

3/85

وزارة التعليم و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT ELECTRONIQUE



PROJET DE FIN D'ETUDES

S U J E T

CONCEPTION ET REALISATION
D'UNE UNITE DE TRAITEMENT
PIPE - LINE

Proposé par :

Mr H. BESSALAH

Etudié par :

A. AMROUN

M. AOUDJIT

Dirigé par :

Mr M. RAMDA

PROMOTION : JANVIER 85

Dédicaces

A mon père, A ma mère
A ma tante et mon Oncle
A mes frères et sœurs
A tous ceux que j'aime

Abdenour

A mon père, à ma mère.
A mes frères et sœurs.
Et à tous ceux qui me sont chers.

M. M. M.

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT ELECTRONIQUE

PROJET DE FIN D'ETUDES

S U J E T

CONCEPTION ET REALISATION
D'UNE UNITE DE TRAITEMENT
PIPE - LINE

Proposé par :

Mr H. BESSALAH

Etudié par :

A. AMROUN

M. AOUDJIT

Dirigé par :

Mr M. RAMDA

PROMOTION : JANVIER 85

INTRODUCTION

Actuellement de nombreux travaux et investissements sont orientés vers le traitement du signal.

On assiste alors à l'extension de l'utilisation des systèmes informatiques de plus en plus spécifiques pour la gestion et le contrôle des systèmes physiques.

L'augmentation de la vitesse de traitement dans les calculateurs numériques est devenue indispensable pour les calculateurs spécifiques (conduite de processus industriels ou traitements numériques du signal en temps réel en raison de la contrainte de la fréquence d'échantillonnage).

Plusieurs domaines font appel au traitement numérique du signal en temps réel tels : l'analyse spectrale, le traitement de la parole et de l'image etc.....

Au commissariat aux énergies nouvelles, les besoins en calcul connaissent un développement extrêmement rapide. Leur croissance nous conduit à la recherche de machines de plus en plus puissantes. L'accroissement des performances peut être recherché soit au niveau de la technologie, soit celui de l'organisation.

Ces dernières années, les progrès de la technologie

ont permis un accroissement très important des performances des machines, mais quelles que soient les technologies utilisées on se heurtera à une limitation.

Puisque les limitations technologiques ne nous permettent pas de gagner en vitesse de traitement, c'est donc de l'organisation interne des machines que l'on peut gagner du terrain dans le domaine des performances.

Dans le cadre du projet Processeur de FFT noyau du système rapide de traitement du signal en voie de réalisation au laboratoire Architecture des Systèmes du CDTA, l'unité de traitement du processeur de FFT nous a été confiée.

Cette unité de traitement présente les particularités suivantes :

- Calcul de l'opération ^{papillon} en un temps inférieur à 160 ns
- Calcul de M opérations papillons ($M = 2^n$).
- Système réalisé entièrement en logique câblée.
- Un nombre d'entrées, sorties limité afin de permettre sa réalisation sur un seul cristal en technologie des circuits prédiffusés ou en technologie VLSI.

Notre travail est axé essentiellement sur deux parties.

- La première partie est consacré aux rappels sur la FFT, à l'étude de l'architecture des systèmes et la technologie VLSI puis aux algorithmes de multiplications.
- La deuxième partie est consacré à la conception et la réalisation de l'U.T.

REMERCIEMENTS.

Nous remercions M^r BESSALAH.H responsable du laboratoire architecture des systèmes pour nous avoir proposé le sujet et pour son aide dans l'accomplissement de ce travail.

Nos vives remerciements à M^r RAMDA.M pour nous avoir suivi durant tout le projet et pour ses conseils qu'il nous a prodigués.

Nous tenons à remercier tout le personnel du laboratoire architecture des systèmes pour sa cordialité.

Que tous ceux qui ont contribué à notre formation trouvent ici l'expression de notre profonde gratitude.

SOMMAIRE.

INTRODUCTION.....	1.
Première partie.	
CHAPITRE.1. DESCRIPTION ET APPLICATIONS DE LA FFT.	
1.1. Notion de transformée de Fourier.....	3.
1.2. La transformation de Fourier discrète.....	5.
1.3. Principe de la FFT.....	7.
CHAPITRE.2. ARCHITECTURE DES MACHINES DE GRANDE PUISSANCE.	
2.1. Architecture pseudo-pipe-line.....	12.
2.2. Architecture pipe-line.....	12.
2.3. Présentation des systèmes de traitement de données par flot de données et flot d'instructions.....	14.
2.4. Conclusion.....	15.
CHAPITRE.3. ASPECTS ALGORITHMIQUES DE LA MULTIPLICATION.	
3.1. Première méthode.....	17.
3.2. Multiplication par passage au module.....	20.
3.3. Multiplication selon l'algorithme de Booth.....	21.
3.4. Méthode de Booth modifiée.....	23.
CHAPITRE.4. ETUDE DE LA TECHNOLOGIE VLSI.	
4.1. Introduction.....	27.
4.2. Paramètres d'un système intégré et conséquences d'une réduction d'échelle.....	29.

4.3. Choix d'une technologie	32
4.4. Données et contrôle dans les structures systématiques	34
4.5. Intégration des systèmes pipe-line en VLSI	36
4.6. La VLSI et les systèmes fortement concurrents	39
4.7. Algorithmes pour les réseaux de processeurs VLSI	44

Deuxième partie.

CHAPITRE 5. CONCEPTION ET REALISATION DE L'UNITE DE TRAITEMENT PIPELINE.

5.1. Introduction	49
5.2. Schéma synoptique et fonctionnement général du système	50
5.3. Multiplieur 16x8 bits	53
5.4. Circuit d'aiguillage des données et file d'attente (FIFO)	63
5.5. Circuit sélecteur et additionneurs soustracteurs	67
5.6. Circuit d'aiguillage des résultats	71
5.7. Circuit de commande	73
5.8. Horloge	77
5.9. Génération du signal "WRITE" de la mémoire extérieure	78
5.10. Extension de l'utilisation de l'unité de traitement à un multiplieur 16x8 rapide	79
CONCLUSION	80
BIBLIOGRAPHIE.	
ANNEXE.	

DESCRIPTION ET APPLICATIONS DE LA FFT (Fast Fourier Transform).

1.1. Notion de transformée de Fourier:

Actuellement, la transformée de Fourier est l'outil principal d'analyse dans plusieurs domaines scientifiques.

Son application a connu un essor considérable notamment dans l'analyse des systèmes linéaires.

On distingue quatre sortes de transformations [3]

- Transformation de Fourier des signaux aperiodiques à temps continu.

Transformation directe:

$$X(f) = \int_{-\infty}^{+\infty} x(t) \exp(-j2\pi ft) dt.$$

Transformation inverse:

$$x(t) = \int_{-\infty}^{+\infty} X(f) \cdot \exp(+j2\pi ft) dt.$$

- Transformation de Fourier des signaux périodiques à temps continu.

4.
Transformation directe:

$$X(n) = \left(\frac{1}{T_0} \right) \int_{T_0} x(t) \cdot \exp(-j2\pi n f_0 t) dt.$$

Transformation inverse.

$$x(t) = \sum_{n=-\infty}^{+\infty} X(n) \cdot \exp(+j2\pi n f_0 t).$$

- Transformation de Fourier des signaux aperiodes à temps discret.

Transformation directe:

$$X(f) = \sum_{k=-\infty}^{+\infty} x(k) \cdot \exp(-j2\pi f k).$$

Transformation inverse:

$$x(k) = \int_{-\frac{1}{2}}^{+\frac{1}{2}} X(f) \cdot \exp(+j2\pi f k) df.$$

- Transformation de Fourier des signaux periodiques à temps discret.

Transformation directe:

$$X(n) = \sum_k x(k) \cdot \exp(-j2\pi n k/N). \quad (1.1)$$

$$n = 0, 1, \dots, N-1.$$

Transformation inverse:

$$x(k) = \left(\frac{1}{N} \right) \sum_n X(n) \cdot \exp(+j2\pi n k/N).$$

$$k = 0, 1, \dots, N-1.$$

Cette dernière transformation constitue la base de la DFT.

1.2. la transformation de Fourier discrète (DFT)

La transformation de Fourier discrète d'un signal est une approximation de la transformée de Fourier exacte [2].

Elle est calculée à partir d'un nombre fini d'échantillons de ce signal.

Elle est définie par:

$$X(nT_1) = \sum_{k=0}^{N-1} x(kT_1) \cdot \exp(-j 2\pi n k / N).$$

$$n = 0, 1, 2, \dots, N-1.$$

$$N = \frac{T_0}{T_1} : \text{nombre d'itérations.}$$

$$T_1 : \text{période d'échantillonnage.}$$

$$T_0 : \text{période de répétition.}$$

Généralement T_1 est pris égal à l'unité.

d'où:

$$X(n) = \sum_{k=0}^{N-1} x(k) \cdot \exp(-j 2\pi n k / N) \quad (1.2)$$

sous forme complexe l'expression (1.2) nous donne:

$$\operatorname{Re} X(n) = \sum_{k=0}^{N-1} x(k) \cdot \cos(2\pi n k / N).$$

$$\operatorname{Im} X(n) = \sum_{k=0}^{N-1} x(k) \cdot \sin(2\pi n k / N).$$

Les coefficients de la DFT correspondent aux affixes des nombres complexes $W = \exp(-j2\pi nk/N)$.

La relation (1.2) s'écrit alors :

$$X(n) = \sum_{k=0}^{N-1} x(k) \cdot W(nk/N).$$

soit sous-forme matricielle.

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(n) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & \dots & W^0 \\ W^0 & W^1 & \dots & W^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ W^0 & W^{N-1} & \dots & W^{(N-1)^2} \end{bmatrix} \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix}$$

avec : $W^i = \exp(-j2\pi i/N)$.

et $W^N = 1$.

$[W]$ est une matrice carrée $N \cdot N$ dont les lignes et les colonnes de même indice ont les mêmes éléments.

La DFT a trouvé relativement peu d'applications à cause du problème de rapidité de calcul.

En effet le calcul de N échantillons d'une période spectrale exige un nombre d'opérations de l'ordre de N^2 .

Toutefois une organisation méthodique de calcul de la DFT est possible, ramenant le nombre des

opérations à effectuer à une valeur de l'ordre de $N \log_2 N$.
L'ensemble des algorithmes permettant d'accélérer le calcul de la DFT, constitue la FFT.

1.3. Principe de la FFT.

1.3.1. Présentation des différents algorithmes de la FFT.

IL existe essentiellement quatre classes [9] d'algorithmes permettant un calcul rapide de la DFT.

- Algorithmes de la classe A:

Le désembrouillage se fait au niveau des entrées. Les séquences de sortie sont obtenues dans l'ordre.

- Algorithmes de la classe B:

On introduit les données normalement mais un réarrangement est nécessaire en fin d'exécution.

- Algorithmes de la classe C:

Le désembrouillage se fait au niveau de la matrice W et les séquences de données et de sorties sont dans l'ordre.

- Algorithmes de la classe D:

Les entrées sont ordonnées, les coefficients de rotation et les sorties sont désembrouillés.

N.B: Le désembroillage consiste à prendre le linéaire refléchi de chaque indice.

1.3.2 Comparaison et Conséquences:

Les algorithmes A, B, C et D s'effectuent en $\log_2 N$ itérations. Chaque itération comprend $\frac{N}{2}$ opérations de base.

Les opérations de base constituant l'algorithme A et D sont symétriques ce qui facilite leur réalisation en hardware. Par contre les algorithmes B et C sont plutôt adaptés à des réalisations programmées.

Pour le calcul de la FFT, nous avons opté pour l'algorithme A qui offre les avantages suivants:

- la symétrie de l'opération papillon.
- l'erreur de troncature est minimale.

La figure 1.1 représente la forme graphique de l'algorithme A et ce dans le but de refléter le processus de calcul. Ce diagramme est constitué essentiellement par des motifs appelés itérations reliant le signal initial à ses états intermédiaires, les coefficients de Fourier sont obtenus à la dernière itération.

Une géométrie identique de toutes les étapes de traitement composées d'unités algorithmiques, semblable est appelée "Opération papillon".

* Les conventions du "papillon". (fig 1.2). [9]

- Le cercle vide (o) définit une opération d'addition - soustraction dans laquelle la somme apparaît en haut et la différence en bas.

- La flèche (\rightarrow) décrit une multiplication avec une valeur située au-dessous de la flèche.

a et b deux nœuds sources.

c = a + bw et d = a - bw deux nœuds puits.

Dans le cas de la figure 1.1, le premier papillon de la première itération a pour nœuds sources les points $x(0)$ et $x(8)$ et pour nœuds puits $x_1(0)$ et $x_1(8)$ reliés aux nœuds sources par la relation:

$$\left. \begin{aligned} X_1(0) &= x(0) + W^0 x(8) \\ X_1(8) &= x(0) - W^0 x(8) \end{aligned} \right\} (1.3)$$

Pour le troisième papillon par exemple.

$$\left. \begin{aligned} X_1(2) &= x(2) + W^0 x(10) \\ X_1(10) &= x(2) - W^0 x(10) \end{aligned} \right\} (1.4)$$

Le premier papillon de la deuxième itération a pour nœuds sources les points $X_1(0)$ et $X_1(4)$ et pour nœuds puits les points $X_2(0)$ et $X_2(4)$ reliés aux nœuds sources par la relation :

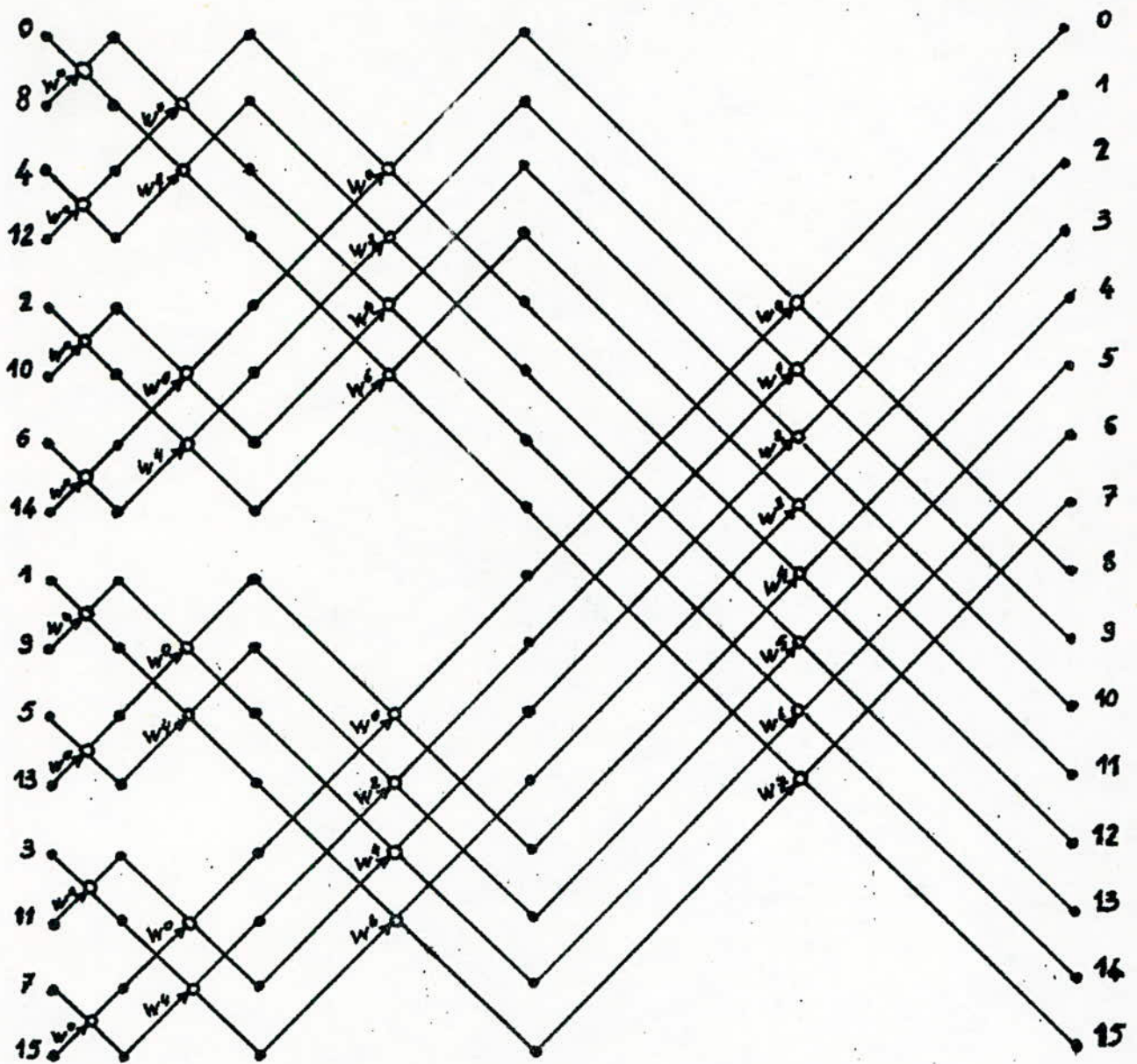


Fig1.1 . Diagramme de FFT sur 16 points
Entrée désembrillée - Sortie ordonnée.

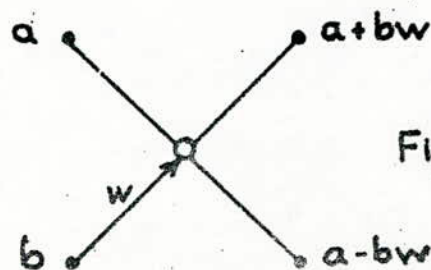


Fig1.2 . Opération papillon

$$\left. \begin{aligned} X_2(0) &= X_1(0) + w^{10} X_1(4) \\ X_2(4) &= X_1(0) - w^{10} X_1(4) \end{aligned} \right\} (1.5)$$

Dans le cas de N points avec $0 \leq k \leq \frac{N}{2} - 1$ et à partir des expressions (1.3), (1.4) et (1.5) on peut déterminer les expressions liant chaque état intermédiaire ($j+1$) avec l'état j .

$$\left. \begin{aligned} X_{j+1}(k) &= X_j(k) + w^P X_j(k + \frac{N}{2}) \\ X_{j+1}(k + \frac{N}{2}) &= X_j(k) - w^P X_j(k + \frac{N}{2}) \end{aligned} \right\} (1.6)$$

En posant $k = K0$ et $k + \frac{N}{2} = K1$.

Les expressions (1.6) deviennent alors:

$$\left. \begin{aligned} X_{j+1}(K0) &= X_j(K0) + w^P X_j(K1) \\ X_{j+1}(K1) &= X_j(K0) - w^P X_j(K1) \end{aligned} \right\} (1.7)$$

avec X_j : Complexes.

Sachant que:

$$w^P = \exp(-j \frac{2\pi}{N} P) = \cos(\frac{2\pi}{N} P) - j \sin(\frac{2\pi}{N} P)$$

En posant $\theta = \frac{2\pi}{N} P$.

On aura pour chaque papillon les expressions suivantes:

$$RE_{j+1}(K0) = RE_j(K0) + [RE_j(K1) \cos \theta + IM_j(K1) \sin \theta]$$

$$RE_{j+1}(K1) = RE_j(K0) - [RE_j(K1) \cos \theta + IM_j(K1) \sin \theta]$$

$$IM_{j+1}(K0) = IM_j(K0) + [IM_j(K1) \cos \theta - RE_j(K1) \sin \theta]$$

(1.8)

$$IM_{j+1}(K1) = IM_j(K0) - [IM_j(K1) \cos \theta - RE_j(K1) \sin \theta]$$

$$\text{avec } K0 = K_{m-1} K_{m-2} \dots 0 \dots K0$$

$$K1 = K_{m-1} K_{m-2} \dots 1 \dots K0$$

1.3.3. Conclusion:

Plusieurs domaines scientifiques, tel le traitement du signal, font appel à la transformation de Fourier rapide, ceci explique l'attention particulière que portent les chercheurs en vue d'un calcul rapide de celle-ci.

Nous sommes alors amenés à concevoir un système performant du point de vue rapidité, consommation et fiabilité.

Plusieurs architectures satisfaisant aux exigences citées ou du moins un compromis entre celles-ci, sont alors possibles.

L'étude qu'on fera dans le prochain chapitre nous permettra d'opter pour une solution optimale.

ARCHITECTURE DES MACHINES DE GRANDE PUISSANCE

La simultanéité de traitement et donc l'utilisation optimale des unités de traitement, semble être la voie suivie ces dernières années dans la perspective d'une amélioration conséquente des performances des machines de grande puissance. Cette orientation est venue suite à la limitation technologique du point de vue temps d'exécution des opérations élémentaires.

Deux solutions principales ont été alors introduites.

2.1. Architecture pseudo-pipe-line.

Cette solution multiplie le hardware mais divise le temps d'exécution. [7]

Cette méthode est illustrée par un schéma fonctionnel (fig 2.1.a) et son diagramme temporel (fig 2.1.b).

2.2. Architecture pipe-line.

Celle-ci repose sur le concept d'anticipation qui permet de disposer des instructions et des opérandes à l'avance; on peut gagner ainsi du temps en lançant une nouvelle opération avant que l'opération précédente ne soit terminée; autrement dit les couples de composantes de

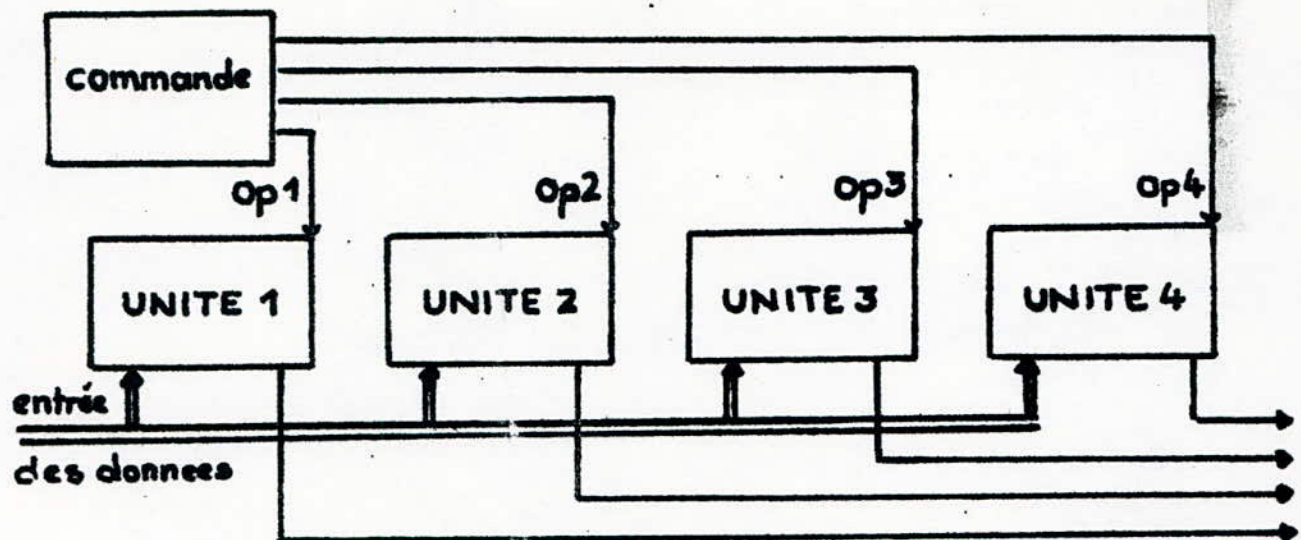


Fig 21-a

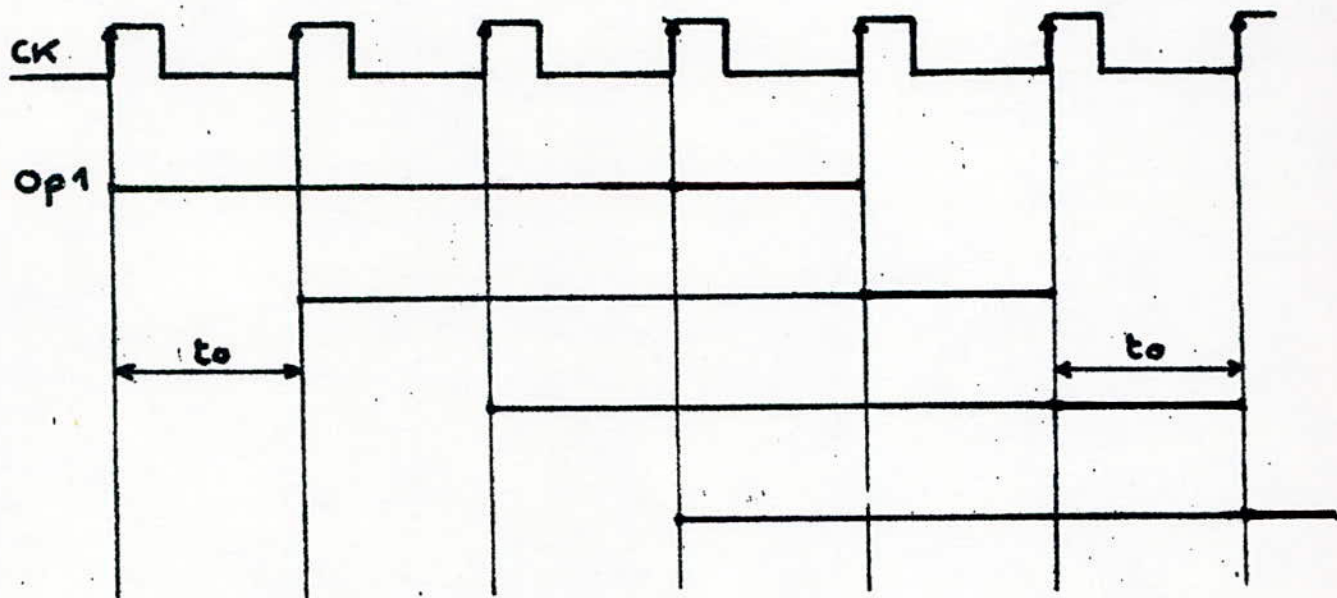


Fig 21-b

13.
même rang de deux vecteurs opérands traversent le pipe-line en libérant au fur et à mesure les sections successives pour le couple de composantes du rang suivant.

Ainsi dans cette structure, l'introduction d'une nouvelle donnée (ou l'exécution d'une opération) peut être effectuée toutes les t_0 (ms) alors que chaque opération dure en fait nt_0 .

n : nombre de sections ou niveaux (fig 2.2).

Le premier résultat est obtenu après le temps $n \cdot t_0$ qui représente l'inertie du système, alors que les suivants sont donnés régulièrement à des intervalles de temps t_0 .

Le temps total de traitement de N opérations sera

$$t = nt_0 + (N-1)t_0$$

Ainsi en diminuant t_0 , cette méthode permet à de très grandes vitesses, le traitement d'un nombre de données élevé; ce qui est le cas dans les systèmes de traitement numérique des signaux uni et bidimensionnel.

L'organisation d'un système en pipe-line exige un découplage en plusieurs sections - correspondant chacune à une étape de l'opération à exécuter; les sections sont séparées par des jeux de registres - capables de mémoriser pour la section suivante, les résultats obtenus dans la section précédente (fig 2.3). Le temps de traversée d'une section est au maximum égal à l'intervalle entre deux tops d'horloge.

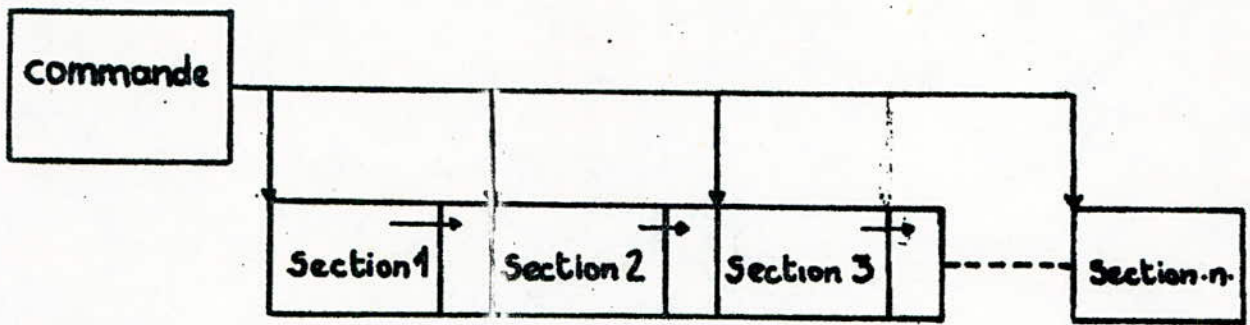


Fig 2.2. Schéma Synoptique d'une Structure Pipe-Line .

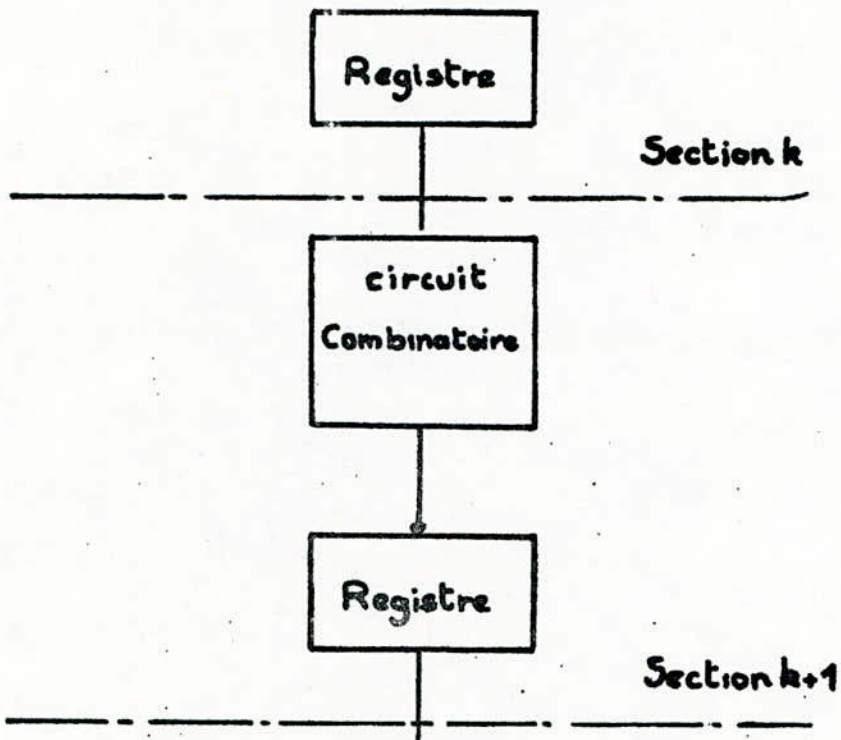


Fig 2.3. Schéma d'une Section du Pipe-Line .

- 14.
- Au top H_1 : l'opération 1 débute dans la section 1.
 - Au top H_2 : l'opération 1 se continue dans la section 2, tandis que la section 1 est libre pour prendre en charge l'opération 2 et ainsi de suite.

C'est ainsi qu'au top H_4 , les opérations 1, 2, 3 et 4 seront en cours de traitement respectivement dans les sections 4, 3, 2 et 1.

2.3. Présentation des systèmes de traitement de données par flots de données et flot d'instructions:

La classification de FLYNN de tous ces systèmes est basée sur l'aspect simple ou multiple des flots d'instructions et de données. [9]

- Systèmes MIMO: (multiple instruction flow, multiple data flow).

Ces systèmes sont difficiles à mettre en œuvre. Leur difficulté de gestion est due à la simultanéité d'exécution de différents flots d'instructions sur différents flots de données.

- Systèmes SISD: (single instruction, single data flow)

Ils ont à simple flot d'instructions et simple flot de données. Ces systèmes sont linéaires donc très lent.

- Systèmes MISD (Multiple instructions flow, single data flow)

C'est une structure où plusieurs instructions s'exécutent

simultanément sur les mêmes données.

- Systèmes SIMD: (single instruction flow, multiple data flow).

Ces systèmes sont à simple flot d'instructions et multiple flots de données.

Ils nous intéressent alors particulièrement, vu que dans les algorithmes de la FFT nous avons un grand nombre d'opérations à traiter et un très faible nombre d'entrées. Dans ce cas plusieurs algorithmes sont utilisés, entre autre celui qui permet le fonctionnement de plusieurs unités de traitement en parallèle.

De ces systèmes sont nées les structure pipe-line et pseudo-pipe-line (étudiées dans le paragraphe 2.1 et 2.2)

2.4. Conclusion:

L'architecture pipe-line est particulièrement intéressante pour des applications à grand nombre de traitements, ce qui est le cas de notre projet où le nombre d'opérations est égal à:

$$k \cdot \frac{N}{2} \log_2 N$$

avec $N = 512$: nombre d'échantillons

k : nombre de signaux à traiter.

Ainsi malgré l'extension du hardware, l'inertie du système reste négligeable devant le temps total de traitement.

Notons enfin que l'amélioration des systèmes pipe-line passe par un compromis entre d'une part l'accroissement de l'aspect pipe-line (ce qui suppose une augmentation de la complexité des algorithmes de contrôle); d'autre part la réduction à la fois du cycle mineur et du prix de revient ce qui à technologie égale s'oppose à cette augmentation de complexité.

L'apparition récente des systèmes VLSI est venue à temps, apportant de nombreux avantages dont l'intégration facile et de ce fait moins coûteuse des systèmes pipe-line.

N.B.: cycle mineur (ou cycle machine): dans les machines pipe-lines le cycle mineur correspond à:

$$t_0 = \left(\frac{1}{\text{fréquence d'horloge}} \right)$$

ASPECTS ALGORITMIQUES DE LA
MULTIPLICATION

Il existe plusieurs méthodes de multiplication de nombres binaires. Dans ce chapitre nous étudierons les principales.

3.1. Première méthode:

soit à multiplier deux nombres X et Y

X et Y peuvent s'écrire sous la forme:

$$\left. \begin{aligned} X &= z_0 2^0 + \sum_{i=1}^n x_i 2^{-i} = z_0 + x \\ Y &= y_0 2^0 + \sum_{i=1}^n y_i 2^{-i} = y_0 + y \end{aligned} \right\} (3.1.)$$

z_0 et y_0 sont respectivement les signes de X et Y .

x et y sont respectivement les mantisses de X et Y

Lors de la multiplication de ces deux nombres, le bit de signe et les bits de la mantisse sont traités séparément.

Le produit sera donc:

$$Z = z_0 2^0 + xy \quad (3.2)$$

avec $z_0 = z_0 \oplus y_0$ est le signe de Z

En utilisant les expressions (3.1) on obtient:

$$Z = (x + z_0)(y + y_0) = xy + xy_0 + z_0 y + z_0 y_0 \quad (3.3)$$

Pour obtenir l'expression (3.2) nous devons faire un circuit de correction qui permet de soustraire $(x_0y + x_1y_0 + x_0y_1)$ de l'expression 3.3.

Cette correction depend des signes x_0 et y_0 respectivement de X et Y (Tableau 1.).

Tableau 1.

x_0	y_0	XY	Correction
0	0	xy	.
0	1	$xy + x_1y_0$	$-x_1y_0$
1	0	$xy + x_0y_1$	$-x_0y_1$
1	1	$xy + x_0y_1 + y_0x_1 + x_0y_0$	$-(x_0y_1 + y_0x_1 + x_0y_0)$

Calcul de $x.y$.

$$xy = y [x_1 2^{-1} + x_2 2^{-2} + \dots + x_{n-1} 2^{-(n-1)} + x_n 2^{-n}]$$

En mettant en facteur 2^{-1} nous obtenons:

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

on pose: $P_1 = (P_0 + y x_n) 2^{-1}$

$$P_2 = (P_1 + y x_{n-1}) 2^{-1}$$

$$P_3 = (P_3 + y x_{n-2}) 2^{-1}$$

⋮

D'où la formule de recurrence suivante:

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

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

On déduit:

$$xy = P_n = (P_{n-1} + yx_1)2^{-1}$$

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

Il existe d'autres méthodes de calcul du produit xy telles que:

- La multiplication par le bit de plus faible poids du multiplicateur avec décalage à gauche du multiplicande, la formule de récurrence est:

$$xy = P_n = P_{n-1} + y_{n-1}x_1$$

avec $P_0 = 0$ pour $i=0$

$$P_{i+1} = P_i + y_i x_{n-i} \quad i = 0, 1, \dots, n$$

- Multiplication par les bits de plus fort poids du multiplicateur et décalage à droite du multiplicande,

la formule de récurrence est:

$$xy = P_n = P_{n-1} + y_n x_n$$

$$\text{avec: } P_i = P_{i-1} + x_i y_i$$

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

$$y_i = y 2^{-i} = y_{i-1} 2^{-1}$$

Conclusion:

Cette méthode est facilement programmable sur ordinateur (cependant le temps de traitement est considérable),

mais difficilement réalisable en hardware vu la complexité des circuits de correction. De plus le traitement séparé des bits de signe et des bits de la mantisse ne convient pas pour des nombres donnés dans le code du complément à deux.

3.2. Multiplication par passage au module:

Dans cette méthode, la multiplication dépend du signe du multiplicateur. Deux cas se présentent: [1]

- Si le multiplicateur est positif, la multiplication s'effectue normalement, le bit signe étant traité comme un bit ordinaire. On propage le signe de chaque produit partiel jusqu'au bit de plus fort poids du dernier produit partiel.

- Si le multiplicateur est négatif, on complémente à deux celui-ci ainsi que le multiplicande. La multiplication s'effectue normalement avec propagation du signe de chaque produit partiel jusqu'au bit de plus fort poids du dernier produit partiel.

Dans les deux cas le résultat est donné dans le code du complément à deux.

Conclusion:

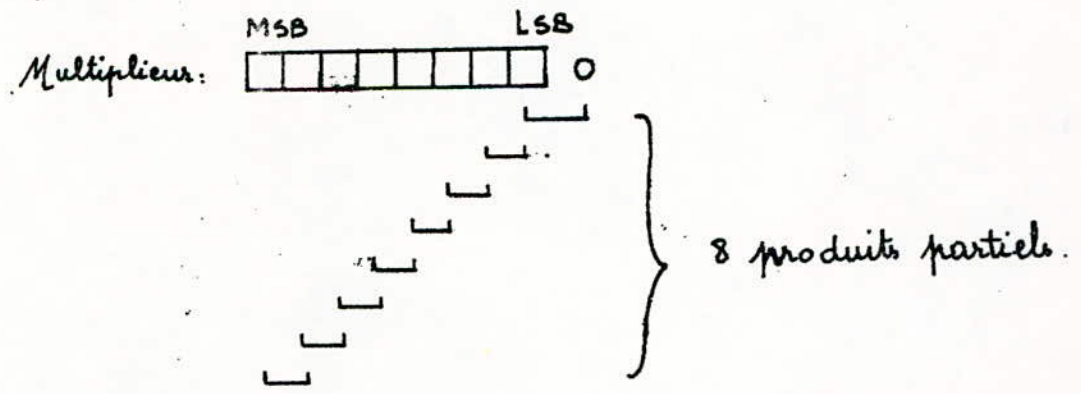
Cette méthode est relativement facile cependant

elle nécessite à l'entrée du système un circuit de complément à deux.

3.3. Multiplication selon l'algorithme de Booth.[1]

Cet algorithme est conçu pour multiplier directement des nombres signés exprimés dans le code du complément à deux.

Dans ce cas le multiplicateur est examiné par groupe de deux bits, chaque groupe ayant un bit en commun avec le précédent. Avant le premier cycle, il est nécessaire d'ajouter un zéro (bit imaginaire) à droite du multiplicateur.



L'exploitation du multiplicateur par groupe de deux bits nous conduit au tableau 2

- Il convient de noter les deux règles suivantes:
- Tout décalage est effectué en conservant le bit de plus fort poids du produit partiel qui est le signe.

X_i	X_{i-1}	Opérations à effectuer.
0	0	Décalage du multiplicande d'un rang vers la droite
0	1	Addition du multiplicande puis décalage d'un rang vers la droite.
1	0	Soustraction du multiplicande puis décalage d'un rang vers la droite.
1	1	Décalage du multiplicande d'un rang vers la droite.

Tableau 2.

23.
si le signe vaut zéro, on rentrera zéro à gauche du produit partiel.

si le signe vaut un, on rentrera un à gauche du produit partiel.

- On ne tient pas compte d'un report ou d'une retenue au delà du bit signe.

Le résultat de l'opération est exprimé dans le code du complément à deux.

Conclusion:

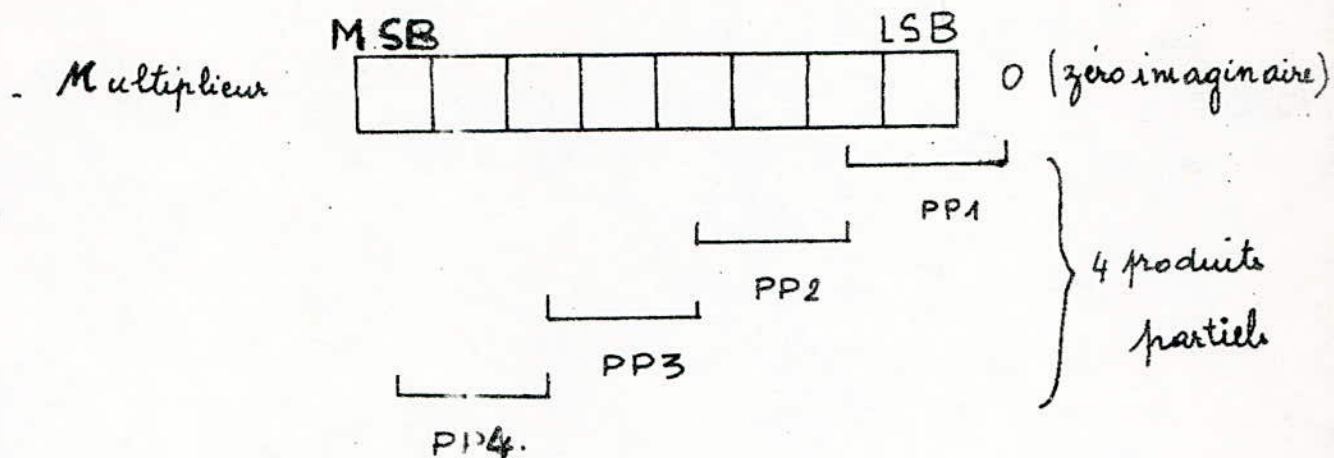
Le fait d'examiner le multiplicateur par groupe de deux bits en ajoutant un zéro imaginaire, ne change pas le nombre de produits partiels, mais permet d'effectuer la multiplication en considérant le bit signe comme un bit ordinaire. De plus elle convient aux nombres donnés dans le code du complément à deux et le résultat est en complément à deux.

L'amélioration de cette méthode consiste à traiter le multiplicateur par groupe de trois bits: C'est la méthode de Booth modifiée (ou méthode combinée).

3.4. Méthode de Booth modifiée.

L'algorithme de cette méthode découle de celui de Booth, toujours dans le but de traiter directement des nombres signés. [8]

Seulement dans ce cas le multiplicateur est traité par groupe de trois bits.



L'algorithme est donné par le tableau 3.

Dans ce cas le décalage s'effectue à gauche et de deux positions. Le bit de plus fort poids de chaque produit partiel est propagé jusqu'au bit de plus fort poids du dernier produit partiel.

Cette méthode s'avère plus rapide que celle utilisée par Booth car le nombre de produits partiels est réduit dans un rapport de deux.

Conclusion:

Pour la réalisation du multiplicateur 16×8 bits, nous avons opté pour cette méthode qui nous réalise le maximum d'avantages à savoir:

- L'adaptation à une structure pipe-line.
- Une réalisation pratique grâce à des circuits

Multiplieurs 4x2 bits.

- Ses données ainsi que les résultats sont exprimés dans le code du complément à deux.
- Plusieurs circuits multiplieurs ont été conçus à base de cet algorithme tels que [5]
 - 67058
 - AM 25558
 - MPV 8HJ
 - MPV 16HJ.

Multipliateurs pas 1			Symbole de l'opération	Pour obtenir le produit partiel
2^{i-1} 2	2^{i-2} 2	2^{i-3} 2		
0	0	0	0	Remplacer le multiplicande par zéro.
0	0	1	+1B	Copier le multiplicande.
0	1	0	+1B	Copier le multiplicande
0	1	1	+2B	décaler le multiplicande d'un bit à gauche.
1	0	0	-2B	Décaler le complément à deux du multiplicande d'un bit à gauche
1	0	1	-1B	Remplacer le multiplicande par son complément à deux
1	1	1	0	Remplacer le multiplicande par zéro.

Tableau 3.

CHAPITRE 4 .

ETUDE DE LA TECHNOLOGIE VLSI

4.1. Introduction:

L'idée d'insérer dans un réseau monocristallin, dont le volume ne dépasse pas quelques centièmes de millimètres cubes, plusieurs milliers de composants, alimentés et interconnectés entre eux; a vu le jour après les inventions du transistor à pointes en 1947, du transistor à jonctions deux ans plus tard et surtout celle des circuits intégrés monolithiques en 1959.

Les perfectionnements apportés depuis, tant dans le domaine de la recherche technologique que dans celui de la conception des circuits ont donné naissance aux circuits logiques SSI, MSI puis LSI et récemment les circuits VLSI qui feront l'objet de ce chapitre. [4]

L'intégration à grande échelle (LSI) est devenue si dense qu'une seule puce de silicium LSI peut contenir plusieurs dizaines de milliers de transistors. Beaucoup de circuits LSI, comme les micro-processeurs se décomposent à l'heure actuelle en multiples sous-systèmes complexes. Ces puces doivent être considérées plus comme

des systèmes que comme des circuits intégrés.

La densité d'intégration double presque tous les ans. Les principes de la physique indiquent que les transistors peuvent être réduits en taille, à moins d'un centième de leur surface actuelle, tout en conservant leur propriété d'éléments de commutation (cette propriété est très utilisée dans les systèmes digitaux).

À la fin des années quatre-vingt il sera possible de fabriquer des circuits de plusieurs millions de transistors. Pour de tels systèmes à très haute échelle d'intégration (VLSI), les composants et leurs interconnexions auront des dimensions de surface plus petites que la longueur d'onde de la lumière. Ces nouvelles techniques de lithographie à haute résolution ont été déjà éprouvées et rendent la fabrication de tels circuits possible.

Il est clair que ces systèmes offrent une meilleure fiabilité en réduisant d'une part les interconnexions et d'autre part la consommation.

Ainsi la structure des systèmes informatiques, leurs méthodes de conception, la relation entre le hardware et le software, la conception des algorithmes seront toutes grandement affectées par les développements considérables

qu'apportent l'intégration à très haute échelle. [6]

4.2. Paramètres d'un système intégré et Conséquences d'une réduction d'échelle:

Soit à faire une réduction d'échelle de facteur -constant α de toutes les dimensions du système, y compris les dimensions perpendiculaires à la surface. Ainsi les tensions sont aussi réduites dans le rapport α . Dans ces conditions tous les champs électriques du circuit restent inchangés de sorte que de nombreux facteurs non linéaires dont dépendent les performances, ne varieront pas alors qu'ils le feraient pour des changements d'échelle plus complexes (le cas où les dimensions du système intégré sont réduites dans un facteur différent).

L'expérience a montré et ce en accord avec la théorie que lorsque le facteur de réduction α augmente, le temps de commutation, la capacité et le courant collecteur (ou courant drain-source pour un MOS) diminuent proportionnellement.

4.2.1. Paramètres d'un système intégré

- Puissance commutée:

Pour un MOS, la puissance commutée est

l'énergie stockée dans la capacité d'un élément, divisée par la période d'horloge (intervalle de temps entre une charge et une décharge successive de la capacité).

La période de l'horloge du système est proportionnelle au temps de transfert de ses plus petits composants.

- Facteur de mérite d'un circuit (f):

f est une mesure importante des performances de ce circuit. Il est défini par le produit de la puissance dissipée, de la fréquence maximale d'horloge et du temps de commutation.

Après une réduction d'échelle de d , f devient f' tel que:

$$f' = \frac{f}{d^3}$$

4.2.2 Conséquences d'une réduction d'échelle:

On remarque que le facteur de réduction d est limité, car si d est très grand la tension de seuil diminue et devient alors proche de la tension de blocage, ce qui serait nuisible au fonctionnement du système.

Exemple:

supposons que nous réduisons complètement le

circuit intégré avec un facteur $d = 10$. Le circuit comporterait alors cent fois plus de composants par unité de surface ($s = L \cdot l \Rightarrow S_d = \frac{L}{d} \cdot \frac{l}{d} = \frac{s}{d^2}$)

Par contre la puissance dissipée par unité de surface resterait constante. Ceci s'explique par le fait que les tensions seraient dix fois plus faibles alors que le courant à fournir par unité de surface serait dix fois plus élevé.

Le temps de commutation serait réduit dix fois (ce qui est considérable)

Le facteur de mérite serait alors divisé par mille.

$$f_d = \frac{f}{d^3} = \frac{f}{10^3}$$

Conclusion:

Cette réduction d'échelle est attrayante, sauf pour les densités de courant.

Le courant doit être distribué à tous les composants du circuit à l'aide de conducteurs métalliques, afin que les chutes de tensions internes ne soient pas excessives.

Il en résulte qu'il faudra bien trouver un moyen pour réduire les besoins en courant du système, à un courant constant par unité de surface.

Les recherches actuelles sont orientées dans deux voies d'une part les technologies à faible consommation (telle la CMOS) et d'autre part les technologies utilisant la logique à interrupteurs.

La limite de la réduction d'échelle est atteinte lorsque les composants sont incapables de commuter.

En outre un certain nombre de phénomènes physiques tel l'effet tunnel ou la migration des impuretés dans les couches appauvries s'opposent au bon fonctionnement des composants de plus petite taille.

4.3. Choix d'une technologie:

Avec l'avance technologique, de plus en plus de fonctions systèmes peuvent être réalisées sur un circuit de même taille.

Le but final est la réalisation de systèmes de grande taille sur une simple pastille de silicium.

Pour atteindre cet objectif, tout signal utilisé dans le système à l'exception des entrées, sorties, alimentations doit être généré dans la technologie du circuit.

Les paramètres qui doivent être considérés dans le choix d'une technologie concernent:

- La densité d'intégration.
- La richesse des fonctions que l'on peut obtenir.
- La performance relative à la consommation.
- La vitesse.
- Les propriétés topologiques des interconnexions.
- Son adéquation à l'intégration d'un système complet.
- La disponibilité des chaînes technologiques.
- Le prix des divers composants.

Toute technologie à intégration à grande échelle passe par l'utilisation de deux types de transistors (pour la technologie bipolaire, PNP et NPN) Ceci afin qu'au moins l'un d'eux se trouve bloqué lorsque l'entrée est portée au potentiel bas.

Actuellement trois technologies dominent dans les systèmes VLSI, ce sont la IL , la CMOS et la TTL ALS.

Les circuits bipolaires sont très rapides car leur temps de transit est déterminé par la finesse de l'épaisseur de la base.

En outre dans cette technologie, le courant circule perpendiculairement à la surface, ainsi le courant et la capacité décroissent avec le même

facteur que les dimensions de la surface du dispositif.

4.4. Données et Contrôle dans les structures Systématiques:

La méthodologie utilisée pour la conception d'un ensemble intégré à grande échelle (VLSI), doit utiliser au maximum les possibilités architecturales offertes par la technologie employée.

Notre objectif est alors de chercher la manière d'assembler les structures systématisées, en ensembles plus grands pour former des sous-systèmes qui, réunis à leur tour donneront le système global.

La forme particulière des circuits présentés, (en VLSI) tend à produire des systèmes avec une topologie d'interconnexions très simple et régulière.

Elle tend ainsi à minimiser les surfaces nécessaires pour implanter les fonctions.

Des réseaux de transistors interrupteurs, pour la logique combinatoire dans des chemins de transfert registres à registre (cas des systèmes pipe-line) sont utilisés à chaque fois que possible pour implanter les fonctions du système.

Cette approche tend à minimiser aussi bien la puissance

dissipées par unité de surface, que les systèmes de propagation par fonction.

La conception de systèmes informatiques utilisant des circuits MSI et LSI disponibles sur le marché utilise implicitement une approche structurée.

On sait qu'on peut réaliser n'importe quel circuit logique à l'aide de portes élémentaires (NAND, NOR etc...). A partir de là, la conception des systèmes intégrés peut utiliser la méthode traditionnelle de conception logique qui est utilisée avec les composants discrets. Actuellement les systèmes intégrés sont souvent réalisés de cette manière.

Il est cependant peu probable que de telles approches anarchiques puissent survivre alors que la technologie tend vers un degré d'intégration maximum avec la VLSI.

Pour tenir compte de la complexité de réalisation des systèmes intégrés, on décompose souvent le problème en différents niveaux.

- Niveau architectural.
- Conception logique.
- Conception du circuit.
- Dessin des masques.

Dans les systèmes VLSI, la surface utilisée sur la puce de silicium par la circuiterie, est beaucoup plus une fonction des propriétés topologiques des interconnexions du circuit, que du nombre de portes logiques élémentaires. L'implantation d'une fonction avec un nombre minimal de portes utilise souvent beaucoup plus de surface de silicium, qu'une autre réalisation utilisant plus de portes, mais avec une topologie d'interconnexions plus simple.

4.5. Intégration des systèmes pipe-line en VLSI:

4.5.1. Horloge:

Un type particulier d'horloge est souvent utilisé: ce sont les horloges biphasées (fig 4.1).

En général le type d'horloge, correspond à la manière de définir les temps pendant lesquels le transfert d'informations est autorisé dans un étage, et à travers les étages consécutifs d'un système, et les intervalles de temps pendant lesquels les étages sont isolés les uns des autres.

On notera que dans un système intégré, le type d'horloge utilisé dépend fortement de la manière d'assembler les circuits et les sous-ensembles. Il y a des conséquences importantes sur l'architecture

qui en résulte.

4.5.2 Registres dynamiques:

Les registres de mémorisation de données jouent un rôle fondamental dans la conception des systèmes digitaux. La juxtaposition de deux transistors inverseurs et d'un transistor interrupteur forme un registre temporaire ou dynamique (fig 4.2).

4.5.3. Transfert registre à registre

Les entrées sont mémorisées dans le registre d'entrée pendant la première phase ϕ_1 de l'horloge, ensuite elles se propagent à travers les circuits logiques combinatoires, et les sorties sont mémorisées dans le registre de sortie pendant ϕ_2 (fig 4.3).

4.5.4. Réalisation de la logique combinatoire:

Elle est réalisée par des opérations sur les charges se déplaçant entre les étages; les transferts étant contrôlés par des transistors interrupteurs.

Actuellement beaucoup de travaux sont orientés vers l'examen des structures et des techniques différentes pour réaliser les opérations logiques élémentaires, avec par exemple des éléments à transfert

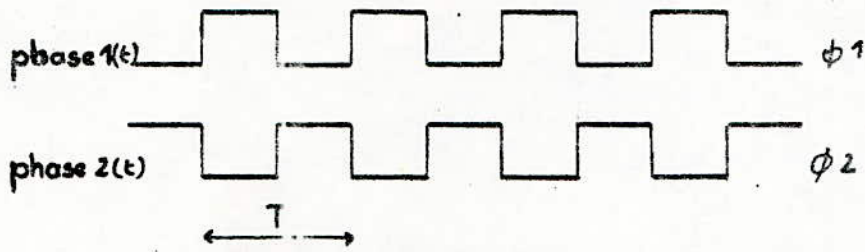


Fig 4.1 Signaux d'horloges à deux phases sans recouvrement.

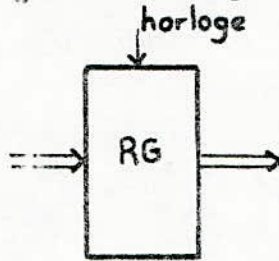


Fig 4.2 Registre dynamique

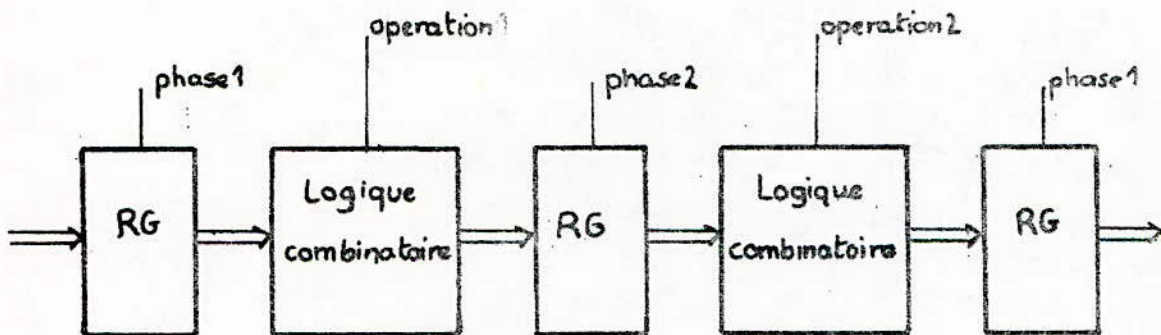


Fig 4.3 Forme générale d'un chemin de données.

de charge (c.c.D). Cependant la synthèse des fonctions complexes est difficilement réalisable avec une structure régulière.

Généralement l'utilisation de "Réseaux logiques programmables" apportent des solutions satisfaisantes à ce problème.

Réseaux logiques programmables (PLA).

Cette technique offre un grand avantage: Les fonctions peuvent subir des modifications significatives sans nécessiter un changement majeur de la conception.

Le réseau logique programmable est une structure qui a toutes les propriétés d'une mémoire pour implanter les fonctions logiques combinatoires.

Toute fonction peut se ramener à la somme de mintermes, une matrice de portes AND et une matrice de portes OR suffisent pour réaliser toute fonction booléenne (fig 4.4).

4.6 La VLSI et les systèmes Fortement Concurrents

Jusqu'ici, nous avons décrit principalement la façon d'arranger le circuits et les interconnexions

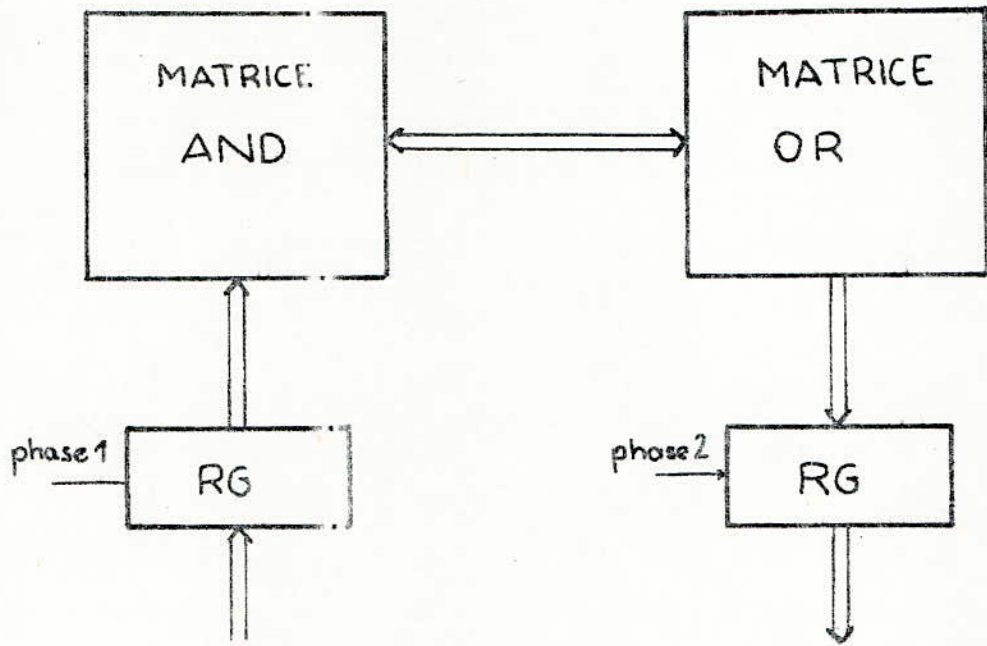


Fig 4.4 Structure d'un PLA

sur la pastille de silicium.

Dans cette partie nous étudierons les applications de la technique VLSI à la solution des problèmes informatiques intéressants.

Nous verrons que la VLSI apporte davantage qu'un moyen économique d'intégrer des processeurs. En effet puisque il est aisé d'intégrer simultanément dans une VLSI des éléments de calcul et des éléments de mémoire.

Il serait intéressant d'utiliser des structures présentant un haut degré de concurrence autrement dit la simultanéité.

4.6.1. Traitement de l'information et Simultanéité dans les calculateurs.

L'architecture des calculateurs conventionnels souffre de deux défauts que l'on cherche à éviter dans la conception de structure de calcul VLSI.

- Le processeur est séparé de sa mémoire par une longue voie de communication (par exemple un bus).

- Les informations sont traitées séquentiellement, ce qui rend ces processeurs très lents.

Les technologies VLSI offrent une grande souplesse que les technologies précédentes (MSI, LSI, SSI) car les mémoires et les structures de traitement de l'information, peuvent être réalisées avec la même technologie et disposées dans l'immédiate proximité les unes des autres.

4.6.2 Coût des communications dans un Calculateur:

Dans un ordinateur, le processeur pointe une instruction dans la mémoire, la décode, l'exécute et reprend le cycle au début.

De nombreuses instructions entraînent des références supplémentaires à la mémoire, pour pointer des opérandes ou stocker des résultats.

Les performances d'un tel -calculateur dépendent d'une façon critique de la vitesse avec laquelle on peut accéder à la mémoire.

- Vitesse d'une mémoire :

si une mémoire de M bits est réalisée sur une pastille, le temps nécessaire pour transmettre l'information d'une cellule mémoire au processeur est proportionnel à la longueur du fil les reliant.

Ainsi plus la distance sur laquelle doit se propager le signal, est grande, plus la capacité parasite du fil est importante,

diminuant alors d'une part la vitesse et d'autre part la fiabilité du système. De plus les connexions consomment beaucoup. Dans la plus part des organisations la majorité des cellules mémoires ainsi que leur câblage restent au repos pendant la plus grande partie du temps. Ce qui constitue un gaspillage vu qu'elles sont alimentées constamment.

Le coût des communications dans les ordinateurs actuels est de ce fait exorbitant. Le plus gros de la dépense en temps et en énergie nécessaire au calcul est consommé par le transport d'informations. Ceci a conduit dans un premier temps à la hiérarchisation des mémoires et dans un deuxième temps à l'apparition des systèmes VLSI.

4.6.3. Systèmes multiprocesseurs et Processeurs pipe-line.

Les concepteurs d'ordinateurs ne se sont pas contentés d'utiliser une hiérarchie des mémoires, ils ont aussi songé à augmenter la simultanéité des traitements.

Un certain nombre de méthodes a été élaboré, nous en présentons deux:

a) - Structure multiprocesseurs:

Cette structure autorise chaque processeur à communiquer avec les autres. Par ailleurs un bus commun permet à chaque processeur d'adresser la mémoire de l'autre.

Nous constatons que le doublement du nombre de processeurs contribue au gain par un facteur deux ou plus car un gain supplémentaire est réalisé vu que chaque processeur adresse une mémoire plus petite.

b) - Processeurs Pipe-line:

Cette structure a déjà été décrite précédemment. Nous ajouterons à cela quelques propriétés concernant les calculateurs utilisant cette architecture.

Un processeur s'occupe du chargement des instructions depuis la mémoire et transmet ensuite au second processeur les informations nécessaires à leur exécution. La vitesse est ainsi doublée.

Grâce aux techniques pipe-line le traitement des données peut être effectué concurremment avec les opérations d'entrée / sortie.

Et ailleurs ce sera le cas dans notre réalisation.

4.7. Algorithmes pour les réseaux de processeurs VLSI:

Ce sont des algorithmes parallèles de hautes performances susceptibles d'être directement exécutés par des dispositifs matériels économiques.

La performance est celle qui caractérise le débit qu'il est possible d'obtenir lorsqu'un périphérique dédié est attribué à un ordinateur hôte d'usage général (ce qui est le cas du système global à réaliser au niveau du labo d'architecture des systèmes).

- Étude d'un processeur ^{produit.} interne élémentaire PIE (fig 4.5).

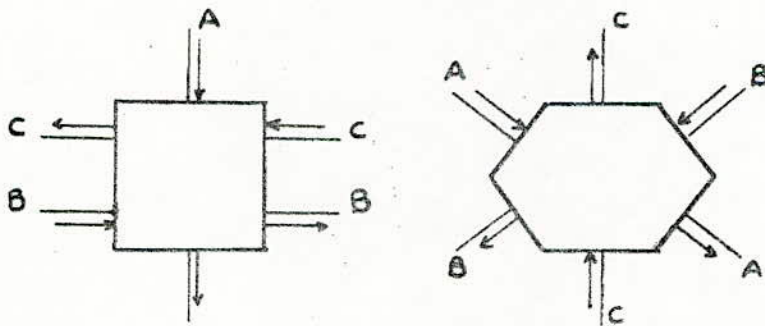


Fig 4.5. Géométrie d'un Processeur Produit Interieur Elémentaire (PIE).

Il réalise l'opération $C \leftarrow C + A.B$.

On possède un processeur à trois registres RA, RB et RC . Chaque registre admet deux accès, un pour l'entrée et un pour la sortie.

Nous appellerons unité de temps, le temps d'exécution d'une opération par ce processeur.

Durant chaque intervalle de temps unité, le processeur charge les données présentées sur les lignes d'entrée A, B, C respectivement dans les registres R_A , R_B et R_C , calcule $R_C \leftarrow R_C + R_A R_B$ et délivre le contenu de R_A de R_B et le nouveau contenu de R_C sur les lignes de sorties.

Toutes les sorties sont verrouillées et la logique est condensée, de telle sorte que lorsqu'un processeur est connecté à un autre, le changement des sorties de l'un pendant l'intervalle de temps unité n'interfère pas avec les entrées d'un autre pendant ce même intervalle.

Actuellement certains algorithmes sont très utilisés dans l'analyse numérique notamment pour effectuer le produit d'une matrice par un vecteur ou le produit de deux matrices.

Pour ce faire, il existe ce qu'on appelle des réseaux maillés ordonnés (fig 4.6) qui sont des dispositifs complets composés de plusieurs processeurs P/E interconnectés. Ses connexions d'un processeur sont à destination d'un processeur immédiatement voisin.

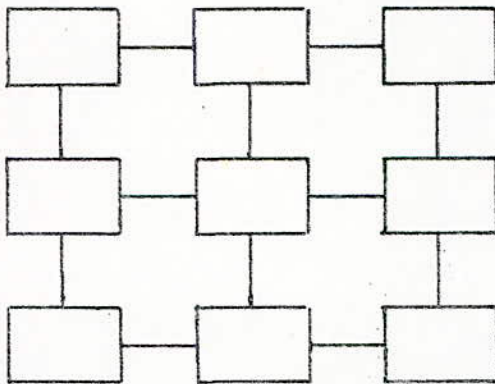
On distingue les réseaux à connexions linéaires et les réseaux à connexions orthogonales. Si de plus des connexions

diagonales sont ajoutées dans une direction, nous dirons du schéma résultat qu'il constitue un réseau maillé hexagonal. Ce dernier est particulièrement indiqué pour la triangulation d'une matrice.

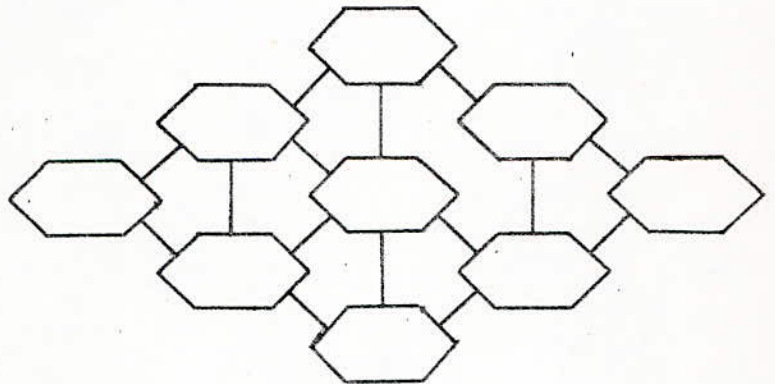
La construction de tels processeurs est très économique car les éléments de calcul sont identiques, les connexions interprocesseurs sont simples et régulières et les connexions externes réduites au maximum.



(a) Connexions Lineaires



(b) Connexions orthogonales



(c) Connexions hexagonales.

Fig 4.6. Structure des réseaux maillés.

Conclusion:

Les techniques VLSI fournissent le moyen de réaliser

économiquement des dispositifs de performances élevées.

Les approches destinées à réduire le coût des communications et à tenter d'exploiter la simultanéité, présentées dans ce chapitre peuvent être associées de diverses manières dans les structures d'ordinateur.

Le tableau suivant résume les gains de vitesse permis par ces techniques estimés à partir des modèles déjà utilisés.

Téchnique	Gain typique en vitesse possible
- Hiérarchie des machines	10
- structure pipe-line	
a) - recouvrement d'instructions	2
b) - algorithmes spécifiques	n.
- Multiprocesseurs	< n

n = nombre de processeurs.

Nous voyons donc que la VLSI tient surtout compte de :

- l'architecture du système
- l'algorithme spécifique.

Notre U.T sera réalisée dans l'architecture pipe-line utilisant un algorithme spécifique.

L'intégration à grande échelle devient alors aisée et nous procure des performances optimales quant à la vitesse,

la fiabilité, la consommation du système.

Le plus grand défi que la VLSI lance peut être à l'informatique est le développement ou non d'une théorie du traitement de l'information qui prenne en compte un modèle de coût plus général.

La réalisation de la VLSI qui se développe actuellement a révélé les faiblesses d'une théorie trop étroitement liée aux propriétés de la seule machine séquentielle.

CHAPITRE 5

CONCEPTION ET REALISATION DE L'UNITE
DE TRAITEMENT PIPE-LINE (U-T).

5.1. Introduction.

Notre projet consiste en la réalisation d'une unité de traitement pipeline faisant le calcul de l'opération "papillon" en un temps inférieur ou égal à 160 ns.

L'objectif principal est l'intégration de cette U.T. En outre cette U.T permet à l'utilisateur d'effectuer les opérations papillon de la FFT quel que soit le nombre d'échantillons (256, 512, 1024, etc....).

Elle délivre un signal qui indique la présence des résultats à sa sortie.

- Les données $RE_j(K1)$, $IM_j(K1)$, $RE_j(K0)$ et $IM_j(K0)$ arrivent (sur 16 bits) dans cet ordre à l'entrée de l'U.T à une fréquence de 25MHz.

- $\cos \theta$ et $\sin \theta$ sont donnés sur 8 bits.

- Les résultats sont obtenus dans le même ordre d'entrée et à la même fréquence.

- Les données et les résultats sont en virgule fixe, exprimés en complément à deux.

5.2. Schéma Synoptique et fonctionnement général du Système

5.2.1. Processus de Calcul de l'opération papillon:

- Pour la conception de l'U.T nous avons disposé les expressions (1.8) de la manière suivante:

$$\begin{cases}
 RE_{j+1}(K0) = [RE_j(K0) + RE_j(K1) \cos \theta] + IM_j(K1) \sin \theta, (a) \\
 RE_{j+1}(K1) = [RE_j(K0) - RE_j(K1) \cos \theta] - IM_j(K1) \sin \theta, (b) \\
 IM_{j+1}(K1) = [IM_j(K0) + RE_j(K1) \sin \theta] - IM_j(K1) \cos \theta, (c) \\
 IM_{j+1}(K0) = [IM_j(K0) - RE_j(K1) \sin \theta] + IM_j(K1) \cos \theta, (d)
 \end{cases}$$

Nous remarquons que les expressions (a) et (b) comportent les mêmes termes. Nous les effectuerons en parallèle et comme suit :

- On effectue respectivement les produits $RE_j(K1) \cos \theta$ et $IM_j(K1) \sin \theta$.

- les expressions :

$$\left. \begin{aligned}
 A_1 &= RE_j(K0) + RE_j(K1) \cos \theta \\
 S_1 &= RE_j(K0) - RE_j(K1) \cos \theta.
 \end{aligned} \right\} (5.2)$$

sont effectuées simultanément.

- Une fois le calcul précédent effectué (c'est à dire au cycle d'horloge suivant) les expressions:

$$\left. \begin{aligned} RE_{j+1}(K0) &= A_1 + IM_j(K1) \sin \theta \\ RE_{j+1}(K1) &= S_1 - IM_j(K1) \sin \theta \end{aligned} \right\} (5.3)$$

sont effectuées simultanément.

Le même procédé est utilisé pour effectuer les expressions (c) et (d)

- On obtient alors successivement $RE_j(K1) \cdot \sin \theta$ et $IM_j(K1) \cdot \cos \theta$.

$$\left. \begin{aligned} - \text{Puis } A_2 &= [IM_j(K0) + RE_j(K1) \sin \theta] \\ \text{simultanément avec:} \\ S_2 &= [IM_j(K0) - RE_j(K1) \sin \theta]. \end{aligned} \right\} (5.4)$$

- Les expressions:

$$\left. \begin{aligned} IM_{j+1}(K1) &= A_2 - IM_j(K1) \cos \theta \\ IM_{j+1}(K0) &= S_2 + IM_j(K1) \cos \theta \end{aligned} \right\} (5.5)$$

sont effectuées simultanément.

5.2.2. Fonctionnement du système:

En tenant compte du processus de calcul décrit précédemment, nous avons élaboré un schéma synoptique (fig 5.1)

- Un circuit multiplexeur permet l'aiguillage des données de telle sorte que les couples:

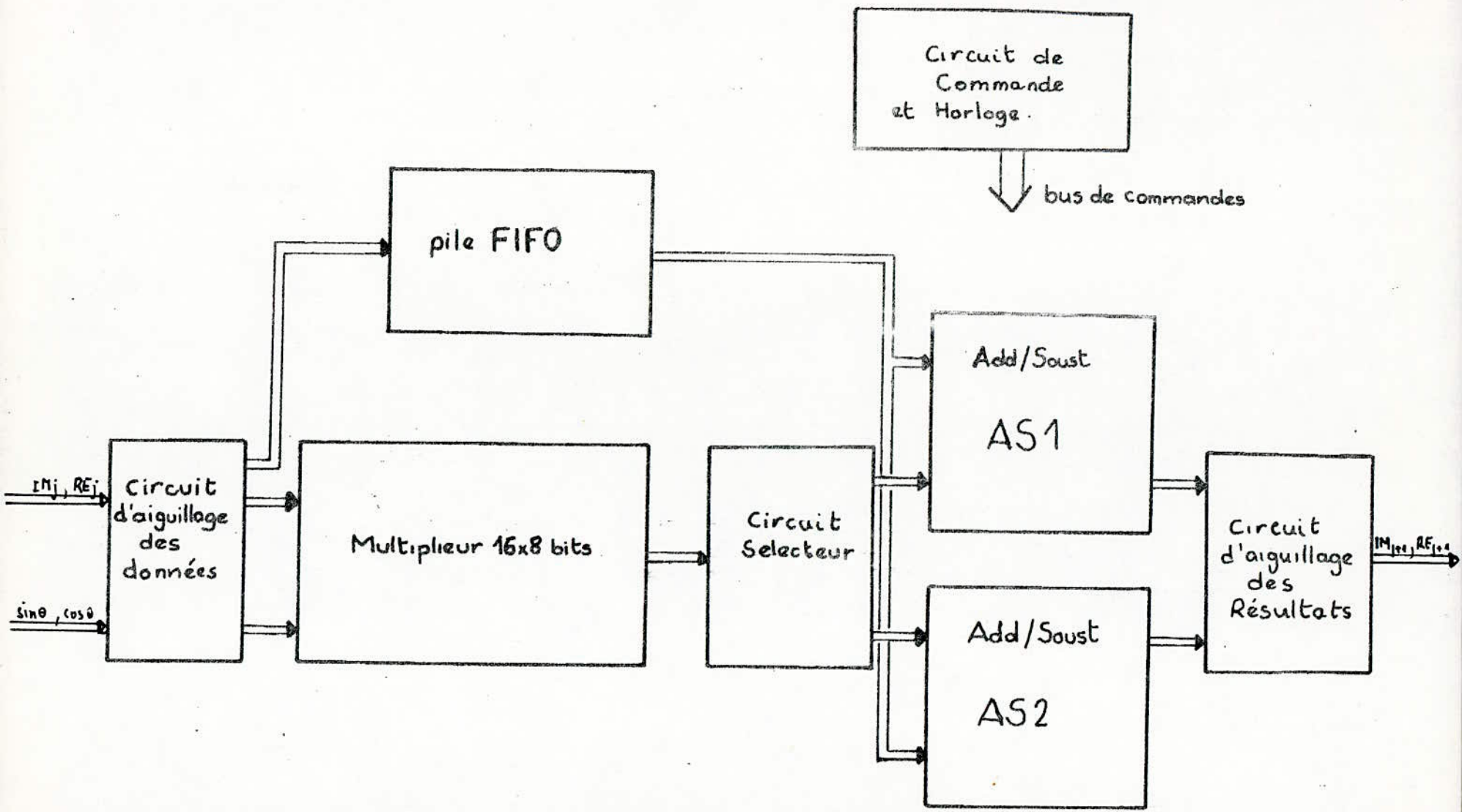


Fig 5.1 Schéma Synoptique général de l'U.T. pipe-line

$$[RE_j(K1), \cos \theta]$$

$$[IM_j(K1), \sin \theta]$$

$$[RE_j(K1), \sin \theta]$$

$$[IM_j(K1), \cos \theta]$$

passent dans le multiplieur dans cet ordre avec une fréquence de 25 MHz.

- Un multiplieur à structure pipe-line fournit à sa sortie un résultat toutes les 40 ns dans l'ordre des couples cités précédemment.

- Les termes $RE_j(K0)$ et $IM_j(K0)$ sont retardés dans la file d'attente (FIFO), afin que leur arrivée au niveau des additionneurs - soustracteurs AS1 et AS2 soit synchrone avec les produits effectués à travers les différents étages du multiplieur. L'introduction de ces termes dans la FIFO est organisée (par le circuit d'aiguillage) de façon à les obtenir en sortie respectivement et simultanément avec $RE_j(K1) \cdot \cos \theta$ et $RE_j(K1) \cdot \sin \theta$, de manière à pouvoir calculer les expressions (5.2) et (5.4).

- Un circuit de sélection disposé après le multiplieur permet d'acheminer alternativement les résultats du multiplieur vers le premier et le deuxième étage de AS1 et AS2 et commande l'additionneur - soustracteur.

- L'additionneur - soustracteur AS_1 calcule $RE_{j+1}(K0)$ et $IM_{j+1}(K1)$ et AS_2 calcule $RE_{j+1}(K1)$ et $IM_{j+1}(K0)$

- Le circuit d'aiguillage de sortie organise les résultats obtenus dans le même ordre que l'entrée des données.

- Le circuit de commande génère tous les signaux nécessaires à la synchronisation et le contrôle des différentes parties du système. Il fournit aussi le signal d'horloge (25 MHz), le signal d'initialisation de l'unité de traitement et le signal "Write" qui indiquent l'apparition du début et de la fin des résultats d'une itération.

5.3. Multiplieur 16 x 8 bits :

Le multiplieur constitue l'élément essentiel de l'U.T. Il est conçu suivant la méthode de Booth modifiée (voir chapitre 3).

L'addition des quatre produits partiels (fig 5-2) est effectuée à l'aide d'un additionneur pipe-line afin d'augmenter la rapidité.

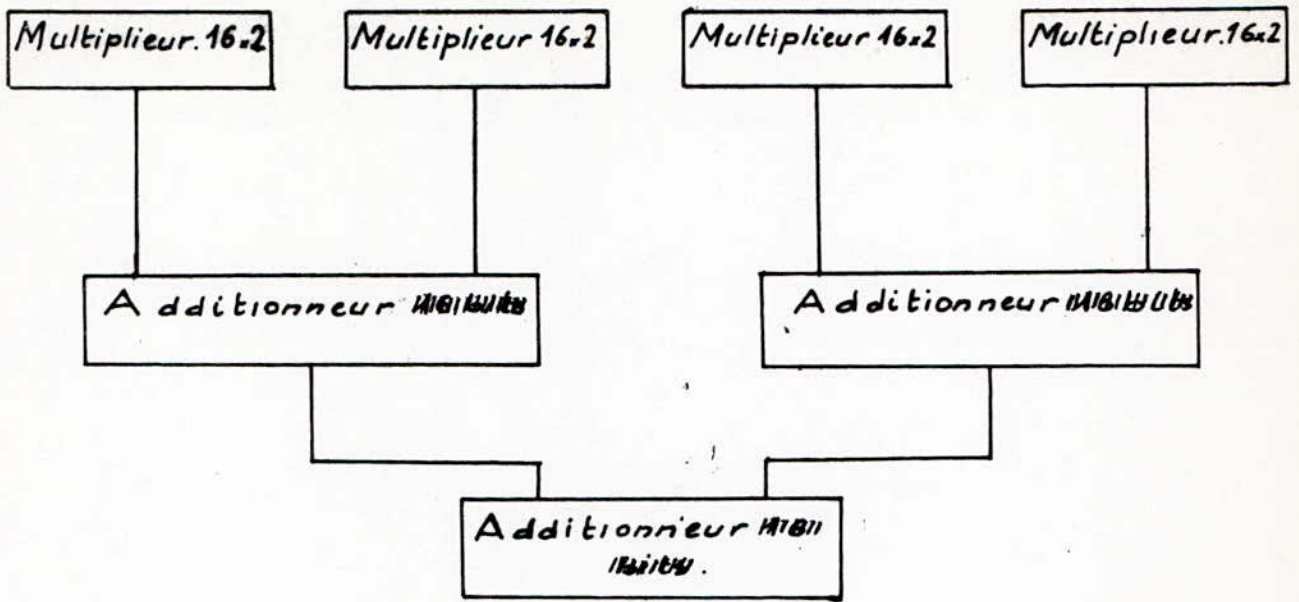


figure 5.2 : Schéma Synoptique du Multiplieur.

5.3.1. Etude d'un additionneur pipe-line:

Nous étudierons un système qui réalise l'addition de deux nombres de 16 bits représentés en virgule fixe, codés en complément à deux.

L'addition ne comporte pas de débordement au delà du signe (car la somme ne dépasse pas 1).

Cet additionneur pipe-line comporte quatre étages (fig 5.3) constitués d'additionneurs 4 bits (fig 5.4) travaillant en parallèle et de registres de découplage. Les retenues obtenues dans un étage sont propagées à l'étage suivant pour y être additionnées.

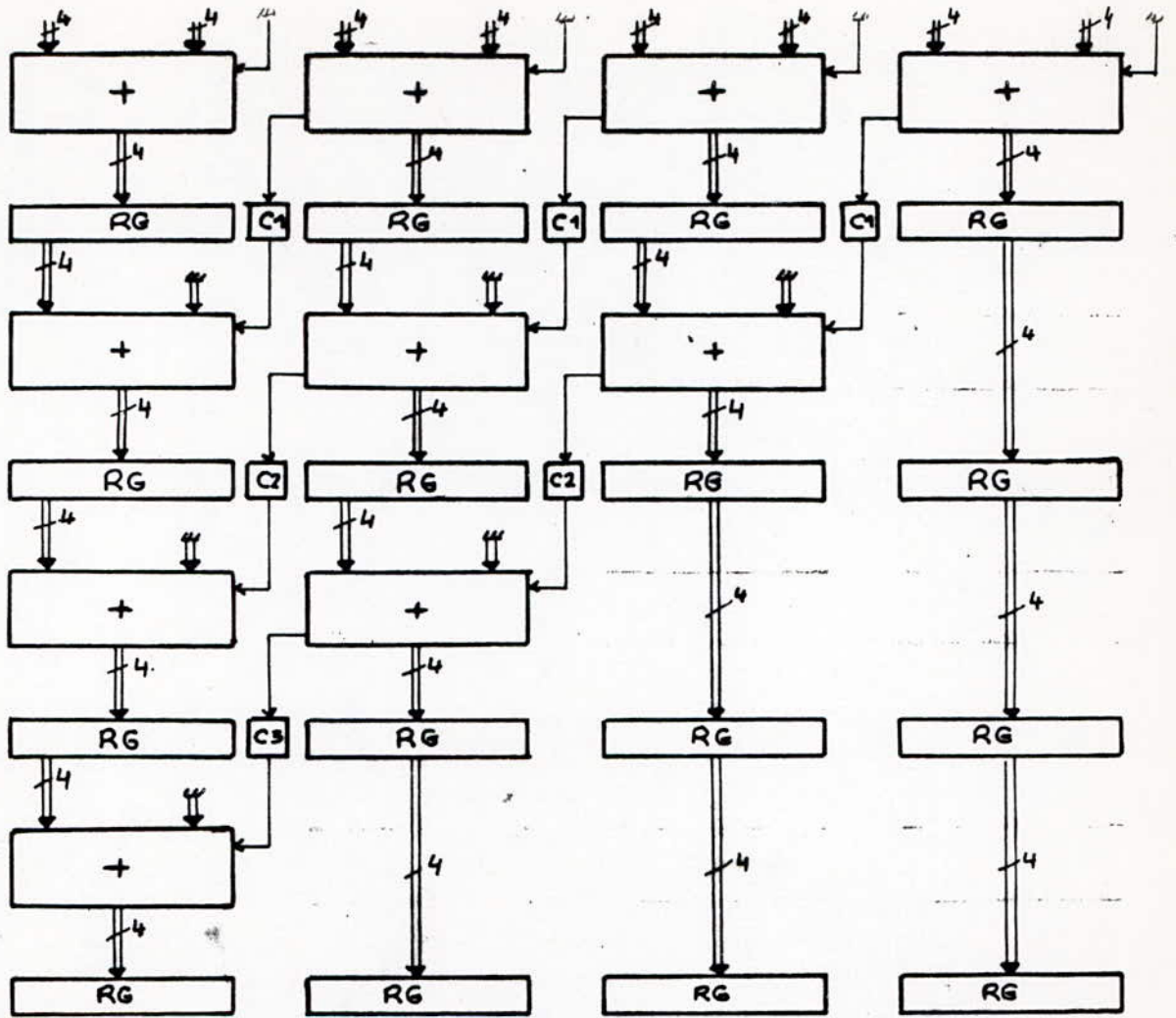


Fig 5.3 Structure d'un Additionneur Pipe-line

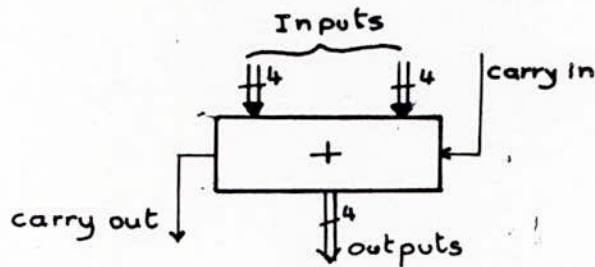


Fig 5.4 Additionneur 4 bits

Nous avons alors utilisé des additionneurs de type 74LS83 et des registres de type 74LS174.

L'exécution d'un nombre important d'additions est possible et ce avec une grande rapidité.

En effet le temps de propagation typique d'un étage est de 26 ns (16 ns dans les LS174 et 10 ns dans les LS83) d'où la possibilité de travailler à des fréquences supérieures à 30 MHz.

En outre une simple extension du système permet le traitement de plusieurs nombres.

5.3.2 Fonctionnement du multiplieur 16.8 bits :

Il est constitué de 8 étages (fig 5.5) dont le premier est destiné au calcul des 4 produits partiels. Les sept autres étages réalisent l'addition de ces derniers en tenant compte, des décalages obtenus directement par câblage.

La mise en parallèle de quatre circuits multiplieurs 4×2 bits (type 74LS261) réalise la multiplication 16×2 bits désirée (fig 5.6).

Afin d'obtenir des résultats en complément à deux il est nécessaire d'ajouter un bit appelé "Rounding bit" délivré grâce à un circuit à deux portes élémentaires (fig 5.7).

Les quatre rounding bits (R_1, R_2, R_3, R_4) sont obtenus parallèlement à l'aide d'un circuit 74LS08 et un circuit 74LS00.

- Deuxième étage :

A ce niveau, on effectue les sommes partielles :

$$S_{21} = PP1 + PP2 + R1.$$

$$S_{22} = PP3 + PP4 + R3.$$

en tenant compte de l'extension du signe.

L'addition de $R1$ à $PP1$ et $PP2$ est réalisé en utilisant l'entrée carry de l'additionneur (A200) des quatre bits de plus faibles poids.

$R2$ ne pouvant être ajouté dans cet étage (toutes les entrées de A200 sont utilisées) est stocké avec le résultat M (23 bits) et les cinq retenues C_1 dans les cinq registres (RG200 à RG204)

$R3$ est ajouté au LSB de $PP3$ (entrée carry de A206).

Les 19 bits du résultat N , les quatre retenues C_2 ainsi que le bit R_4 sont mémorisés dans les registres (RG205 à RG208).

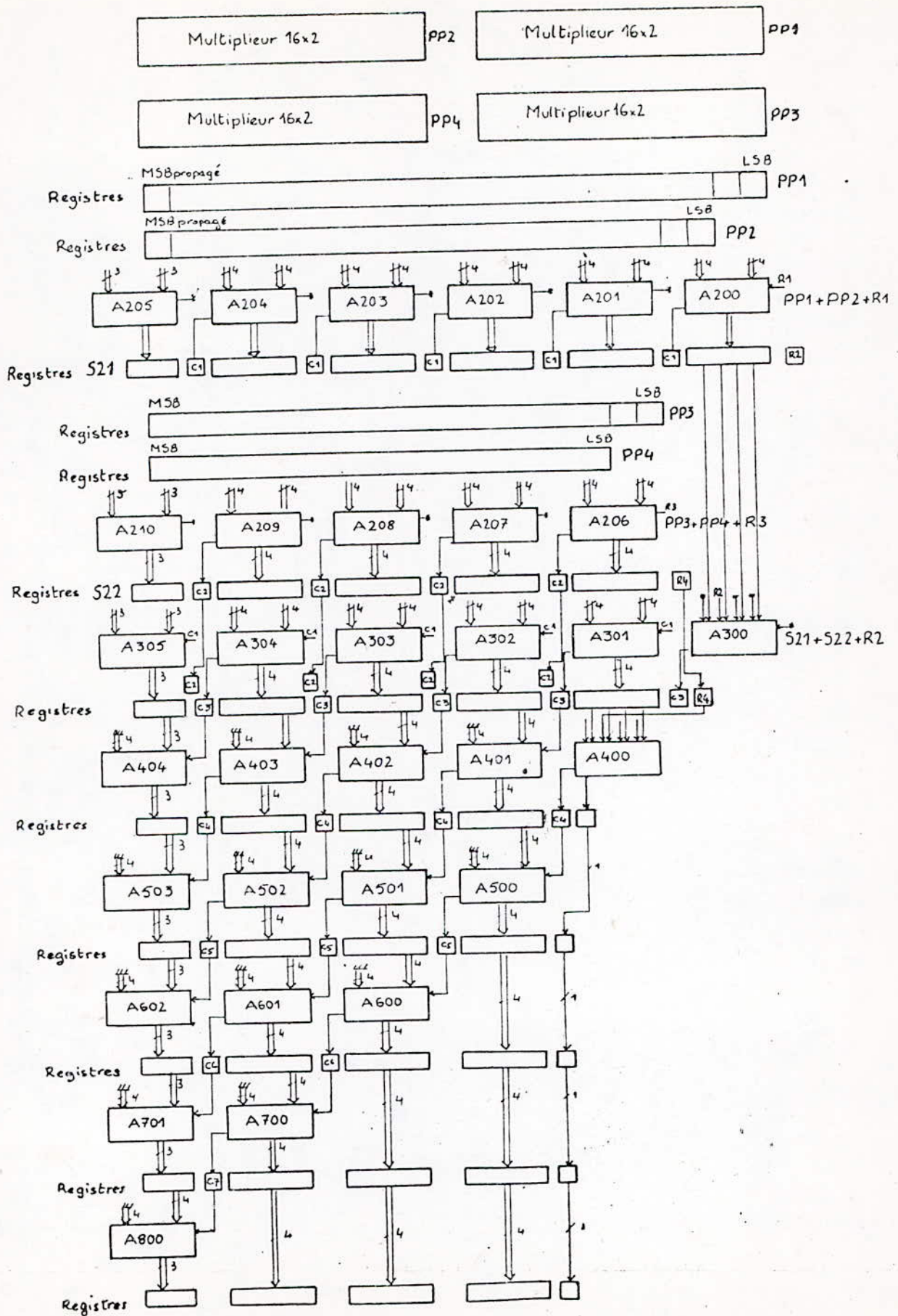


Fig 5-5 Schéma Synoptique du MULTIPLIEUR 16x8.

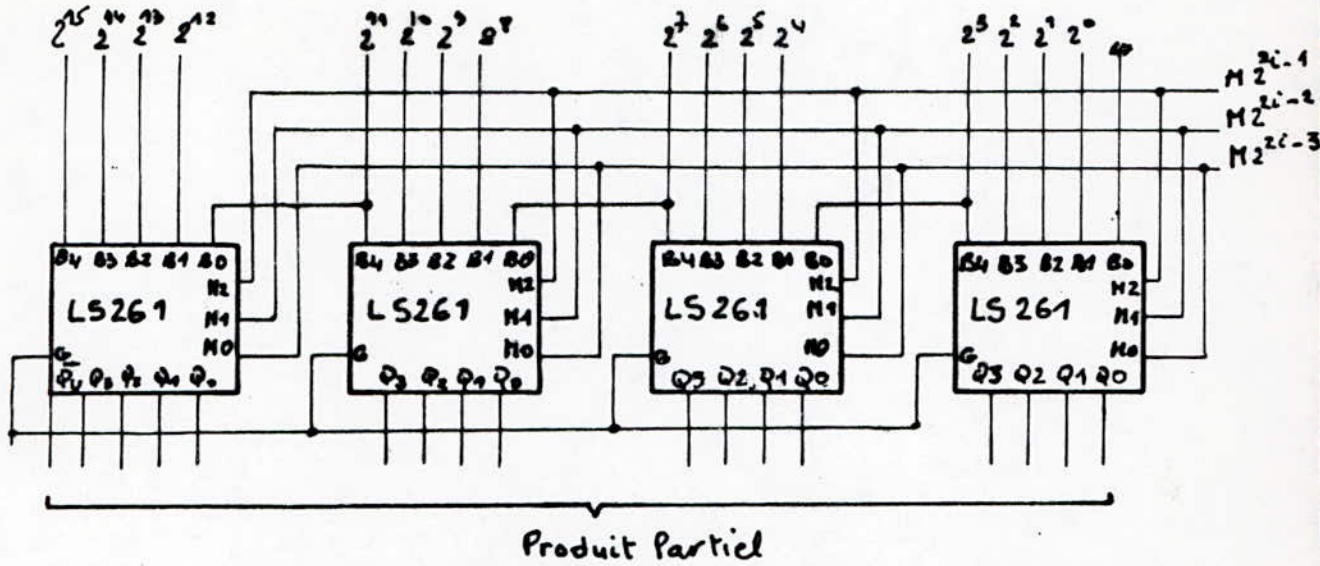
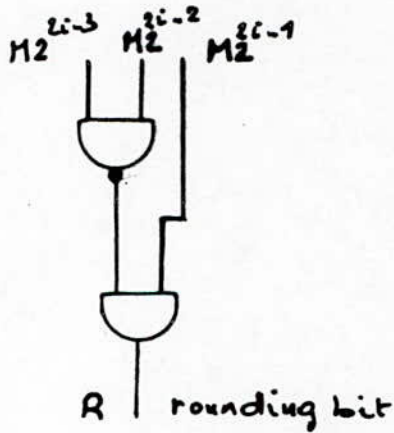


Fig 5.6 Multiplieur 16x2 bits



Groupe de 3 bits de M1			R
2^{2i-1}	2^{2i-2}	2^{2i-3}	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Fig 5.7 .Circuit et table de vérité du Rounding bit.

- * Les principales caractéristiques des circuits LS 261. (fig 5.8) sont:
- Multiplication 4.2 bits en 26 ns type.
 - Puissance dissipée: 110 mW type.
 - Sorties verrouillées (très utile pour la synchronisation).
 - Compatibilité avec toutes les familles TTL.
 - Les données ainsi que les résultats sont en complément à deux.

Remarque:

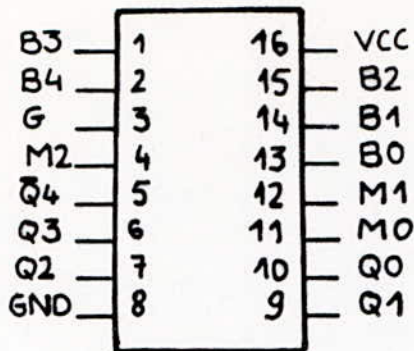
Le cinquième bit (bit signe) de sortie est inversé. L'obtention du résultat correct nécessite alors l'inversion de celui-ci.

Le schéma électrique est illustré par la figure 5.9.

- Premier étage:

Les quatre produits partiels (PP1, PP2, PP3, PP4) sont donnés simultanément puis stockés dans des registres de type 74LS174.

Les quatre bits de signe devant être inversés sont mis dans un registre 74LS175. Ce registre de quatre bits est identique au 74LS174 du point de vue rapidité mais offre en plus des sorties symétriques Q et \bar{Q} .



B0, B1, B2, B3, B4 : Multiplicande

M0, M1, M2 : Multiplicateur

Q0, Q1, Q2, Q3, Q4 : Résultat

G : Verrouillage des Sorties .

INPUTS				OUTPUTS				
Latch Control G	Multiplicateur			Q4	Q3	Q2	Q1	Q0
	M2	M1	M0					
L	X	X	X	Q4	Q3	Q2	Q1	Q0
H	L	L	L	H	L	L	L	L
H	L	L	H	B4	B4	B3	B2	B1
H	L	L	L	B4	B4	B3	B2	B1
H	L	H	H	B4	B3	B2	B1	B0
H	H	L	L	B4	B3	B2	B1	B0
H	H	L	H	B4	B3	B2	B1	B0
H	H	H	L	B4	B3	B2	B1	B0
H	H	H	H	H	L	L	L	L

Fig 5.8 Schéma de brochage et table de vérité du LS261

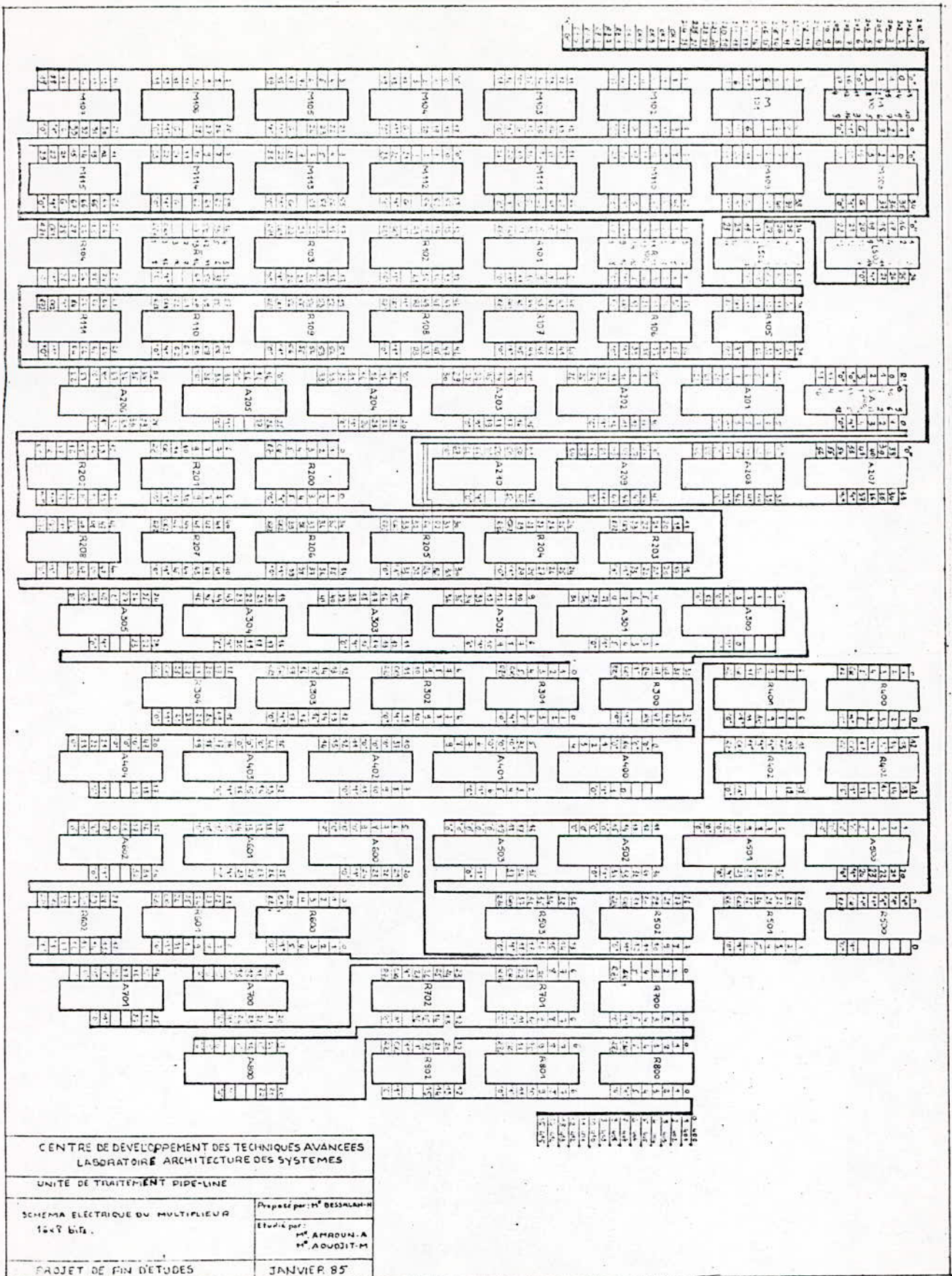


Fig 5.9

N.B. Voir agrandissement en annexe

Pour la même raison que R_2 , R_4 sera propagé vers le troisième puis le quatrième étage où il sera additionné à la somme partielle.

- Troisième étage:

A l'arrivée du top d'horloge suivant les résultats de l'étage précédent sont présents à l'entrée des additionneurs A_{3j} qui à leurs sorties fournissent la somme S_3 avec:

$$S_3 = S_{21} + S_{22} + \text{les retenues } C_1 + R_2^*$$

* : le bit R_2 est ajouté au LSB de PP_2 et ce au niveau de A_{300} , dont les quatre bits de sortie sont définitifs. Cependant ils ne sont pas stockés, cette troncature est due au fait que le résultat final est donné sur 16 bits.

- Quatrième étage:

On procède maintenant à l'addition des retenues (C_2 et C_3), du résultat partiel S_3 obtenu précédemment et du bit R_4 . Ce dernier correspond au LSB de PP_4 .

Encore une fois il y aura troncature; en effet les trois premiers bits donnés par A_{400} ne seront pas

Conservés : ils représentent le 17^e, 18^e et 19^e bit du résultat général qui contient 23 bits et dont on ne retient que les 16 de plus forts poids.

- Au delà du Cinquième étage :

A partir du cinquième étage le circuit devient un additionneur pipe-line classique à propagation normale des retenues (cette structure à été étudiée dans le paragraphe 5.3.1).

on notera que les bits définitifs obtenus au fur et à mesure dans les 4^e, 5^e, 6^e et 7^e étages seront propagés jusqu'au huitième, où l'on obtient le résultat final sur 16 bits.

Conclusion :

Comparée à la méthode manuelle, la méthode de Booth modifiée divise le pipe-line par deux (du point de vue nombre d'étages).

La gamme des multiplicateurs performants existant sur le marché est donné par le tableau 4. [5]

Type	Fonction	Vitesse (ns)		Consommation (mW)	Boitier
		type (25°C)	max 0 à 70°C		
25 S 557 et 74 S 557	Multiplieur 8.8 bits parallèle avec registre tampon en sortie.	45	80	1400	40 br
25 S 558 et 74 S 558	Multiplieur 8.8 bits parallèle (sans registre tampon en sortie)	45	80	1400	40 br
29 S 16 et MPY-16NJ	Multiplieur 16x16 parallèle avec possibilité supplé- mentaire de multiplexage en sortie (4 signaux d'horloge de commande)	40	65	4000	64 br

Tableau 4.

Un multiplieur 16x16 bits parallèle en boitier 64 broches est référencé par 74 ALS 1616. Les performances annoncées le situent comme le plus performant multiplieur du marché:

- Temps de multiplication : 40 ns type .
- Consommation : 1,5 W max .
- Possède un registre permettant de mémoriser les operands et le résultat .

Pour le multiplieur qu'on a réalisé, le temps typique

de propagation dans un étage est de 36 ns ($f = 28 \text{ MHz}$).

La consommation calculée à partir des puissances types respectives dissipées dans chaque circuit intégré utilisé est de 6 watts. Ce qui est très proche de la consommation des multiplieurs totalement intégrés existant sur le marché.

Lors des tests le courant mesuré est de 1,3 A ce qui correspond à une puissance de 6,5 W.

L'horloge (25 MHz) commandant les registres 74LS174 et 74LS175 est distribuée par des portes trigger de type 74LS14

5.4. Circuit d'aiguillage des données et file d'attente (FIFO):

5.4.1. Circuit d'aiguillage des données:

Le multiplieur effectue les produits dans cet ordre:

$$\left. \begin{array}{l} - RE_j(K1) \cdot \cos \theta \\ - IM_j(K1) \cdot \sin \theta \\ - RE_j(K1) \cdot \sin \theta \\ - IM_j(K1) \cdot \cos \theta \end{array} \right\} (5.6).$$

L'utilisation de $RE_j(K0)$ et $IM_j(K0)$ comme indiqué sur le diagramme de la figure 5.10 est obtenue en mémorisant $IM_j(K0)$ pendant un cycle d'horloge (afin de le retarder pendant ce cycle), alors que $RE_j(K0)$ passe directement dans la FIFO.

Les entrées de RG_1 , RG_2 , B_0 et RG_5 étant reliées entre elles, reçoivent respectivement les données $RE_j(K1)$, $IM_j(K1)$, $RE_j(K0)$ et $IM_j(K0)$ (fig 5.11)

L'entrée d'un circuit est validée seulement quand l'information lui est destinée.

Il en sera de même pour RG_3 et RG_4 qui reçoivent respectivement $-\cos\theta$ et $\sin\theta$.

En compatibilité avec les conditions citées ci-dessus et en tenant compte du temps de propagation des circuits dont nous disposons, nous avons opté pour des registres de type 74LS373 (à sortie trois états et entrée commandable) et des buffers trois états de type 74LS244.

Les signaux de commande relatifs à l'entrée et à la sortie des circuits sont représentés par les chronogrammes de la figure 5.12.

Le schéma électrique est donné par la figure 5.13.

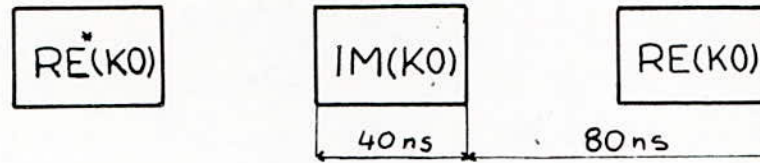


Fig 5.10. Organisation des données dans la FIFO

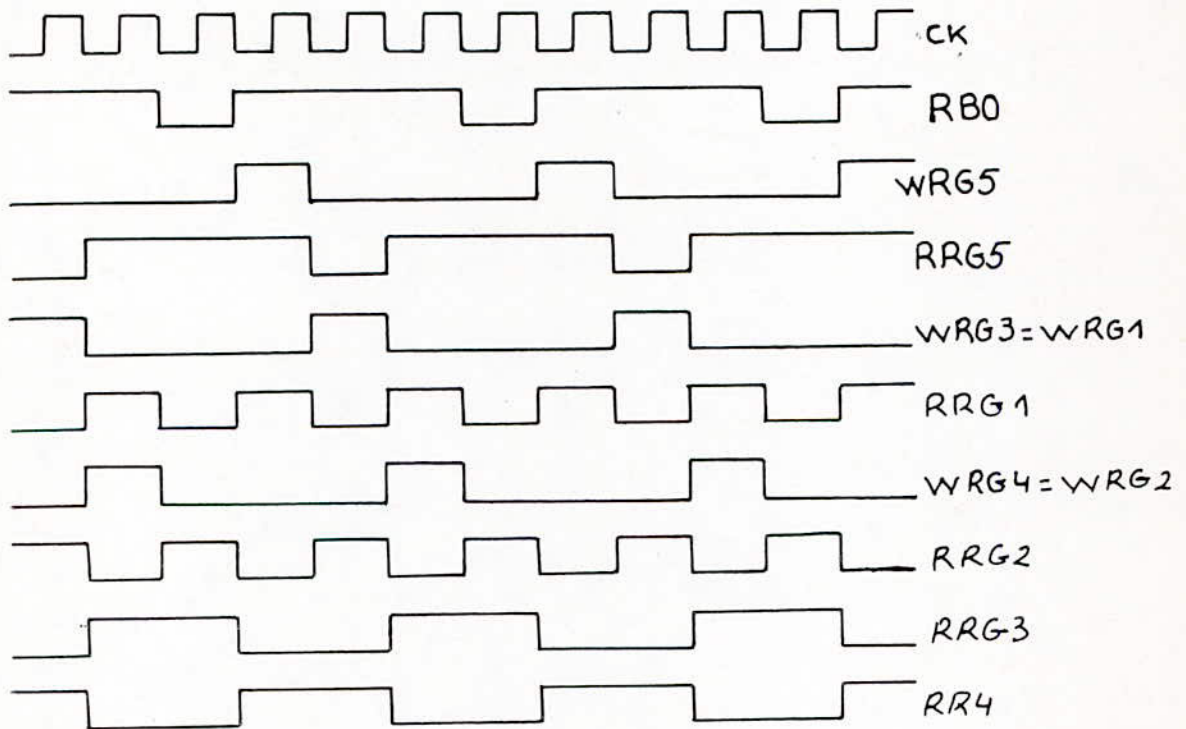


Fig 5.12. Chronogrammes des Signaux de Commande du Circuit d'aiguillage des données

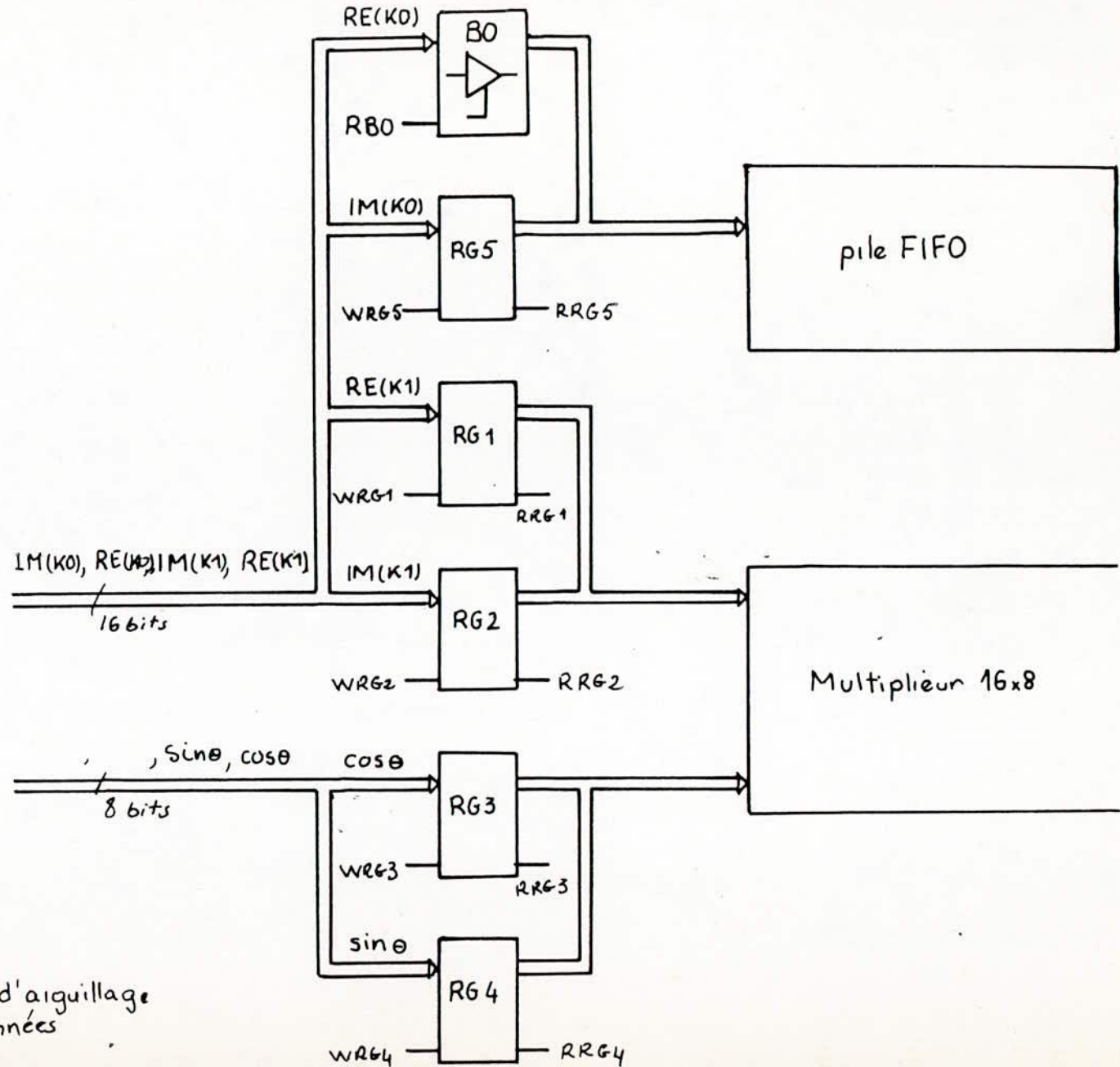


Fig 5.11 Circuit d'arbitrage des données

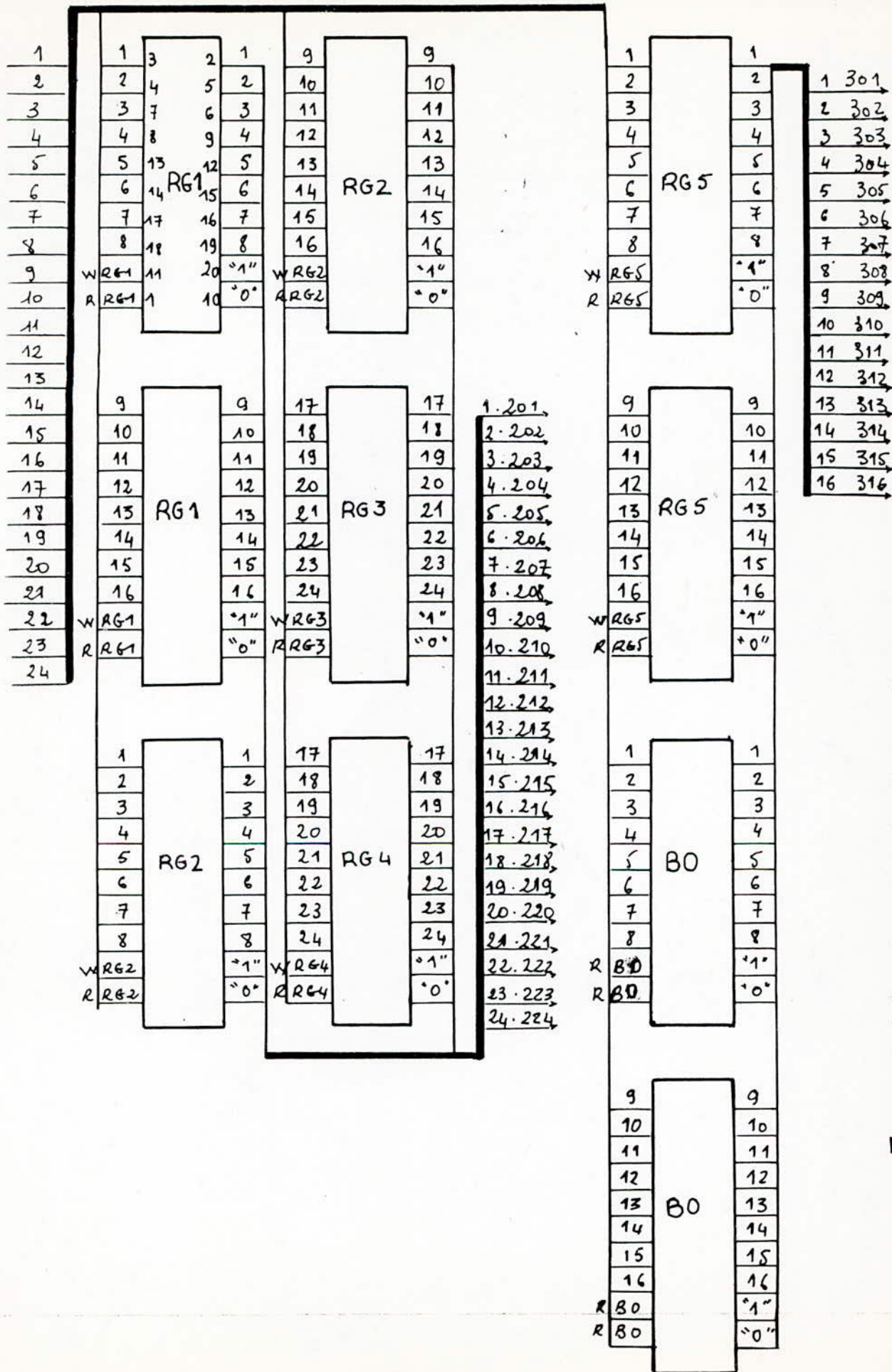


Fig. 5.13. Schéma électrique du circuit d'aiguillage des données.

En outre seuls les niveaux des signaux nous intéressent pour la commande de chaque circuit.

5.4.2 La FIFO:

$RE_j(K0)$ arrive en retard de deux cycles d'horloge (ce qui équivaut à deux étages du multiplieur) par rapport à $RE_j(K1)$ qui traversera les huit étages du multiplieur (paragraphe 5.3).

Ainsi la pile doit contenir six étages et fonctionnent à la même fréquence que le multiplieur.

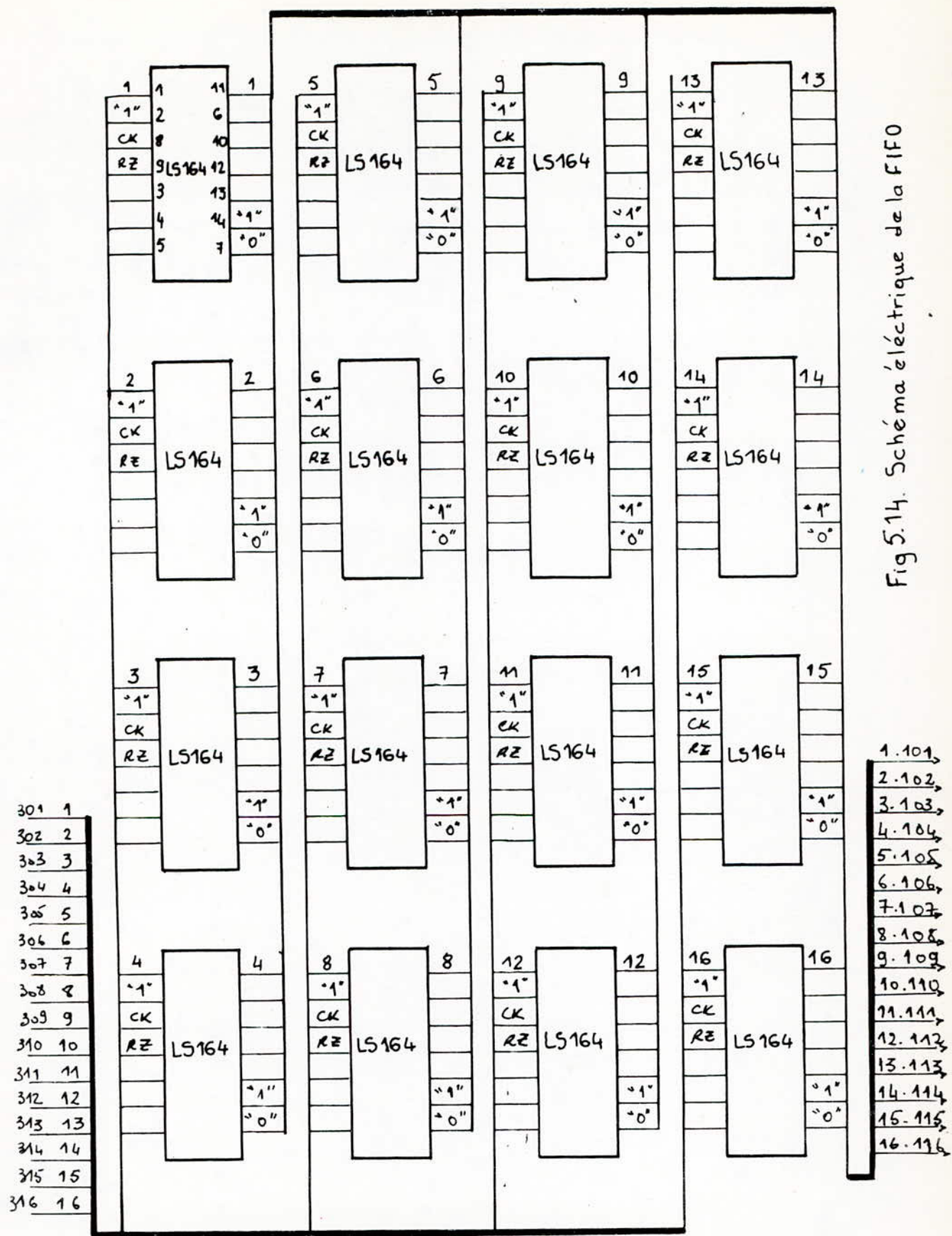
Dans la réalisation nous avons utilisé 16 registres (car $RE_j(K0)$ et $IM_j(K0)$ sont donnés sur 16 bits) à décalage de 8 bits (type 74LS164). Ceux-ci sont placés en parallèle; chacun correspond à un bit.

Les sorties sont prises au niveau des sixièmes bascules des registres.

Le schéma électrique est illustré par la figure 5.14.

Les tests effectués sur cette pile réalisée à base de registres à décalage de type 74LS164.

Fig 5.14. Schéma électrique de la FIFO



ont donné de bons résultats à des fréquences dépassant 25 MHz

5.5. Circuit Sélecteur et Additionneurs-Soustracteurs:

Les deux additionneurs-soustracteurs AS1 et AS2 travaillent en parallèlement dans le but d'augmenter la rapidité. Ainsi,

AS1 calcule $RE_{j+1}(K1)$ et $IM_{j+1}(K0)$
et

AS2 calcule $RE_{j+1}(K0)$ et $IM_{j+1}(K1)$.

AS1 et AS2 sont deux additionneurs pipe-line (voir paragraphe 5.3).

Le traitement de nombres codés en complément à deux, transforme au besoin ceux-ci en soustracteurs.

La soustraction de deux nombres A et B donnés dans le code du complément à deux peut se ramener à une addition.

$$A + (-B) = A + B^*$$

B^* : complément à deux de B. ($B^* = \bar{B} + 1$)

B^* est obtenu en inversant B au niveau du sélecteur et en ajoutant un 1 au niveau de l'additionneur (entrée-carry).

Nous avons à réaliser la synthèse de deux additionneurs pipe-line calculant chacun la somme de trois nombres A, B et C.

Au premier étage, on commence l'opération $A+B$, le résultat obtenu est ajouté à C au niveau du deuxième étage durant le cycle d'horloge suivant.

Pour additionner trois nombres donnés sur 16 bits, il faudrait cinq étages.

Au niveau du premier étage de AS1 et AS2 sont effectués respectivement les expressions (5.2)

Au top d'horloge suivant et pendant que le deuxième effectue les expressions (5.3) dans AS1 et AS2, le premier étage est au repos.

Durant les deux cycles suivants; AS1 et AS2 effectuent dans les deux premiers étages respectifs les expressions (c) et (d) de (5.1) de la même manière que précédemment.

On constate alors que chaque étage de AS1 et AS2 fonctionne pendant un cycle sur deux, ce qui conduit à une arrivée des résultats par couples de deux nombres toutes les 80 ns.

Le circuit sélecteur aiguille les données à

additionner suivant quatre viers vers le premier ou le deuxième étage de A51 et A52 (fig 5.15).

- Les buffers trois états inverseurs sont de type 74LS240.

- Les buffers trois états non inverseurs sont de type 74LS244.

- Pour les portes AND, nous avons utilisé des circuits de type 74LS08.

Les signaux de commande RBX et RAND (chronogrammes fig 5.18) sont distribués à l'aide de circuits de type 74LS265.

A51 et A52 sont construites de la même manière que dans le multiplieur de la figure 5.5.

Le schéma électrique du circuit sélecteur est illustré par la figure 5.17

Pour l'ensemble de ces circuits les tests ont donné de bons résultats jusqu'à la fréquence $f = 18 \text{ MHz}$

L'entrée "carry in", du premier étage de A51 et A52 est mise respectivement à l'état "0" et "1".

A l'entrée "carry in" du deuxième étage de A51 et A52 arrivent respectivement les signaux $\bar{A}/51$ et $\bar{A}52$.

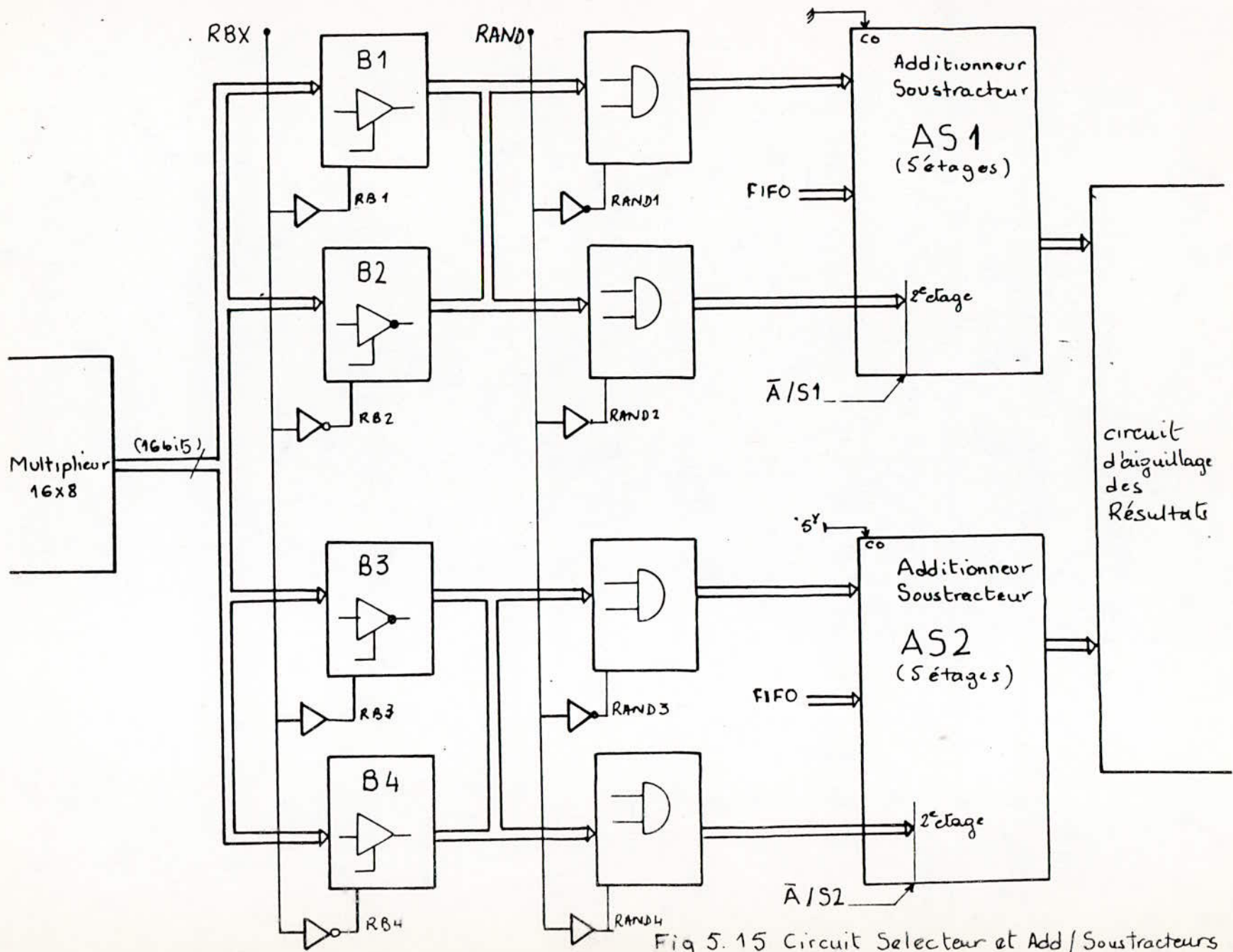


Fig 5.15 Circuit Selecteur et Add/Soustracteurs

Les signaux RBX et $RAND$ commandent respectivement les ensembles de buffers (B_1, B_2, B_3, B_4) et les portes AND.

Les chronogrammes de ces signaux sont donnés par la figure 5.18.

ona: $\bar{A}/S1 = RBX.$

$$\bar{A}/S2$$

$$RAND.$$

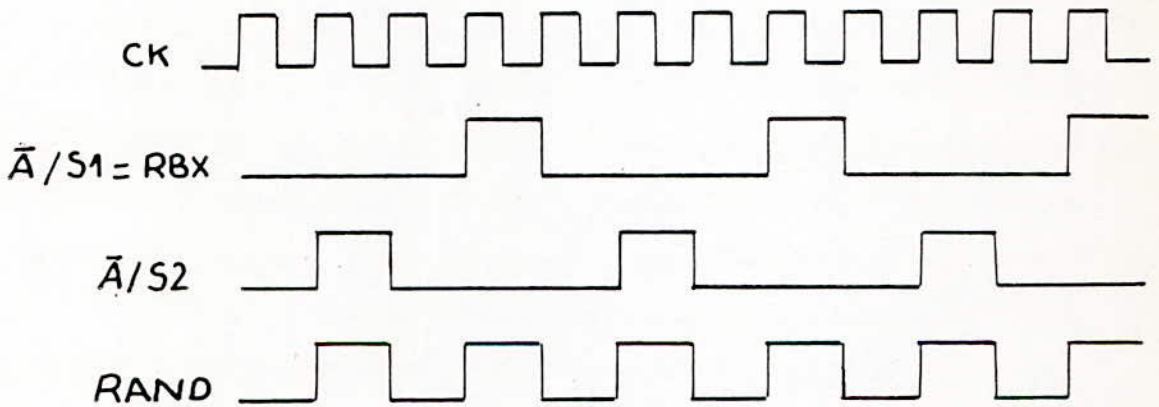


Fig 5.18. Chronogrammes des signaux de commande.

5.6. Circuits d'aiguillage des résultats:

Les résultats obtenus lors d'une itération, à la sortie de l'U.T, serviront à l'entrée pour la prochaine itération.

Leur organisation à la sortie devrait être identique à celle de l'entrée des données.

Comme le montre le schéma synoptique de la figure 5.20, chaque résultat correspond à un registre.

Ainsi à l'arrivée des couples de résultats délivrés par AS1 et AS2, on valide l'entrée des deux registres correspondants.

L'écart de 80 ns (équivalent à deux cycles d'horloge du pipe-line) séparant deux couples de résultats fournis par AS1 et AS2 fait qu'on mémorisera le tout premier couple ($RE_1(K0)$, $RE_1(K1)$) pendant un cycle d'horloge.

Au cycle d'horloge suivant la sortie de RG8 est validée, le résultat $RE_1(K1)$ est alors disponible à la sortie de l'U.T.

A partir de ce cycle le phénomène devient périodique (le système délivrera un résultat toutes les 40 ns).

Pour les mêmes raisons citées dans le paragraphe 5.4, nous avons utilisé comme registres commandables à l'entrée et à la sortie les 74LS373.

Ses signaux de commande relatifs à ce circuit d'aiguillage sont représentés par leur chronogrammes respectifs (fig 5.20).

Le schéma électrique est illustré par la figure 5.21.

Les tests ont donné un fonctionnement parfait jusqu'à une fréquence de 19 MHz.

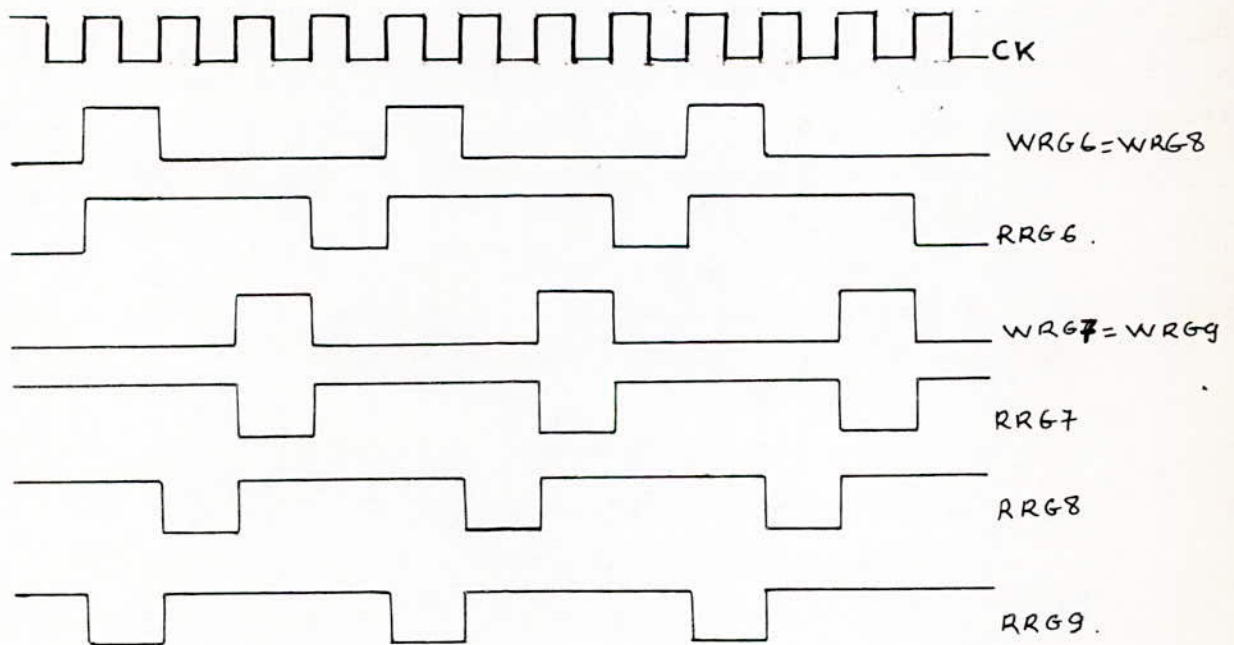
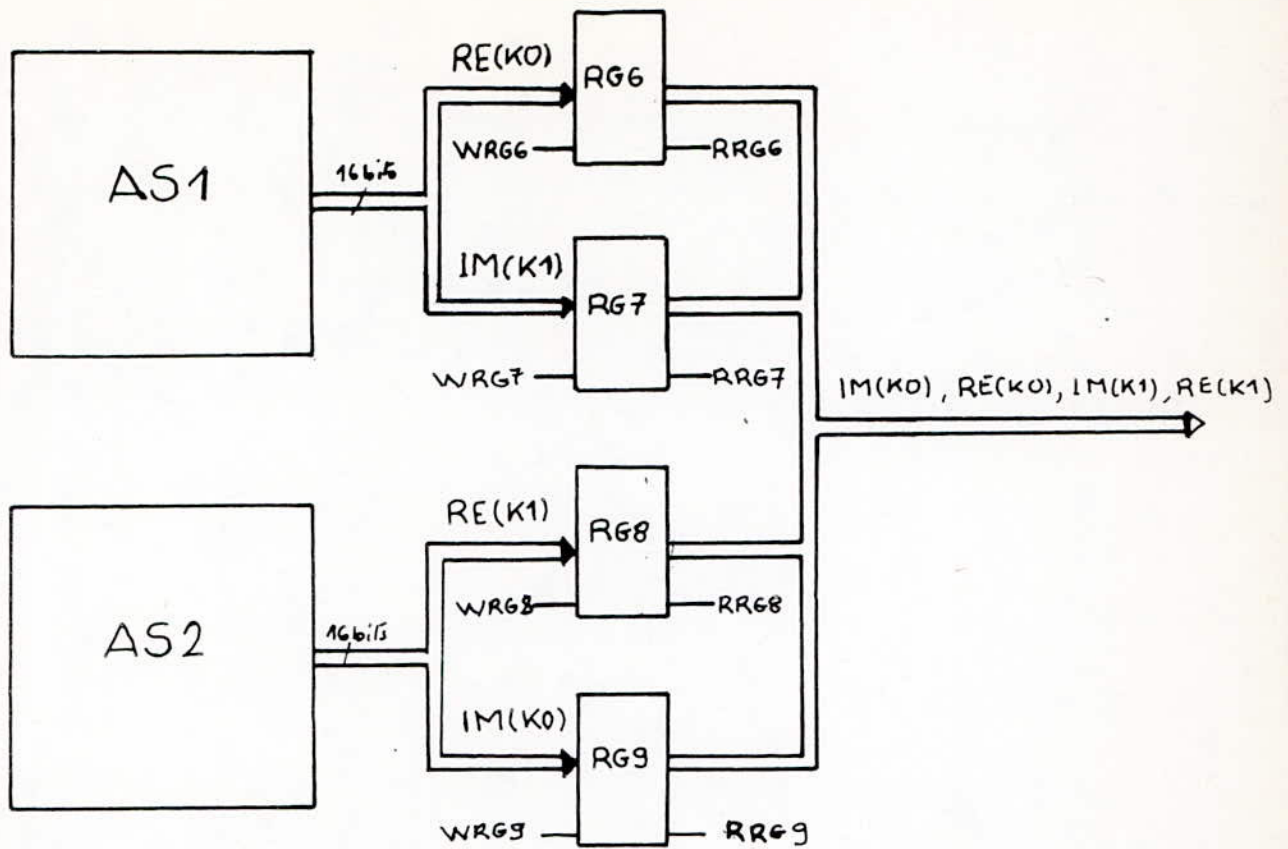


Fig 5.20 Schéma fonctionnel et chronogrammes des Signaux de commande du Circuit d'aiguillage des résultats :

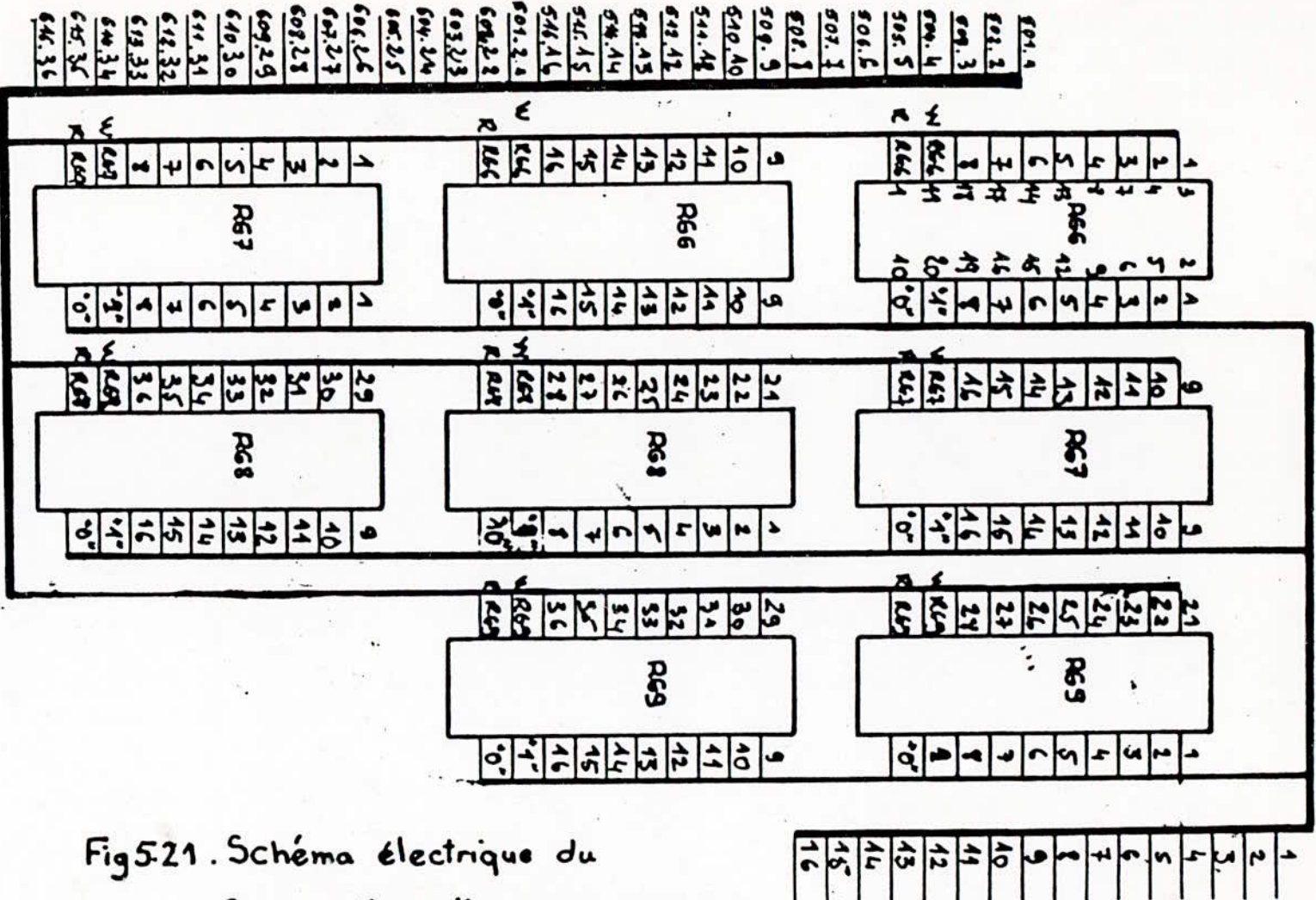


Fig 5.21. Schéma électrique du Circuit d'aiguillage de résultats

5.7. Circuit de commande:

Dans chaque partie étudiée précédemment, notre circuit faisait appel à des signaux de commande.

Dans le but d'avoir une bonne synchronisation, tous les signaux seront fournis par le même circuit.

On notera que seul le niveau du signal commande nos circuits.

Un tableau récapitulatif de tous les signaux est donné dans la figure. 5.22.

Il apparaît ainsi que plusieurs signaux sont identiques entre-eux.

En effet :

$$RB0 = RRG8 \quad \text{on le note } H1.$$

$$WRG5 = RBX = \bar{A}/S1 = WRG7 = WRG9 \quad \text{on le note } H2$$

$$RRG5 = RRG6 \quad \text{on le note } H7.$$

$$WRG1 = WRG3 \quad \text{" " " } H3.$$

$$RRG1 = RAND \quad \text{" " " } H4.$$

$$\bar{A}/S2 = WRG2 = WRG4 = WRG6 = WRG8 \quad \text{on le note } H5$$

$$\text{on notera } RRG3 \text{ par } H6.$$

Puis on remarque que :

$$H7 = \bar{H}3$$

$$RRG2 = \bar{H}4$$

$$RRG7 = \bar{H}2$$

WRG9	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
RRG9	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0
WRG1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
RRG1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
WRG2	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
RRG2	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
WRG3	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
RRG3	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
WRG4	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
RRG4	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
WRG5	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
RRG5	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1
RBO	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
RBX	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
RAND	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
A/S1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
A/S2	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
WRG6	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
RRG6	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1
WRG7	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
RRG7	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
WRG8	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
RRG8	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0

Fig 5-22: Tableau récapitulatif des signaux de Commande

$$RRG9 = \bar{H}5$$

$$RRG4 = \bar{H}6$$

Ce qui limite le nombre de signaux à fournir à six, il s'agit de $H1, H2, H3, H4, H5$ et $H6$.

L'utilisation combinée des sorties Q_A et Q_B d'un compteurs à quatre bits nous conduit aux formules suivantes :

$$* H2 = Q_A \cdot Q_B$$

$$\text{or } RRG7 = \bar{H}2$$

$$\text{d'où } RRG7 = \bar{H}2 = \overline{Q_A \cdot Q_B}$$

$$* H4 = Q_A$$

$$\text{or } RRG2 = \bar{H}4$$

$$\text{d'où } RRG2 = \bar{Q}_A$$

$$* H1 = \overline{Q_A \cdot Q_B}$$

$$* H5 = \bar{H}2 \cdot H4 = \overline{Q_A \cdot Q_B} \cdot Q_A = (\bar{Q}_A + \bar{Q}_B) Q_A = Q_A \cdot \bar{Q}_B$$

$$\text{or } RRG9 = \bar{H}5 = \overline{Q_A \cdot \bar{Q}_B}$$

$$* H3 = \bar{H}5 \cdot \bar{Q}_B = \overline{Q_A \cdot \bar{Q}_B} \cdot \bar{Q}_B = (\bar{Q}_A + Q_B) \bar{Q}_B = \bar{Q}_A \cdot \bar{Q}_B$$

$$\text{or } H7 = \bar{H}3 = \overline{\bar{Q}_A \cdot \bar{Q}_B}$$

$$* H6 = H7 \cdot \bar{H}2 = \overline{\bar{Q}_A \cdot \bar{Q}_B} \cdot \overline{Q_A \cdot Q_B}$$

$$= (Q_A + Q_B) (\bar{Q}_A + \bar{Q}_B)$$

$$= Q_A \bar{Q}_B + Q_B \bar{Q}_A = Q_A \oplus Q_B$$

$$\alpha RRG4 = \bar{H6}$$

$$\text{d'où } RRG4 = \overline{QA \oplus QB}$$

Le compteur utilisé est du type 74LS 163.

Nous obtenons ainsi le circuit de la figure 5.23.

Le compteur est attaqué par une horloge de 25 MHz.

Cette même horloge retardée (du même temps que met le circuit de commande pour générer les six signaux) est distribuée à travers tout le système pour la commande des registres du pipe-line et de la FIFO.

La distribution de l'horloge principale est faite à l'aide de portes triggers (type 74LS 14).

Les portes à sorties symétriques de type LS 265 sont utilisées pour la distribution des signaux de commande.

Conclusion :

Les tests effectués sur notre montage ont donné de bons résultats.

En effet les signaux obtenus sont conformes à ceux recherchés pour une fréquence allant jusqu'à 25 MHz.

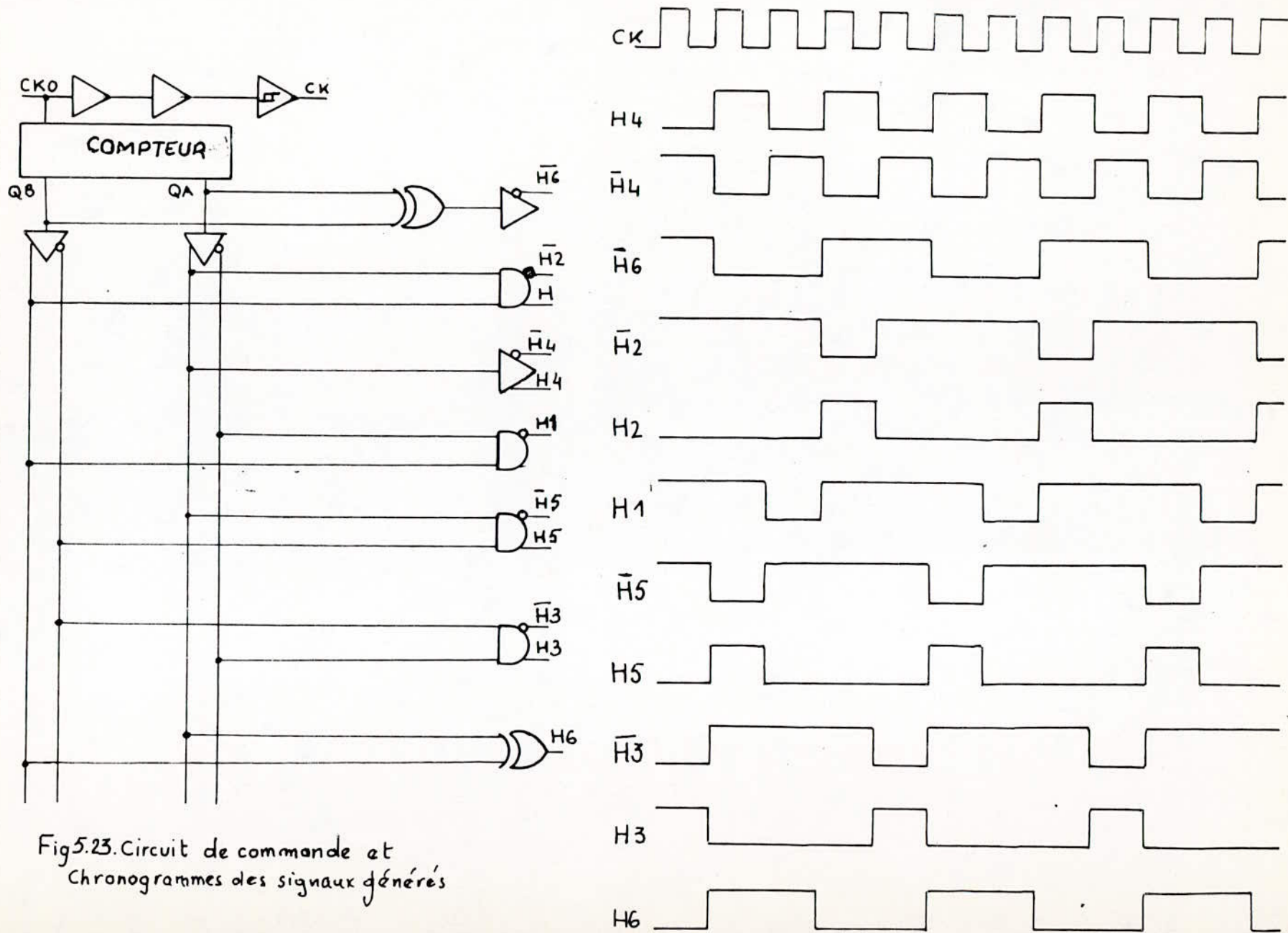


Fig.5.23. Circuit de commande et
Chronogrammes des signaux g n r s

5.8. Horloge.

Notre système étant appelé à être intégré, il est alors nécessaire de prévoir un circuit d'horloge pour assurer l'autonomie de l'U.T.

Ce circuit est réalisé à base d'un oscillateur à fréquence variable (de type 74LS124).

Une capacité est reliée entre les broches 4 et 5 (fig 5.24) du LS124 permet d'avoir la fréquence désirée suivant la formule:

$$f_0 = \frac{1 \times 10^{-4}}{C_{ext} (F)}$$

(MHz)

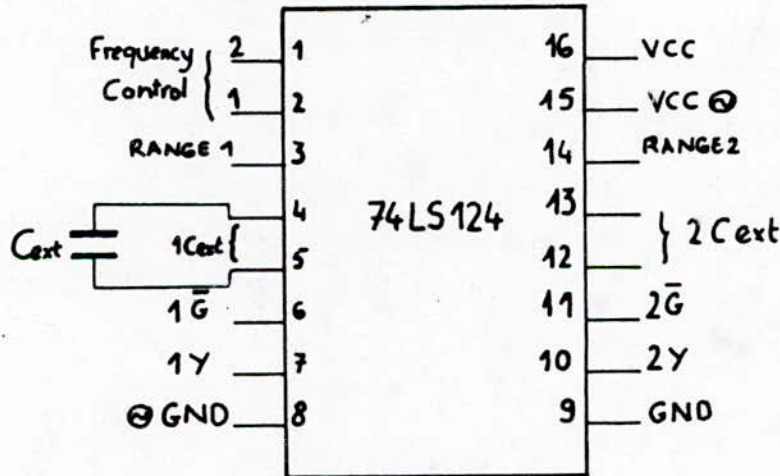


Fig 5.24

* Calcul de la capacité nécessaire pour une fréquence de 25 MHz.

$$f_0 = \frac{10^{-4}}{C_{ext}} = 25 \text{ MHz} \Rightarrow C_{ext} = \frac{10^{-4}}{25 \cdot 10^6} = 4 \text{ pF}$$

Cette capacité peut être obtenue à l'aide d'un condensateur ou d'un Quartz (pour une meilleure stabilité).

5.9. Génération du signal "WRITE" de la mémoire extérieure.

Ce signal annonce le début et la fin du traitement d'une itération.

Le signal "WRITE" est obtenu en retardant le signal READ à l'aide d'un registre à décalage de 14 bits (fig 5.26).

En effet le système comprend 14 étages. Il faut donc 14 cycles d'horloge pour le traitement d'une donnée. soit Δt l'inertie du système.

$$\Delta t = 14 \cdot 40 = 560 \text{ ns.}$$

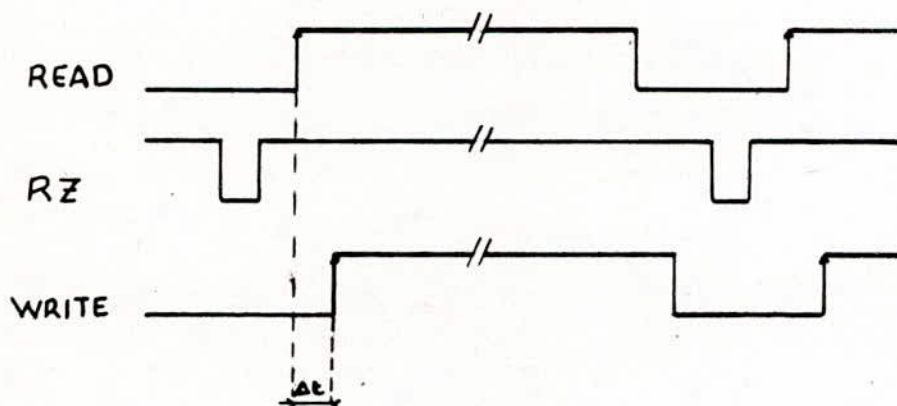



Fig 5.26

5.10. Extension de l'utilisation de l'U.T à un Multiplieur 16x8. bits rapide.

Dans le but d'augmenter la rentabilité de l'U.T, une extension de celle-ci en multiplieur 16x8 est possible à l'aide d'un  multiplexeur de mots (fig 5.25) facilement intégrable.

La sélection est faite à l'aide d'un signal de commande M/ \bar{F} .

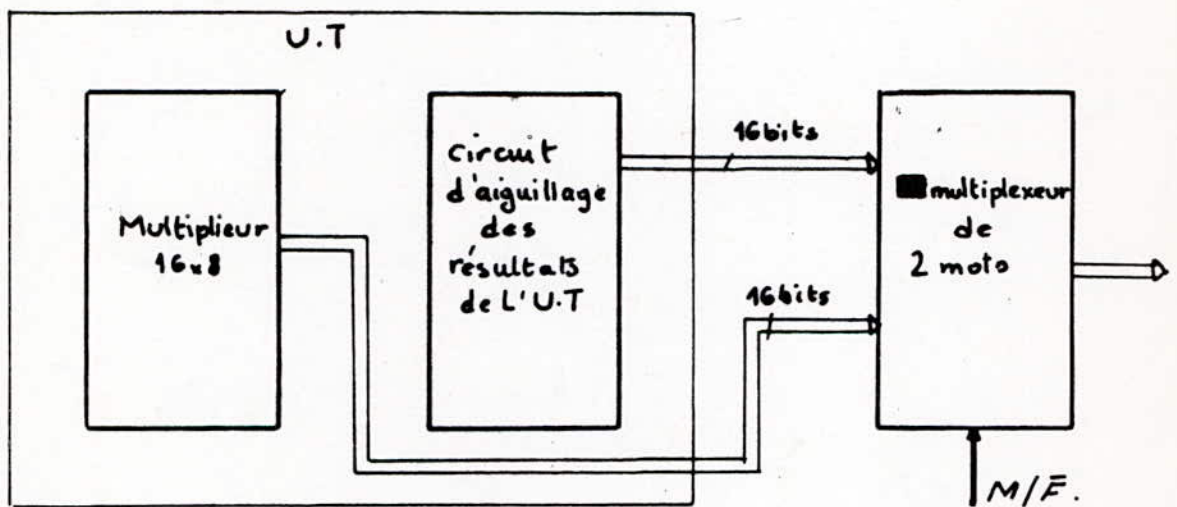


Fig 5.25

CONCLUSION.

Notre objectif principal était la conception d'une U.T rapide tenant compte de la consommation et de la fiabilité.

L'étude d'un algorithme de traitement spécifique facilement adaptable sur une architecture pipe-line, a permis de réaliser un compromis entre la vitesse, la consommation et la fiabilité. L'U.T a donné des satisfactions quant aux résultats obtenus lors des tests.

Toutefois, une amélioration sensible peut être apportée à notre circuit dans l'avenir grâce à l'utilisation des circuits de la technologie TTLALS, surtout quand on sait que cette dernière est appelée à remplacer la TTLLS (du point de vue prix et fonctions). De plus la TTLALS est très utilisée actuellement pour l'intégration en VLSI des systèmes rapides. Ce qui promet des performances très intéressantes une fois notre système totalement intégré.

Cette unité de traitement offre la possibilité d'extension en :

- Multiplicateur 16x8 bits rapide (25 MHz)
- En unité de calcul d'expressions de la forme :

$$S_1 = AB + C \quad \text{ou} \quad S_2 = AB + CD.$$

Le projet de fin d'étude nous a permis de mettre en œuvre les connaissances acquises lors de notre formation et par la même occasion d'avoir une bonne initiation à la vie professionnelle.

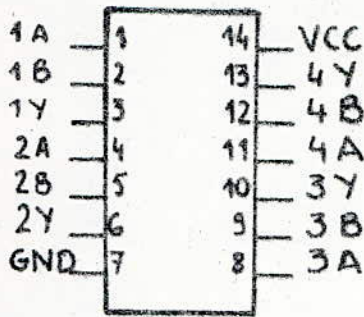
BIBLIOGRAPHIE

- [1] . Aumiaux.M. Logique binaire et ordinateur - Tome 1.
Edition : Masson. - 1974.
- [2] . Brigham. The Fast Fourier Transform.
Edition : Prentice Hall.
- [3] . Lifermann.J. Les méthodes rapides de transformation
de Fourier, Fourier, Walsh, Hadamard, Haar
Edition : Masson. 1980.
- [4] . Lyon.Caen. Circuits Intégrés LSI et MSI
Edition : Masson. 1976.
- [5] . Maurice.B. Multiplieurs Numériques.
Revue Mini.Micro. n° 136 et 188.
- [6] . Mead.conway. Introduction aux Systèmes VLSI.
- [7] . Menadier.J.p. Structure et Fonctionnement des Ordinateurs
Larousse (série informatique)
- [8] . Profit.A. Structure et technologie des ordinateurs.
Arman.Collin.
- [9] . Ramda.M. Mouzali.F. Conception et réalisation d'un
processeur pipe-line de FFT.
Thèse d'ingénieur - janvier 1984.
- [10] Texas Instruments Incorporated : Designing with TTL
Integrated Circuits international
Student edition.
- [11] . Data Book TTL.

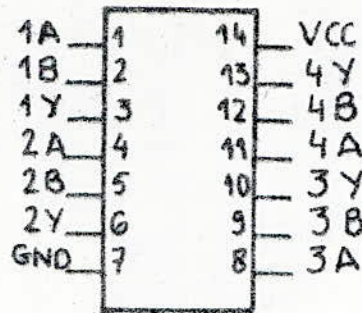
ANNEXE

Designation	quantité	prix unitaire en FF	Prix total en FF
74LS00	1	3.06	3.06
74LS261	16	14	224
74LS 175	1	4.85	4.85
74LS174	59	4.85	286.15
74LS 83	48	5.70	273.6
74LS 124	1	16	16
74LS 14	2	4.51	9.02
74LS244	6	9.16	54.96
74LS240	4	9.16	36.64
74LS373	16	17.98	287.68
74LS164	16	12	192.
74LS265	6	7.71	46.26
74LS08	17	3.23	54.91
74LS 163	1	16	16
74LS 86	1	4.41	4.41
Connecteur 20 pins	1	67.70	67.70
" 50 pins	2	67.70	135.40
" 40 pins	2	67.70	135.40
Support 20 pins	26	12	312
" 16 pins	132	12	1584
" 14 pins	36	12	432
Quartz	1	40	40
-Carte	3	300	900
PRIX TOTAL			5116.04

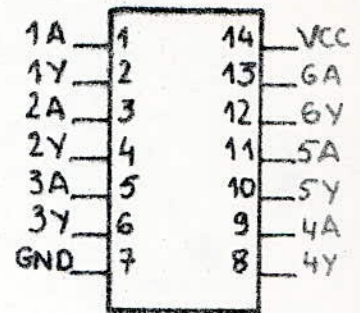
Prix de revient de l'unité de traitement.



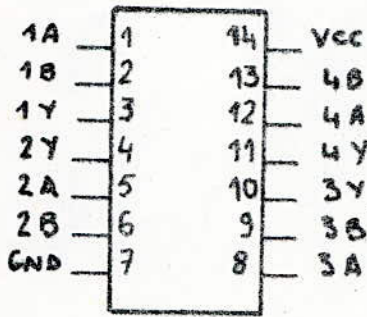
74LS00



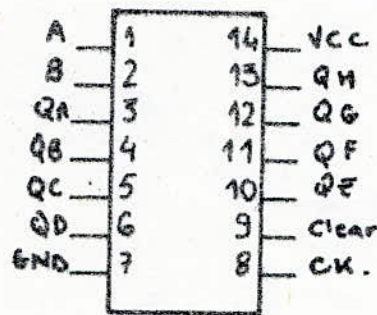
74LS08



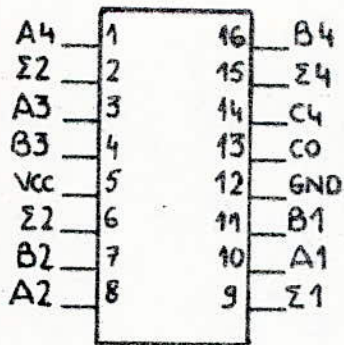
74LS14



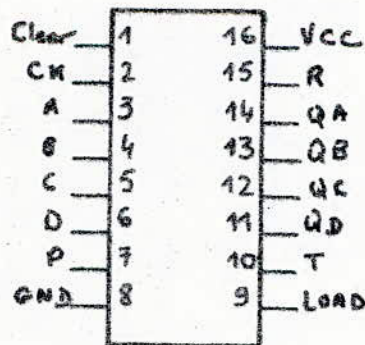
74LS86N



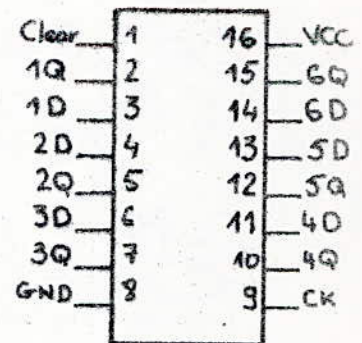
74LS164



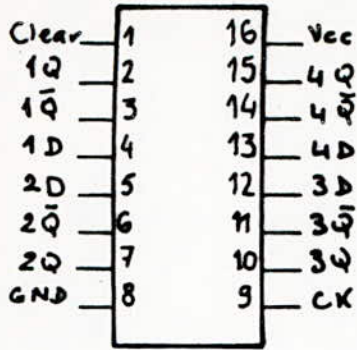
74LS83



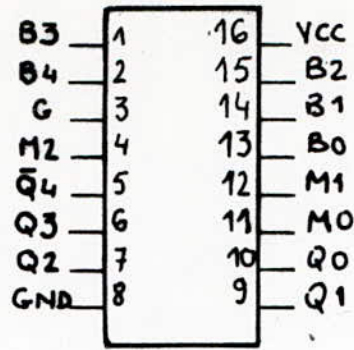
74LS163



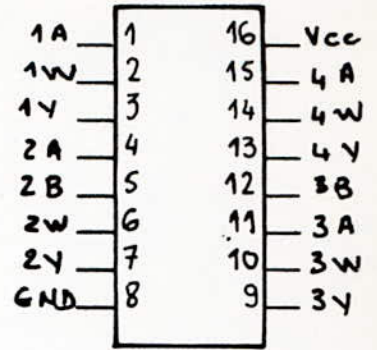
74LS174



74LS175



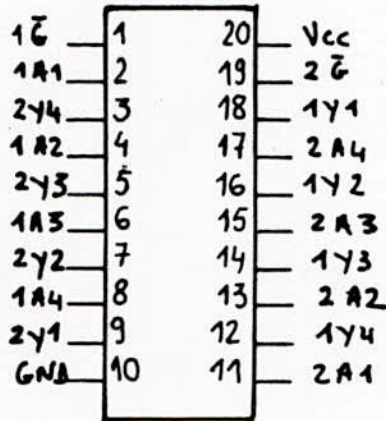
74LS261



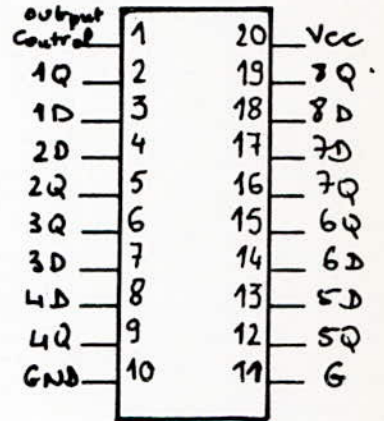
74LS265



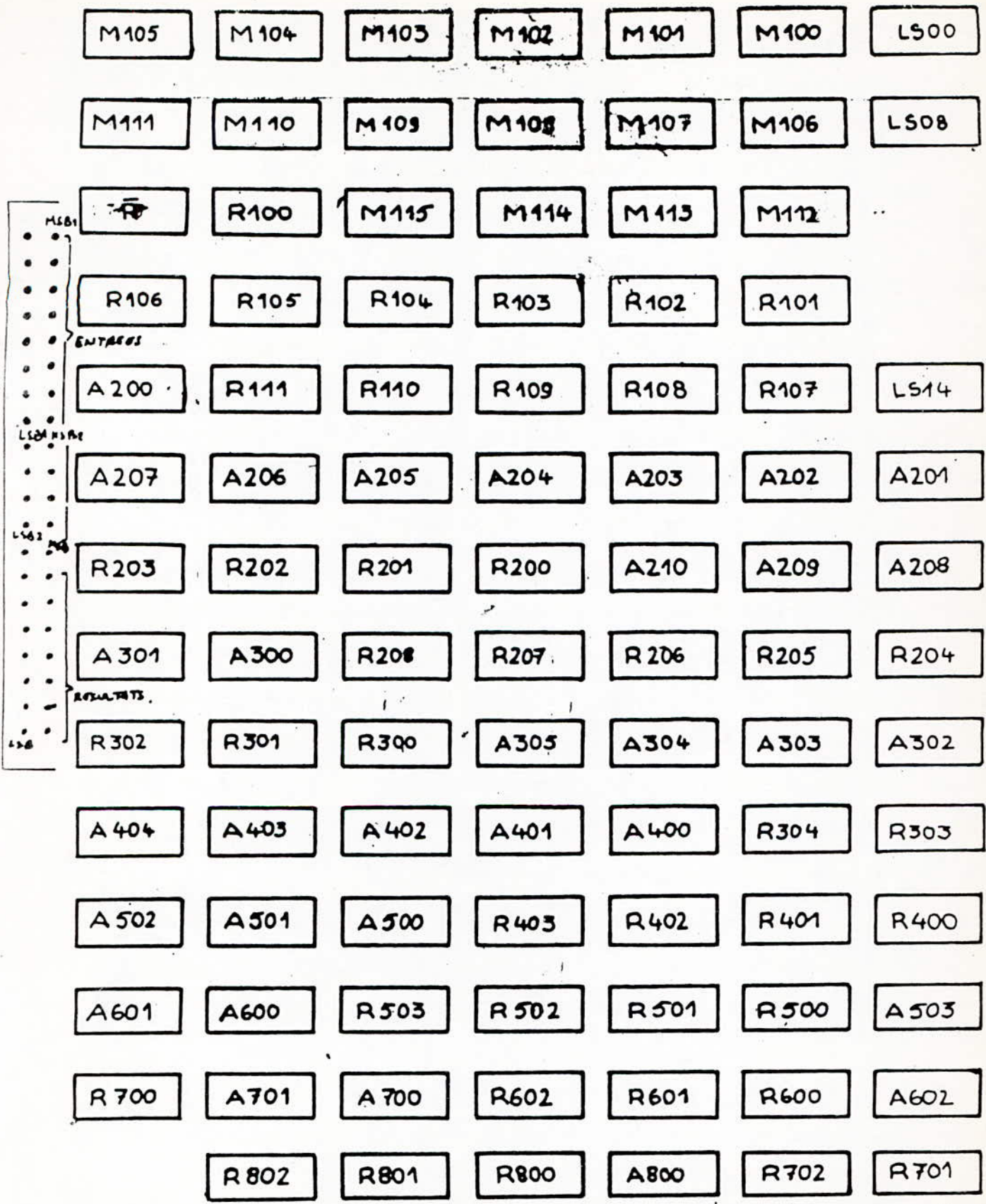
74LS240



74LS244



74LS373



SCHEM D'IMPLANTATION DE LA CARTE MULTIPLEXEUR 16x8



CENTRE DE DEVELOPPEMENT DES TECHNIQUES AVANCEES
LABORATOIRE ARCHITECTURE DES SYSTEMES

UNITE DE TRAITEMENT PIPE-LINE

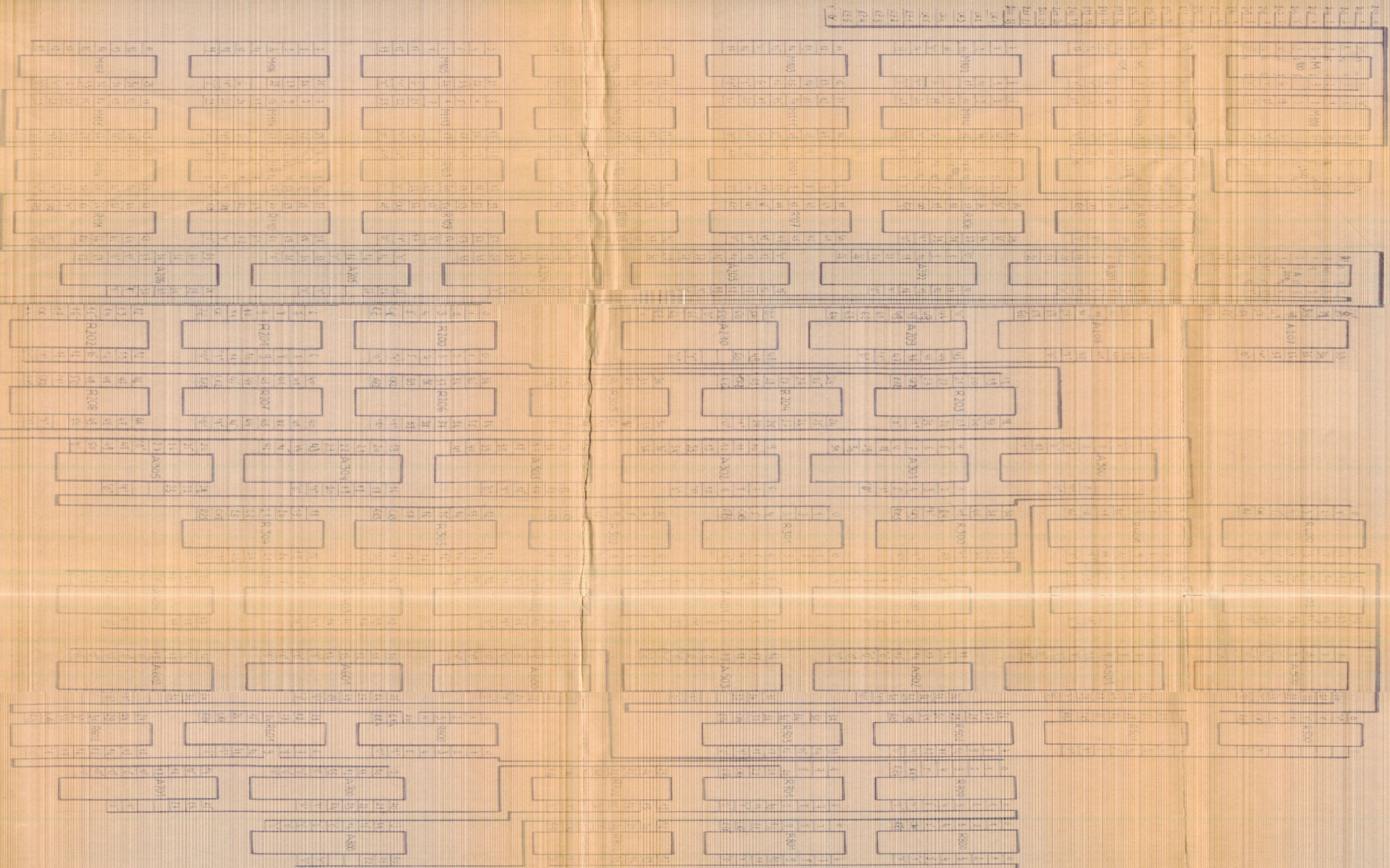
SCHEMA ELECTRIQUE DU CIRCUIT SELECTEUR
ET DES ADDITIONNEURS/SOUSTRACTEURS

PROJET DE FIN D'ETUDES

Proposé par : M^r BESSALAH

Etudié par :
M^r AMROUN A
M^r ADOUJIT M

JANVIER 85



CENTRE DE DEVELOPPEMENT DES TECHNIQUES AVANCEES
LABORATOIRE ARCHITECTURE DES SYSTEMES

UNITE DE TRAITEMENT PIPE-LINE

SCHEMA ELECTRIQUE DV MULTIPLEXIA
1C-5 614

Preparé par M^r BESSALIM H

Etudié par M^r AMROUN A

et M^r ASOUJIT H

PROJET DE FIN DE VOIES

JANVIER 85

0-015
1-015
2-015
3-015
4-015
5-015
6-015
7-015
8-015
9-015
10-015
11-015
12-015
13-015
14-015
15-015