W(m,n) = W(\sqrt{m^2 + n^2})$

$W(m)$  et  $W(n)$  étant des fenêtres unidimensionnelles

$W(m,n) = W(m) \cdot W(n)$  donne un filtre carré.

$W(m,n) = W(\sqrt{m^2 + n^2})$  donne un filtre approximativement circulaire.

La série tronquée sera donc

$$\{C_0(m,n)\} = \{C(m,n) \cdot W(m,n)\}$$

$\{C_0(m,n)\}$  est la séquence réponse impulsionnelle finie.

### 2.5.2 - DIFFERENTES CATEGORIES DE FENETRES

Les fenêtres de troncature se divisent en deux catégories : les fenêtres paramétriques et les fenêtres non paramétriques.

Dans le cadre des filtres numériques, les principales fenêtres non paramétriques sont définies par les relations suivantes :

.../...

- fenêtre triangulaire :

$$W(n) = 1 - |n|/M+1$$

$$W(n) = 0$$

$$|n| \leq M$$

$$|n| > M$$

Cette fenêtre a la particularité suivante : elle est considérée comme la convolution de deux fenêtres rectangulaires, sa TF est non négative, ce qui entraîne que la réponse d'un filtre tend vers la réponse idéale d'une manière monotone.

- Hanning : (pour un filtre de longueur impaire)

$$W(n) = 0,5 \left( 1 + \cos \frac{\pi n}{M+1} \right)$$

$$M = \frac{N-1}{2}$$

Elle a des lobes secondaires d'amplitude beaucoup plus faible que la fenêtre rectangulaire, leur décroissance est également plus rapide.

- Blackman :

$$W(n) = 0,42 + 0,5 \cos \frac{\pi n}{M+1} + 0,08 \cos \frac{2\pi n}{M+1}$$

$$\text{pour } n = -M, M$$

$$W(n) = 0$$

$$\text{pour } n > M$$

Elle a des lobes secondaires beaucoup plus faibles au détriment de la largeur du lobe central.

Les fenêtres citées ci-dessus ont été retenues, pour leur simplicité (exemple : leur mise au point) et leur efficacité (due à leurs particularités).

2.6 - TRANSFORMATION DE MC CLELLAN POUR FILTRES DE DIMENSIONS  
 $2N \times 2N$  (5)

Mc Clellan a proposé une transformation qui permet d'obtenir un filtre bidimensionnel à partir d'un filtre unidimensionnel. Considérons le filtre digital unidimensionnel à phase linéaire de longueur  $2N$ .

La réponse impulsionnelle  $h(n)$  du filtre est non nulle pour  
 $-N \leq n \leq N$

Puisque le filtre est à phase linéaire,  $h(n)$  obéit à la contrainte symétrique.

$$h(n) = h(1-n) \text{ pour } -N+1 \leq n < N \quad (1)$$

La réponse en fréquence du filtre est la suivante :

$$A(e^{j\omega}) = e^{-j\frac{\omega}{2}} \sum_{n=1}^N h(n) \cdot 2 \cdot \cos \omega \left( n - \frac{1}{2} \right) \quad (2)$$

La relation (2) montre que  $A(e^{j\pi}) = 0$ . Ainsi on peut écrire

$$A(z) = (1+z^{-1}) \hat{A}(z) \quad (3)$$

Similairement, si  $h(m,n)$  est la réponse impulsionnelle du filtre numérique de dimensions  $2M \times 2M$ , à phase linéaire, qui est non nulle pour  
 $-M+1 \leq m \leq M$   
 $-M+1 \leq n \leq M$

La réponse en fréquence d'un tel filtre est la suivante :

$$A(e^{j\omega_1}, e^{j\omega_2}) = e^{-j\left(\frac{\omega_1 + \omega_2}{2}\right)} \sum_{m=1}^M \sum_{n=1}^M h(m, n) \cos \omega_1 \left(m - \frac{1}{2}\right) \cos \omega_2 \left(n - \frac{1}{2}\right) \quad (4)$$

La relation (4) montre que :

$$A(e^{j\pi}, e^{j\omega_2}) = A(e^{j\omega_1}, e^{j\pi}) = 0$$

Ainsi, on peut écrire :

$$A(z_1, z_2) = (1 + z_1^{-1}) (1 + z_2^{-1}) \hat{A}(z_1, z_2) \quad (5)$$

$\hat{A}(z)$  est de longueur  $2N-1$ , similairement

$\hat{A}(z_1, z_2)$  est de dimensions  $(2M-1) \times (2M-1)$ .

Ainsi, il est possible de définir des filtres  $A(z_1, z_2)$  à partir des filtres  $A(z)$  par utilisation de la transformation suivante :

$$\cos w = \sum_{p=0}^P \sum_{q=0}^Q t(p, q) \cos p\omega_1 \cdot \cos q\omega_2 \quad \text{avec } P=Q=1$$

d'où

$$\cos w = t(0,0) + t(0,1) \cos \omega_2 + t(1,0) \cos \omega_1 + t(1,1) \cos \omega_1 \cos \omega_2$$

(Les filtres à symétrie circulaire peuvent être approxiés par  $t(1,1) = t(0,1) = t(1,0) = -t(0,0) = 0,5$ )

D'où l'algorithme suivant :

- i-1 Détermination de  $2N$  termes de  $h(n)$ ,
- i-2 De  $h(n)$  on détermine une séquence  $\hat{h}(n)$  ayant  $(2N-1)$  termes, où  $\hat{h}(n)$  est la transformée en  $z$  inverse de  $\hat{A}(z)$  définie en (3).

i-3 En utilisant la transformation de Mc Clellan, on transformera  $\hat{h}(n)$  en  $\hat{h}(m,n)$ , le filtre résultant est de dimensions  $(2M-1) \times (2M-1)$

i-4 La convolution de  $\hat{h}(M,n)$  par  $p(m,n)$  donne  $h(n,n)$   
 $p(m,n)$  étant la transformée en Z inverse de  $p(z_1, z_2)$

$$P(z_1, z_2) = (1 + z_1^{-1}) (1 + z_2^{-1})$$

Le filtre résultant de cet algorithme est un filtre à phase linéaire et de dimensions  $2N \times 2N$ .

Cette procédure nous conduit aux filtres optimaux seulement dans le cas des contours de la transformation utilisée en (i-3), correspondant aux contours d'égalité d'amplitude.

$$|P(e^{j\omega_1}, e^{j\omega_2})| = Cte$$

## 2.7 - CONCLUSION

L'objectif de ce chapitre était l'étude et la maîtrise des méthodes de détermination des coefficients du filtre.

Le calcul des coefficients n'a pas été fait parce que la partie acquisition de données ne fait pas l'objet de ce travail.

L'étude précédente montre que le filtre non récursif est caractérisé par sa réponse impulsionnelle ou sa réponse en fréquence, ceci s'explique par le fait que :

Les termes de la réponse impulsionnelle sont les coefficients du filtre,

Connaissant la réponse en fréquence, nous pouvons déterminer les coefficients du filtre (par exemple en effectuant une décomposition en série de Fourier).

Les filtres non récursifs restent les plus adaptés à la déconvolution d'images comme nous l'avons vu précédemment.

### CHAPITRE 3

#### ETUDE DES METHODES ALGORITHMIQUES ET STRUCTURELLES DE REALISATION DE L'ALGORITHME DE BASE

##### 3.1 INTRODUCTION

Dans ce chapitre sera faite l'étude des facteurs les plus pondérants dans le choix de la structure du processeur d'amélioration d'images. Il est évident que les algorithmes présentés dans les chapitres précédents peuvent être programmés et mis en oeuvre sur un ordinateur universel, seulement, le coût et le temps de traitement seront excessivement élevés. Une augmentation de la vitesse de traitement peut être obtenue par l'utilisation de processeurs spécialisés, réalisant les différentes fonctions nécessaires à l'amélioration d'images.

Suivant la relation :

$$Z(m, n) = \sum_{k=-b}^{b-1} \sum_{l=-a}^{a-1} X(m-k, n-l) \cdot Y(k, l)$$

les algorithmes d'amélioration d'images ont deux particularités qui militent en faveur de leur réalisation sur des processeurs spécialisés :

1 - L'opération de base à réaliser est du type :

$$\sum x_i y_i$$

.../...

2 - Le nombre d'entrées-sorties est relativement faible par rapport aux nombres d'opérations arithmétiques.

Ces particularités signifient que la charge essentielle retombe sur l'unité arithmétique d'où la nécessité de concevoir et de réaliser un processeur très performant du point de vue vitesse de traitement, ainsi que du point de vue rapidité et compatibilité de transmission de données entre la mémoire et l'unité de traitement.

Toutes ces réflexions nous ont amené à faire une étude approfondie des aspects algorithmiques et structurels de réalisation de l'opération de base.

### 3.2 - GENERALITES SUR LA REPRESENTATION DES MICROALGORITHMES :

L'opération de base est la multiplication dont l'exécution consiste à réaliser, une suite d'opérations élémentaires (des micro-opérations) sur des registres, des additionneurs et des compteurs.

On peut citer différents types de micro-instructions nécessaires à l'exécution de l'opération de base :

- Transfert de données (TD),

exemple : d'un registre vers un autre registre ou vers une mémoire.

- Transfert de données complémentaires: (TDC),

(en complément à 1 ou à 2)

- Initialisation à zéro : (CLR)

(d'un registre, d'un compteur, etc...)

.../...

- Réception de données directes : (RD)
- Réception de données inversées : (RI)
- Décalage à droite dans un registre : (ASR)
- Décalage à gauche dans un registre : (ASL)
- Incrémentation de l'accumulateur : (INC)
- Décrémentement de l'accumulateur : (DEC)

Une suite de micro-instructions dont l'exécution permet la réalisation de l'opération considérée est appelée "micro-algorithme"

La réalisation de ces micro-algorithmes peut se faire à l'aide de commandes micro-programmées ou câblées.

### 3.3 - METHODES CONVENTIONNELLES DE MULTIPLICATION ET LEURS MICROALGORITHMES :

Lors de la multiplication en mode direct de deux nombres, le bit du signe et les bits de la mantisse sont traités séparément.

La multiplication des mantisses peut être effectuée soit par les bits de plus fort poids (MSB), ou par les bits de plus faible poids (LSB) du multiplieur. Dans les deux cas, nous pouvons faire un décalage soit des produits partiels, soit du multiplicande.

La combinaison de ces possibilités nous permet d'en déduire quatre méthodes de multiplication qu'on appellera "méthodes conventionnelles".

## 3.3.1 - PREMIERE METHODE

Par X et Y on désigne respectivement le multiplieur et le multiplicande.

d'où le produit Z de X par Y se définit par la relation :

$$Z = X \cdot Y = Y [x_1 \cdot 2^{-1} + x_2 \cdot 2^{-2} + \dots + x_{n-1} \cdot 2^{-n+1} + x_n \cdot 2^{-n}]$$

Suivant la formule de HORNER (en prenant à chaque fois  $2^{-i}$  en facteur) la dernière expression prend la forme suivante :

$$Z = 2^{-1} [y x_1 + 2^{-1} (y x_2 + 2^{-1} (y x_3 + \dots + 2^{-1} (y x_n + 0))) \dots]$$

Posons :

$$\begin{aligned} P_1 &= (P_0 + y x_n) 2^{-1} \\ P_2 &= (P_1 + y x_{n-1}) 2^{-1} \\ P_3 &= (P_2 + y x_{n-2}) 2^{-1} \end{aligned}$$

et tirons la formule de récurrence :

$$P_{i+1} = (P_i + y x_{n-i}) 2^{-1} \quad ; \quad i = 0, 1, \dots, n.$$

$$P_0 = 0, \text{ pour } i = 0.$$

on déduit :

$$Z = P_n = (P_{n-1} + y x_1) 2^{-1}$$

Cette méthode est dite multiplication par le bit de plus faible poids du multiplieur et décalage à droite du produit partiel.

La réalisation de cette méthode nécessite l'utilisation :

- d'un registre (RGY) pour le multiplicande sur lequel arrivent les signaux de commande qui réalisent la multiplication par  $x_{n-1} \dots x_{n-i}$ .

.../...

- d'un additionneur pour la réalisation de l'addition de  $P_{i-1}$  avec  $y_{x-n-i}$
- d'un registre (RGZ) pour le produit Z ayant une longueur double de celle du multiplicateur sur lequel arrivent les signaux de décalage à droite ainsi que le stockage de  $P_{i-1}$
- d'un registre à décalage (RGX) pour l'opérande X qui permet de stocker la valeur du multiplicateur et son décalage à droite.

Pour la réalisation de cette méthode, on propose un micro-algorithme  $\mu m_1$  qui peut être facilement traduit en organigramme.

$$\mu m_1 = (\text{DEBT}) (\text{CLR. COMPT}) \downarrow (\text{TD. RGY}) (X_n = 0) \uparrow (\text{RD. RGZ}) \downarrow \\ (\text{ASR. RGZ}) (\text{ASR. RGX}) (\text{INC. COMPT}) (\text{COMPT} = n) \uparrow (\text{END})$$

L'interprétation de ce micro-algorithme est la suivante :

- 1) Initialisation à zéro du compteur de cycles
- 2) Transmission des données dans (RGY) (transmission du multiplicande)
- 3) Test du dernier bit de plus faible poids du multiplicateur
  - s'il est égal à (1), nous avons une transmission du multiplicande dans le registre du produit (RGZ)
  - s'il est égal à (0), nous suivons la flèche (1) c'est à dire :
    - décalage d'un bit à droite du produit partiel
    - décalage d'un bit à droite du multiplicateur
- 3 - Incrémentation du compteur d'une unité et comparaison de son contenu avec le nombre n (nbre de bits du multiplicateur X) :

-s'il est inférieur à (n) on suit la flèche (2), on reprend le cycle.

(s'il est égal à (n) on arrête le processus.

Ce micro-algorithme peut être donné sous forme d'organigramme (voit fig. 3.3. 1.a)

Nous illustrons cette méthode par l'exemple suivant :

Soit deux nombres binaires non signés :

$$Y = 0110$$

$$X = 0101$$

Pour l'exécution de la multiplication se référer à la figure 3.3. 1.b.

### 3.3.2 - DEUXIEME METHODE

Le produit Z s'écrit :

$$Z = XY = 2^{-n} (y_1 2^{n-1} x_1 + y_2 2^{n-2} x_2 + \dots + y_{n-1} 2^1 x_{n-1} + y_n 2^0 x_n)$$

avec un décalage de Y

$$Z = 2^{-n} (y_{n-1} x_1 + y_{n-2} x_2 + \dots + y_1 x_{n-1} + y_0 x_n)$$

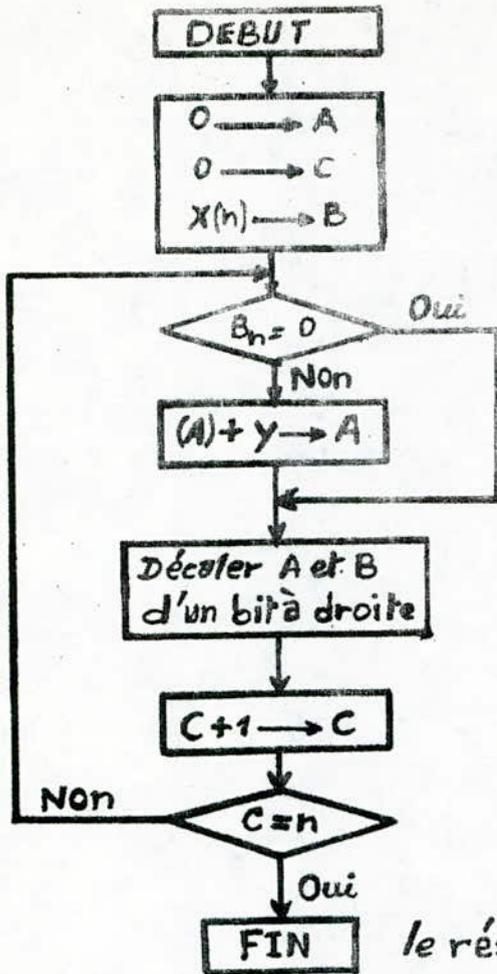
où  $y_i = y 2^i = y_{i-1} \cdot 2$  ; (décalage à gauche de Y)

Formule de récurrence

$$\begin{cases} P_{i+1} = P_i + y_i x_{n-i} & , \quad i = 0, 1, \dots, n. \\ P_i = 0 & , \quad \text{pour } i = 0 \end{cases}$$

Cette méthode est dite multiplication par le bit de plus faible poids du multiplieur avec décalage à gauche du multiplicande.

.../...



A = accumulateur (sur n bits).  
 B = " " " " "  
 Y = multiplicande (n bits).  
 X = multiplieur (n bits).  
 C = compteur.

le résultat est sur A et B.

a) organigramme de l'algorithme d'addition décalage.  
 (nombres non signés)

Faire  $(A) + Y \rightarrow A$   
 Décalage à droite  
 $(A) + 0 \rightarrow A$   
 Décalage à droite  
 $(A) + Y \rightarrow A$   
 Décalage à droite  
 $(A) + 0 \rightarrow A$   
 Décalage à droite

A				B			
0	0	0	0	0	1	0	1
0	1	1	0	0	1	0	1
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	0	1	1	0	0	1
0	1	1	1	1	0	0	1
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	0	1	1	1	1	0

b) illustration de l'algorithme d'addition décalage  
 (deux nombres binaires sur 4 bits).

Fig. 3.3.1. Illustration de la 1<sup>ère</sup> méthode conventionnelle.

Micro-algorithme :

$M_{m_2} = (\text{DEBT}) (\text{CLR. COMPT}) \downarrow (\text{TD. RG Z}) (\text{TD. RG Y}) (X_n = 0) \uparrow$   
 $(\text{RD. RG Z}) \downarrow (\text{ASL. RG Y}) (\text{ASR. RG X}) (\text{INC. COMPT})$   
 $(\text{COMPT} = n) \uparrow (\text{END}).$

### 3.3.3 - TROISIEME METHODE

Cette méthode consiste à scruter le multiplieur bit par bit en partant des bits de plus fort poids dans le but de générer n produits partiels et de décaler à chaque fois le produit partiel d'un bit à gauche.

Le multiplieur X est défini par :

$$X = \sum_{i=1}^n x_i 2^{n-i}$$

d'où le produit Z :

$$Z = X Y = 2^{-n} \left( \dots \left( (y x_1 2^{n-1}) + y x_2 2^{n-2} \right) + \dots + y x_{n-1} 2^1 + y x_n \right)$$

Formule de récurrence :

$$\begin{cases} P_{i+1} = P_i \cdot 2 + y x_{i+1} + 1 \\ P_i = 0, \text{ pour } i = 0 \end{cases}$$

Micro-Algorithme :

$M_{m_3} = (\text{DEBT}) (\text{CLR. COMPT}) \downarrow (\text{TD. RG Y}) (\text{TD. RG Z}) (X_n = 0) \uparrow$   
 $(\text{RD. RG Z}) \downarrow (\text{ASL. RG Z}) (\text{ASL. RG X}) (\text{INC. COMPT})$   
 $(\text{COMPT} = n) \uparrow (\text{END}).$

## 3.3.4 - QUATRIEME METHODE

Cette méthode consiste à effectuer  $n$  itérations du type :

$$\begin{cases} P_i = P_{i-1} + x_i y_i \\ y_i = y 2^{-i} = y_{i-1} 2^{-1} \\ P_i = 0, \text{ pour } i=0. \end{cases}$$

Le produit est donné par la relation suivante :

$$Z = XY = x_1 y_1 + x_2 y_2 + \dots + x_{n-1} y_{n-1} + x_n y_n$$

Micro-algorithme :

$$M_{nb} = (\text{DEBT}) (\text{CLR.COMPT}) \downarrow^2 (\text{TD.RGX}) (\text{TD.RGZ}) (x_1=0) \uparrow^1 \\ (\text{RD.RGZ}) \downarrow^1 (\text{ASL.RGX}) (\text{ASR.RGY}) (\text{INC.COMPT}) \\ (\text{COMPT} = n) \uparrow^1 (\text{END}).$$

Cette méthode consiste à faire la multiplication par les bits de plus fort poids du multiplieur, et décalage à droite du multiplicande.

L'analyse comparative de ces quatre méthodes permet de faire les conclusions suivantes :

Du point de vue matériel, vitesse et efficacité, l'unité de commande réalisée à partir de la première méthode est plus simple, alors que celle de la quatrième est la plus efficace ; ceci explique le fait que la première et la quatrième méthodes de multiplication sont les plus utilisées.

.../...

L'un des problèmes rencontrés lors de la multiplication de deux nombres est celui de l'arrondissement des résultats.

A l'exception de la rapidité obtenue lors de l'utilisation de la première et quatrième méthode, des dernières permettent de réaliser les opérations du type  $\sum x_i y_i$  et ce, en décalant le produit partiel dans l'une et le multiplicande dans l'autre.

### 3.4 - METHODES ALGORITHMIQUES D'ACCELERATION DE LA MULTIPLICATION.

3.4.1 - Une première méthode consiste à analyser le multiplieur dans le but de chercher les "1" dont nous tiendrons compte et, les "0" que nous omettons. ET, de ce fait, le nombre des produits partiels est égal à N tel que N représente le nombre de "1" contenus dans le multiplieur. La probabilité d'avoir 1 ou 0 est la même.

Pour cette méthode, on peut proposer le micro-algorithme suivant :

$$M_m = (\text{DEBT}) (\text{CLR. COMPT}) \downarrow (x_n = 0) \uparrow (\text{TD RGY} \times \text{RD. RGZ}) \downarrow$$

$$(\text{ASR. RBY}) (\text{ASR. RGZ}) (\text{INC. COMPT})$$

$$(\text{COMPT} \times N) \uparrow^2 (\text{END}).$$

Dans ce cas, le temps d'une multiplication est égal à :

$$T_m = N \left( \frac{t_a}{2} + t_c \right)$$

Tels que :

$\frac{t_a}{2}$  : temps d'addition

$t_c$  : temps d'analyse de  $x_n$ .

### 3.4.2 - METHODE DE BOOTH (2) :

C'est un algorithme proposé, pour multiplier directement des nombres signés (code complément à 2).

Cet algorithme utilise un recodage du multiplieur. Dans ce cas le multiplieur n'est plus examiné bit par bit mais par groupes de deux bits; chaque groupe ayant un bit en commun avec le précédent.

La représentation d'un mot de 8 bits suivant ce code est illustrée par la figure 3.4.2.

L'exploitation de ces deux bits se fait comme indiqué dans le tableau 1.

Le fait d'examiner le multiplieur par groupes de deux bits, ne change pas le nombre de produits partiels mais permet de traiter le bit signe de la même façon que les autres bits.

Un algorithme dérivé de celui de BOOTH a été proposé, toujours dans le but du traitement direct des nombres signés.

Le multiplieur dans ce cas, est examiné par groupes de trois bits, ce qui permet de réduire le nombre de produits partiels dans un rapport de deux. Les trois bits sont exploités tel que précisé dans le tableau 2. Cet algorithme est réalisé à l'aide de circuits multiplieurs binaires parallèles 2 x 4 bits.

.../...

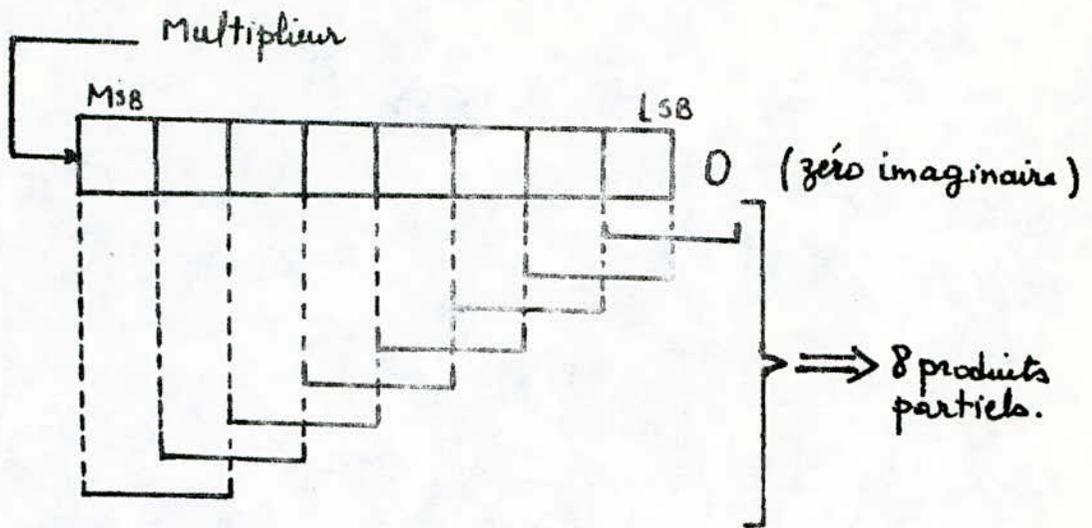


Fig. 3.4.2. REPRESENTATION DU MULTIPLIEUR  
DANS LA METHODE DE BOOTH.

$X_i$	$X_{i-1}$	Symbole	Action.
0	0	0	Faire $Z + 0 \rightarrow Z$
0	1	(+1)	Faire $Z + Y \rightarrow Z$
1	0	(-1)	Faire $Z - Y \rightarrow Z$
1	1	0	Faire $Z - 0 \rightarrow Z$

Tableau 1: EXAMEN DU MULTIPLIEUR  
PAR GROUPES DE DEUX BITS.

$X_{i+1}$	$X_i$	$X_{i-1}$	Symb	Action.
0	0	0	0	$Z + 0 \rightarrow Z$
0	0	1	(+1)	$Z + Y \rightarrow Z$
0	1	0	(+1)	$Z + Y \rightarrow Z$
0	1	1	(+2)	$Z + 2Y \rightarrow Z$
1	0	0	(-2)	$Z - 2Y \rightarrow Z$
1	0	1	(-1)	$Z - Y \rightarrow Z$
1	1	0	(-1)	$Z - Y \rightarrow Z$
1	1	1	0	$Z - 0 \rightarrow Z$

Tableau 2: EXAMEN DU MULTIPLIEUR  
PAR GROUPE DE TROIS BITS.

Cette méthode s'avère très rapide du fait qu'on a un produit de cinq bits en 26 ns. Mais elle demande beaucoup de matériel du fait que pour 16 x 16 bits, il faut environ cent circuits.

Cette méthode appliquée dans un système diminuerait sa fiabilité et le rendrait encombrant.

Le temps d'exécution d'une opération est égal au temps d'analyse du multiplieur ta plus le temps de l'opération propre ; pour N opérations on aura :

$$T = N (ta + top)$$

3.4.3 - METHODE DE RECHERCHE EN TABLE. (TABULAIRE) : ( )

Le principe consiste ici à stocker tous les résultats possibles dans une mémoire morte (ROM), l'adresse étant constituée par l'ensemble des deux opérandes.

Pour effectuer une multiplication m x n bits la taille de la mémoire nécessaire pour assurer un résultat exact est de  $2^{(m+n)}$  mots de (m+n) bits.

Nous pouvons voir que cette méthode n'est concevable que pour des petits formats vu que le nombre de mots mémoire accroit considérablement avec le nombre de bits des mots à multiplier.

Le temps d'exécution d'une opération dépend du temps d'accès à la mémoire ; plus ce dernier est court plus l'exécution d'une opération est rapide.

Si on désigne par :

$t_e$  : le temps d'accès à la mémoire

$t$  : le temps d'addition

$N$  : le nombre d'opérations à effectuer

Le temps d'exécution de ces  $N$  opérations est donné par la relation suivante :

$$T = N (t + t_e)$$

Posons  $t + t_e = t_{op}$  ( temps d'une opération )

Le temps d'exécution de  $N$  opérations est :

$$T = N t_{op}$$

### 3.5 - ASPECTS STRUCTURELS D'AMELIORATION DE LA VITESSE D'EXECUTION DE LA MULTIPLICATION

Dans le but d'améliorer la rapidité d'exécution de l'opération de base, nous citons les trois structures suivantes :

#### 3.5.1 - STRUCTURE TABULAIRE

Le principe est simple : Il se résume à une sommation séquentielle des produits partiels issus de la mémoire (par accès à cette dernière) ou de circuits multiplicateurs disposés en parallèle.

Les circuits considérés effectuent la multiplication  $4 \times 4$  bits ce qui nécessite la subdivision des nombres à multiplier en groupes de 4 bits.

.../...

Examinons la structure pour deux nombres de 8 bits, soit les deux nombres :

$$X = X_1 + X_2 \quad , \text{ multiplieur}$$

$$\text{avec } X_1 = x_1 \ x_2 \ x_3 \ x_4 \ 0000$$

$$X_2 = 0 \ 0 \ 0 \ 0 \ x_5 \ x_6 \ x_7 \ x_8$$

$$Y = Y_1 + Y_2 \quad , \text{ multiplicande}$$

$$\text{avec : } Y_1 = y_1 \ y_2 \ y_3 \ y_4 \ 0000$$

$$Y_2 = 0000 \ y_5 \ y_6 \ y_7 \ y_8$$

d'où le produit de y par x :

$$Z = X Y = (X_1 + X_2) (Y_1 + Y_2) = X_1 Y_1 + X_1 Y_2 + X_2 Y_1 + X_2 Y_2$$

$X_1 Y_1$  : (MSB) plus fort poids

$X_2 Y_2$  : (LSB) plus faible poids

où  $X_i Y_j$  : produit partiel sur 8 bits.

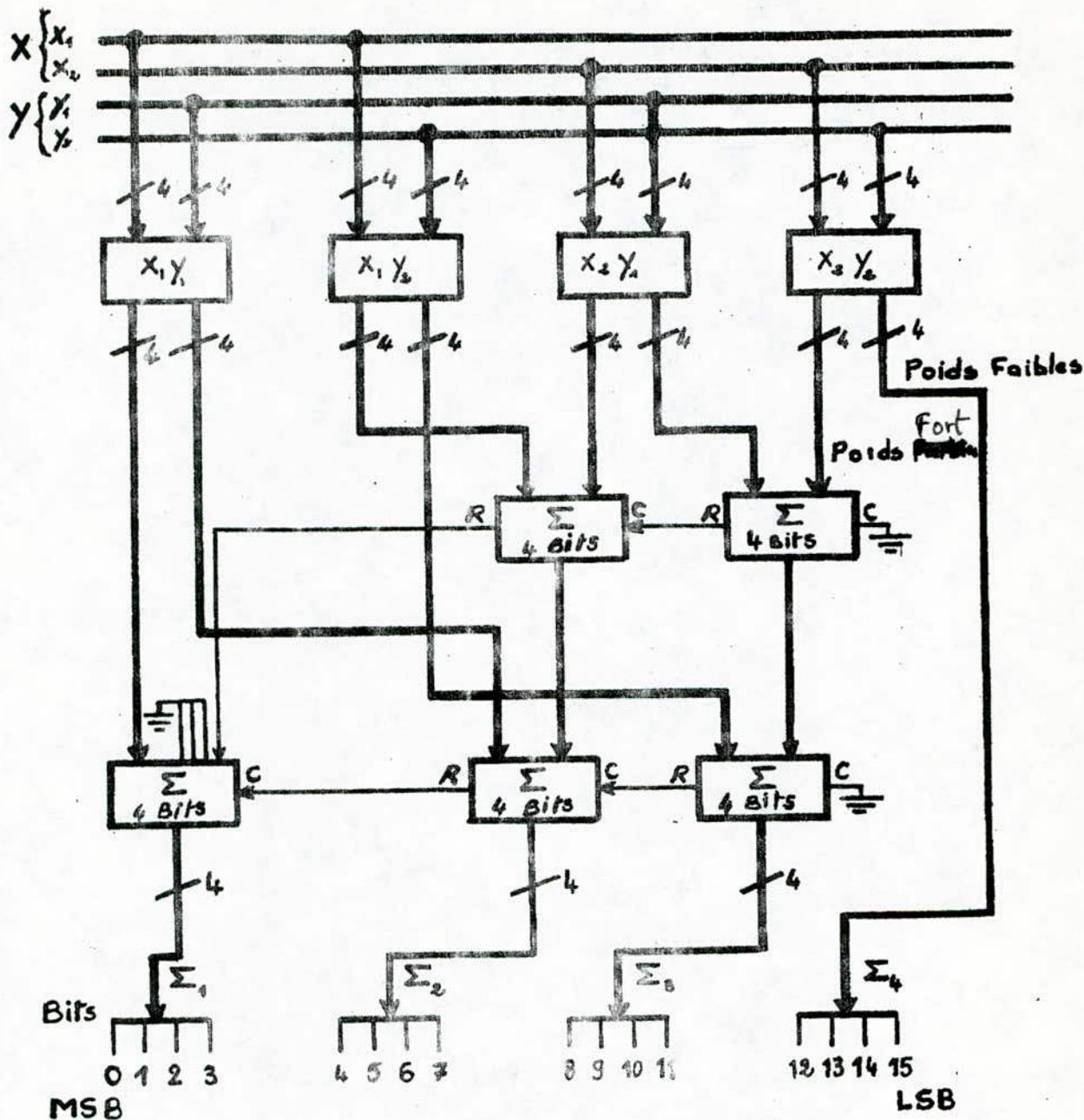
Cette procédure est illustrée par la figure (3.5.1)

Le temps nécessaire pour l'exécution d'une opération dépend de la rapidité des circuits.

Si on doit faire N opérations, le temps total nécessaire est N fois le temps d'exécution d'une opération ( $\tau$ )

$$T = N \tau$$

Donc avec cette structure, pour lancer une deuxième opération il faut attendre que la précédente soit exécutée.



Multiplication de deux nombres de 8bits

Fig.3.5.1 . STRUCTURE TABULAIRE.

### 3.5.2 - STRUCTURE MATRICIELLE: (6)

Cette structure telle que schématisée par la figure (3.5.2.a) peut se déduire directement à partir de l'algorithme de multiplication habituel.

Il s'agit d'une structure combinatoire formée d'une cascade d'étages d'additionneurs complets et de portes ET (AND) utilisées pour le calcul des produits élémentaires.

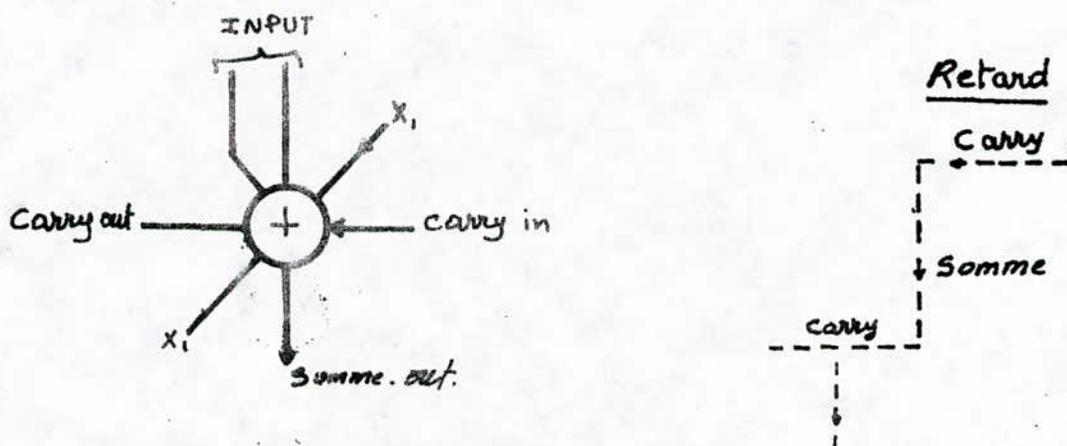
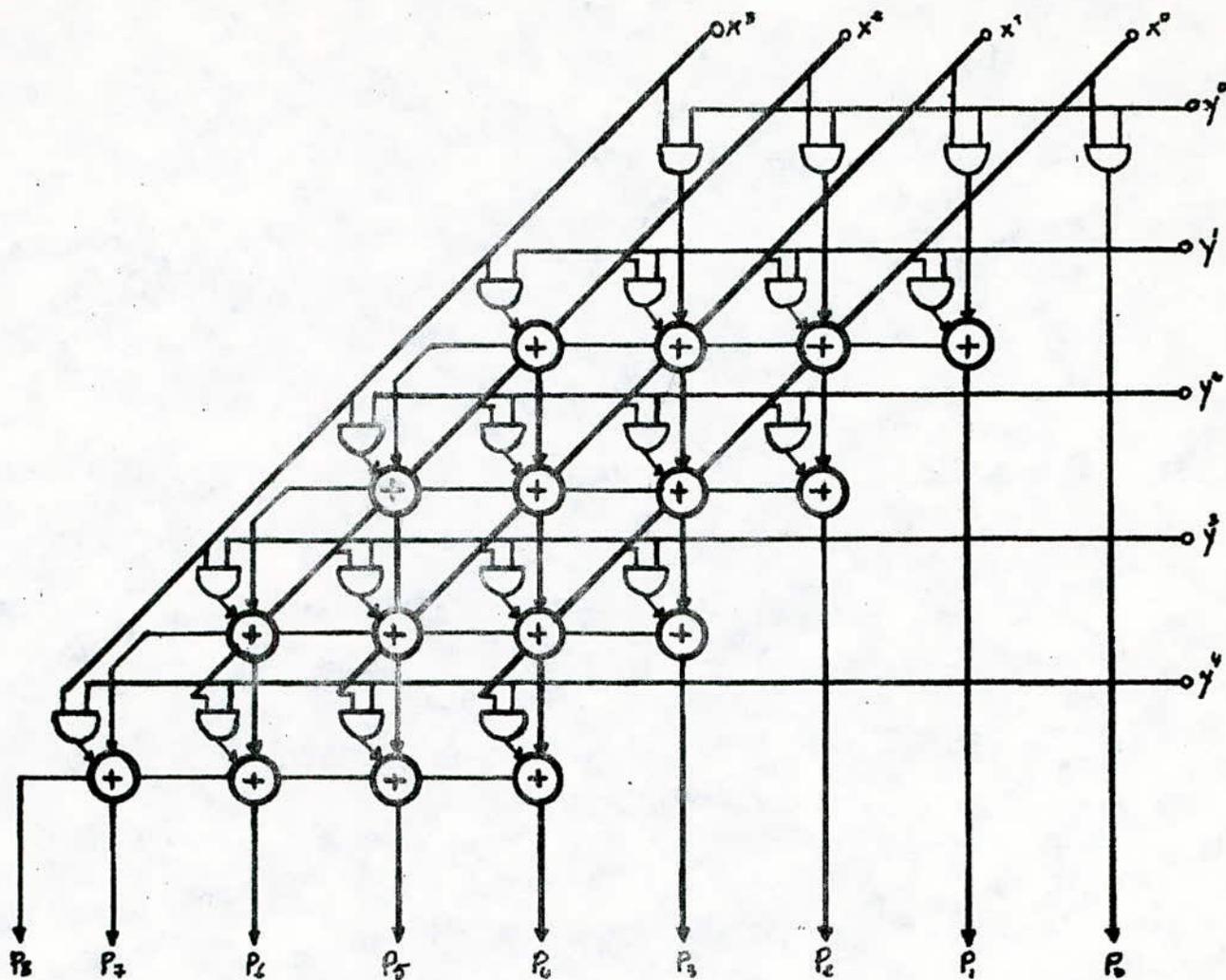
La rapidité de cette structure, bien que nettement supérieure à celle des multiplicateurs série-série ou série-parallèle, est néanmoins limitée par la transmission du carry dans chaque étage et par la transmission des sommes partielles d'un étage à l'autre, (voir dessin fig.(3.5.2.a)).

Cet inconvénient peut être évité par la transmission simultanées (en parallèle) des retenues et des sommes partielles d'un étage à l'autre (retenues transmises en diagonales) sauf pour le dernier étage où la retenue doit se propager horizontalement si on ne prévoit pas un carry look-ahead (voir fig. 3.5.2.b).

Dans ce cas, une compétition apparaît dans la transmission verticale des sommes partielles et diagonales des retenues.

Dans les deux structures considérées, les sommes partielles des produits de même poids sont réalisées successivement le long d'une même colonne du tableau.

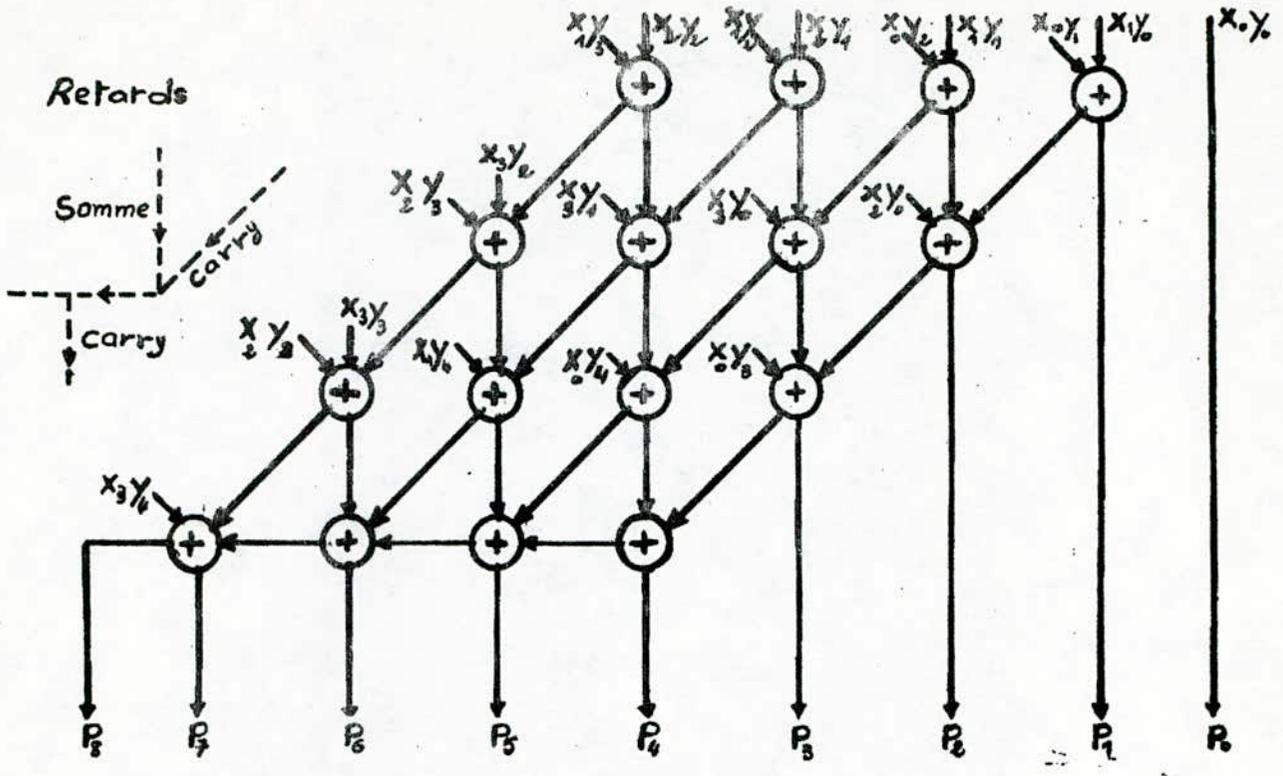
.../...



a) STRUCTURE MATRICIELLE

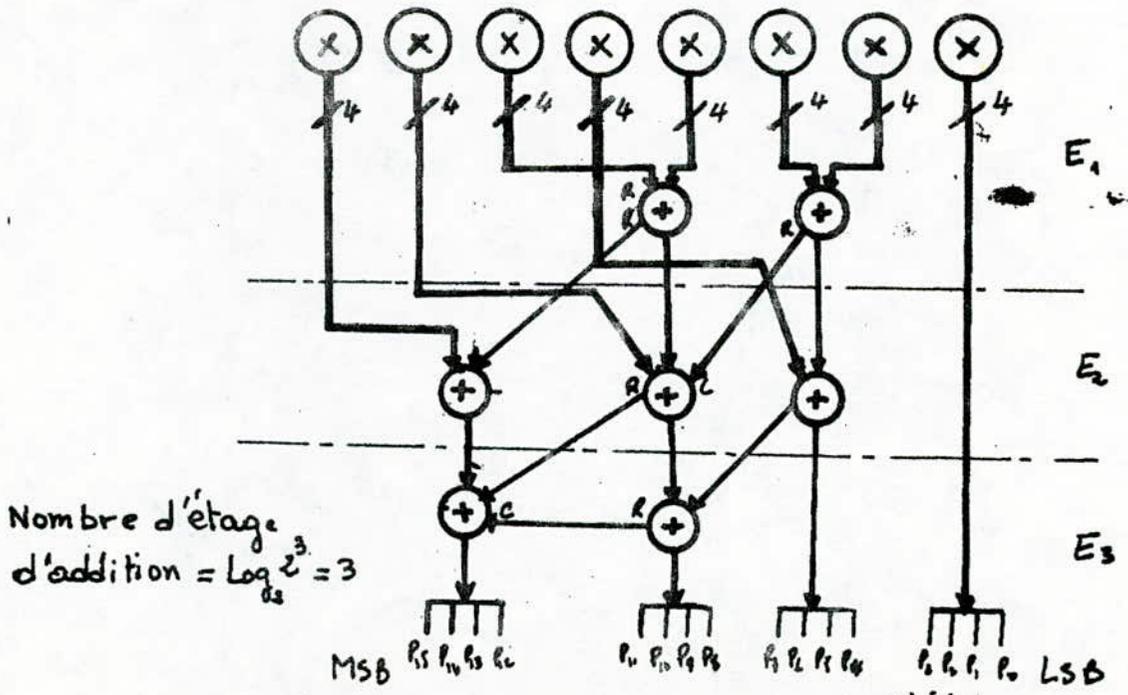
Multiplicateur 4x5 bits - Carry-horizontale.

Fig. 3.5.2.



**b) STRUCTURE MATRICIELLE**

Multiplicateur 4x5 bits - Carry diagonal



**STRUCTURE EN ARBRE. (8x8 bits)**

FIG: 35.2.

Dans le cas où le retard causé par la transmission verticale des sommes partielles est plus grand que le retard dû à la transmission diagonale du carry, l'introduction d'une structure en arbre sur les sommes partielles permet de réduire le temps nécessaire pour effectuer celle-ci et d'augmenter la vitesse du circuit.

Dans la structure en arbre, on somme simultanément deux à deux, les  $n/2$  sommes partielles obtenues et, ainsi de suite ; le nombre d'étage d'addition se réduit à  $\log_2 n$ , (voir fig. 3.5.2.c)

### 3.5.3 - STRUCTURE EN PIPELINE : ( )

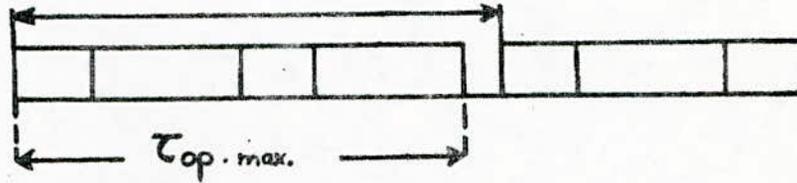
La notion de structure en pipeline d'une opération peut être expliquée de façon générale avec les dessins et les diagrammes de la figure (3.5.3 a,b)

Dans la structure :

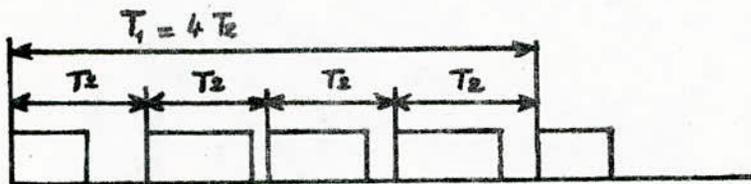
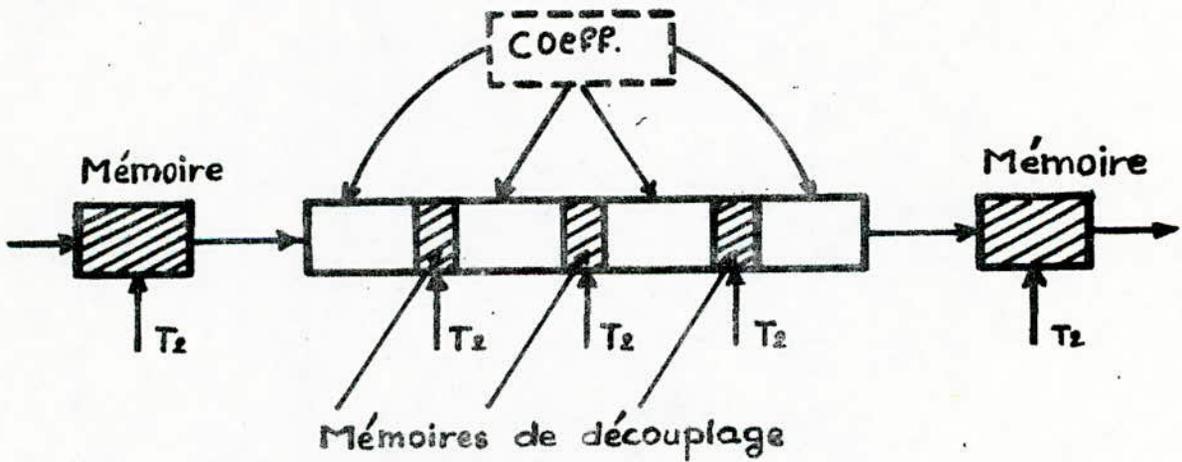
a) Sans pipeline : le temps nécessaire pour effectuer toute l'opération étant au pire des cas égale à  $T_{opmax}$ , la période de l'horloge qui commande le début de chaque nouvelle opération est fixée à  $T_1 \gg T_{opmax}$ .

b) Avec pipeline : l'idée fondamentale du "pipeline" consiste à séparer, par l'introduction de mémoire de découplage dans le matériel qui réalise l'opération, l'opération en sous-opérations commandées par une période d'horloge  $T_2$  plus courte que  $T_1$ .

.../...



a) opération sans pipeline.



b) opération avec pipeline.

Fig 3.5.3

Bien que le temps total nécessaire pour effectuer toute l'opération devienne supérieur à  $T_1$ , la présence de mémoire de découplage dans la ligne d'opération permet d'introduire une nouvelle donnée après chaque période  $T_2$ , avant même que le traitement des précédentes soit terminé (voir fig. 3.5.3.b).

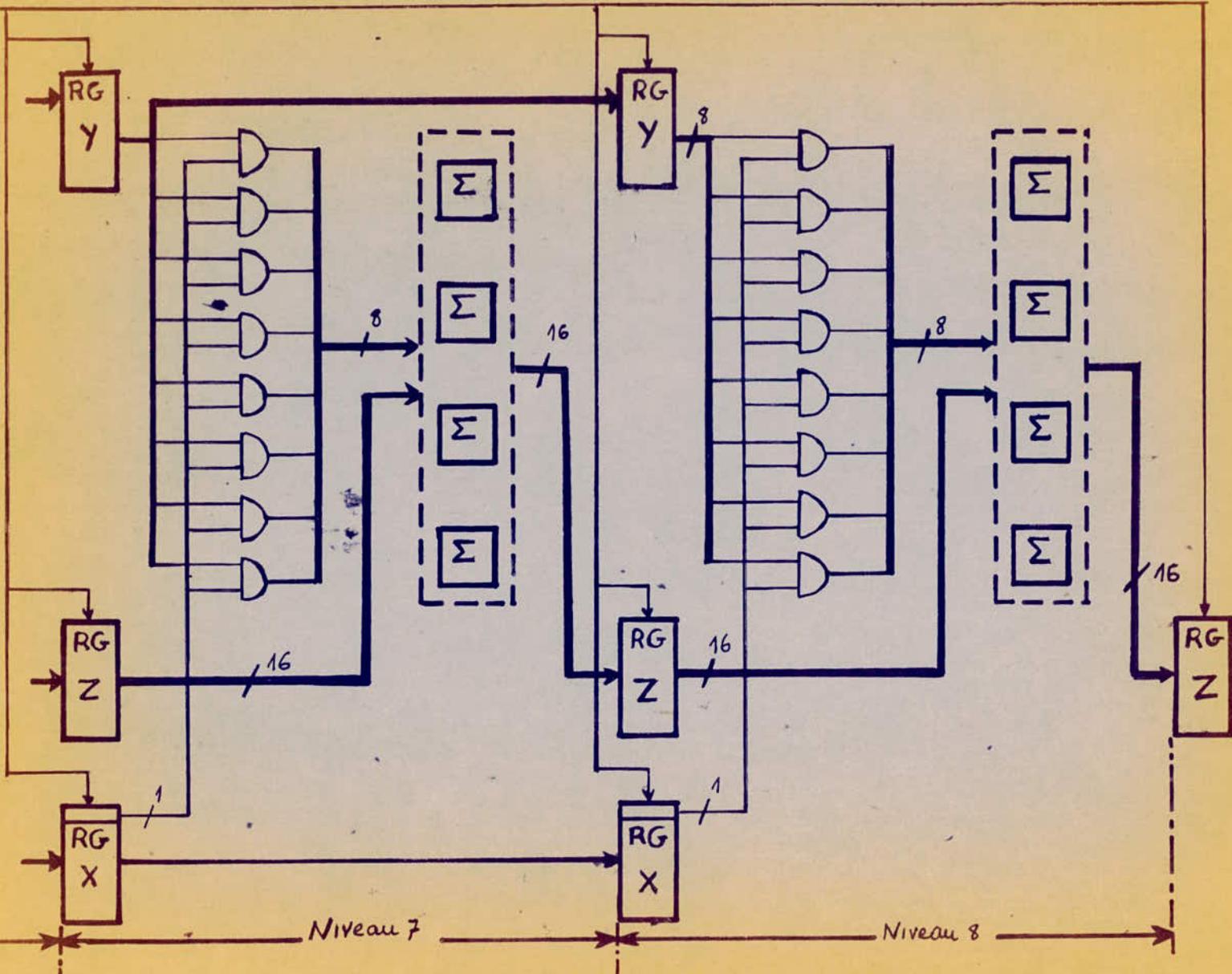
Ainsi, la structure en pipeline permet d'augmenter sensiblement la fréquence de travail, sans devoir multiplier le matériel. Cette notion est très utile et intéressante dans la réalisation des multiplieurs en pipeline.

Illustrons la structure pipeline classique, constituée de registres, additionneurs et portes (AND) pour effectuer la multiplication élémentaire (1x1 bit).

- le multiplicande (Y) et le multiplieur (X) sont contenus dans des registres (8 bits), (registre à décalage pour le multiplieur)
- le produit partiel doit être stocké dans un registre à décalage (16 bits)
- Des portes (AND) pour la multiplication
- Un accumulateur pour effectuer la somme partielle

Dans la structure pipeline l'opération est divisée en un nombre de sous-opérations égal au nombre de bit du multiplieur ( $n$  bits). On appelle niveau une sous-opération. Par exemple, si le multiplieur est donné sur huit bits on a huit sous-opérations (niveaux). Cette structure est bien expliquée dans la figure (3.5.3.c).

.../...



LINE - MULTIPLICATION DE DEUX MOTS (SUR 8 bits).

Chaîne de synchron.

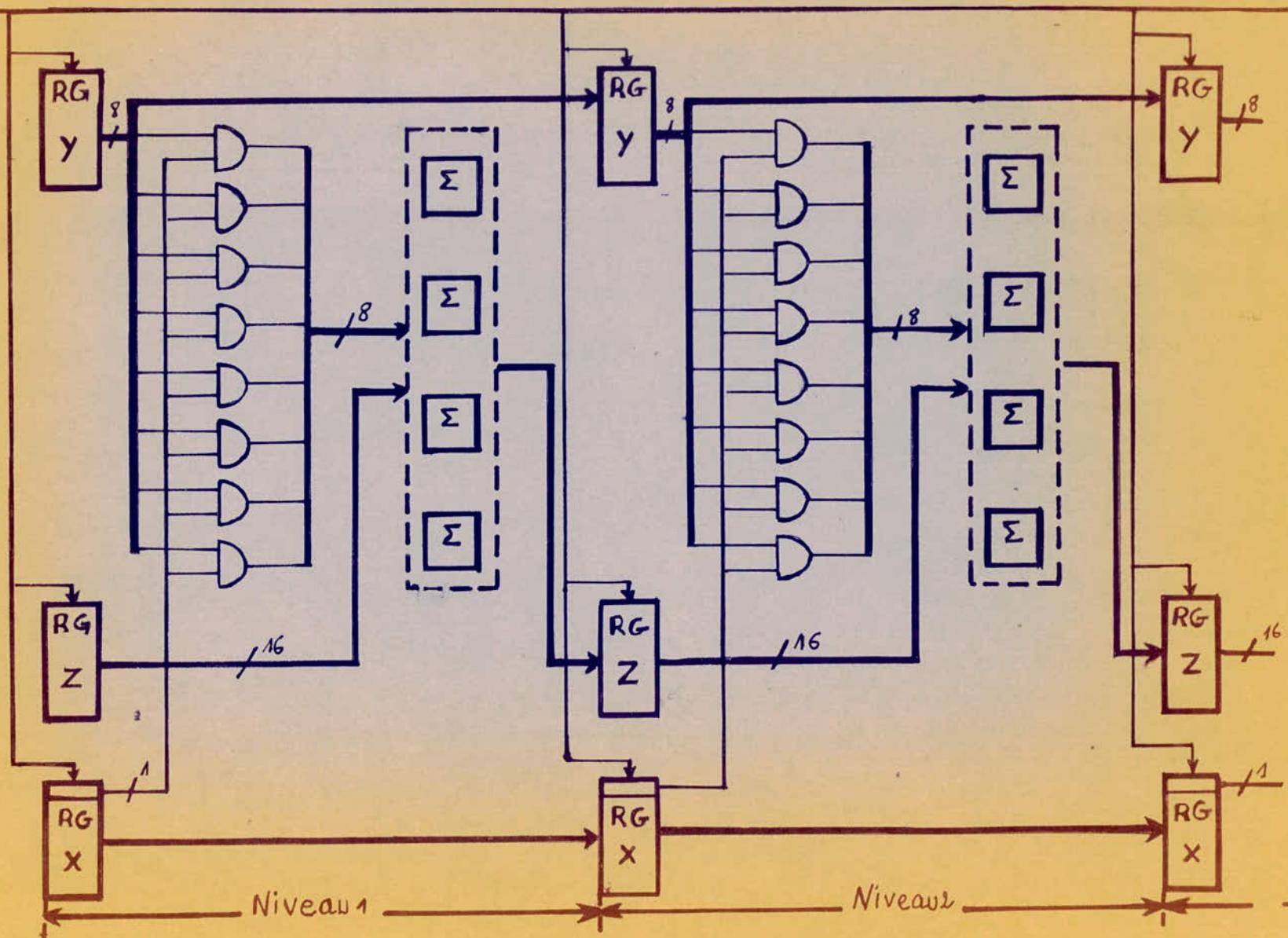


Fig. 3.5.3.C SCHEMA DE LA STRUCTURE PIP

D'après cette structure classique on a le premier produit (résultat) après 8 niveaux ; le deuxième résultat est obtenu après la neuvième période d'horloge. On comprend de ceci que le temps d'exécution de la deuxième opération est couvert par celui de la première et ainsi de suite.

D'où, on conclut que si le temps d'exécution d'une sous-opération est  $T_{s.op}$  et, le premier résultat est obtenu au bout de  $t_1$ ,  $t_1 = n T_{s.op}$ , le deuxième au bout de  $t_2 = t_1 + T_{s.op}$ . Et, si on a  $N$  opérations à effectuer, le temps nécessaire est :

$$T = n T_{s.op} + (N-1) T_{s.op}$$

On peut conclure que les temps de calcul se recouvrent ceci est illustré par la figure(3.5.3.d)

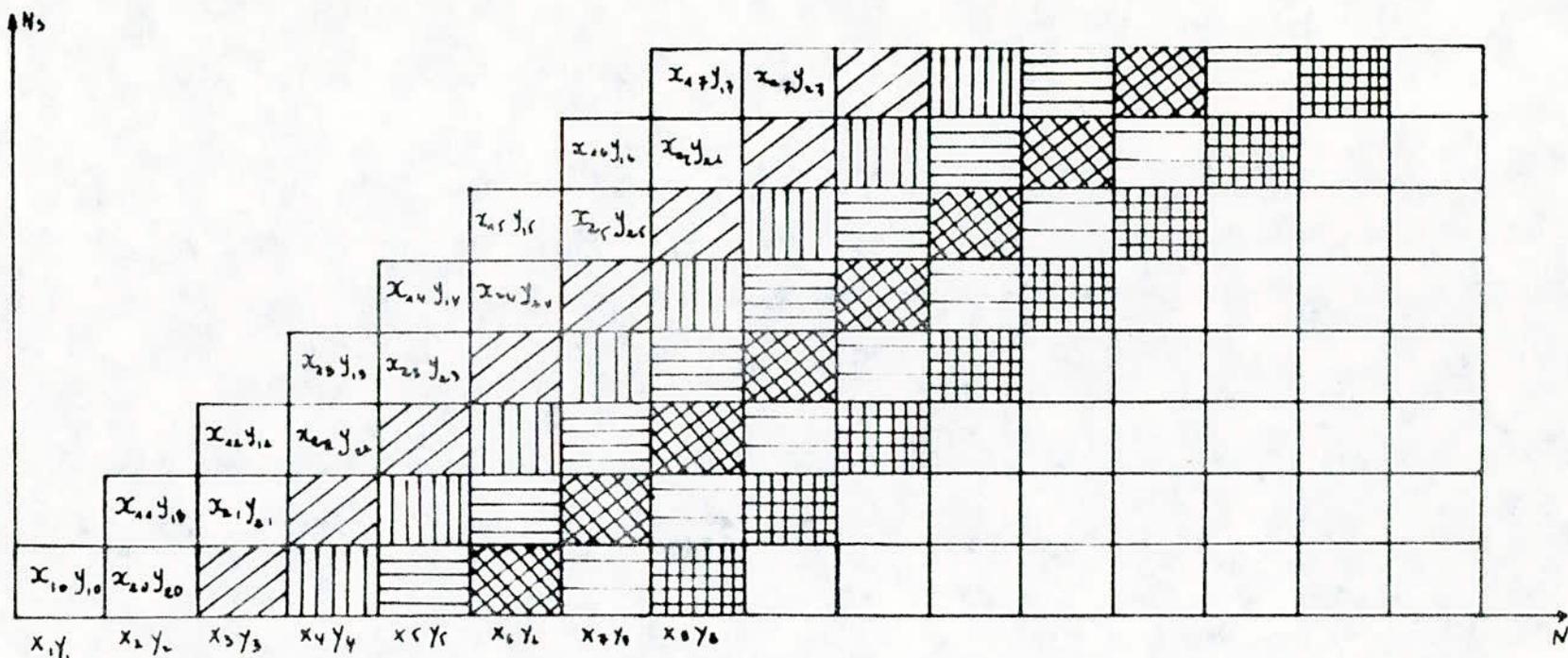
En ce qui concerne la somme partielle, on utilise des additionneurs complets où la retenue de propage en série d'un additionneur à l'autre.

Nous avons dit que le nombre de niveaux est égal au nombre de bits du multiplicateur, autrement dit, on calcul à chaque niveau un produit partiel.

Examinons le matériel nécessaire pour un niveau :

- Un registre (8bits) pour le multiplicande,
- Un registre (8bits) pour le multiplicateur,
- Un registre (16bits) pour le produit
- Deux quadruples porte AND pour effectuer les produits élémentaires.

.../...



51

$\overbrace{\hspace{1cm}}^{t_{sop}}$  temps d'exécution d'une sous-opération.  
 $\overbrace{\hspace{3cm}}^{t_1 = 8 t_{sop}}$  temps d'exécution d'une opération (la première).  
 $\overbrace{\hspace{4cm}}^{t_2 = t_1 + t_{sop}}$  temps d'exécution de la deuxième opération.

$$T_N = t_1 + (N-1)t_{sop} = Ns t_{sop} + (N-1)t_{sop}$$

$N_s$ : nombre de sous-opérations.  
 $N$ : nombre d'opérations à effectuer (dans notre cas  $N=8$ ).

$$\begin{aligned}
 X_i &= x_{i0} x_{i1} x_{i2} x_{i3} x_{i4} x_{i5} x_{i6} x_{i7} \\
 Y_i &= y_{i0} y_{i1} y_{i2} y_{i3} y_{i4} y_{i5} y_{i6} y_{i7}
 \end{aligned}$$

Fig: 3.5.3.d

- Quatre additionneurs (4bits) pour la sommation partielle.

Nous pouvons remarquer aisément que cette structure nécessite beaucoup de matériel bien que simple pour réaliser l'opération de base considérée.

#### CONCLUSION

L'étude des différentes méthodes algorithmiques et structurelles d'accélération de l'opération de multiplication, nous a permis de constater les différents avantages et inconvénients de chacune d'elles. Une analyse comparative dans le chapitre suivant nous permettra de faire le choix d'une structure et d'un algorithme adéquat à notre application.

## CHAPITRE 4

## ANALYSE COMPARATIVE ET SOLUTION RETENUE

Il importe avant d'aborder l'analyse comparative des structures et des algorithmes de l'opération de base, de situer " le système d'amélioration d'images " au sein du système de traitement du signal.

## 4.1 - CONFIGURATION DU SYSTEME DE TRAITEMENT DU SIGNAL

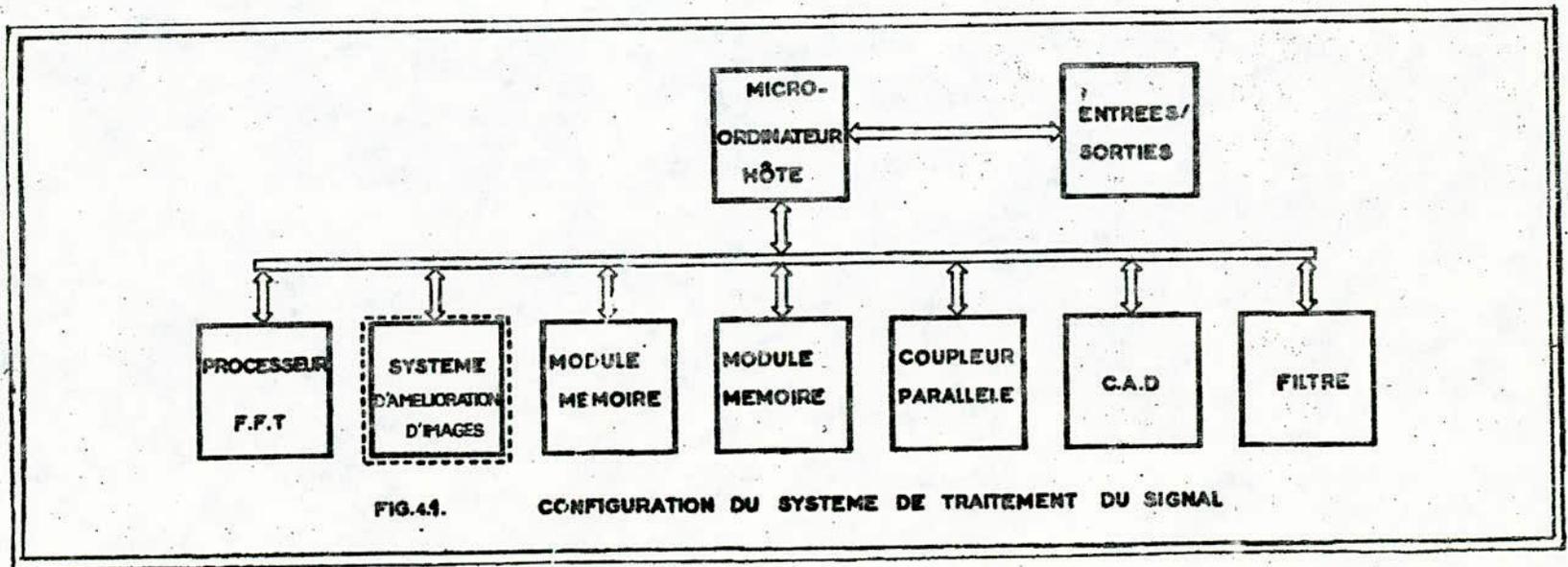
Le système d'amélioration d'images qu'on se propose de réaliser, constitue un des blocs importants du système de traitement du signal.

Notre système procède au traitement des données digitalisées en provenance du module mémoire. Une fois traitées, ces dernières sont transférées dans le module mémoire en vue d'autres traitements éventuels. De plus, il est relié au micro-ordinateur hôte qui le gère ainsi que tous les autres constituants du système de traitement du signal;

Filtre, CAD (convertisseur analogique digital), coupleur parallèle et processeur de la transformée de fourier rapide  
(PROCESSEUR FFT)

La configuration du système de traitement du signal est illustrée par la figure 4.1.

.../...



## 4.2 - ANALYSE COMPARATIVE DES DIFFERENTES STRUCTURES DE REALISATION DE L'OPERATION DE BASE.

L'étude faite dans le chapitre précédent nous a permis de voir les différentes structures et algorithmes effectuant l'opération de base. Chaque structure présente les avantages et les inconvénients que nous citerons ci-dessous.

### 4.2.1 - LA STRUCTURE MATRICIELLE

Cette structure présente l'avantage de la rapidité d'exécution de l'opération de multiplication dans le cas des nombres de longueur réduite.

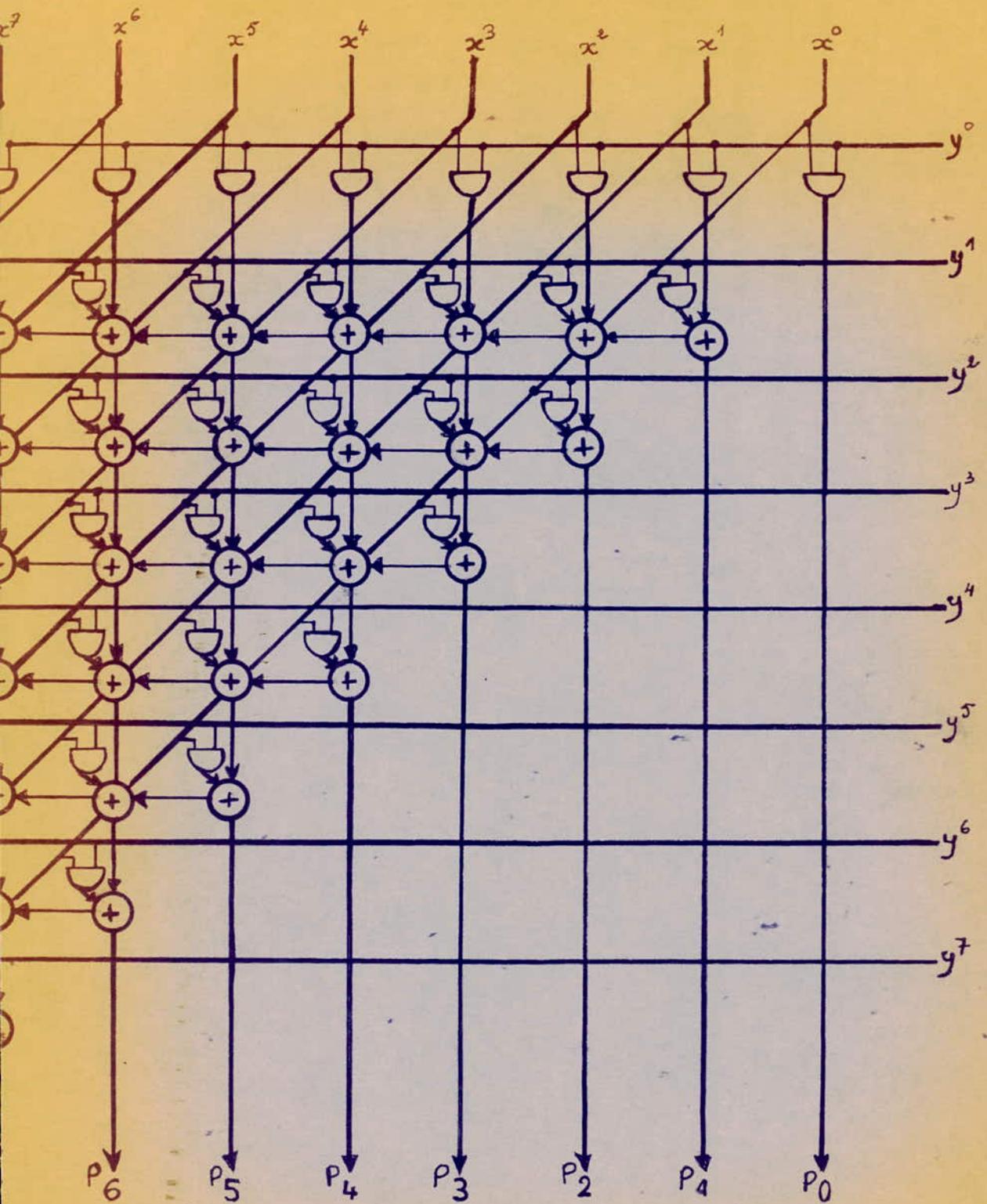
Cependant, sa complexité (nombre important de matériel) augmente avec le nombre de bits des mots considérés. Ceci est illustré par la figure 4.2.1 pour la multiplication de deux mots de (8 bits)

D'après la figure, nous constatons que cette structure nécessite :

- 56 additionneurs pour les produits partiels,
- et 64 portes AND pour effectuer les produits élémentaires.

Ce nombre considérable de circuits nécessaires à la réalisation uniquement de la multiplication rendrait l'architecture de l'unité de traitement du système très encombrants et entraînerait une diminution considérable de la fiabilité du système.

.../...



LE - MULTIPLICATION 8X8bits.

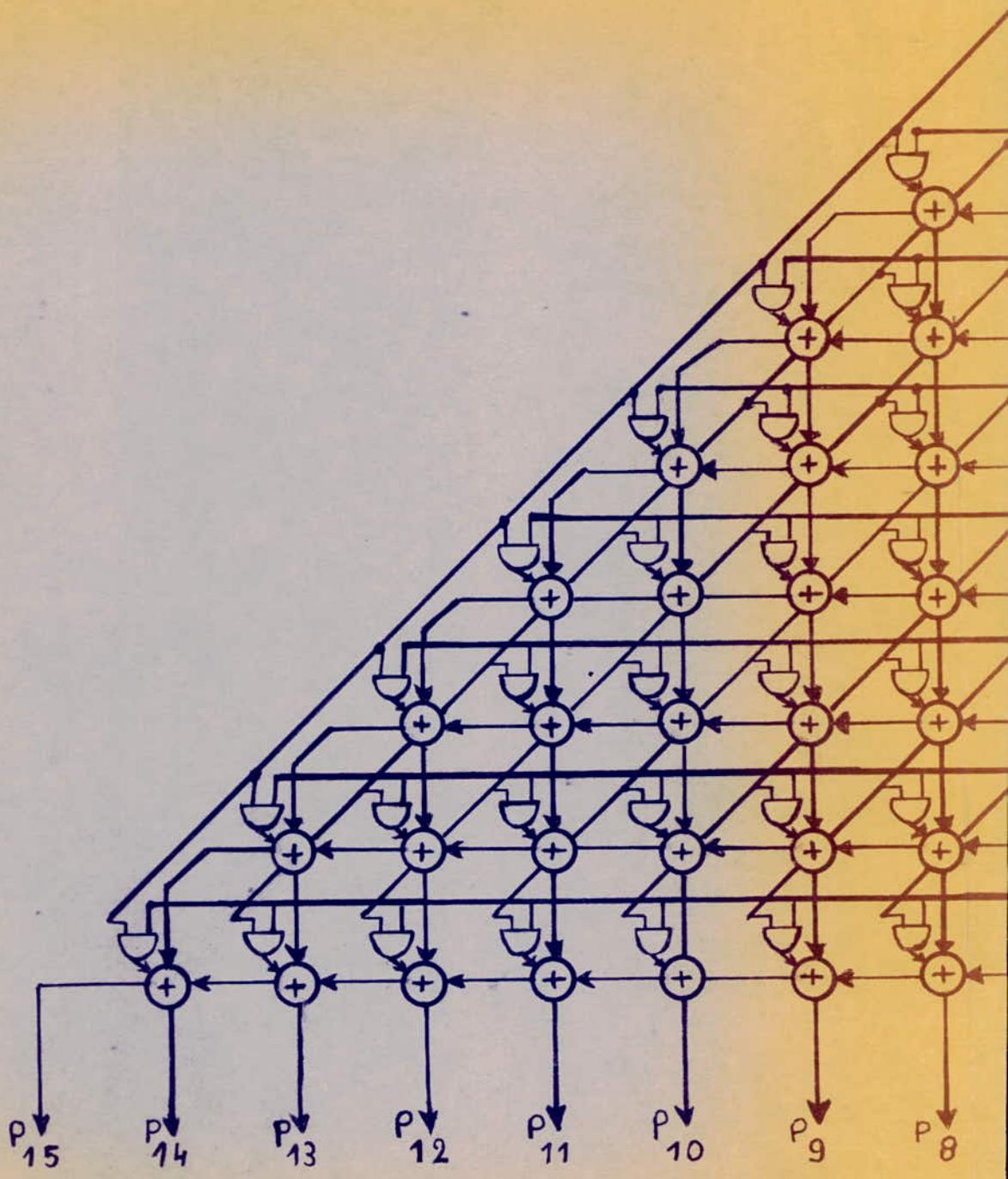


Fig. 4.2.1. STRUCTURE MATR

Donc l'utilisation de cette structure est limitée et n'est applicable que dans le cas où le format des pixels est très limité.

Pour ces raisons et pour d'autres liés à la rapidité (dans le cas des nombres de longueur moyenne) nous concluons que l'utilisation de cette structure dans la réalisation du système d'amélioration d'images, qui se veut rapide et économique, nous paraît peu avantageuse.

#### 4.2.2 - LA STRUCTURE TABULAIRE

Son principe consiste à stocker, tous les résultats dans une mémoire, l'adresse étant constituée par l'ensemble des deux opérandes.

Pour la multiplication de deux nombres respectivement  $X_1$  ( $n_1$  bits),  $X_2$  ( $n_2$  bits), il faut prévoir une mémoire de  $2^{n_1 + n_2}$  mots de  $(n_1 + n_2)$  bits ; de ce fait, nous atteignons vite une limite.

En effet, pour la multiplication de deux mots de (4 bits), il faut une mémoire de  $2^8$  mots de 8 bits, pour des mots (sur 8 bits), une mémoire de  $2^{16}$  mots de 16 bits.

Il est clair qu'à chaque fois que  $n_1$  ou  $n_2$  augmente, on double le nombre des mots mémoire.

Dans le but de minimiser le volume de la mémoire, il fallait faire appel à une organisation utilisant des mémoires (qui effectuent) l'opération 4 x 4 bits) et des additionneurs. Pour plus de précision se référer à la figure (3.4.1) du chapitre précédent.

.../...

Cette structure comme nous pouvons facilement le constater n'est pas complexe du point de vue matériel.

En effet, pour sa réalisation (pour 2 mots de 8 bits) il faut :

- 4 mémoires,
- 5 additionneurs,

Cependant, sa rapidité d'exécution de l'opération dépend du temps d'accès de la mémoire utilisée.

On peut en conclure que l'utilisation de cette structure dans le cas qui nous intéresse, nous paraît efficace et avantageuse du fait qu'on n'a pas de problème de fiabilité, d'encombrement matériel; et de plus sa rapidité peut être augmentée par l'emploi des circuits plus performants (mémoires rapides, circuits multiplicateurs très rapides).

#### 4.2.3 - LA STRUCTURE PIPELINE

C'est la structure idéale pour augmenter la rapidité d'exécution des opérations. Elle permet d'effectuer la multiplication selon un processus continu. Le temps d'exécution de la première opération couvre celui de la deuxième et ainsi de suite. Par conséquent, on gagne du point de vue temps puisqu'on peut lire un résultat alors qu'il y a plusieurs produits en cours, d'où l'intérêt majeur de cette structure.

Toutefois, cette structure demande énormément de matériel (mémoires de découplage, portes AND, additionneurs).

.../...

En effet, pour effectuer un produit partiel (1x8 bits dans notre cas) il faut prévoir :

- 4 mémoires de découplage,
- 2 quadruples de porte AND,
- 4 additionneurs (4 bits),

et ce matériel est multiplié par le nombre de bits du multiplieur pour effectuer une opération (8 x 8 bits)

Cette structure réalisée avec des portes AND, des circuits commercialisés reviendrait très chère et offrirait ainsi peut d'intérêt.

Mais elle se révèle très efficace (très importante) pour une perspective de mise en oeuvre de circuits prédiffusés.

Cette analyse nous permet d'opter pour une structure dérivée des structures tabulaire et pipeline, et, ceci dans le but de remédier au problème d'encombrement matériel et d'augmenter la rapidité d'exécution de l'opération considérée.

Cette structure réalisée avec des mémoires rapides permet d'augmenter sensiblement la fréquence de travail sans avoir à augmenter le matériel. Ceci explique l'intérêt suscité par cette structure .

#### 4.3 - PRESENTATION DE LA SOLUTION RETENUE : SYSTEME PIPELINE - TABULAIRE D'AMELIORATION D'IMAGES.

L'analyse précédente nous a permis d'opter pour une structure

.../...

qui sera à la base de la conception et la réalisation du système d'amélioration d'images et, plus précisément de son unité de traitement.

Le système d'amélioration d'images illustré par le synoptique (fig. 4.3), comporte les parties suivantes dont nous décrivons brièvement le rôle et le fonctionnement.

6 Séquenceurs dont le rôle consiste à fournir les adresses aux mémoires destinées à recevoir les pixels et à la mémoire ~~convi-~~  
~~ténant~~ en principe les coefficients des différents masques.

- Une mémoire PROM contenant les différents coefficients de masques destinés à l'amélioration d'images. Ces coefficients sont acheminés vers l'unité de traitement.

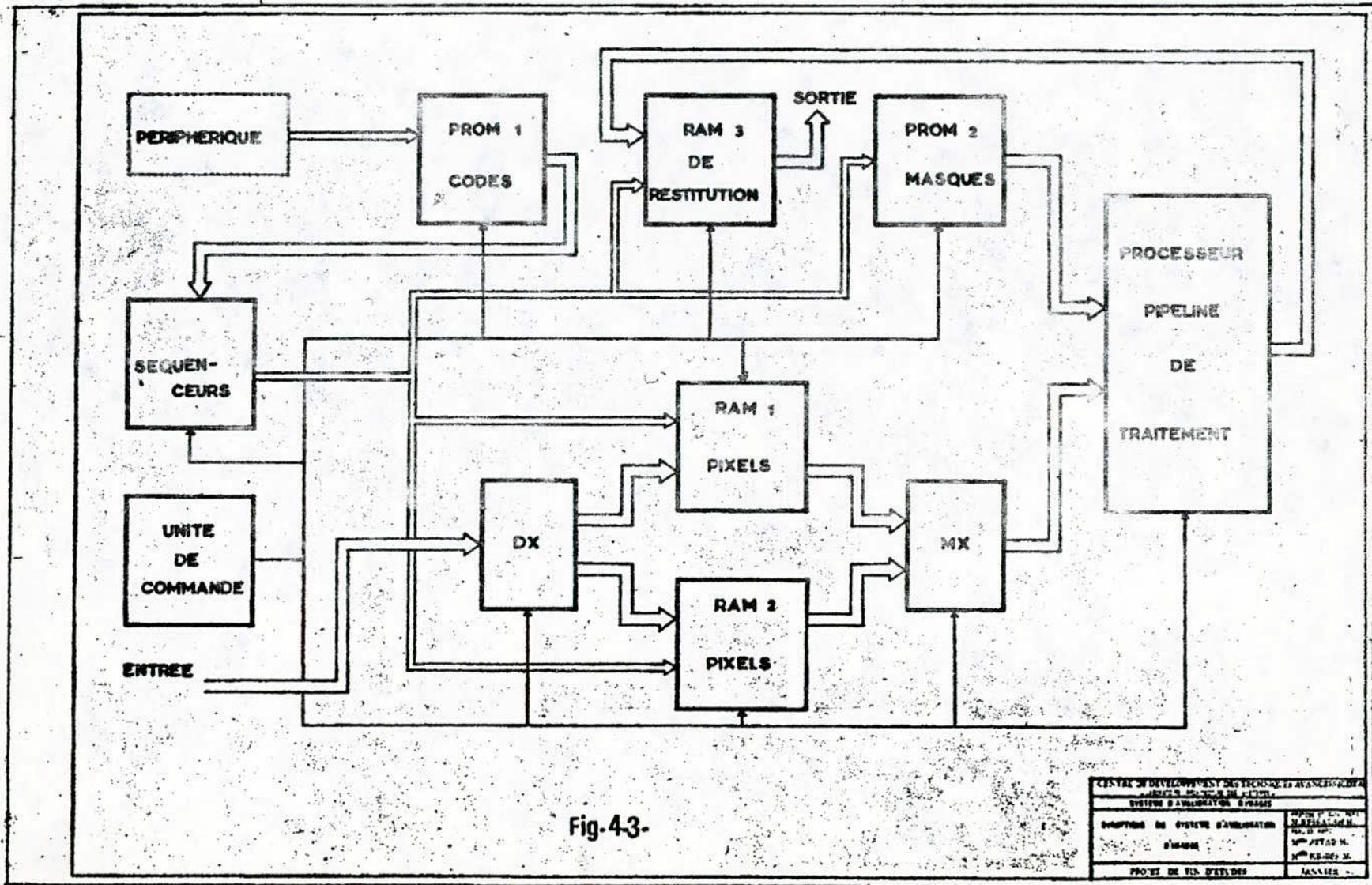
- Deux mémoires RAM destinées à recevoir les pixels en provenance du plan mémoire. Elles travaillent par intermittence entre elles avec l'unité de traitement, ce qui augmente sensiblement la vitesse de traitement.

- L'unité de traitement conçue selon la structure adoptée est constituée essentiellement de mémoires PROM contenant les produits partiels, d'additionneurs, d'accumulateur et de mémoires de découplage insérées entre étages.

L'unité de traitement ainsi présentée effectue l'algorithme cité auparavant :

$$\sum_i \sum_j X_i Y_j$$

.../...



69

Fig-43-

CENTRE DE DEVELOPPEMENT DES TECHNIQUES AVANCEES	
SYSTEME D'ALIMENTATION VIDEO	
DESCRIPTION DU SYSTEME D'ALIMENTATION	REF. P. 1. 101
D'UNITE	N. 101. 101. 101
	N. 101. 101. 101
PROJET DE FIN D'ETUDES	JANVIER -

Ces résultats issus de l'unité de traitement sont transférés au plan mémoire.

- L'unité de commande qui assure la synchronisation de toutes les unités citées ci-dessus.

#### CONCLUSION

Grâce à la structure adoptée, nous pouvons mettre au point un système très performant du fait que :

- L'unité de traitement fonctionne en pipeline,
- Les mémoires, dont les données sont acheminées vers l'unité de traitement, travaillent par intermittence ce qui donne un aspect continu à l'acheminement des données (pixels) vers l'unité de traitement.

CHAPITRE 5SYSTEME D'AMELIORATION D'IMAGES

## 5.1 - INTRODUCTION

L'étude faite dans les chapitres précédents nous a permis de tracer les grandes lignes permettant la conception et la réalisation d'un système d'amélioration d'images, performant du point de vue vitesse.

Au début de ce chapitre, nous donnerons l'architecture du système d'amélioration d'images. Ensuite, nous examinerons en détail le rôle, l'architecture et le fonctionnement de chaque partie du système.

## 5.2 - ARCHITECTURE DU SYSTEME D'AMELIORATION D'IMAGES

L'architecture du système d'amélioration d'images est illustré par la figure 5-1.

Le système d'amélioration d'images comporte essentiellement;

- le séquenceur,
- l'unité de commande,
- l'unité de traitement.

L'unique fonction du séquenceur est la formation des adresses des pixels et des coefficients qui sont stockés respectivement dans les RAM et la PROM.

L'unité de commande fournit les signaux de lecture-écriture des mémoires pixels, validation du latched résultat et de remise à zéro de l'accumulateur. D'autre part, elle délivre les signaux de synchronisation h1 et h3 et le signal horloge du séquenceur.

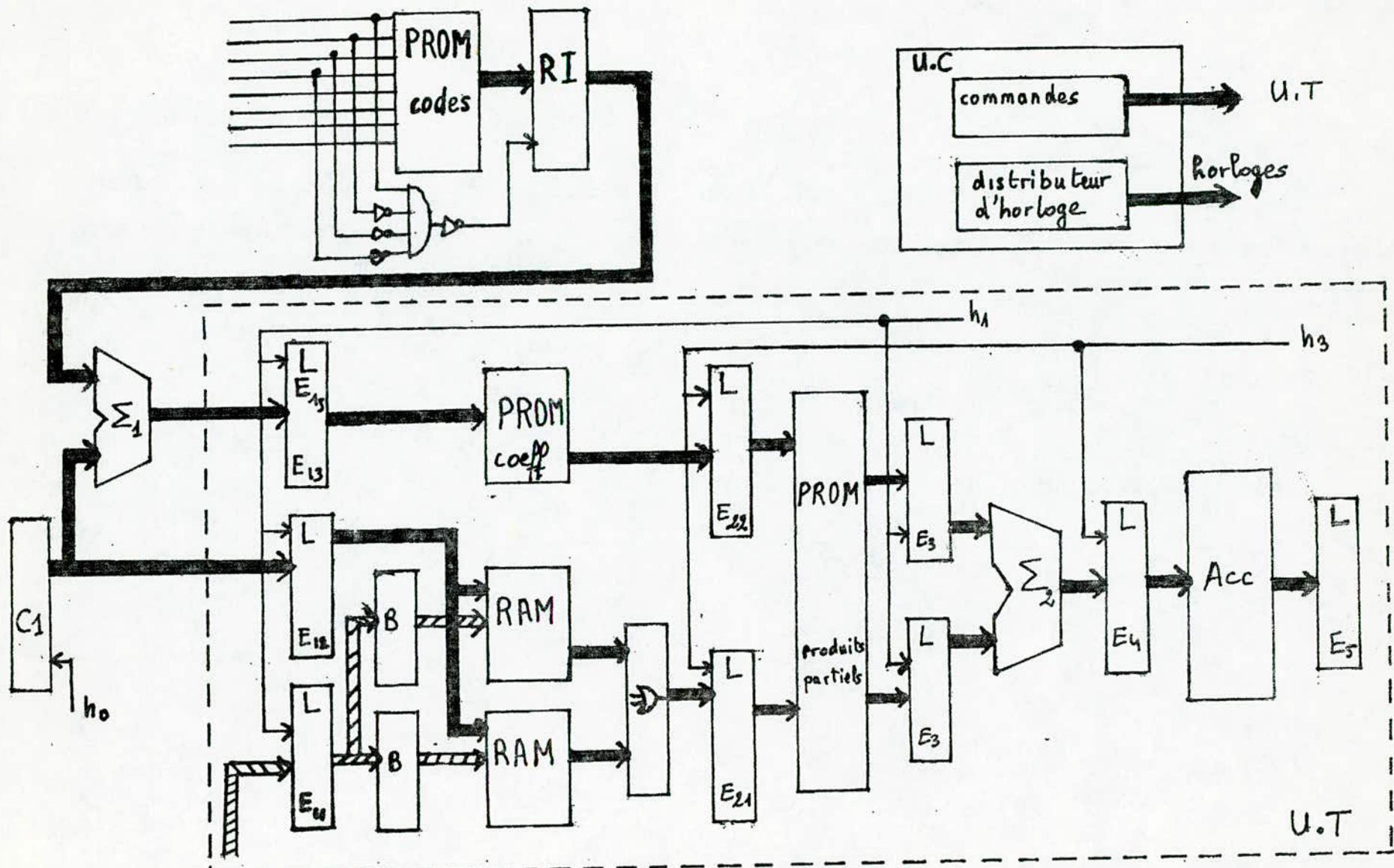


Fig.: 5-1. Architecture du système d'amélioration d'images

L'unité de traitement effectue l'opération d'amélioration selon l'algorithme donné précédemment. Cette opération est effectuée sur des mots de 8 bits (coefficients et pixels), en virgule fixe.

Ce système utilise deux chaînes de synchronisation dans le but d'éviter le recouvrement de données de deux étages successifs : par exemple, si E21, E22 et E3 avaient le même signal de synchronisation, E3 risque de mémoriser des données aléatoires se trouvant sur les sorties des mémoires PROM qui le précède.

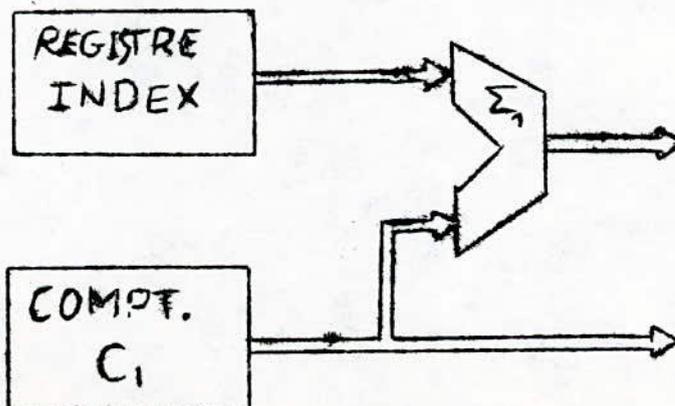
Le fait d'utiliser deux chaînes de synchronisation diminue la vitesse du système mais augmente sa fiabilité vu qu'il n'y a pas de risque de recouvrement de données.

### 5.3 - LE SEQUENCEUR

La structure et le fonctionnement du séquenceur dépendent du mode d'adressage des mémoires adoptés.

Les coefficients et les pixels sont stockés dans des mémoires dont chaque cellule est caractérisée par son adresse. Nous disposons de plusieurs modes d'adressage pour atteindre une cellule.

Dans ce système, les adresses sont fournies selon deux modes d'adressage comme l'indique le schéma suivant :



Dans le cas des pixels, l'adressage est direct car les adresses sont données par le compteur C1.

En ce qui concerne les coefficients, les adresses sont données suivant un mode d'adressage indexé.

Nous avons choisi le mode d'adressage indexé parce qu'il est particulièrement adapté, au traitement de tableau de données par incrémentation.

L'objectif de ce mode d'adressage est une bonne utilisation de l'espace mémoire de la PROM.

Le registre index (R.I) fournit l'adresse du premier coefficient du masque utilisé pour l'opération de traitement. Le contenu de R.I. additionné au contenu de C1 donne les adresses des autres coefficients du masque.

Pour la formation des adresses des premiers coefficients des masques est utilisée une PROM de codes.

Le contenu de cette PROM est déterminé à partir de la répartition des éléments des masques dans la PROM coefficients. Pour les masques 16 x 16 les coefficients sont répartis suivant les adresses de 00 à FF (en hexadécimal), alors que pour les autres masques d'amélioration d'images (masques 3x3) leurs coefficients sont répartis dans la mémoire coefficient comme suit :

- LAPLACIEN 1 de 101 à 109
- LAPLACIEN 2 de 10A à 112
- LAPLACIEN 3 de 113 à 11B

.../...

- GRADIENT 1 de 11 C à 124
- GRADIENT 2 de 125 à 12 D
- GRADIENT 3 de 12 E à 136
- GRADIENT 4 de 137 à 13 F
- GRADIENT 5 de 14 Ø à 148
- GRADIENT 6 de 149 à 151

Le contenu de la PROM de codes est le suivant :

- Ø Ø
- Ø 1
- Ø A
- 13
- 1 C
- 25
- 2 E)
- 37
- 4 Ø
- 49

Dans le but d'un test interactif de notre système, nous utilisons le code ASCII. C'est ainsi que les fonctions effectuées par les différents opérateurs seront désignées par les mnémoniques suivantes :

- masques 16 x 16 → A
- masques 3 x 3 :

LAPACIEN 1 → B

LAPLACIEN 2 → C

LAPLACIEN 3 → D

...../.....

GRADIENT 1 ----- E  
 GRADIENT 2 ----- F  
 GRADIENT 3 ----- G  
 GRADIENT 4 ----- H  
 GRADIENT 5 ----- I  
 GRADIENT 6 ----- J

Ces différentes mnémoniques constitueront les adresses du premier coefficient de chaque masque.

D'après la table du code ASCII, illustrée par la figure 5-2, nous pouvons faire les remarques suivantes :

- Toutes les mnémoniques ont le MSB à "0" et les 3 bits de plus fort poids forment le nombre 4 (100 en binaire)

- Elles diffèrent entre elles par les 4 bits de plus faible poids.

Ce fait, nous permet d'effectuer un adressage indexé de la mémoire PROM où sont stockés les coefficients des différents masques.

La mémoire PROM où sont stockées les adresses des premiers coefficients est adressée par l'intermédiaire d'un clavier. Donc, au niveau du bus d'adresse de la mémoire sont prélevés les 4 bits de plus faible poids avec lesquels est généré le signal U 1. Ceci est illustré par la figure 5.3 où U 1 est à "0" dans le cas du masque 16 x 16 et à "1" dans le cas des autres masques 3 x 3.

Une fois adressée, cette mémoire transmet ses données dans le registre index, le signal U 1 est appliqué au bit de plus fort poids de ce registre.

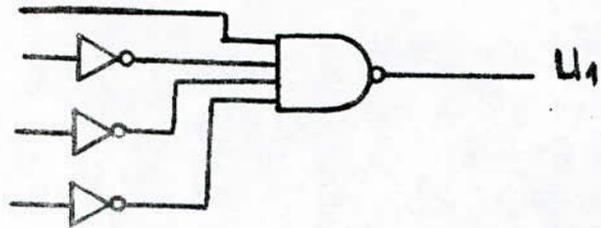
.../...

Sits ou chiffres les plus significatifs

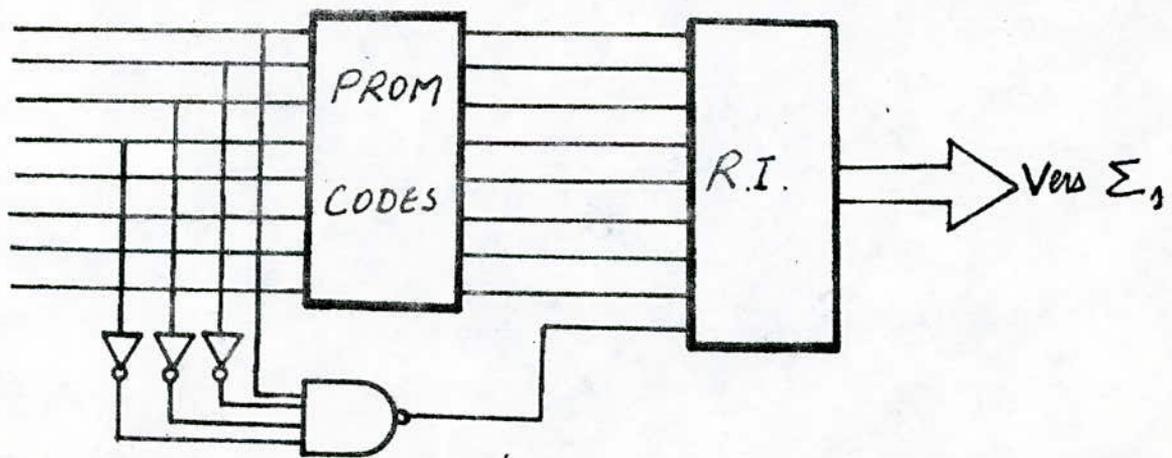
Binaire	0000	0001	0010	0011	0100	0101	0110	0111
	0	1	2	3	4	5	6	7
0000	0 NUL NOTHING	DATA LINK ESCAPE CONTROL P	SPACE	0	0	P	'	p
0001	1 START OF HEADING CONTROL A	DEVICE CONT. 1 CONTROL Q*		1	A	Q	q	q
0010	2 START OF TEXT CONTROL B	DEVICE CONT. 2 CONTROL R	-	2	B	R	b	r
0011	3 END OF TEXT CONTROL C	DEVICE CONT. 3 CONTROL S†	#	3	C	S	c	s
0100	4 END OF TRANSMISSION CONTROL D	DEVICE CONT. 4 CONTROL T	\$	4	D	T	d	t
0101	5 ENQUIRY CONTROL E	NOT ACKNOWLEDGED CONTROL U	%	5	E	U	e	u
0110	6 ACKNOWLEDGE CONTROL F	SYNCHRONIZE CONTROL V	&	6	F	V	f	v
0111	7 BELL CONTROL G	END OF TRANS. BLOCK CONTROL W		7	G	W	g	w
1000	8 BACK SPACE CONTROL H	CANCEL CONTROL X		8	H	X	h	x
1001	9 HORIZ TAB CONTROL I	END OF MEDIUM CONTROL Y	)	9	I	Y	i	y
1010	A LINE FEED CONTROL J	SUBSTITUTE CONTROL Z	.	:	J	Z	j	z
1011	B VERT TAB CONTROL K	ESCAPE	+	:	K		k	{
1100	C FORM FEED CONTROL L	FILE SEPARATOR	.	<	L	\	l	
1101	D CARRAGE RET CONTROL M	GROUP SEPARATOR	-	=	M		m	}
1110	E SHIFT OUT CONTROL N	RECORD SEPARATOR	.	>	N		n	~
1111	F SHIFT IN CONTROL O	UNIT SEPARATOR	/	?	O	-	o	DELETE

\* oussi XON  
† oussi XOFF

Fig. 5.2. Table de codage ASCII.



a/ Circuit de génération du signal U1.



b/ Circuit de génération de l'adresse du premier coefficient

Fig. 5.3

Le compteur C 1 est composé de trois circuits du type SN 74 LS 163 qui sont des compteurs binaires à 4 étages à retenu, affichage et remise à zéro synchrone. Le schéma fonctionnel de C 1 est donné par la figure 5.4. Les circuits SN 74 LS 163 ont été montés en série afin de pouvoir compter jusqu'à 255 et ceci dans le cas des masques 16 x 16. Les entrées d'affichage A, B, C, D des circuits V 8<sub>1</sub>, V 8<sub>2</sub> et V 8<sub>3</sub> qui composent C 1 sont reliées à la masse. Les entrées load (chargement) sont mises au potentiel Vcc = + 5V pour ne pas avoir de chargement parallèle. Les sorties de V 8<sub>1</sub> et V 8<sub>2</sub> sont reliées au bus d'adresses des mémoires pixels et à l'additionneur comme l'indique la figure 5.1

A la mise sous tension, comme tout circuit séquentiel, le compteur C 1 prend un état quelconque. Dans le système d'amélioration d'images C 1 est prévu pour démarrer de l'état "0".

Soit U, le signal d'initialisation. Il peut être généré par un trigger de schmitt dont on prendra la sortie inversée, voir figure 5.5.

C'est la constante de temps RC qui fixe la durée de l'impulsion d du signal délivré par le trigger de schmitt.

$$R = \quad ; \quad C =$$

Le système utilise des masques 16 x 16 et des masques 3 x 3 donc un plus de l'initialisation, une remise à zéro sera prévue dans le cas des masques 3 x 3 chaque fois que le compteur est à l'état 8 (C 1 = 0000 1000)

Soit U<sub>2</sub> le signal formé à partir des états de C 1 :

U<sub>2</sub> = 1 correspond à l'état 8

U<sub>2</sub> = 0 correspond aux autres états .

U<sub>2</sub> est issu de la sortie Q<sub>D</sub> du circuit V 8<sub>1</sub> de C 1.

Le processus de remise à zéro est représenté par le diagramme des temps illustré par la figure 5.6.

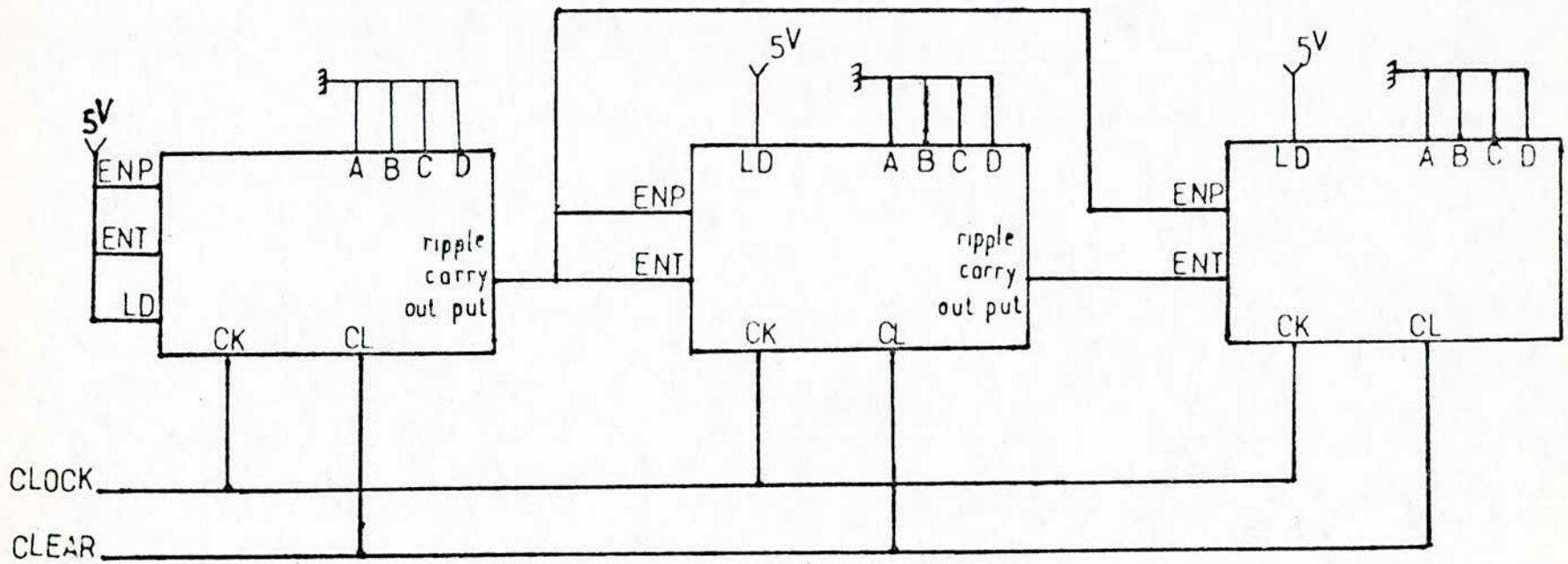
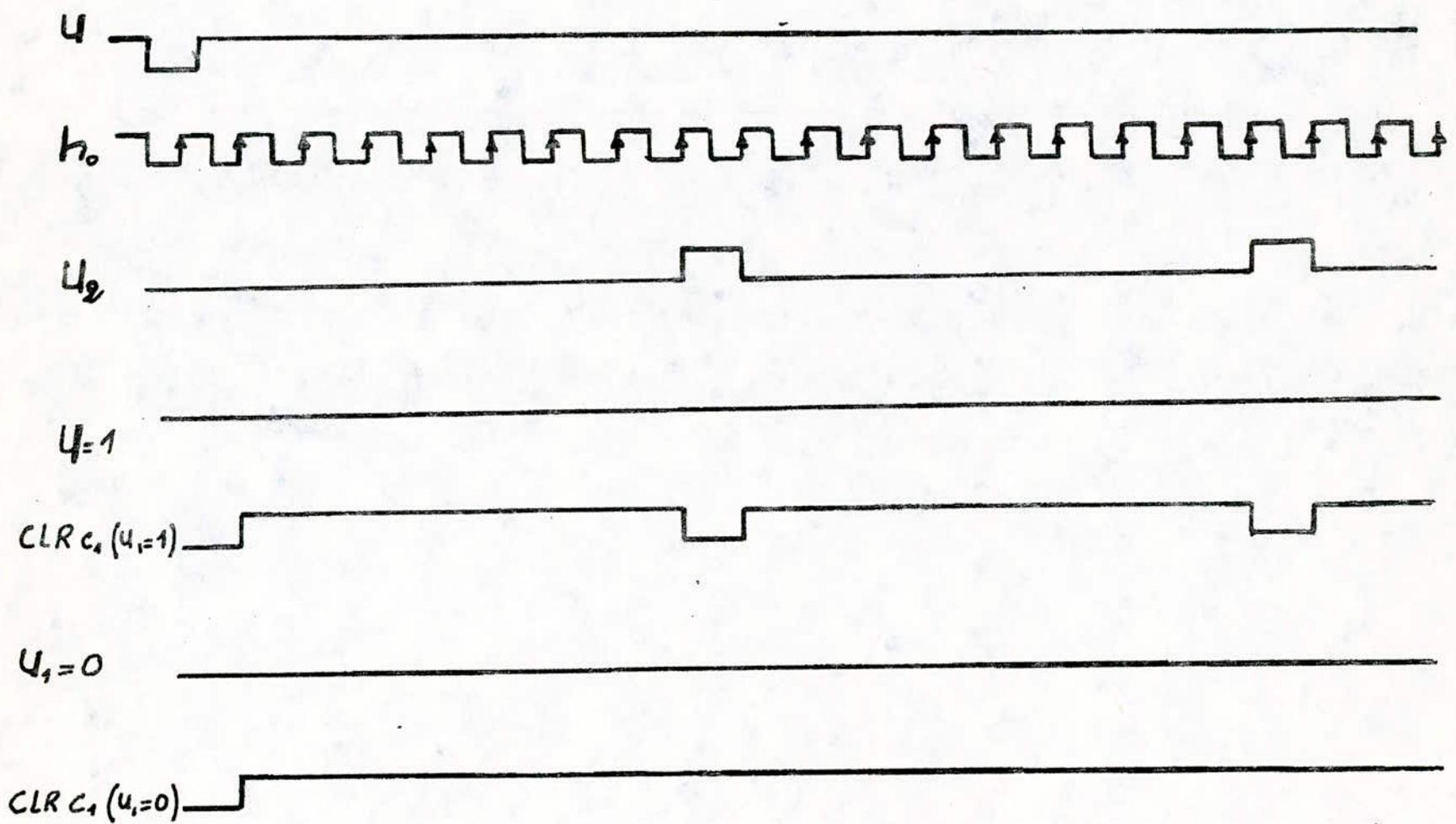


Fig. 5.4 . schéma fonctionnel de C1



79

Fig. 5.6. Diagramme des temps du processus de remise à zéro du compteur  $C_1$ .

Du diagramme donné par la figure 5.6, nous tirons la table de vérité représenté par la figure 5.7.a.

La fonction  $\overline{\text{CLR}}$  de C 1 étant complètement spécifiée, elle sera simplifiée en utilisant la table de Karnaugh, représentée par la figure 5-7.a.

CLR C 1 est donné par l'expression suivante :

$$\text{CLR C 1} = \overline{U_2 \bar{U}} + U_1 U_2 U$$

Un tel signal peut être obtenu par le circuit de la fig. 5.

#### 5.4 - UNITE DE COMMANDE

Cette unité est composée de :

- Un distributeur d'horloge,
- Un générateur de commandes.

##### 5.4.1 - LE DISTRIBUTEUR D'HORLOGE.

Le schéma fonctionnel est donné par la figure 5.8. Les sorties 0,1,2,3 du circuit SN 7442 fournissent des temps disjoints (voir figure 5.9) parfaitement réguliers.

Les signaux issus de ce distributeur sont utilisés comme suit :

h<sub>0</sub> est le signal horloge de C 1, h<sub>1</sub> et h<sub>3</sub> servent à synchroniser les latches comme l'indique la figure 5.1.

h<sub>2</sub> est le signal horloge du compteur C 2 qui sert à la remise à zéro de l'accumulateur circuits du type SN 74 52 81 associés à des circuits du type SN 74 182 , et à valider le lache résultat E 5.

$u_1$	$u_2$	$u$	$\overline{CLR}$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

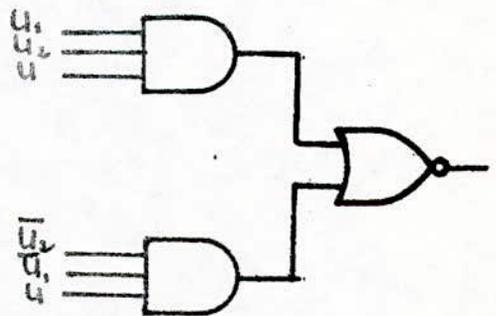
← remise à zéro.

← pas de remise à zéro.

a/ Table de vérité de la fonction  $\overline{CLR}$  de  $e_1$ .

$u_1, u_2$ $u$	00	01	10	11
0	1	0	1	0
1	0	0	0	1

b/ Table de KARNAUGH.



c/ Circuit de génération du signal  $\overline{CLR}$  de  $e_1$ .

Fig. 5.7.

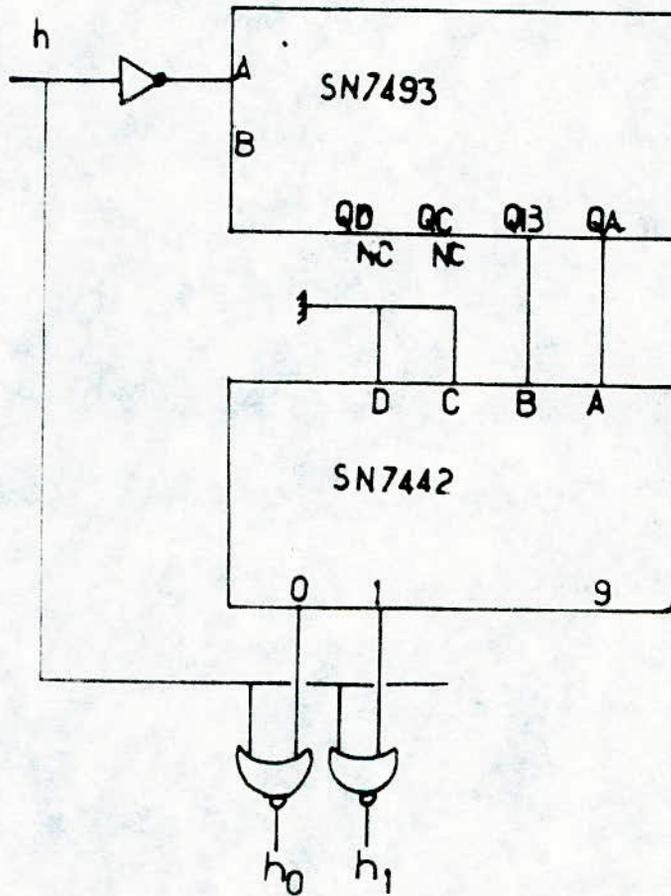


Fig 5.8 schema fonctionnel du distributeur d'horloge

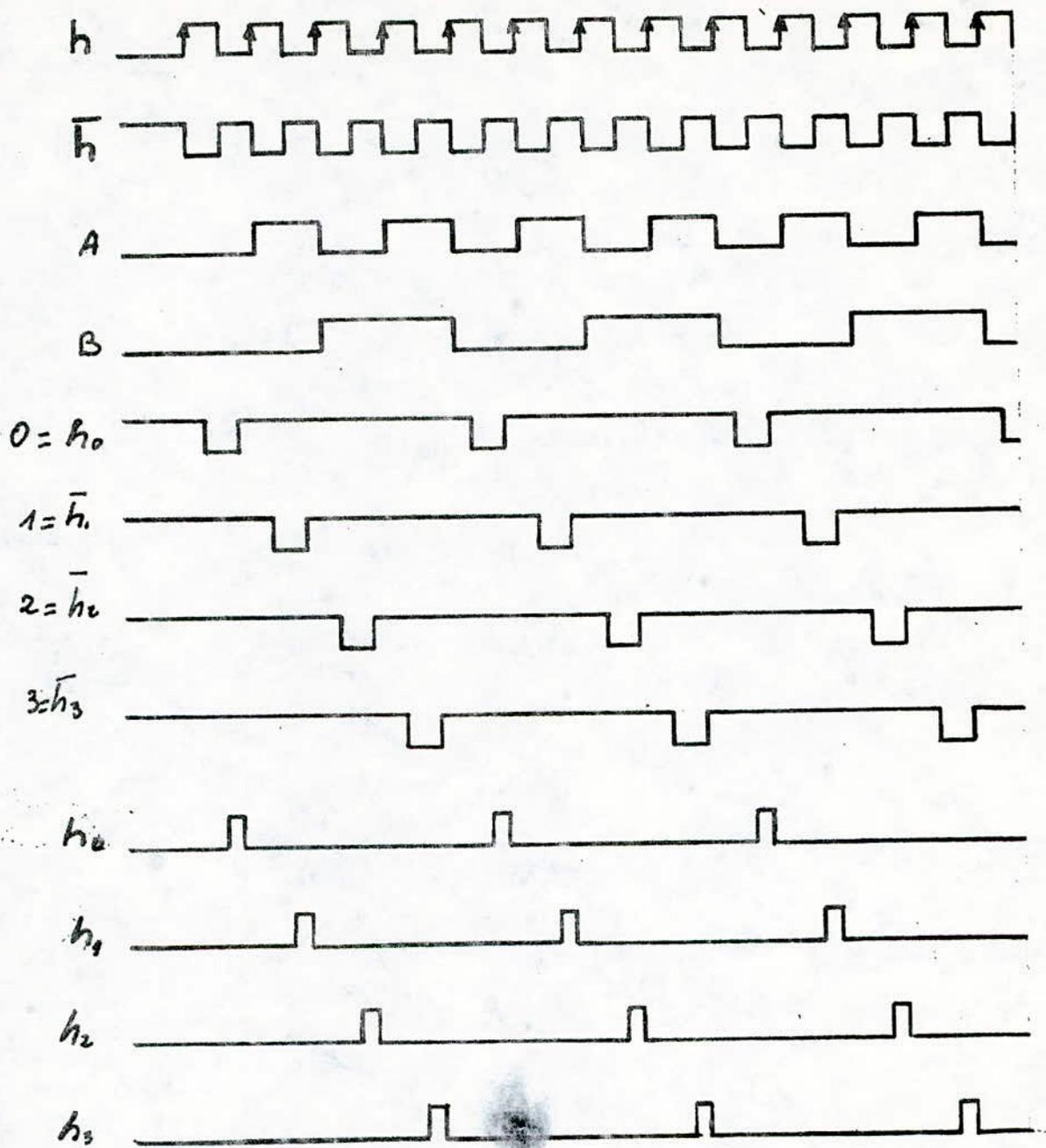


Fig. 5.9. Diagramme des temps du montage de la figure 5.8. cycle d'horloge à temps disjoint.

La valeur de la fréquence de l'horloge h (voir schéma de la figure 5-8) est obtenue comme suit :

Soit t le temps mis par les pixels pour passer du latch E 1 au latch E 2.

D'après le diagramme représenté par la figure 5-9, nous remarquons que les impulsions des horloges h 1 et h 3, sont espacées d'une durée d'égale à 7 d (d étant la durée de l'impulsion horloge h 1 ou h 3)

$$d = \frac{1}{4} T$$

T étant la période du signal horloge h.

$$d' = \frac{7}{4} T$$

d'où

$$f = \frac{1}{T} = \frac{7}{4d'}$$

désignons par :

- t 1 le temps d'accès de la RAM
- t 2 le temps de propagation dans les portes OR
- t 3 le temps de propagation dans les buffers.

Le temps t est la somme de t 1, t 2 et t 3

$$t = t 1 + t 2 + t 3$$

$$t = 55 + 7 + 18$$

$$t = 80 \text{ ns}$$

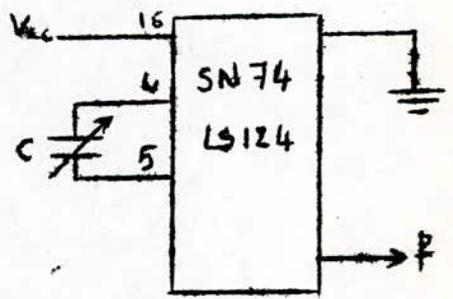
t doit être supérieur ou égal à d'

Pour t = d', nous avons :

$$f = 21,87 \text{ MHz}$$

Cette fréquence est donnée par le circuit ci-contre.

$$f = 1.10^{-4} / C_{ext} \quad d'où C_{ext} = 4,57 \text{ PF}$$



.../...

Soit  $f_t$  la fréquence de travail

$$f_t = f_{h0} = f_{h1} = f_{h2} = f_{h3}$$

$$f_{h0} = \frac{1}{4} f$$

d'où

$$f_t = 5,46 \text{ MHz}$$

### 5.4.2 - COMMANDE DE LECTURE - ECRITURE DES MEMOIRES PIXELS

C 1 est un compteur modulo 256 dans le cas des masques 16 x 16. La sortie de plus faible poids (LSB) de V 83 (voir figure 5.4) donne le signal  $R/\overline{W}$  des RAM.

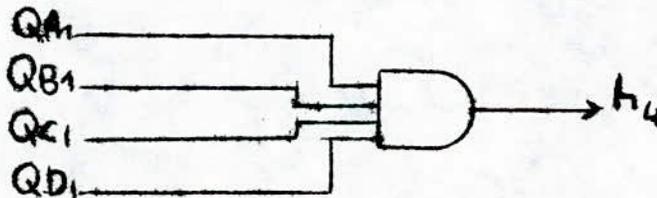
En ce qui concerne les masques 3 x 3, C 1 est un compteur modulo 9. Dans ce cas, l'unité de contrôle fournit un autre signal  $R/\overline{W}$ . Ce signal est issu de la sortie de plus faible poids d'un compteur (C 3) modulo 16, car elle est alternativement à "0" ou "1", durant 9 impulsions d'horloge  $h_0$ .

Le signal horloge de C 3 est obtenu à partir des états de C 1.

$$h_4 = 1 \text{ pour } C_1 = 0$$

$$h_4 = 0 \text{ pour } C_1 \neq 0$$

$h_4$  est généré comme suit :



$Q_{A1}$ ,  $Q_{B1}$ ,  $Q_{C1}$ ,  $Q_{D1}$  étant les sorties de plus faible poids de  $C1$ , Le signal CLR  $C3$  est le signal donné dans l'étude de  $C1$ .

Le diagramme des temps donnant le signal  $R / \bar{W}$  est représenté par la figure 5-10.

Il y a deux signaux  $R / \bar{W}$  des mémoires RAM (cas des masques  $3 \times 3$  et cas des masques  $16 \times 16$ ), d'où la nécessité d'un multiplexeur qui sélectionnera l'un ou l'autre des signaux.

La table de vérité du multiplexeur est la suivante :

Entrées				Sorties
A	B	Sélection	Strobe	
L	X	0	0	L
H	X	0	0	H
X	L	1	0	L
X	H	1	0	H

L = 0  
H = 1

D'après la table précédente, nous remarquons que nous pouvons appliquer le signal  $\bar{C1}$  sur l'entrée de sélection du multiplexeur.

Les signaux issus de  $C3$  et  $C1$  sont respectivement appliqués sur les entrées A et B du multiplexeur.

La sortie de ce multiplexeur donnera le signal  $R / \bar{W}$ . Ce signal est appliqué sur l'entrée commande de l'une des RAM alors que son inverse est appliqué sur l'entrée de commande  $R / \bar{W}$  de l'autre RAM afin d'avoir des mémoires qui travaillent par intermittence.

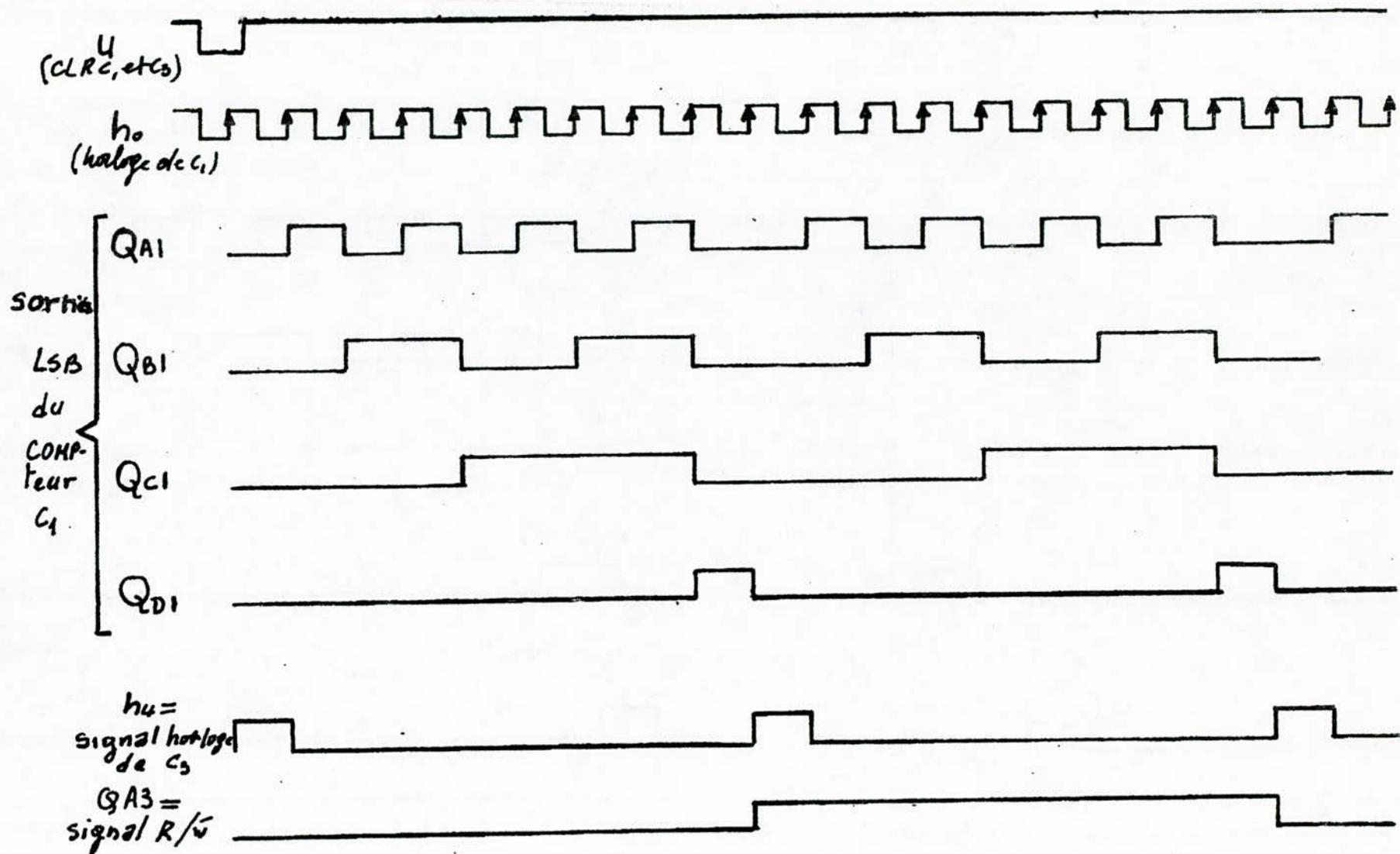


Fig. 5.10. Diagramme des temps donnant le signal R/ $\bar{W}$  (masques 3x3)

## 5.4.3 - REMISE A ZERO DE L'ACCUMULATEUR

D'après le chronogramme représenté par la figure 5-9 et la figure 5-1 représentant l'architecture du système d'amélioration d'images, nous remarquons que la valeur du premier produit partiel est mémorisé dans le latche E 4 pendant la durée de la deuxième impulsion de l'horloge h3. A ce moment, l'accumulateur doit être initialisé pour le préparer à recevoir le premier produit partiel. La première accumulation ("0" de l'accumulateur plus la valeur du premier produit partiel) se fait pendant la durée d', c'est-à-dire quand C1 est à l'état "1" (C1 = 0000 0001).

La remise à zéro de l'accumulateur doit avoir lieu aussi après chaque 256 accumulations dans le cas des masques 16 x 16 ou 9 accumulations dans le cas des masques 3 x 3. Un compteur (C2) sera utilisé pour compter le nombre d'accumulations et commander la remise à zéro de l'accumulateur.

Le compteur C 2 est composé de deux circuits du type SN 74 L S 163 dont le schéma fonctionnel est donné par la figure 5.11.a.

Le signal de remise à zéro de l'accumulateur est formé à partir des états de C 2. Puisque le compteur C 2 est prévu pour compter le nombre d'accumulations, la remise à zéro a lieu chaque fois que C 2 est à l'état "0". Le circuit de génération du signal CLR de C 2 est donné par la figure 5.11.b et sont les sorties du compteur C 2.

CLR accumulateur = U 6 dans le cas des masques 3 x 3

CLR accumulateur = U 5 dans le cas des masques 16 x 16

Il y a deux signaux CLR d'où la nécessité d'un multiplexeur (circuit du type SN 74 157).

.../....

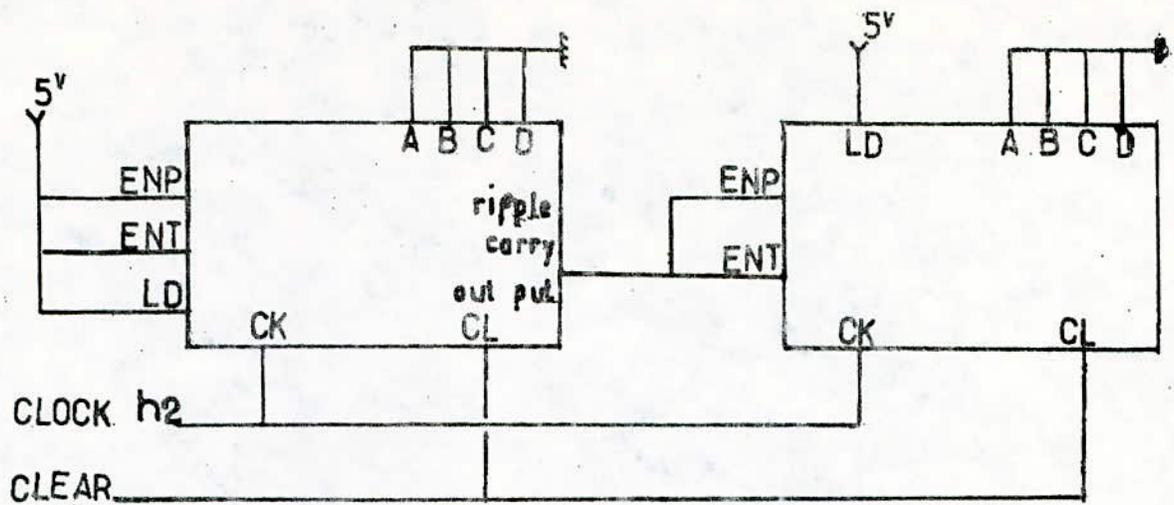


Fig. 5-11.a

schéma fonctionnel du compteur c2

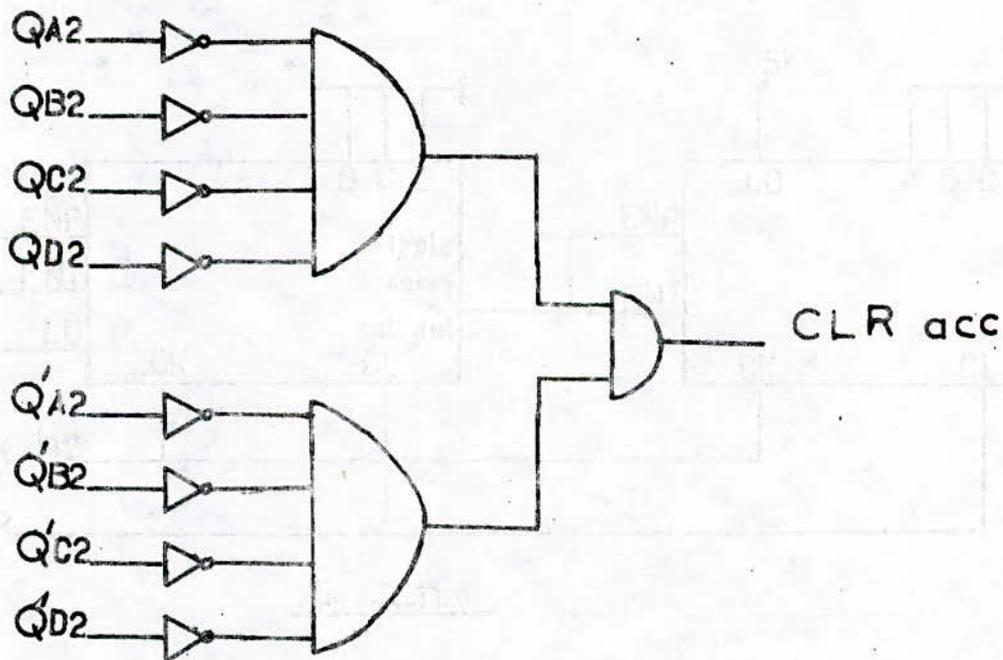


Fig 5:11.b

Circuit de génération du signal CLR  
de l'accumulateur

U 1 est appliqué sur la commande S du multiplexeur.

U 6 est appliqué sur l'entrée B du multiplexeur.

U 5 est appliqué sur l'entrée B du multiplexeur.

Le signal issu de la sortie du multiplexeur est le signal CLR de l'accumulateur. Il est appliqué sur l'entrée M (mode control) des circuits SN 74 S 281 qui composent l'accumulateur.

Remise à zéro du compteur C 2 : à la mise sous tension C 2 prend un état aléatoire. C 2 étant prévu pour compter le nombre d'accumulations, il doit démarrer de l'état "0". C 2 est remis à zéro chaque fois qu'il comptera 256 accumulations (256 impulsions horloge h2) dans le cas des masques 16 x 16 ou 9 accumulations (9 impulsions horloge h2) dans le cas des masques 3 x 3.

Soit U 4, le signal d'initialisation. Il peut être généré par un trigger de schmitt dont on prendra la sortie inversée.

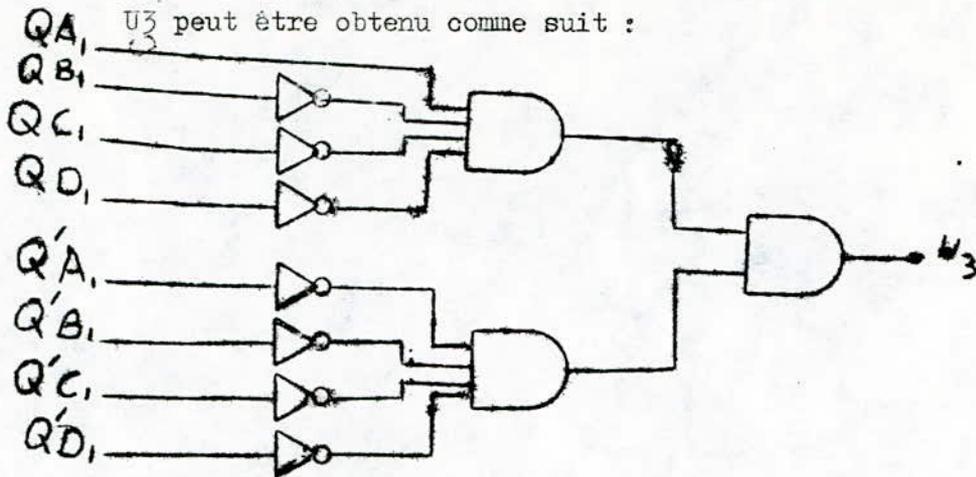
Soit le signal U'2 issu de QD2 sortie de plus fort poids du circuit V 9<sub>1</sub> de C 2.

Le signal U 3 est donné par les états de C 1

U 3 = 1 pour C 1 = 0000 0001

U 3 = 0 pour C 1 ≠ 0000 0001

.../...



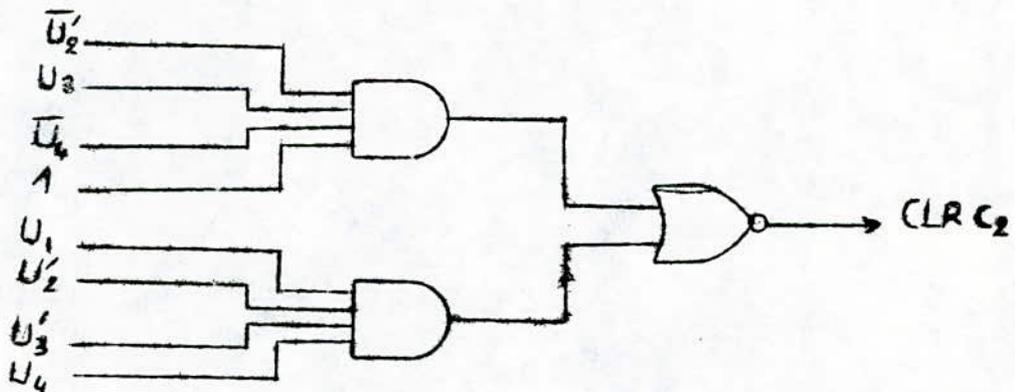
Le processus de remise à zéro est représenté sur le chronogramme.  
De ce diagramme, nous tirons la table de vérité représentée par la figure 5-12.a.

La fonction  $\overline{CLR C 2}$  étant complètement spécifiée, elle sera simplifiée en utilisant la table de Karnaugh voir figure 5-12.b.

Le signal CLR C 2 est donné par l'expression suivante :

$$CLR C 2 = \overline{U_2} U_3 \overline{U_4} + U_1 U_2 \overline{U_3} U_4$$

Ce signal est obtenu comme suit :



.../...

$U_1$	$U_2$	$U_3$	$U_4$	$\overline{CLR}$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

← Remise à zéro

← pas de remise à zéro.

a/. Tableau de vérité du  $\overline{CLR}$   $C_2$

$U_1, U_2$ $U_3, U_4$	00	01	10	11
00	0	0	0	0
01	0	0	0	1
10	1	0	1	0
11	0	0	0	0

b/. Table de KARNAUGH.

#### 5.4.4 - VALIDATION DU LATCHE RESULTAT.

Le compteur C 2 est utilisé pour valider le latche résultat E 5. Ce latche est validé après l'obtention du résultat d'un point traité. C'est-à-dire, après la 256 ème accumulation dans le cas des masques 16 x 16 ou la 9ème accumulation dans le cas des masques 3 x 3.

Dans le cas des masques 16 x 16 le signal de validation est issu de la sortie carry du circuit  $V_{92}$  de C 2.

Dans ce cas

En ce qui concerne les masques 3 x 3, le signal de validation est le signal disponible sur la sortie Q D du circuit  $V_{91}$  de C 2.

La présence de deux signaux (de validation) nécessite l'emploi d'un multiplexeur. Le signal issu de la sortie du multiplexeur est le signal de validation du latche résultat. Il est appliqué sur les entrées output control et enable des circuits du type SN 74LS373 qui composent Es.

### 5.5 - L'UNITE DE TRAITEMENT

L'unité de traitement représente le noyau du système d'amélioration d'images. Elle fait l'objet d'une étude particulière déjà entamée dans les chapitres précédents.

Dans cette partie, nous examinerons la structure de l'unité de traitement et son fonctionnement.

#### 5.5.1 - STRUCTURE DE L'UNITE DE TRAITEMENT

L'unité de traitement a été conçue suivant une structure pipeline-tubulaire, dans le but d'augmenter la fiabilité, la performance et diminuer le coût du système.

..../....

De cette structure dérive l'appellation "Processeur pipeline de traitement". Ce processeur est illustré par le schéma fonctionnel de la figure (5.13). Il est composé de quatre niveaux :

- Le premier est composé de latches (du type SN 74 LS 373) de données et d'adresses, de mémoires RAM (du type SN 74 S 208) destinées à recevoir par intermittence une fenêtre (16 x 16) éléments de l'image à traiter dans le cas de la convolution et (3x3) éléments dans le cas de l'amélioration d'image.

Ces signaux d'entrée (les pixels en provenance du plan mémoire) sont amplifiés par des buffers unidirectionnels (du type SN 74 LS 244), qui jouent le rôle de démultiplexeur.

Les signaux de sortie des mémoires attaquent directement les entrées des portes OR du type SN 74 LS 32 qui jouent le rôle de multiplexeur. De plus, il est composé d'une mémoire PROM du type SN 74 S 475 où sont stockés les coefficients des différents masques.

- Le deuxième est composé de latches de pixels et coefficients des masques destinés à l'amélioration d'images et de mémoires PROM du type SN 74 S 471 où sont stockés les produits partiels.

- Le troisième est composé de latches qui servent d'intermédiaire entre les mémoires PROM citées ci-dessus et d'additionneurs du type SN 74 83 qui effectuent l'addition des produits partiels.

- Le quatrième est constitué de latches destinés à recevoir le résultat de la multiplication et d'accumulateur du type SN 74 S 281 associés à des circuits carry lookahead du type SN 74 182 qui augmente la vitesse de propagation de la retenue d'un circuit à l'autre ; cet accumulateur termine l'exécution de l'algorithme vu précédemment.

.../...

(Schéma de brochage et connection avec le SN 74 182 se référer à la fig. Annexe).

Il est clair que les niveaux sont découplés à l'aide de latches dont le rôle est très important dans la réalisation du système pipeline.

## 5.5.2 - FONCTIONNEMENT DE L'UNITE DE TRAITEMENT

### L'UNITE MEMOIRE

Les mémoires RAM, adressées par le compteur C 1, reçoivent les données à travers des buffers unidirectionnels. Ces buffers sont utilisés du fait que les mémoires travaillent par intermittence. Lorsque l'une des mémoires est en état de lecture, cas où la sortie du buffer correspondant est en haute impédance la deuxième mémoire est en état d'écriture d'où transfert des données du buffer vers cette dernière.

Il est facile de remarquer que ces buffers jouent le rôle de démultiplexeur. Ceci peut être expliqué par le fait qu'ils reçoivent les données qui sont transférées à la mémoire considérée en fonction des commandes d'écriture/lecture.

Pour que les adresses et les pixels arrivent en même temps aux mémoires, ils sont transmis auparavant dans des latches synchrones.

La mémoire PROM où sont stockés les coefficients des masques ( de convolution et d'amélioration d'images) est adressée selon le mode indéré par le compteur C 1. Les coefficients des masques sont transférés en même temps que les pixels dans l'unité arithmétique à travers les latches du niveau suivant (voir schéma fonctionnel du processeur de traitement figure 5.13).

## L'UNITE ARITHMETIQUE

Les données provenant de l'unité mémoire formeront les adresses des mémoires PROM contenant les produits partiels.

La multiplication de deux nombres de 8 bits nécessite l'utilisation de quatre de ces mémoires. A la sortie de chacune d'elles, il y a un produit de l'opération effectuée (4 x 4 bits), du fait que chaque mot est subdivisé en deux groupes de 4 bits (cf fig. 5.14)

Une fois les mémoires adressées, les produits partiels sont disponibles à leurs sorties et sont acheminées vers l'étage suivant où s'effectue l'addition dans le but de générer un résultat final exact.

Avec cette méthode de multiplication, le signe du résultat est traité séparément. Des pixels et coefficients de masque qui arrivent à l'unité arithmétique, le bit le plus significatif dans les deux cas (bit représentant le signe) est envoyé directement dans une porte XOR du type SN 74 LS 86. Le bit signe du résultat est obtenu suivant la table de vérité (voir annexe fig ).

Le signe ainsi traité est envoyé en même temps que le résultat de la multiplication dans l'accumulateur.

Le principe pipeline est réalisé grâce à l'utilisation de latches intercalaires entre les différents niveaux.

Les produits

Les produits issus du multiplicateur sont accumulés séquentiellement. Dans toute opération d'accumulation, il y a risque de dépassement ce qui fausse le résultat. Dans le but d'éviter ce problème, la normalisation des résultats de la multiplication paraît indispensable. Ce procédé permet d'éviter la

.../...

perte d'information qui se présente quand le problème de dépassement est évité par décalage.

Dans notre cas, une division par  $2^8$  est effectuée sur le résultat de la multiplication.

De ce fait, au lieu d'avoir une accumulation sur 16 bits, celle-ci s'effectuera sur 24 bits. Le résultat final est multiplié par le même facteur de normalisation. Ce résultat transféré dans le latche dit "latche résultat" est donné sur 8 bits, les plus significatifs. Le fait d'omettre les autres bits de plus faible poids ne constitue pas une erreur importante.

Le pixel de l'image traitée est transféré du latche résultat à la mémoire de restitution.

Le processus pipeline adopté dans le système d'amélioration d'images est illustré par la figure (5.13) qui représente le schéma fonctionnel du processeur pipeline de traitement.

En se référant au chronogramme du distributeur d'horloges, le processus pipeline peut-être expliqué comme suit :

- A la première impulsion d'horloge  $h_1$ , (qui valide les latches de niveaux impairs) les données  $E_n$  (Pixels et coefficients des masques) sont automatiquement envoyées dans l'unité arithmétique où, dès que la première impulsion de  $h_3$  arrive les produits partiels sont disponibles aux sorties des PROM.

- A la deuxième impulsion, de  $h_1$  s'effectue l'addition des produits partiels.

- A la deuxième impulsion de  $h_3$ , le premier produit s'accumule alors qu'un autre produit se prépare.

.../...

Ainsi le processus est continu du fait de la structure pipeline et les mémoires qui travaillent par intermittence, donc pas de coupure dans les informations.

Ceci peut être illustré par le diagramme espace-temps (figure 5.15) pour l'obtention de quelques points traités (par exemple 3 points).

D'après le diagramme espace-temps, on remarque qu'on a un résultat d'accumulation à chaque impulsion de  $h3$ .

D'après le calcul de la fréquence de travail effectué précédemment une accumulation est faite tous les 182,85 ns.

Le temps d'accumulation est couvert par le temps de la multiplication. Si  $\tau$  est le temps d'un niveau,  $N$  le nombre d'opérations à effectuer pour avoir un point de l'image traitée, le temps  $T$  de traitement d'un point est donné par la relation:

$$T = 2\tau + (N - 1)\tau = (N + 1)\tau$$

Pour la fréquence calculée et si  $N = 256$

$$T = 257 \cdot 182,85 = 46,99 \mu s$$

$T_{lp} = 46,99 \mu s$
------------------------

### 5.5.3 - CONCLUSION

Ce temps est estimé considérable du fait qu'on voulait réaliser un système très rapide. Mais ceci est expliqué par le choix de la chaîne de synchronisation des différents latches. Cette chaîne a été choisie de façon à éviter

.../...

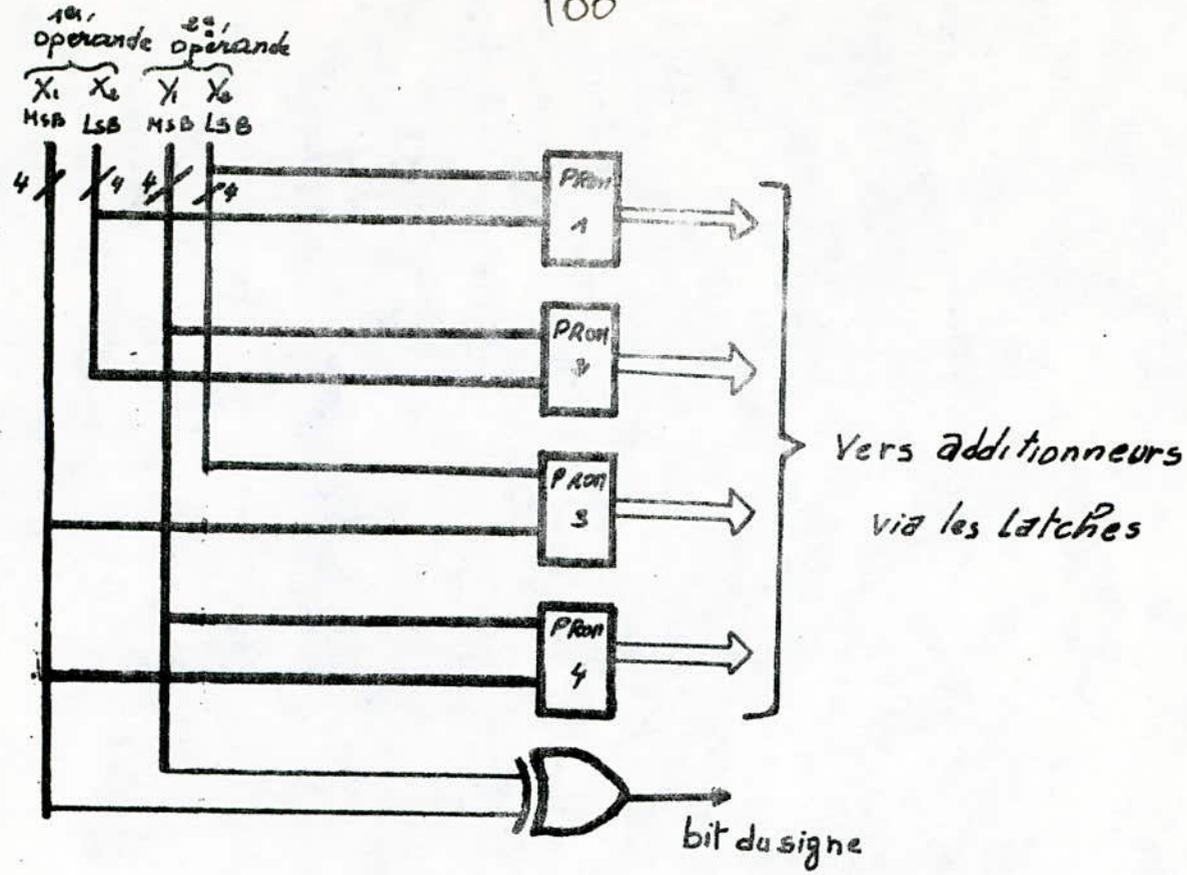
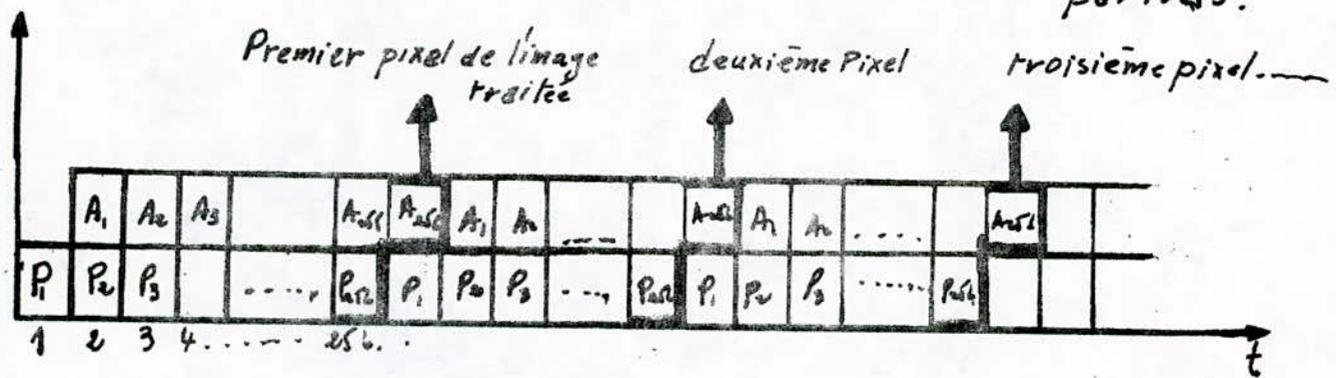


Fig 5.14. disposition des table contenant les produits partiels.



A chaque impulsion de h<sub>s</sub> un produit se prépare  
 A la 2<sup>e</sup> impulsion la 1<sup>er</sup> accumulation a lieu.  
 Apres 256 impulsion le 1<sup>er</sup> pixel de l'image traitée est disponible

{ P: designe le produit  
 { A: l'accumulation

Fig. 5.15. Diagramme espace-temps du processus pipeline de traitement.

tout phénomène de recouvrement des données dans les différents niveaux du système.

Ce temps peut-être amélioré en utilisant des mémoires à accès très rapide.

## 5.6 - REALISATION D'UN SYSTEME D'AMELIORATION D'IMAGES

La réalisation du système conçu précédemment ne peut être mise au point avec les composants cités dans les paragraphes précédents. Ces derniers n'étant pas disponibles, nous avons été amené à utiliser leurs équivalents du point de vue fonctionnel. En ce qui concerne la rapidité de traitement les composants utilisés ont un temps de réponse très élevé.

La structure du système d'amélioration d'images est la même à quelques modifications près comme indiqué dans la figure 5.16.

Les modifications effectuées sont précisées dans le tableau suivant :

COMPOSANTS UTILISES DANS :

	IA CONCEPTION	IA REALISATION
RAM	SN 74 S 208	TMM 2016
Buffers	SN 74 LS 244	MC 688 9 8 T 28
PROM	SN 74 S 475	MCM 2716
PROM	SN 74 S 471	
ACC	SN 74 S 281	UAL SN 74 LS 181 + circuits SN 74 LS 163;

Ce tableau peut être expliqué comme suit :  
Ce Tableau peut être expliqué comme suit :

.../....

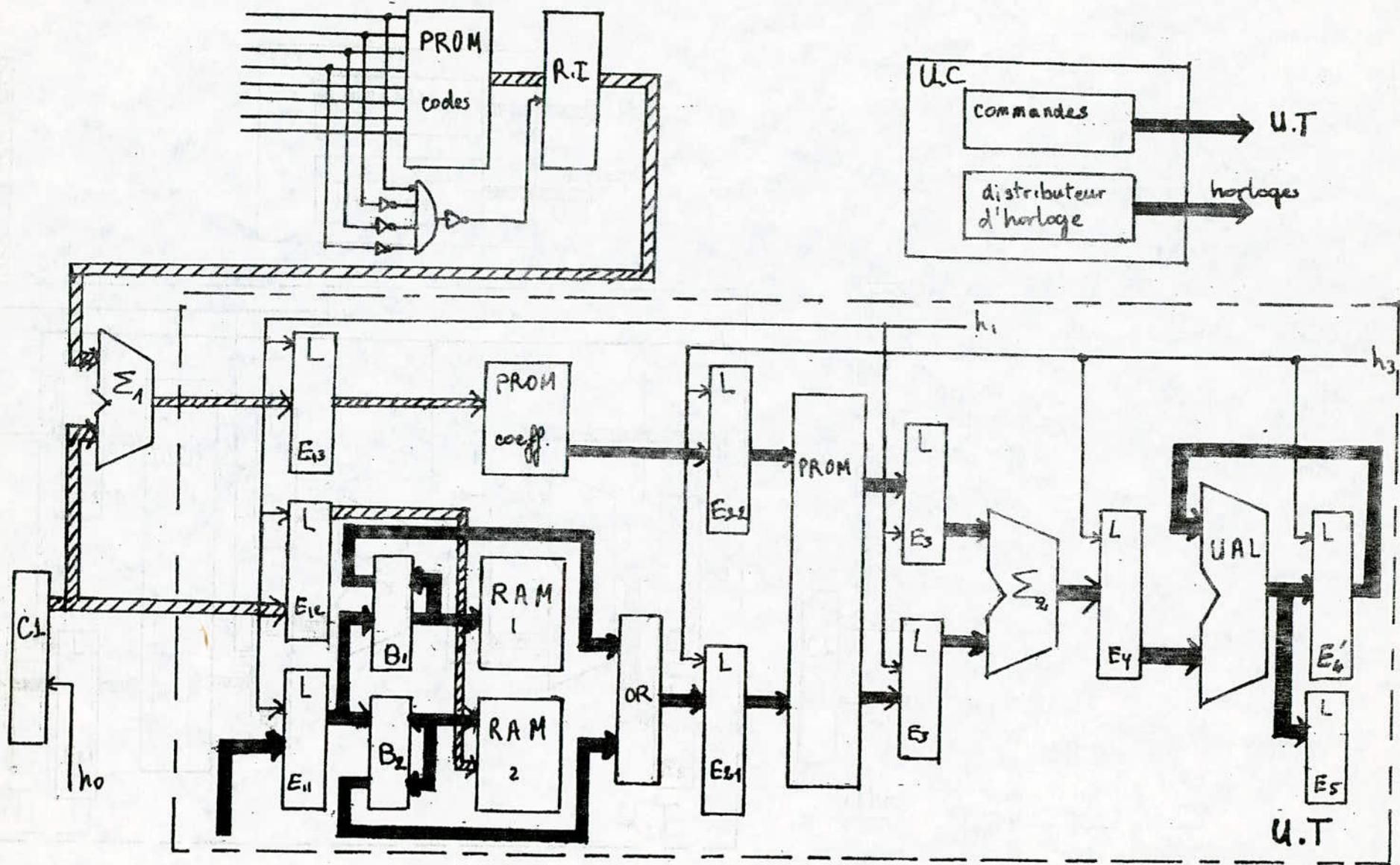


Fig. 5-16 Architecture du système d'amélioration d'images (réalisation)

1031

é L'accumulateur de la réalisation est illustré par la figure (5.17). Le latche sert à réintroduire les résultats des accumulations précédentes dans l'unité arithmétique et logique (UAL). Ce latche est composé de six circuits du type SN 74 LS 163 dont les entrées "Load" sont mises à zéro, ce qui permet un chargement en parallèle des données qui viennent de l'UAL.

Ces circuits sont utilisés dans le but d'effectuer la remise à zéro de l'accumulateur considéré. Les latches d'entrée/sortie de l'accumulateur ont le même signal de synchronisation ce qui permet aux données d'arriver en même temps aux entrées de l'UAL où s'effectue l'accumulation.

- Les mémoires RAM TMM 2016 sont associées à des buffers bidirectionnels du fait qu'elles ont un bus d'entrée - sortie commun. (Voir le schéma fonctionnel fig. 5.38)

- L'utilisation de la structure tabulaire dans la réalisation du système d'amélioration d'images, nous a amené à programmer les produits partiels dans les mémoires EPROM MCM 2716 (2 K x 8 bits)

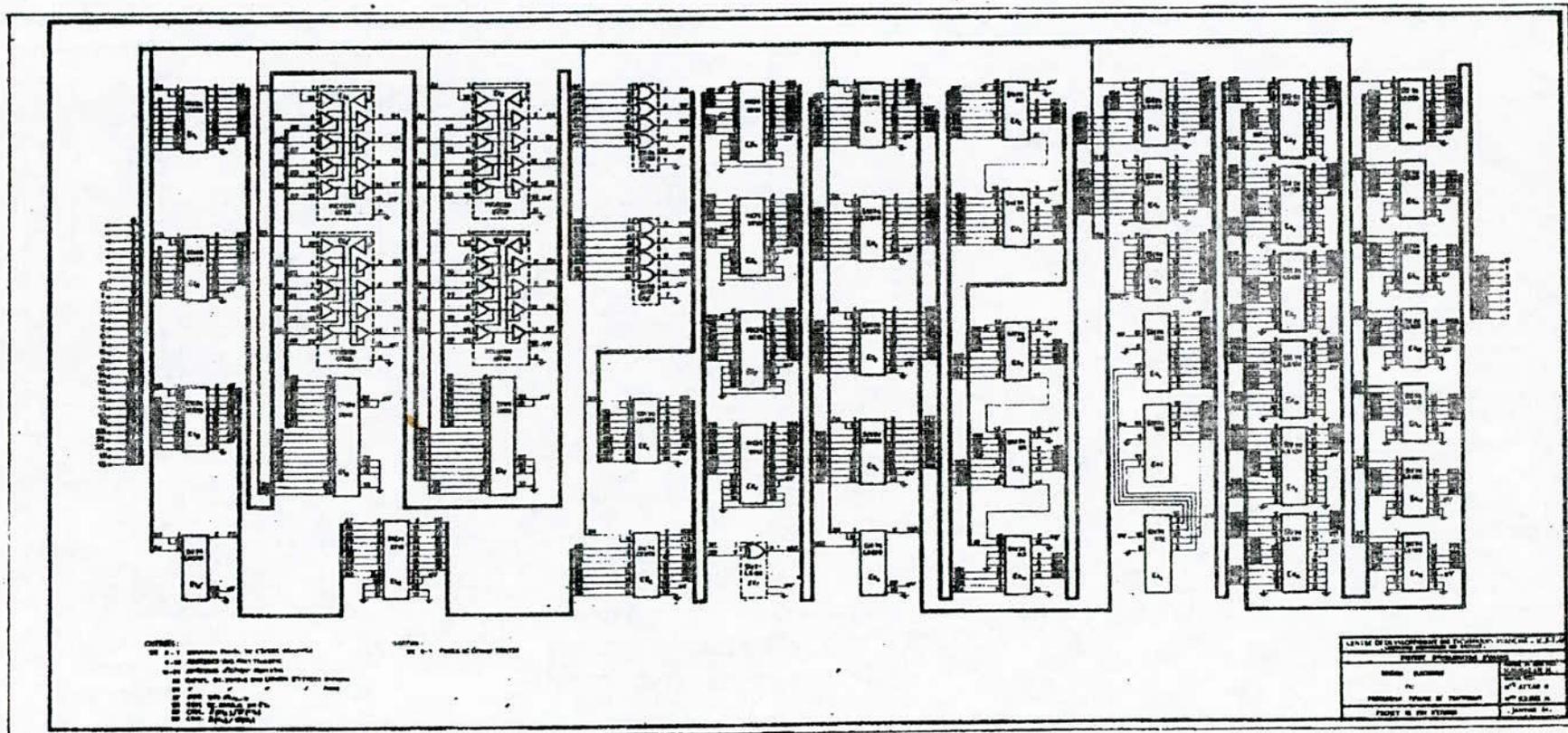
Nous effectuons la multiplication 4 x 4 bits de N par B tel que :

$$A \in \{0, 1, \dots, F\}$$

$$B \in \{0, 1, \dots, F\}$$

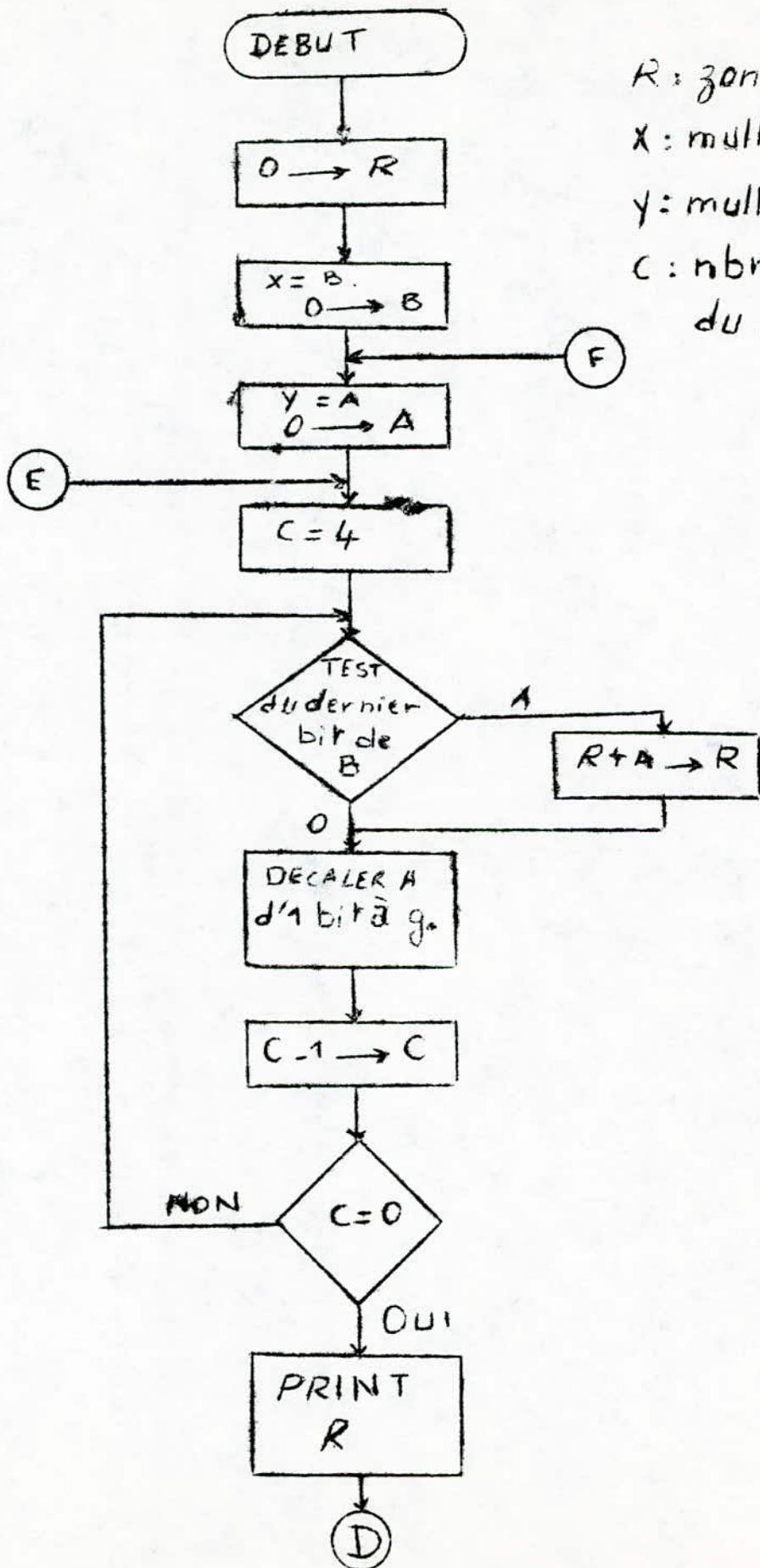
Les deux nombres varient de "0" à "F" (15 en hexadécimal), seulement A varie 16 fois plus vite que B. De ce fait, nous effectuons 256 opérations dont les résultats sont stockés dans l'EPROM.

L'organigramme de la multiplication est illustré par la figure 5.19. Le programme en assembleur correspondant à cet organigramme est donné par la fig. 5.20.



105

Fig. 5.18 Schéma fonctionnel de l'Unité de traitement (réalisation).



$R$ : zone résultat  
 $x$ : multiplieur  
 $y$ : multiplicande  
 $c$ : nbre de bits  
 du multiplieur.

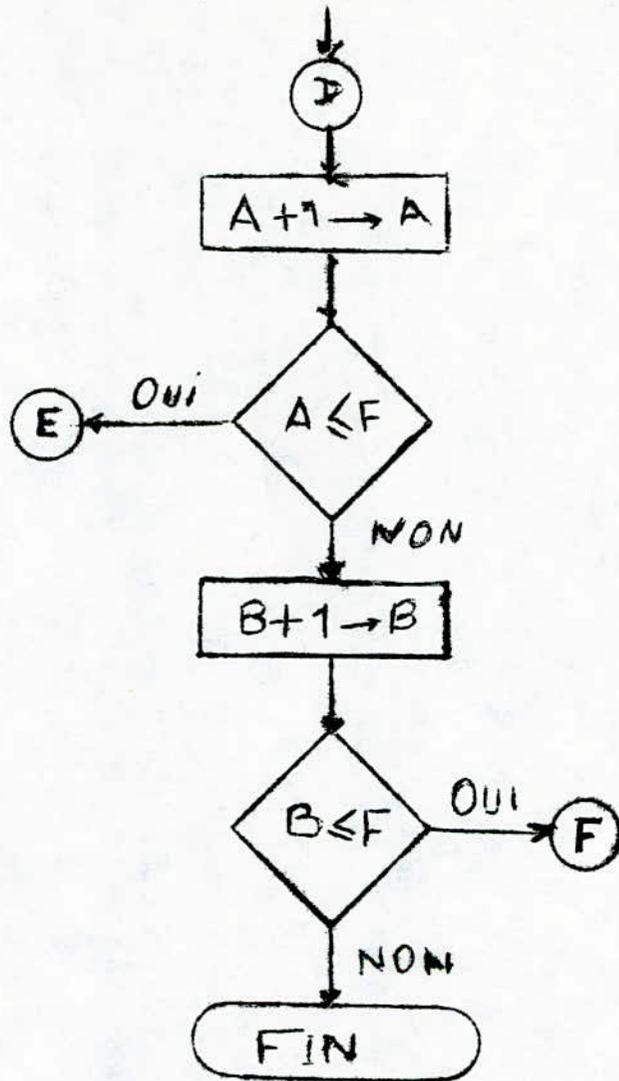


Fig. 5.18. Organigramme de la  
1 19 Multiplication 4x4 bits.

108

Fig. 520. PROGRAMME DE LA MULTIPLICATION

Etiquette	PROG. en L.A	ADRESSE	PROG. en L.M.	Commentaires
	ØRC \$ 4 Ø Ø Ø			4 Ø Ø Ø adresse du début du prog.
	LDS # \$ Ø 1 FF	4 Ø Ø Ø	8 E Ø 1 FF	
	LDX \$ 68 Ø Ø	4 Ø Ø 3	C E 68 Ø Ø	
	CLR \$ 6 Ø 18	4 Ø Ø 6	7 F 6 Ø 18	Ø Ø → 6 Ø 18, (MD)
	CLR B	4 Ø Ø 9	5 F	Ø Ø → B
BETA	CLR \$ 6 Ø 13	4 Ø Ø A	7 F 6 Ø 13	Ø Ø → 6 Ø 13
	CLR \$ 6 Ø 14	4 Ø Ø D	7 F 6 Ø 14	Ø Ø → 6 Ø 13
TETA	LDAA # \$ Ø 4	4 Ø 1 Ø	86 Ø 4	Ø 4 → A (nbre de bits du MR)
	STAA \$ 6 Ø 15	4 0 12	B7 6 Ø 15	(A) → 6 Ø 15
	LDAB \$ 6 Ø 18	4 Ø 15	F6 6 Ø 18	(6 Ø 18) → B
	CLR A	4 Ø 18	4 F	Ø Ø → A
BOUCLE	LSR B	4 Ø 19	54	
	BCC GAMMA	4 Ø 1 A	24 Ø 3	Branchement à GAMMA si C = 0
	ADDA \$ 6 Ø 14	4 Ø 1 C	BB 6 Ø 14	(A) + (6 Ø 14) → A
GAMMA	ASL \$ 6 Ø 14	4 Ø 1 F	78 6 Ø Ø 14	DECALAGE A GAUCHE
	DEC \$ 6 Ø 15	4 Ø 22	7 A 6 Ø 15	(6 0 15) - 1 → 6 Ø 15
	BNE BOUCHE	4 Ø 25	26 F 2	Branchement si ≠ 0
	STAA Ø Ø, X	4 Ø 2 7	A 7 Ø Ø	Ø Ø → RI
	INX	4 Ø 29	Ø 8	(R.I) + 1 → RI
	LDAA \$ 6 Ø 13	4 Ø 2 A	B6 6 Ø 13	(6 Ø 13) → A
	INC A	4 Ø 2 D	4 C	(A) + 1 → A
	STAA \$ 6 Ø 13	4 Ø 2 E	B7 6 Ø 13	(A) → 6 Ø 13
	STAA \$ 6 Ø 14	4 Ø 31	B7 6 Ø 14	(A) → 6 Ø 13
	CMPA # \$ Ø F	4 Ø 34	8 1 Ø F	COMPARAISON
	BLS TETA	4 Ø 36	23 D8	Branch. si <
	LDAB \$ 6 Ø 18	4 Ø 38	F 6 6 Ø 18	(6 Ø 18) → B
	INC B	4 Ø 3 B	5 C	(B) + 1 → DB
	STAB \$ 6 Ø 18	4 Ø 3 Ø	F 7 6 Ø 18	(B) → 6 Ø 18
	CMPB # \$ Ø F	4 Ø 3 F	C 1 Ø F	comparaison de (B) avec F
	BLS BETA	4 Ø 4 1	23 CF	Branch. si <
	SWI	4 Ø 43	3 F	Fin de programme.

Après l'exécution de ce programme, les résultats sont stockés dans une zone mémoire RAM. Le transfert du contenu de cette zone dans l'EPROM est effectué à l'aide du programme de commande suivant :

```

EXBUUG 1.2 MAID
* E 8 0 0 ; C
MDOS 02
= PROMPROG
P MCM 2716
P 0 FSET 0
P PROG 5 0 0 0, 5 0 F F, 0
P VERIF 5 0 0 0, 5 0 F F, 0
P READ 6 0 2 0, 6 11 F, 0

```

Dans la réalisation, nous travaillons à une fréquence basse. La valeur de la fréquence de l'horloge mère qui sert à générer le cycle d'horloges à temps disjoints est la suivante :

$t_1$  = temps d'accès de la RAM  
 $t_2$  = temps de propagation dans le buffer  
 $t_3$  = temps de propagation dans les portes OR

$$d' = \sum_{i=1}^3 t_i = 450 + 17 + 7 = 474 \text{ ns}$$

$$f = \frac{7}{4d'} = 3,69 \text{ MHz}$$

La fréquence de travail est :

$$f_t = \frac{3,69}{4} = 0,92 \text{ MHz}$$

## C O N C L U S I O N

L'étude faite sur le traitement des images a influencé notre choix concernant la conception et la réalisation du système d'amélioration d'images.

Les manipulations d'amélioration d'images sont caractérisées par un flot important d'opérations identiques du type convolution discrète.

Ce fait a défini l'objectif de notre travail comme suit : la conception et la réalisation d'un système performant du point de vue rapidité, encombrement et coût. C'est pourquoi une étude des différentes méthodes algorithmiques et structurelles d'accélération de l'opération de multiplication (opération de base) a été faite.

Cette étude a été déterminante dans le choix de la structure pipeline tabulaire du système conçu à base de composants rapides.

Donc une conception finie d'un système d'amélioration d'images a été effectuée.

Seulement, vu la non disponibilité des composants rapides, dans la partie réalisation, nous avons utilisé des composants équivalents mais dont le temps de réponse est plus grand, et ce, pour vérifier l'aspect fonctionnel du système.

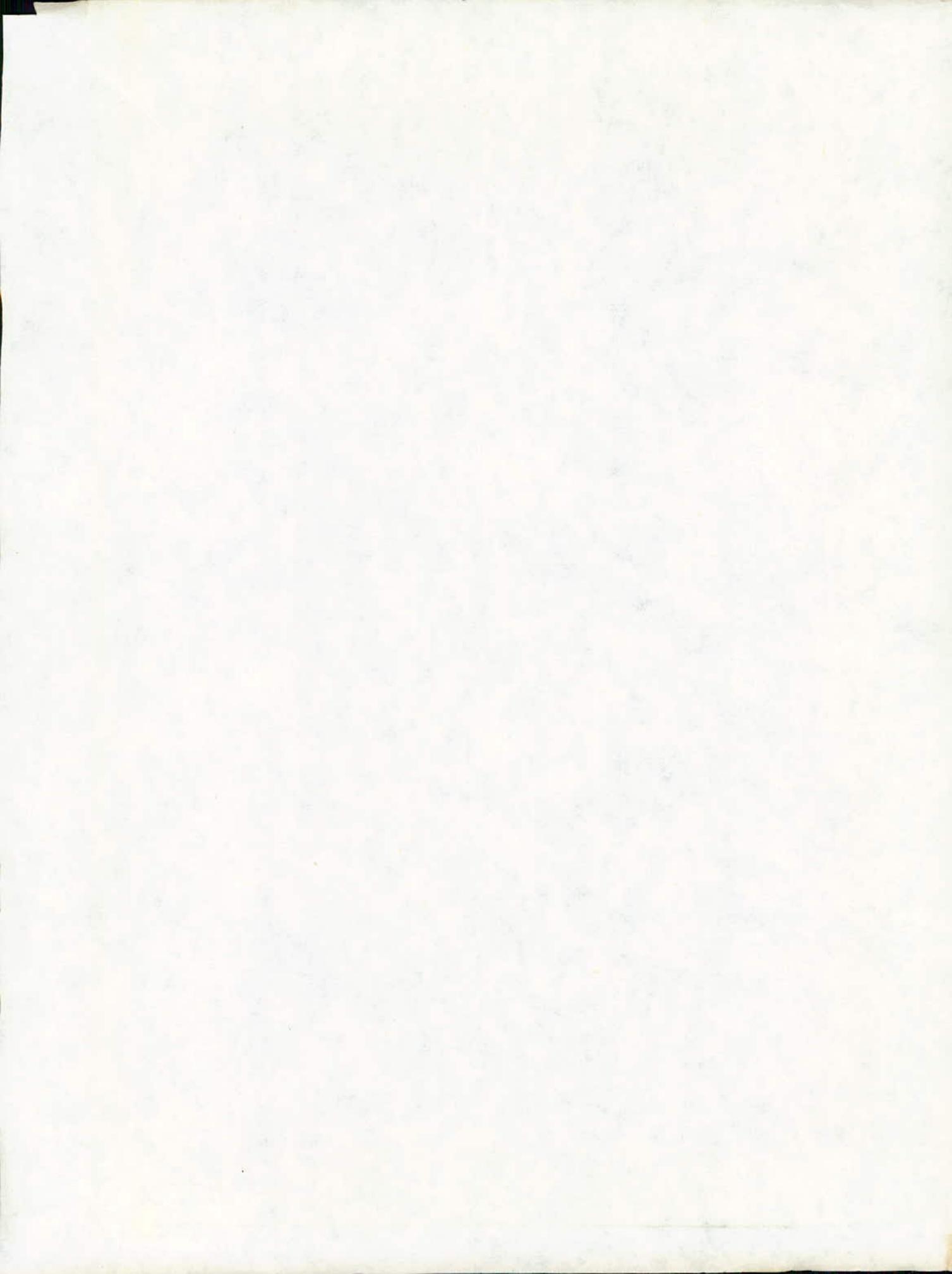
Les parties acquisition et restitution des données ne font pas l'objet de notre travail.

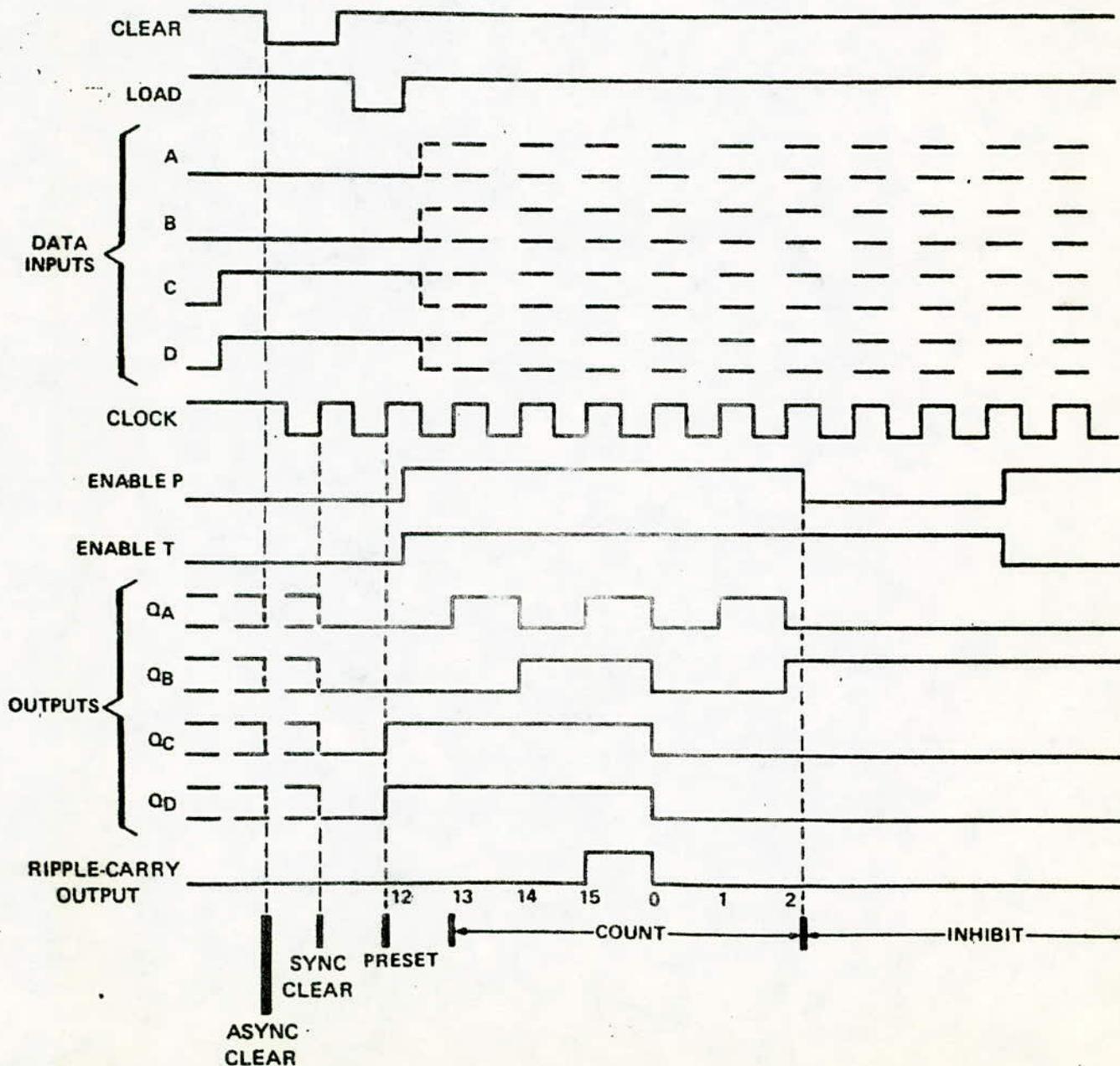
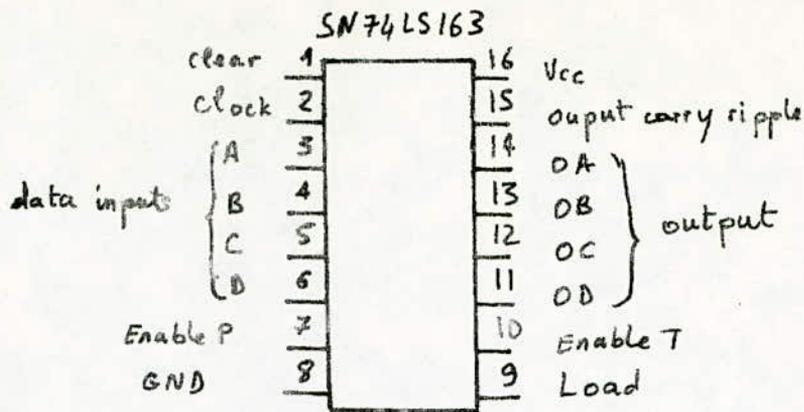
Le système d'amélioration d'images proposé peut être utilisé dans toutes les applications nécessitant un traitement bidimensionnel de données tel que le calcul matriciel.

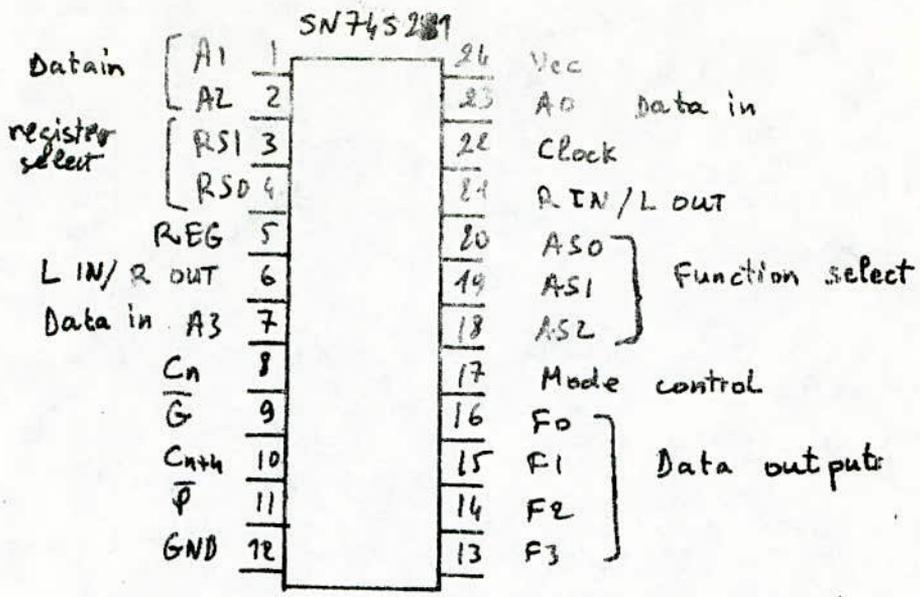
-----

A N N E X E S

A N N E X E S





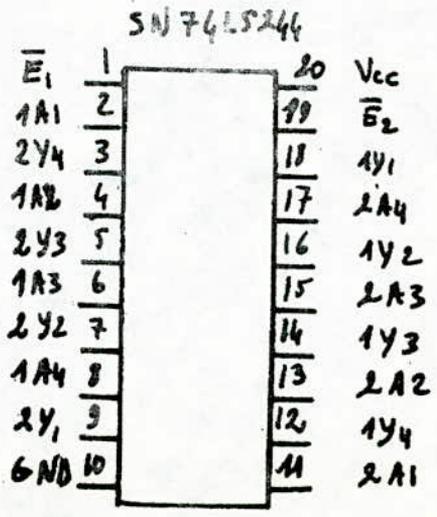
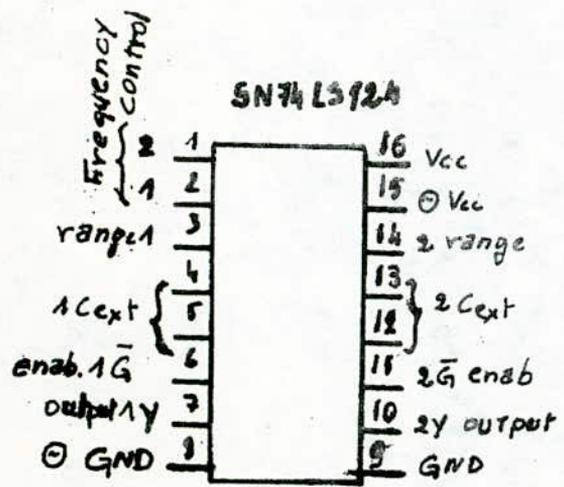
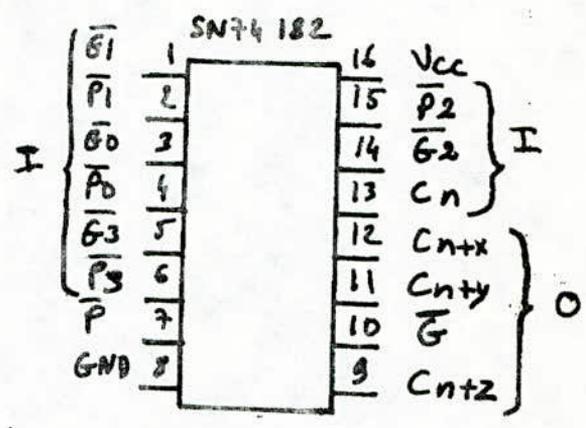
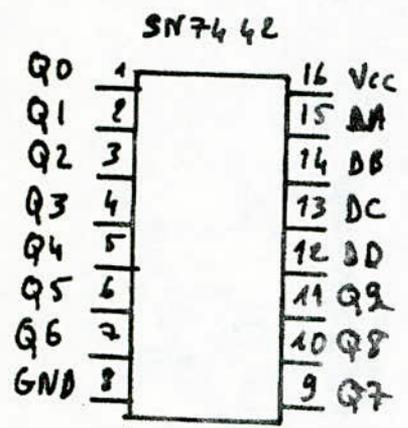
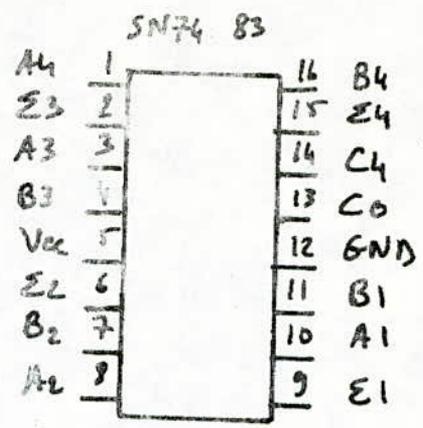
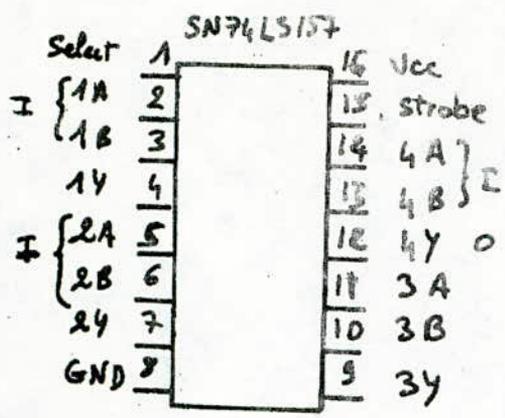


**TABLE 1-ARITHMETIC FUNCTIONS**  
Mode Control (M) = Low

ALU SELECTION			ACTIVE-HIGH DATA	
AS2	AS1	AS0	C <sub>n</sub> = H (with carry)	C <sub>n</sub> = L (no carry)
L	L	L	F <sub>0</sub> = L, F <sub>1</sub> = F <sub>2</sub> = F <sub>3</sub> = H	F <sub>n</sub> = H
L	L	H	F = B MINUS A	F = B MINUS A MINUS 1
L	H	L	F = A MINUS B	F = A MINUS B MINUS 1
L	H	H	F = A PLUS B PLUS 1	F = A PLUS B
H	L	L	F = B PLUS 1	F <sub>n</sub> = B <sub>n</sub>
H	L	H	F = B̄ PLUS 1	F <sub>n</sub> = B̄ <sub>n</sub>
H	H	L	F = A PLUS 1	F <sub>n</sub> = A <sub>n</sub>
H	H	H	F = Ā PLUS 1	F <sub>n</sub> = Ā <sub>n</sub>

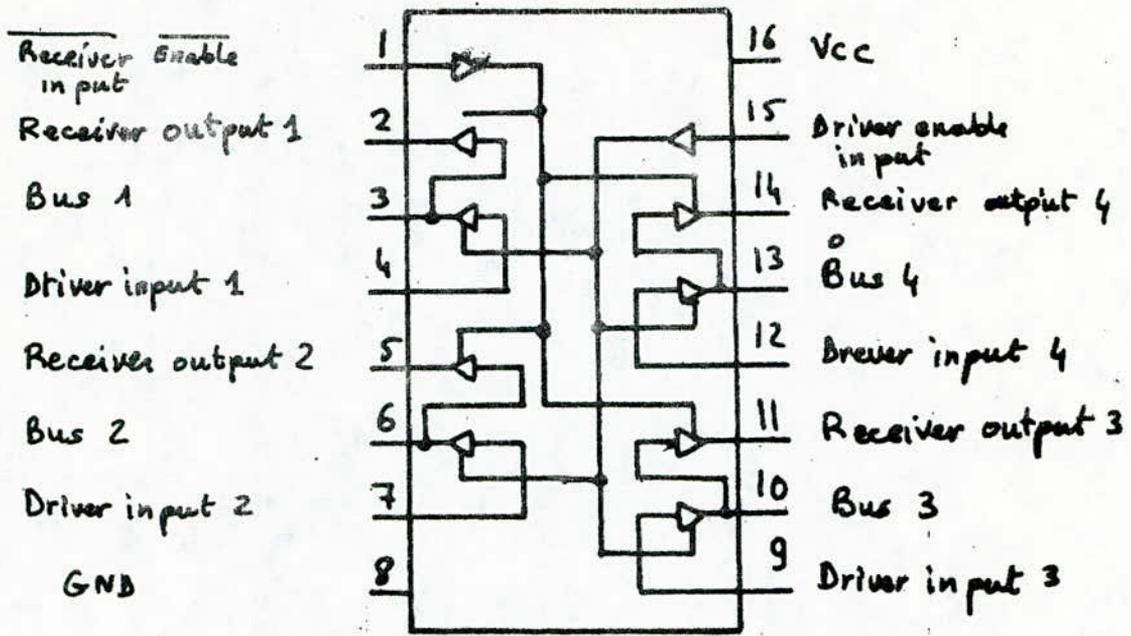
**TABLE 2-LOGIC FUNCTIONS**  
Mode Control (M) = High  
Carry Input (C<sub>n</sub>) = X (Irrelevant)

ALU SELECTION			ACTIVE-HIGH DATA FUNCTION
AS2	AS1	AS0	
L	L	L	F <sub>n</sub> = L
L	X	H	F <sub>n</sub> = A <sub>n</sub> ⊕ B <sub>n</sub>
L	H	L	F <sub>n</sub> = A <sub>n</sub> ⊕ B <sub>n</sub>
H	L	L	F <sub>n</sub> = A <sub>n</sub> B <sub>n</sub>
H	L	H	F <sub>n</sub> = A <sub>n</sub> + B <sub>n</sub>
H	H	L	F <sub>n</sub> = A <sub>n</sub> B <sub>n</sub>
H	H	H	F <sub>n</sub> = A <sub>n</sub> + B <sub>n</sub>



inputs		output
E1	E2	D
L	L	L
L	H	H
H	X	(Z)
H	X	(Z)

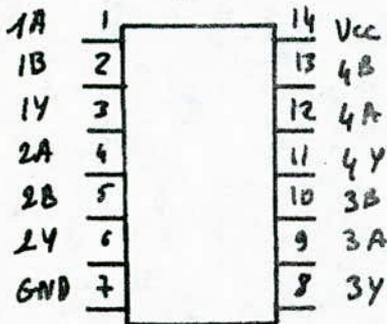
### MC 6989/MC 8T28



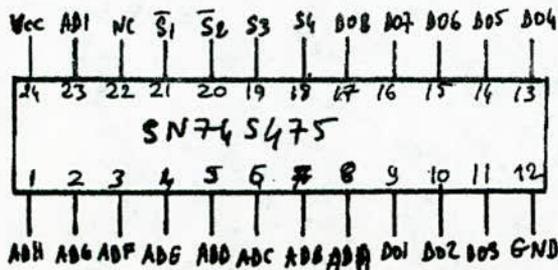
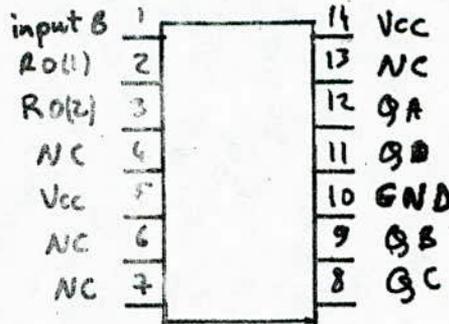
les commande  $\bar{L}$ ,  $P_1$  et  $P_2$  ont le même signal.

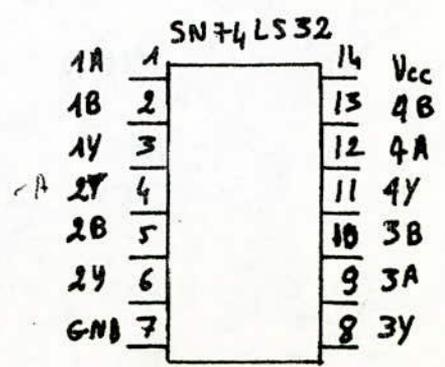
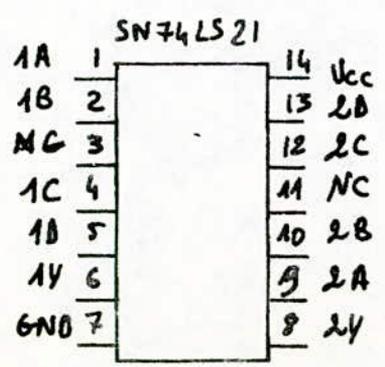
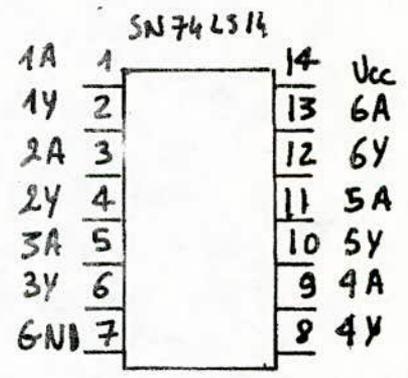
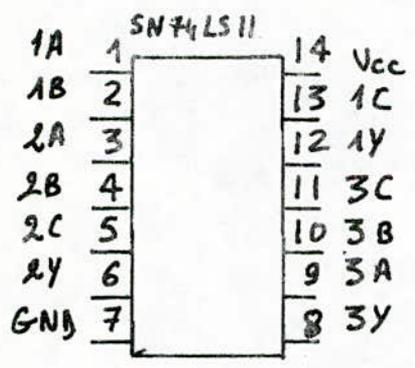
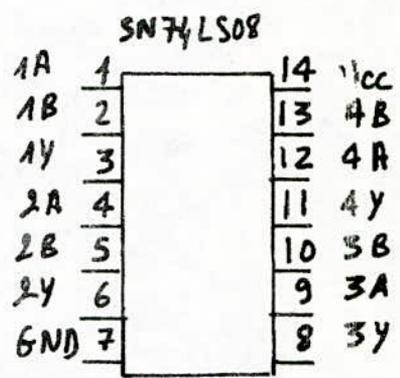
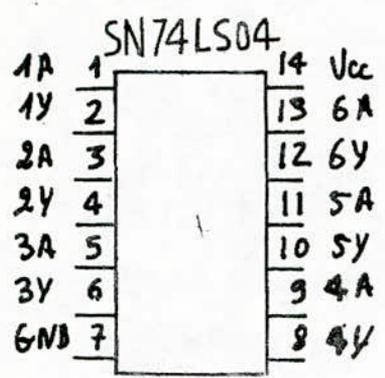
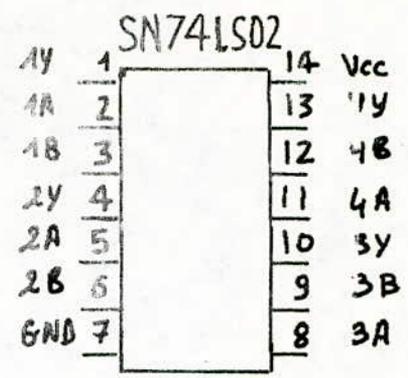
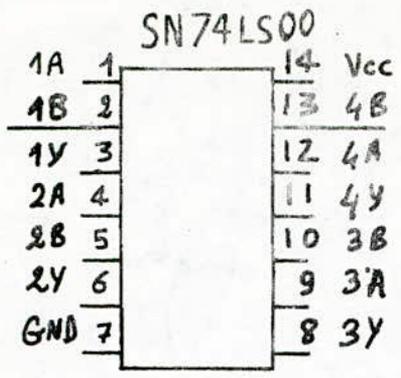
	$\bar{L}$	$P_1$	$P_2$	$P_3$	
RAM 1	0	1	1	1	→ écriture
RAM 2	1	0	0	0	→ lecture
RAM 1	1	0	0	0	→ lecture
RAM 2	0	1	1	1	→ écriture

### SN74LS86

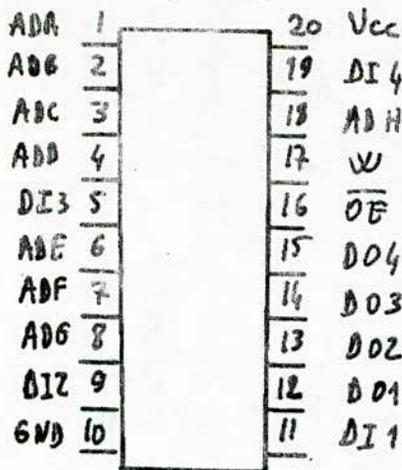


### SN74LS93

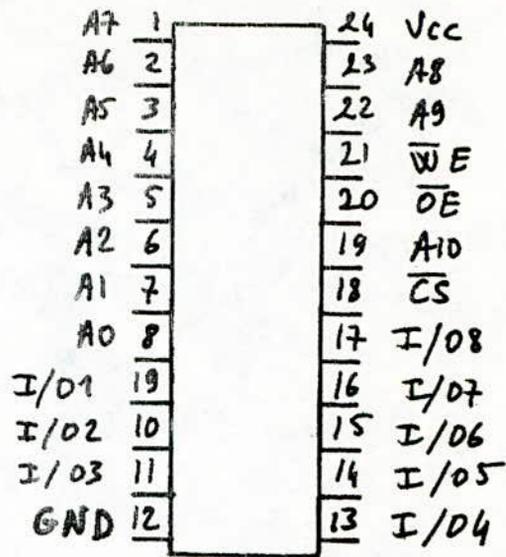




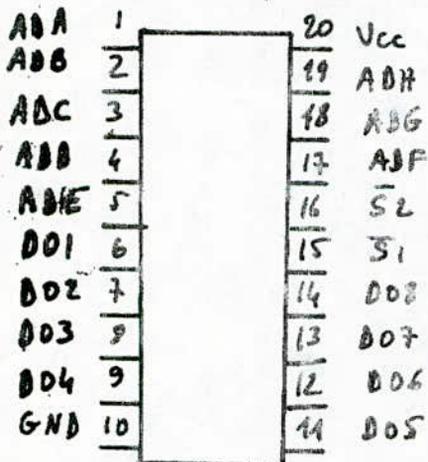
SN74S 208



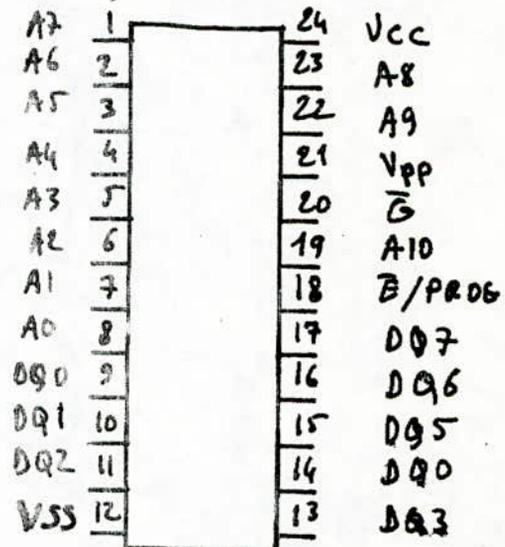
TMM 2016



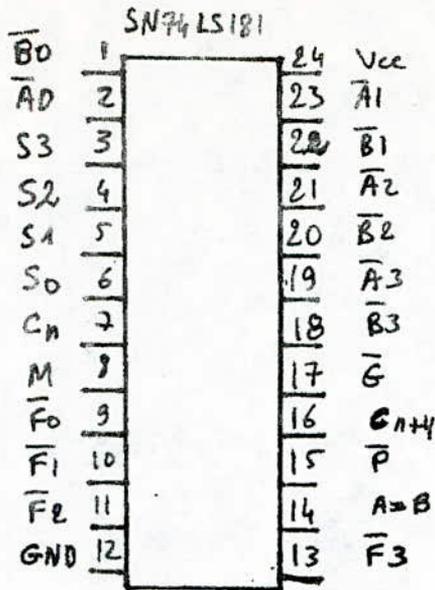
SN74S471



MCN 2716



DO<sub>n</sub> : data input/output  
 $\overline{E/PROG}$  : chip enable / prog  
 $\overline{G}$  : output enable



SELECTION					ACTIVE-HIGH DATA		
					M = H LOGIC FUNCTIONS	M = L; ARITHMETIC OPERATIONS	
						C <sub>n</sub> = H (no carry)	C <sub>n</sub> = L (with carry)
S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>				
L	L	L	L	$F = \overline{A}$	$F = A$	$F = A \text{ PLUS } 1$	
L	L	L	H	$F = \overline{A+B}$	$F = A+B$	$F = (A+B) \text{ PLUS } 1$	
L	L	H	L	$F = \overline{AB}$	$F = A+B$	$F = (A+B) \text{ PLUS } 1$	
L	L	H	H	$F = 0$	$F = \text{MINUS } 1 \text{ (2's COMPL)}$	$F = \text{ZERO}$	
L	H	L	L	$F = \overline{AB}$	$F = A \text{ PLUS } \overline{AB}$	$F = A \text{ PLUS } \overline{AB} \text{ PLUS } 1$	
L	H	L	H	$F = \overline{B}$	$F = (A+B) \text{ PLUS } \overline{AB}$	$F = (A+B) \text{ PLUS } \overline{AB} \text{ PLUS } 1$	
L	H	H	L	$F = A \oplus B$	$F = A \text{ MINUS } B \text{ MINUS } 1$	$F = A \text{ MINUS } B$	
L	H	H	H	$F = \overline{AB}$	$F = \overline{AB} \text{ MINUS } 1$	$F = \overline{AB}$	
H	L	L	L	$F = \overline{A+B}$	$F = A \text{ PLUS } AB$	$F = A \text{ PLUS } AB \text{ PLUS } 1$	
H	L	L	H	$F = A \oplus B$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B \text{ PLUS } 1$	
H	L	H	L	$F = B$	$F = (A+B) \text{ PLUS } AB$	$F = (A+B) \text{ PLUS } AB \text{ PLUS } 1$	
H	L	H	H	$F = \overline{AB}$	$F = \overline{AB} \text{ MINUS } 1$	$F = \overline{AB}$	
H	H	L	L	$F = 1$	$F = A \text{ PLUS } A^*$	$F = A \text{ PLUS } A \text{ PLUS } 1$	
H	H	L	H	$F = A+B$	$F = (A+B) \text{ PLUS } A$	$F = (A+B) \text{ PLUS } A \text{ PLUS } 1$	
H	H	H	L	$F = A+B$	$F = (A+B) \text{ PLUS } A$	$F = (A+B) \text{ PLUS } A \text{ PLUS } 1$	
H	H	H	H	$F = A$	$F = A \text{ MINUS } 1$	$F = A$	



SN 7442

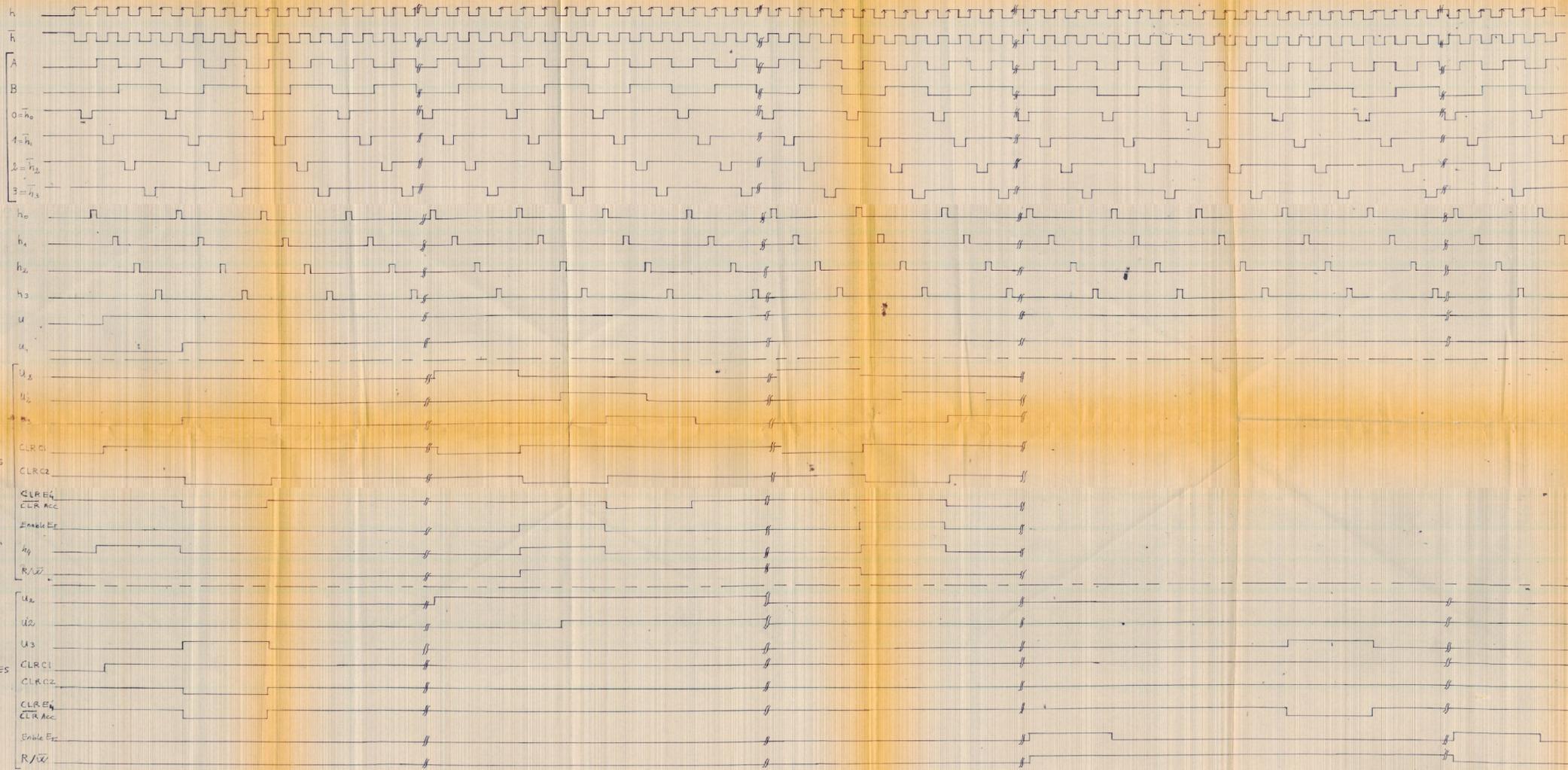
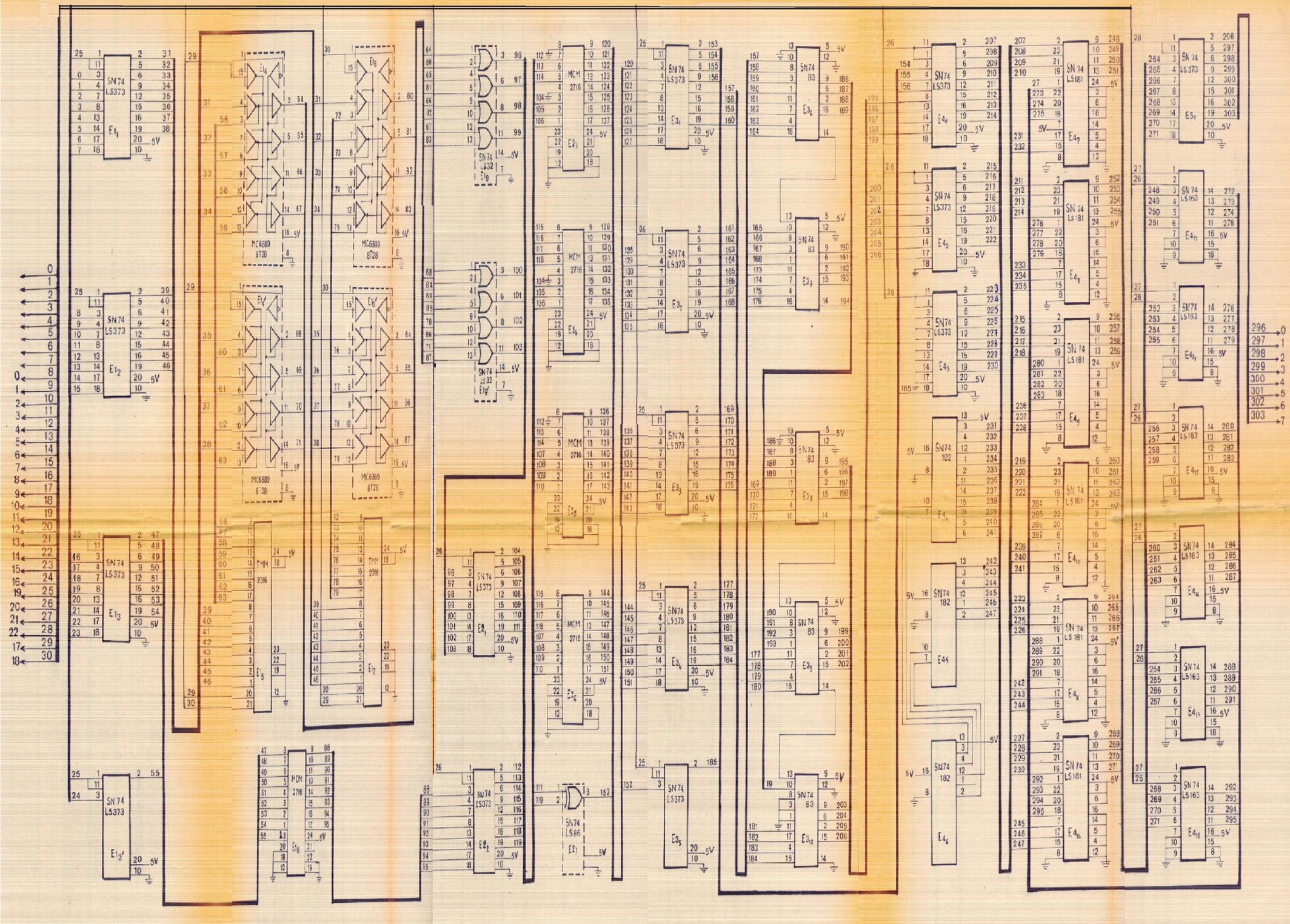


Diagramme de fonctionnement.



- ENTREES:
- DE 0-7 DONNEES PIXELS DE L'IMAGE ORIGINALE
  - 8-15 ADRESSES DES RAM THM 20M
  - 16-24 ADRESSES D'EPRON THM 2216
  - 25 COMM. DC ENAB.G DES LATCHES D'ETAGES IMPAIRS
  - 26 C H H H PAIRS
  - 27 PAZ DES E<sub>0</sub>-E<sub>7</sub>
  - 28 COM. DC ENAB.G DU E<sub>0</sub>
  - 29 COM. R(E<sub>0</sub>)/W(E<sub>0</sub>)
  - 30 COM. R(E<sub>1</sub>)/W(E<sub>1</sub>)

- SORTIES:
- DE 0-7 PIXELS DE L'IMAGE TRAITEE

CENTRE DE DEVELOPPEMENT DES TECHNIQUES AVANCEES - (C.D.T.A.)  
LABORATOIRE ARCHITECTURE DES SYSTEMES

SYSTEME D'AMELIORATION D'IMAGES

SCHEMA ELECTRIQUE	PROFES. ET SUIV. PARS M.BESSALAH H.
EPU	M <sup>de</sup> ATTAB H.
PROCESSEUR PPELNE DE TRAITEMENT	M <sup>de</sup> KALDES M.
PROJET DE FIN D'ETUDES	JANVIER 84.