

10/97

*République Algérienne Démocratique et Populaire*  
Ministère de l'enseignement Supérieur et de la Recherche Scientifique

*Ecole Nationale Polytechnique*

*D.E.R de Génie Electrique et Informatique*

*Filière : Electronique*



المدرسة الوطنية للتكنولوجيا  
Ecole Nationale Polytechnique

**PROJET DE FIN D'ETUDES**

*Pour l'obtention du diplôme d'ingénieur d'état  
en Electronique*

**THEME**

**POSITIONNEMENT D'UN HÉLIOSTAT  
PAR CALCUL DE LA POSITION DU  
SOLEIL**

**Proposé et Dirigé par :**

M. HADDADI  
F. KIRAT

**Etudié par :**

DEBECHE Saber  
SALHI Redouane

*Promotion : Septembre 1997*

E.N.P. 10, AVENUE HASSEN BADI - EL-HARRACH - ALGER

*République Algérienne Démocratique et Populaire*  
Ministère de l'enseignement Supérieur et de la Recherche Scientifique

*Ecole Nationale Polytechnique*

*D.E.R de Génie Electrique et Informatique*

*Filière : Electronique*



المدرسة الوطنية للتقنيات  
Ecole Nationale Polytechnique

**PROJET DE FIN D'ETUDES**

*Pour l'obtention du diplôme d'ingénieur d'état  
en Electronique*

**THEME**

**POSITIONNEMENT D'UN HÉLIOSTAT  
PAR CALCUL DE LA POSITION DU  
SOLEIL**

**Proposé et Dirigé par :**

M. HADDADI  
F. KIRAT

**Etudié par :**

DEBECHE Saber  
SALHI Redouane

**Promotion : Septembre 1997**

**E.N.P. 10, AVENUE HASSEN BADI - EL-HARRACH - ALGER**

ملخص:

نقدم في هذا العمل جهازاً أساسه الدارة المدمجة 68705P3. تتمثل وظيفة هذا الجهاز في جعل الاقطة الشمسي يتعقب مسار الشمس لزيادة المردود الطاقوي. يعتمد التحكم في توجيه الاقطة على تعيين إحداثيات الوضعية الظاهرية للشمس بواسطة المعادلات الفلكية.

---

ABSTRACT:

In this work we have presented a system based on the microcontrolor 68705P3. It's task is to improve the energetic rate by making the solar sensor tracking the sun. The solar sensor orientation control is based on the determination of the position coordinates using cosmologic equations.

---

RÉSUMÉ:

Dans ce travail nous présentons un système à base d'un microcontrôleur 68705P3. Ce système permet aux capteurs solaires de suivre le déplacement du soleil, et ce pour l'augmentation du rendement énergétique. La commande de l'orientation du capteur est basée sur la détermination des coordonnées de la position apparente du soleil à l'aide des équations cosmologiques.

## DÉDICACES

*Je dédie ce modeste travail à :*



- Mes très chères parents que dieu le tout puissant les récompense.
- Mes très chères grands parents, pour leur tendresse, bonté, et compréhension . Que ce mémoire soit l'expression de la profonde affection que je leur porte .
- Mes soeurs Nabila et Lamia.
- Mon cher frère Adel .
- Tous ceux que j'aime .

*Saber*

*Je dédie ce modeste travail à :*

- Mes très chères parents .
- Mes très chères grands parents .
- Mes soeurs et frères .
- Tous mes amis .
- toute la famille .

*Redouane*

## REMERCIEMENTS

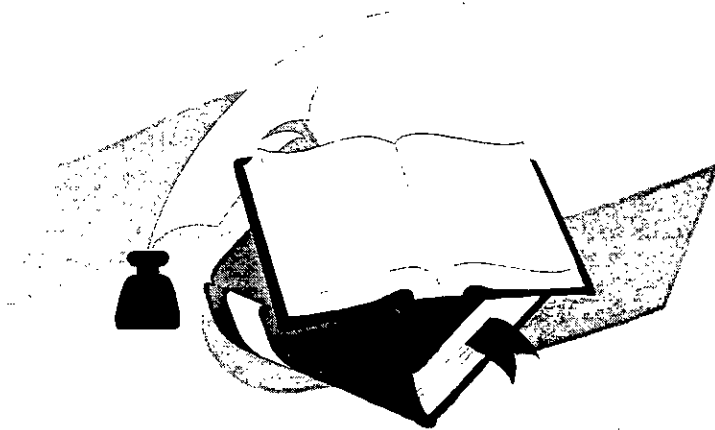


Notre gratitude et nos plus sincères remerciements vont tout d'abords, au Docteur M. HADDADI dont la bienveillante direction et les précieuses orientations nous ont constamment accompagnée et soutenue durant le développement de cette étude.

Notre remerciement vont aussi, à Monsieur F.KIRAT, pour ces conseils avisés.

Nous tenons également à remercier Madame HAMMAMI pour son aide précieuse.

Nous remercions enfin tous ceux qui ont contribué à l'élaboration de ce travail.



# Sommaire

## INTRODUCTION

1

### Chapitre I

#### DESCRIPTION GENERALE DU SYSTEME

1	BLOC DE CALCUL ET DE GESTION	2
2	OSCILLATEUR SINUSOÏDAL BASSE FREQUENCE	2
3	CONVERSION ANALOGIQUE NUMERIQUE	3
4	LE CLAVIER	3
5	L'AFFICHAGE	3
6	LE BLOC DE PUISSANCE	3

### Chapitre II

#### CALCUL DE LA POSITION DU SOLEIL

1	INTRODUCTION	4
2	LE TEMPS	5
3	LONGITUDE DU SOLEIL	6
4	HAUTEUR DROITE ET DECLINAISON	6
5	LE TEMPS SIDERAL	7
6	L'ELEVATION LOCALE ET L'AZIMUT	8

### Chapitre III

#### OSCILLATEUR SINUSOÏDAL BASSE FREQUENCE

1	INTRODUCTION	10
2	L'OSCILLATEUR A QUARTZ DE 4.43 MHZ	10
3	DIVISEUR DE FREQUENCE	11
4	LE FILTRE PASSE BAS	11

### Chapitre IV

#### LA CONVERSION ANALOGIQUE-NUMERIQUE

1	CONVERTISSEURS ANALOGIQUES-NUMERIQUES	15
1.1	Convertisseur à double rampe	15
1.2	Convertisseur à approximations successives	17
1.3	Convertisseur à comptage	18
1.4	Convertisseur parallèle	19
2	L'ADC 0804	20

### Chapitre V

#### BLOC DE CALCUL ET DE GESTION

1	LE MICROCONTROLEUR 68705P3	24
1.1	L'unité centrale	24
1.2	Les entrées/sorties parallèles	25
1.3	Organisation de la mémoire du 68705P3	26
1.4	Mode d'adressage	27
1.4.1	<i>L'adressage inhérent</i>	27
1.4.2	<i>L'adressage immédiat</i>	27
1.4.3	<i>L'adressage direct</i>	27
1.4.4	<i>L'adressage étendu</i>	27

1.4.5	<i>L'adressage indexé</i>	27
1.4.6	<i>L'adressage relatif</i>	28
1.4.7	<i>L'adressage bit ou manipulation de bit</i>	29
2	DESCRIPTION TECHNIQUE	30

## Chapitre VI

### LE CLAVIER HEXADECIMAL

## Chapitre VII

### BLOC DE PUISSANCE

1	LE MOTEUR PAS-A-PAS	36
1.1	Définition	36
1.2	Classification des moteurs pas-à-pas	36
1.2.1	<i>Moteur pas-à-pas à réluctance variable</i>	37
1.2.2	<i>Moteur pas-à-pas à aimant permanent</i>	39
1.2.3	<i>Moteur pas-à-pas hybride</i>	39
1.3	Alimentation	40
1.4	Amélioration des performances aux démarrages	43
1.5	Accélération et décélération	43
1.6	Mode d'excitation	44
1.6.1	<i>Mode d'excitation monophasé</i>	44
1.6.2	<i>Mode d'excitation double phase</i>	45
1.6.3	<i>Mode d'excitation à demi-pas</i>	45
2	CONCEPTION DES L'ETAGE DE PUISSANCE	46
3	FONCTIONNEMENT DE L'ETAGE DE PUISSANCE	47
4	CALCUL PRATIQUE	47

## Chapitre VIII

### PROGRAMME DE CALCUL ET DE GESTION DU SYSTEME

1	CALCUL DE LA POSITION OPTIMALE	50
1.1	Représentation des nombres réels	50
1.2	L'addition des nombres réels	50
1.3	Le complément à deux	51
1.4	La soustraction des nombres réels	53
1.5	La conversion BCD-Héxadécimal	53
1.6	La conversion Héxadécimal-BCD	54
1.7	La multiplication des nombres réels signés	55
1.8	La division des nombres réels signés	58
1.9	La fonction trigonométrique sinus	61
1.10	La fonction trigonométrique cosinus	63
1.11	La fonction trigonométrique arctangente	65
1.12	L'algorithme de calcul de la position optimale	67
2	COMMANDE DU MOTEUR PAS-A-PAS	68
2.1	Détermination du nombre exact de pas	68
2.2	La commande du moteur	70

**CONCLUSION** 72

**ANNEXES** 73

## INTRODUCTION



Le soleil a probablement depuis les temps immémoriaux suscité chez l'humain les sentiments les plus nuancés.

L'étonnement, la curiosité, le respect voire l'idolâtrie ont, sans doute, accompagné les réflexions à son propos. Ce qui était et reste encore indéniablement admis par tous, puisque mettant en jeu les facultés dont est doté l'humain que représentent la sensation de la chaleur et de la vision, est le fait que notre astre du jour est essentiellement source de lumière et surtout de chaleur.

Ces premières constatations, transposées dans un esprit plus évolué dans sa dialectique, génèrent un désir d'exploitation de la dite chaleur; ce qui fut vraisemblablement le cas chez l'élite scientifique des différents civilisations qu'a connus l'histoire. Mais, faute de moyens techniques, les savants n'ont pas pu arriver à leurs fins, sinon imparfaitement.

Le vingtième siècle est connu pour la somme gigantesque de savoir accumulée dans les différents domaines d'activités couverts par ce qu'on nomme prosaïquement la science.

Les techniques solaires étant partie intégrante de cette science, elles ont bénéficié d'un essor tout aussi important.

C'est ainsi qu'ont vu le jour tous les procédés, dénommés classiques exploitant la formidable énergie solaire.

Pour profiter pleinement de cette énergie, les scientifiques doivent encore résoudre un nombre de problèmes, essentiellement liés au rendement relativement faible des cellules photovoltaïques (de l'ordre de 15%) [16] et aux pertes de positionnement des panneaux.

Ces capteurs nécessitent généralement d'être orientés de telle sorte que le rayonnement solaire concentré se focalise en un lieu où se fixe le capteur du système de conversion de l'énergie solaire. Ce problème fait appel à des appareillages de poursuites du soleil ou héliostats. En effet, un capteur plan de cellules photovoltaïques muni d'un système d'orientation est environ 40 à 43% plus rentable que s'il était fixe [16].

Cependant, cette idée se trouve confrontée à un problème de rentabilité du système car en pratique cela pourrait mettre en jeu des puissances pouvant dépasser celles collectées par les panneaux.

Une alternative viserait à obtenir un compromis entre un système statique et un système de poursuite. Celle-ci serait de positionner un panneau suivant un angle optimal présentant un taux d'ensoleillement journalier maximal. Le calcul de cette position est effectué à partir d'équations cosmologiques qui prennent en compte la date et le lieu d'installation du panneau.

Le travail qui nous a été confié consiste en l'étude et la réalisation d'un système basé sur le procédé ainsi décrit, géré par un microcontrôleur 68705P3 de Motorola.



**Chapitre (I)****DESCRIPTION GENERALE DU SYSTEME.**

Notre réalisation consiste à commander un panneau solaire suivant un angle optimal. Le synoptique général de ce système est présenté sur la figure(I-1).

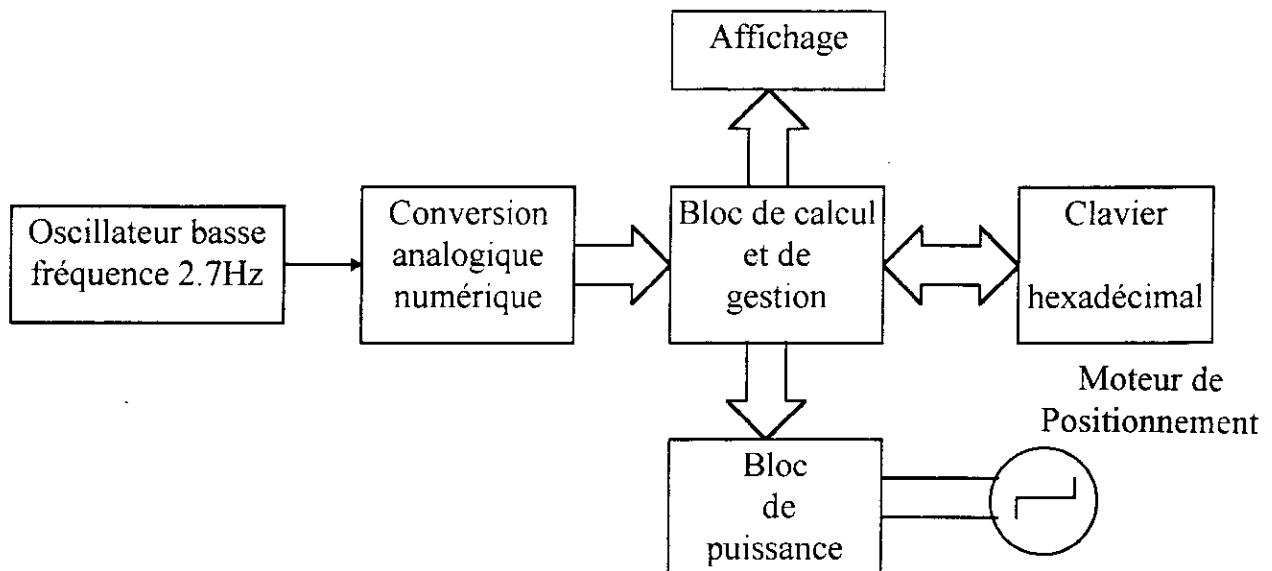


Fig (I-1) : Synoptique générale du système.

**1. BLOC DE CALCUL ET DE GESTION :**

Ce bloc est réalisé à l'aide d'un microcontrôleur 68705P3. C'est la partie principale du système qui effectue la gestion des ressources externes et les calculs nécessaires pour déterminer la position du soleil. Après la détermination de la position optimale, le microcontrôleur commande le panneau solaire à l'aide d'un moteur pas à pas.

**2. OSCILLATEUR SINUSOÏDAL BASSE FREQUENCE :**

Dans notre réalisation, on a besoin des fonctions trigonométriques sinus et cosinus. Mais comme les opérations réalisées par l'unité centrale de ce microcontrôleur s'arrêtent à l'addition binaire, la soustraction binaire et quelques fonctions de décalage et rotatif, il faut réaliser ces fonctions trigonométriques par des programmes. Dans ce cas, le problème qui apparaît est que ces fonctions consomment trop d'espace mémoire, et comme la capacité du microcontrôleur utilisé dans notre réalisation est limitée (inférieure à 2k octets) la solution proposée pour résoudre ce problème est d'utiliser un oscillateur sinusoïdal pour calculer ces fonctions.

Le principe de cette méthode consiste à assimiler la période du signal sinusoïdal divisée par 360 ( $T/360$ ) à l'équivalent d'un degré. Par exemple pour déterminer le sinus d'un angle  $\alpha$  donné, il faut créer une temporisation de durée  $\alpha.T/360$  à partir de l'instant où le signal sinusoïdal passe par zéro dans l'alternance positive. A la fin de cette temporisation on génère l'ordre de conversion dirigé vers le CAN. Celui-ci fournit le code de la tension à cet instant. La mise en forme de ce code va nous donner le sinus de l'angle  $\alpha$ .

### **3. CONVERSION ANALOGIQUE NUMERIQUE :**

la tension délivrée par l'oscillateur est un signal analogique alors que le microcontrôleur est un système numérique, donc cette tension devra être convertie en numérique. Pour cela on utilise un convertisseur analogique numérique.

### **4. LE CLAVIER :**

La position du soleil est fonction de la longitude et de la date. A la mise sous tension du système ces données doivent être transmises au microcontrôleur. Pour réaliser cela on a utilisé un clavier binaire.

### **5. L'AFFICHAGE :**

Pour que l'utilisateur du système puisse contrôler la touche enfoncée, on a utilisé des afficheurs sept segments. A chaque appui sur une touche le code correspondant est affiché sur un afficheur sept segment. En plus après le calcul de la position optimale le microcontrôleur affiche cet angle.

### **6. LE BLOC DE PUISSANCE :**

Les signaux issus du port d'Entrée/Sortie du microcontrôleur sont insuffisants pour commander un moteur pas à pas. Pour cela, il faut insérer un étage amplificateur entre le microcontrôleur et le moteur.

**Chapitre ( II )****CALCUL DE LA POSITION DU SOLEIL****1. INTRODUCTION: [ 5 ]**

Le plan équatorial de la terre fait un angle de  $23,5^\circ$  avec le plan de l'orbite de la terre autour du soleil, comme représenté sur la figure(II.1). C'est cette déclinaison qui est la cause de la variation de l'élévation du soleil d'un bout à l'autre à midi durant l'année, engendre les saisons, et complique le calcul de la position du soleil pour un temps donné.

La figure(II.1) pourrait être utilisée pour nous aider à visualiser comment on doit déterminer l'emplacement du soleil dans le ciel. Cependant ce n'est pas le meilleur point de départ pour appréhender le problème. Parce que en regardant le ciel, on a tendance à croire que la terre est fixe, et le soleil, la lune, les planètes et les étoiles tournent autour de la terre. Imaginons maintenant la sphère céleste ayant un rayon virtuel infini avec la terre fixé à son centre comme représenté sur la figure(II.1).

Plutôt que d'avoir la terre tourner sur son axe, laissant la sphère céleste tourner autour de la terre.

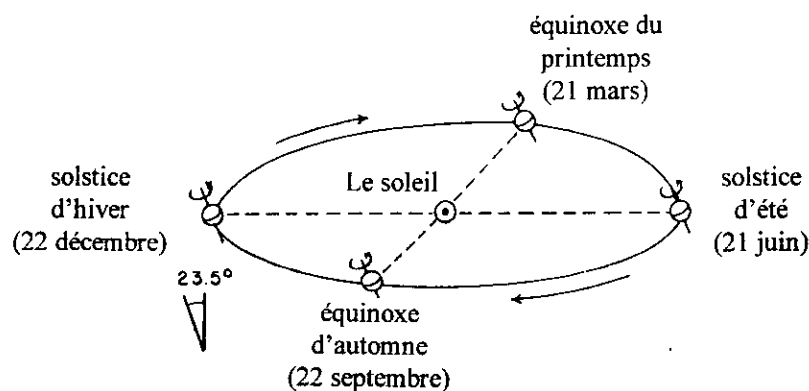


Fig (II-1) : L'orbite de la terre autour du soleil.

L'intersection du plan équatorial de la terre avec la sphère céleste est appelé l'équateur céleste. Le mouvement de la terre autour du soleil [représenté sur la figure(II.1)] est représenté sur la sphère céleste de la figure( II.2 ) par une orbite circulaire appelée l'ellipse, laquelle est approximativement inclinée de  $23,5^\circ$  avec l'équateur céleste. Ainsi le soleil fait une révolution à travers l'ellipse par an dans la direction opposée à la direction du rotation de la sphère céleste due au mouvement de la terre autour du soleil.

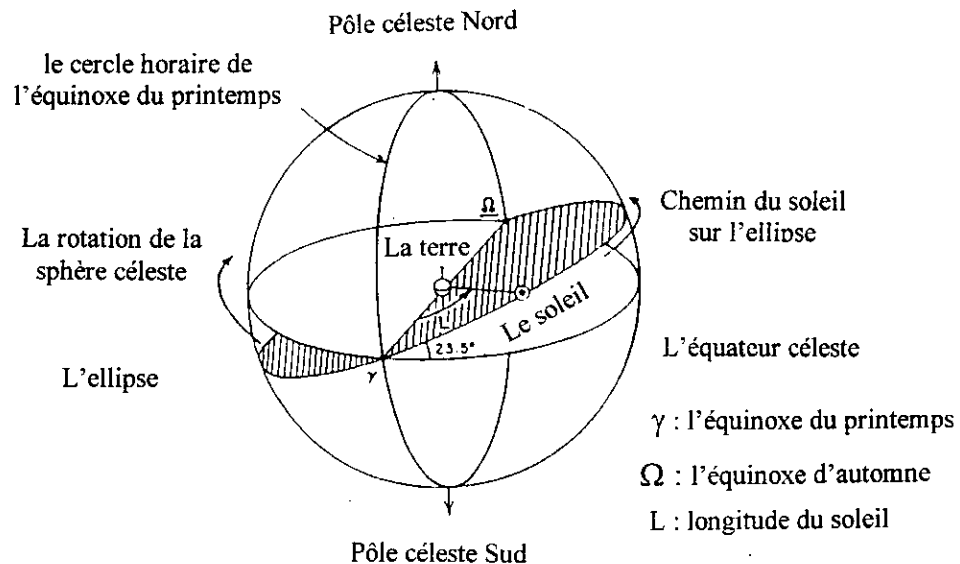


Fig (II.2) : La sphère céleste.

Le soleil traverse l'équateur céleste à l'équinoxe du printemps le 21 Mars et à l'équinoxe d'automne le 23 Septembre. le soleil est à sa plus grande distance de l'équateur céleste en solstice d'été en 21 Juin et en solstice d'hiver en 22 Décembre. Le plus grand cercle qui passe à travers le céleste du pole nord et l'équinoxe du printemps est appelé le cercle horaire de l'équinoxe du printemps.

## 2. LE TEMPS: [ 5 ]

Dans les paragraphes qui suivent, le temps est compté en jours a partir de 1 Janvier 1980 à l'heure moyenne de Greenwich.

L'algorithme suivant nous donne le temps correct, utilisé dans les équations qui suivent.

$$\Delta = \text{année} - 1980$$

$$\text{LEAP} = \Delta / 4 \text{ ( Mod } 4 \text{ )}$$

$$\text{Temps} = \Delta \times 365 + \text{LEAP} + \text{Jour} - 1 + (\text{Heure} + (\text{Minute} + \text{Second} / 60) / 60) \times 24$$

$$\text{si ( } \Delta = \text{LEAP} \times 4 \text{ ) Temps} = \text{Temps} - 1$$

$$\text{si ( } \Delta < 0 \text{ et ( } \Delta \neq \text{LEAP} \times 4 \text{ ) ) Temps} = \text{Temps} - 1$$

où **Jour** est le nombre de jours compté à partir du 1<sup>er</sup> Janvier, exception pour l'année bissextile ou il faut retrancher 1 du nombre de jours s'il est avant le 1 Mars.

### 3. LONGITUDE DU SOLEIL: [ 5 ]

La position du soleil dans la sphère céleste peut être obtenue en spécifiant seulement un paramètre, la longitude  $L$ , comme représentée sur la figure ( II.2 ). Puisque l'orbite de la terre autour du soleil n'est pas circulaire, la vitesse de déplacement du soleil sur l'orbite elliptique varie légèrement dans l'année, ainsi l'expression de la longitude est quelque peu compliquée. La longitude du soleil en radian est donnée par les équations suivantes :

$$\theta = 2 \pi \times \text{Temps} / 365.25$$

$$g = -0.031271 - (4.53963 \times 10^{-7}) \times \text{Temps} + \theta$$

$$L = 4.900968 + (3.67474 \times 10^{-7}) \times \text{Temps} + (0.033434 - 2.3 \times 10^{-9} \times \text{Temps}) \times \sin(g) + 0.00349 \times \sin(2.g) + \theta$$

$g$  étant appelé l'anomalie moyenne de la terre.

### 4. HAUTEUR DROITE ET DECLINAISONS: [ 5 ]

Un autre moyen d'exprimer la position du soleil sur la sphère céleste est en terme de la hauteur droite  $\alpha$  et de la déclinaison  $\delta$  représentées sur la figure( II.3.a ). La hauteur droite est l'angle entre le cercle horaire de l'équinoxe du printemps et le cercle horaire du soleil mesuré à partir de l'équinoxe de printemps dans la direction de l'est.

La déclinaison est la distance angulaire à partir du soleil vers l'équateur céleste le long du cercle horaire. La déclinaison est positive si le soleil est dans l'hémisphère nord de la sphère céleste et négative si le soleil est dans l'hémisphère sud.

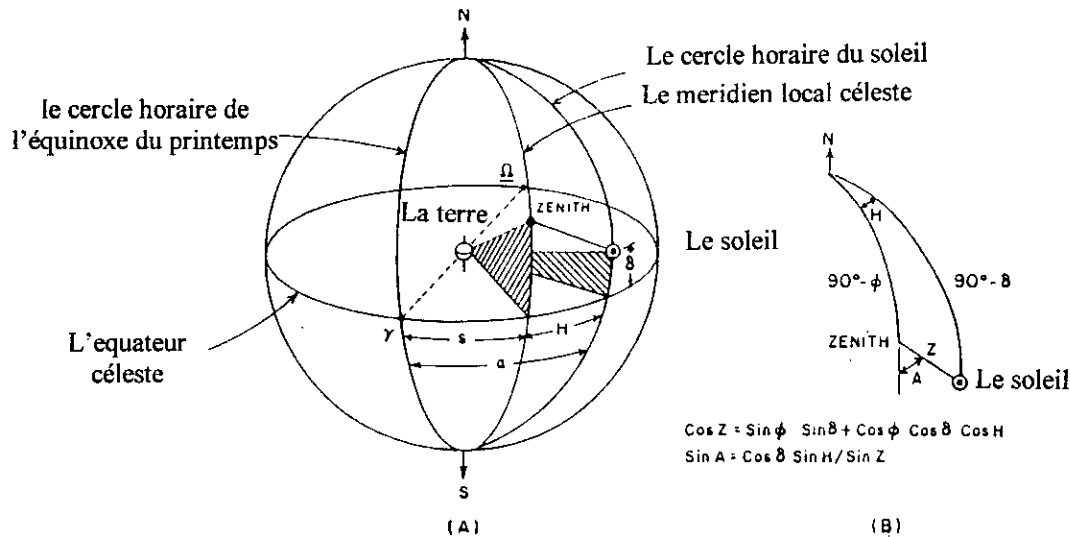
En utilisant le cercle trigonométrique on peut démontrer que la hauteur droite et la déclinaison du soleil sont reliés à la longitude du soleil par la relation:

$$\text{tg}(\alpha) = \cos(\varepsilon) \times \text{tg}(L)$$

$$\sin(\delta) = \sin(\varepsilon) \times \sin(L)$$

avec  $\varepsilon$  est l'angle entre le plan de l'ellipse et le plan de l'équateur céleste, la valeur précise de  $\varepsilon$  est :

$$\varepsilon = 23.442 - (3.56^\circ \times 10^{-7}) \times \text{temps}$$



Fig(II.3): (a): La sphère céleste qui montre les définitions de  $\delta$ , S, H et  $\alpha$ .  
 (b):Le triangle sphérique de la sphère céleste définit la distance zénith z et l'angle azimutal A,  $\phi$  est l'altitude locale.

### 5. LE TEMPS SIDERAL: [ 5 ]

Pour déterminer la position du soleil dans le ciel pour un temps donné, il est nécessaire de connaître non seulement la position du soleil dans la sphère céleste, mais aussi la position de la sphère céleste par rapport à la terre à cet instant. La rotation de la sphère céleste est mesurée dans le temps sidéral relatif au méridien local céleste.

Le méridien local céleste, comme représenté sur la figure (II-3), est le plus grand cercle traversant le pôle céleste et le zénith ( le point sur la sphère céleste directement au dessus d'un observateur sur la terre ).

Le jour sidéral est le temps entre deux transitions successives du cercle horaire de l'équinoxe de printemps à travers le méridien céleste.

L'heure sidérale est de zéro à l'instant où le cercle horaire de l'équinoxe de printemps traverse le méridien local céleste. Ainsi la localisation dans la terre avec différentes longitude nous donne différents temps sidéraux à un instant donné.

Le temps sidéral dans l'heure de Greenwich à 0 h du temps universel de Greenwich ( GUT ) est donné par :

$$ST = 6.720265 + 24 ( \text{Temps} / 365.25 - ( \text{Année} - 1980 ) ) + 0.000001411 \times \text{Temps}$$

Le temps local sidéral à 0 h GUT est obtenu par soustraction de ST la longitude local LONG en heures ouest de Greenwich ( 1 hr = 15° ).

Nous voyons que la sphère céleste tourne autour de la terre de manière légèrement plus rapide que le soleil. En fait :

$$1 \text{ jour standard} = 1.0027379 \text{ jours sidéral.}$$

En outre le temps standard local ( LST ) est relié à ( GUT ) par la relation :

$$GUT = LST + Zone - C$$

où Zone est le temps de Zone internationale donné dans le tableau (II.1) et C est égale à zéro sauf s'il fait jour dans laquelle est égale à un.

Le temps local sidéral dans n'importe quel LST est par conséquent égal au temps local sidéral à 0 h plus  $1.0027379 \times (LST + Zone - C)$ .

En combinant l'effet de la longitude local et du LST, on obtient l'expression suivante du temps sidéral :

$$S = ST - LONG + 1.0027379 \times (LST + Zone - C)$$

## 6. L'ELEVATION LOCALE ET L'AZIMUT: [ 5 ]

Quand on a les paramètres suivants : le temps sidéral local, la hauteur droite, la déclinaison du soleil, et l'altitude locale, alors l'azimut et l'élévation locale du soleil peuvent être déterminés à partir de relations trigonométriques. L'angle azimutal local A et la distance zénith sont définis comme indiqué sur la figure (II.3.b)

L'angle azimutal local du soleil est l'angle entre le méridien local et la ligne entre le zénith et le soleil. Par convention l'angle azimutal local est positif quand le soleil est à l'est ou au sud. La distance zénith Z est la distance angulaire à partir du zénith au soleil. L'élévation locale E du soleil est alors:

$$E = 90^\circ - Z$$

L'expression désirée de l'azimut local et l'élévation sorte de relations trigonométriques représentées sur la figure (II.3.b) où  $\phi$  est l'altitude locale ( entre la distance angulaire du zénith vers l'équateur céleste ), et l'angle horaire H, cette expression est simplement la hauteur droite moins le temps sidéral, comme représenter sur la figure (II.3.a)

$$h = \alpha - S$$

Les transformées trigonométriques dans la figure (II.3.b) donne :

$$\cos(Z) = \sin(\phi) \sin(\delta) + \cos(\phi) \cos(\delta) \cos(H)$$

$$\sin(A) = \cos(\delta) \sin(H) / \sin(Z)$$

Un problème peut survenir quand A est calculé, la fonction Arcsin donne des valeurs qui sont par convention dans l'intervalle de  $-90^\circ$  à  $+90^\circ$ . Si A est en dehors de cette intervalle, elle doit être corrigée.

L'algorithme de correction de A est par conséquent :

si  $\cos(Z) < \sin(\delta) / \sin(\phi)$  donc      si  $A < 0$  donc  $A = A + 360^\circ$ .

$$A = 180^\circ - A.$$

Noter que si la déclinaison solaire est plus grande que l'altitude locale, cette correction doit se faire pour toutes les valeurs de A.

zone de temps	Temps de zone (hr)
Temps standard Atlantique	4
Temps standard Estern	5
Temps standard Central	6
Temps standard Mountain	7
Temps standard Pacifique	8
Temps standard Hawaii	10

Tableau (II.1)

Les temps de zone international pour plusieurs zones de temps.



## **Chapitre (III)**

# **L'OSCILLATEUR SINUSOIDAL BASSE FREQUENCE**

## **1. INTRODUCTION :**

Dans notre réalisation, on a besoin d'un oscillateur sinusoïdal basse fréquence. Comme on utilise cet oscillateur pour calculer les fonctions trigonométriques sinus et cosinus, la fréquence et l'amplitude de cet oscillateur devront être aussi stable que possible.

Le moyen le plus simple pour réaliser un oscillateur de très grande stabilité en fréquence est d'utiliser un quartz pour piloter l'oscillateur. Dans ce cas, la fréquence d'oscillation se trouve en quelque sorte asservie à la fréquence de résonance mécanique du cristal.

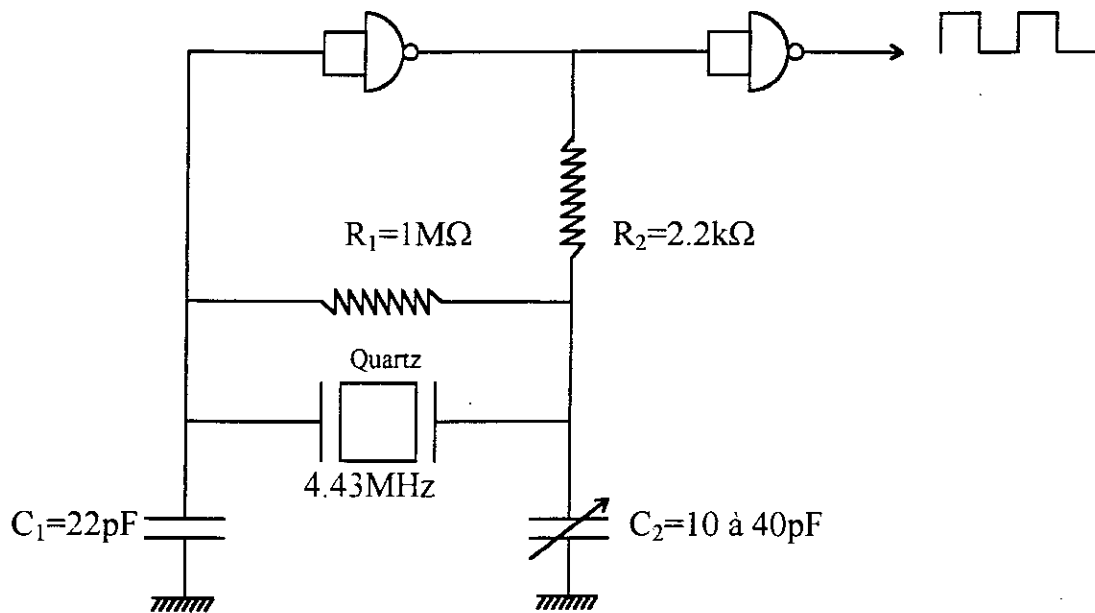
Le fonctionnement d'un cristal de quartz utilise le principe de la piezoélectricité. Toute déformation mécanique d'un cristal provoque l'apparition d'un champ électrique. Un cristal soumis à un champ électrique se déforme mécaniquement. Si le champ électrique en question est alternatif, les déformations mécaniques vont se transformer en oscillations mécaniques de même fréquence que le champ électrique appliqué. Lorsque la fréquence de la tension d'excitation s'approche de la fréquence de résonance mécanique du quartz, l'amplitude des oscillations mécaniques de ce dernier augmente. Ce qui conduit à un accroissement du courant dans le circuit d'excitation avec un maximum à la fréquence de résonance du quartz.

Comme la majorité des oscillateurs à quartz ne fonctionnent que dans la gamme des MHz, la solution la plus simple pour réaliser un oscillateur sinusoïdal basse fréquence très stable sera d'utiliser un oscillateur à quartz à haute fréquence puis le signal résultant sera amené en basse fréquence par des diviseurs de fréquence. Comme la sortie du diviseur est un signal carré, il faudra utiliser un filtre passe bas pour éliminer les harmoniques supérieures au fondamentale.

## **2. L'OSCILLATEUR A QUARTZ DE 4.43 MHZ :**

On a choisi un quartz de 4.43 MHz car par une division de  $2^{14}$  puis par deux fois par 10 on trouve le signal carré de 2.7 Hz (360ms) qui convient bien à notre application.

Pour réaliser cet oscillateur on utilise des portes NAND (4011). Le schéma électrique de cet oscillateur est indiqué sur la figure (III-1).



Fig(III-1) : Oscillateur 4,43MHz à quartz.

### 3. DIVISEUR DE FREQUENCE :

Pour ramener le signal carré de 4.43 Mhz à 2.7 Hz on a utilisé 3 diviseurs de fréquence :

- Le premier est le diviseur 4020 réalisé en technologie CMOS et effectue une division jusqu'à  $2^{14}$ .
- Les deux autres diviseurs sont à base du circuit 74LS90 réalisé en technologie TTL et qui effectue une division par 10.

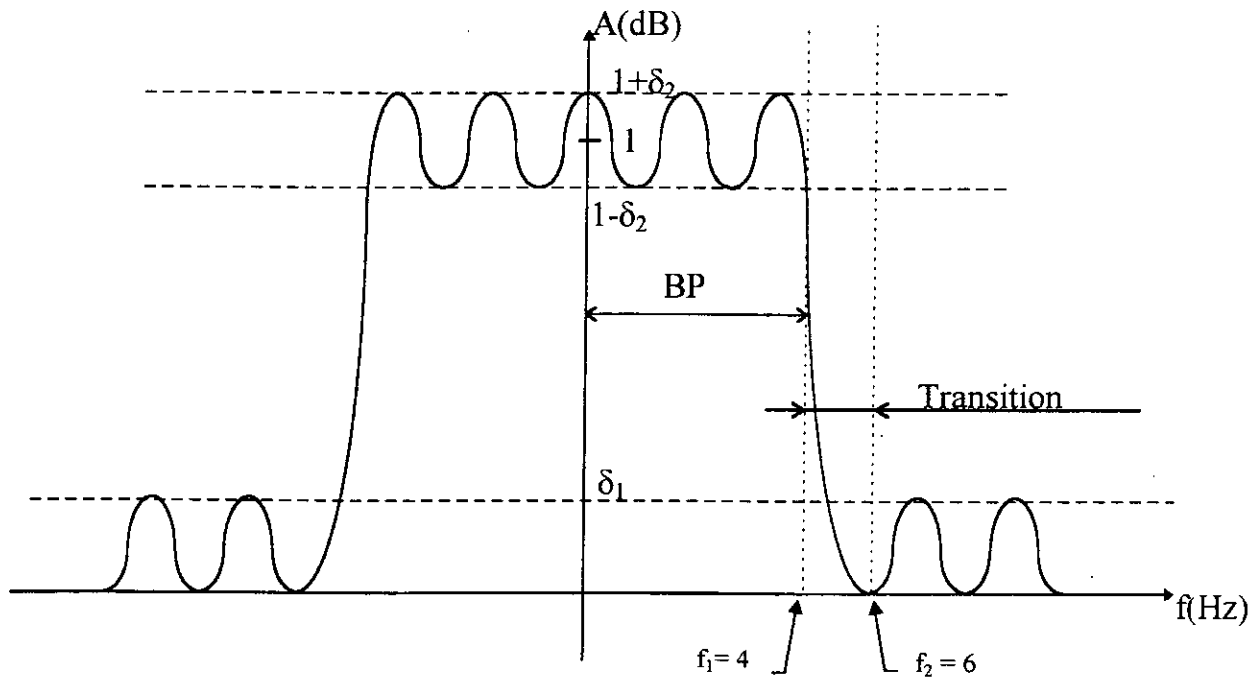
### 4. LE FILTRE PASSE BAS : [ 10 ]

Le signal issu du diviseur de fréquence est un signal carré de fréquence 2.7 Hz (360ms). Son développement en série de Fourier est :

$$x(t) = \frac{5}{2} + \frac{10}{\pi} \sum_{n=1}^{+\infty} \frac{(-1)^{n-1}}{(2n-1)} \cos(2n-1)\omega t.$$

Pour éliminer les harmoniques du signal carré la solution qui vient à l'esprit est l'utilisation d'un filtre sélectif accordé sur la fréquence du fondamental, mais, comme on est en basse fréquence ce type de filtre nécessite l'utilisation de bobines et de capacités pratiquement irréalisables. Donc la solution la plus simple est d'utiliser un filtre actif passe bas qui ne laisse passer que le fondamental.

La courbe d'affaiblissement devra être située dans le gabarit représenté sur la figure (III-2).



Fig(III-2) : Gabarit du filtre actif.

Dans notre réalisation, on utilise un filtre passe bas de Tchebycheff d'ordre 4 qui est très sélectif, et qui possède un affaiblissement minimal de 30db.

A partir des polynômes de Tchebycheff, la fonction de transmission  $f_{TR}(p)$  correspondant à  $N=4$  est :

$$f_{TR}(p) = (1.0137 p^2 + 0.28289 p + 1)(3.5791 p^2 + 2.4114 p + 1) \quad (III-1)$$

La fonction de transfert  $f_T(p)$  est donc :

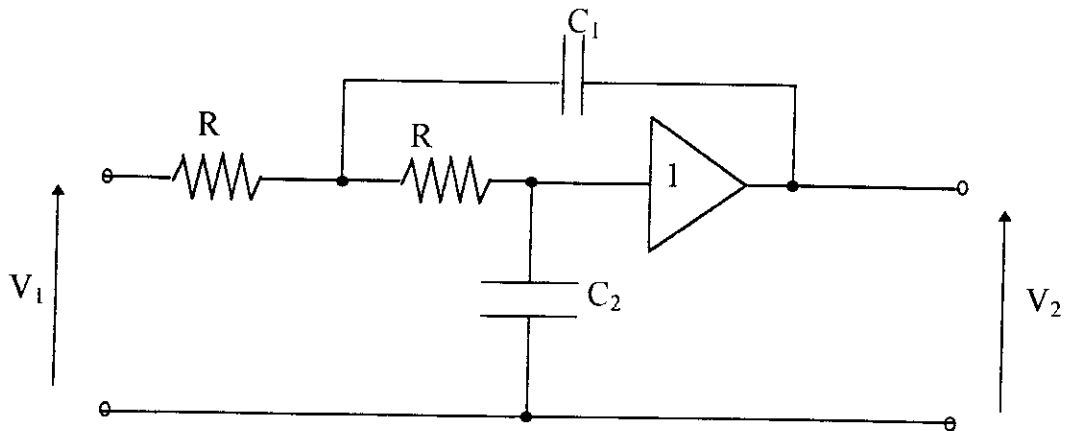
$$f_T(p) = \frac{1}{(1.0137 p^2 + 0.28289 p + 1)(3.5791 p^2 + 2.4114 p + 1)} \quad (III-2)$$

#### • synthèse du filtre :

La cellule élémentaire utilisée dans cette réalisation est celle de la figure (III-3).

Pour  $\beta=1$  on a :

$$\frac{V_2}{V_1} = \frac{1}{R^2 C_1 C_2 p^2 + 2RC_2 p + 1} \quad (III-3)$$



Fig(III-3) : La cellule élémentaire du filtre actif.

Le schéma théorique du filtre est donc le suivant :

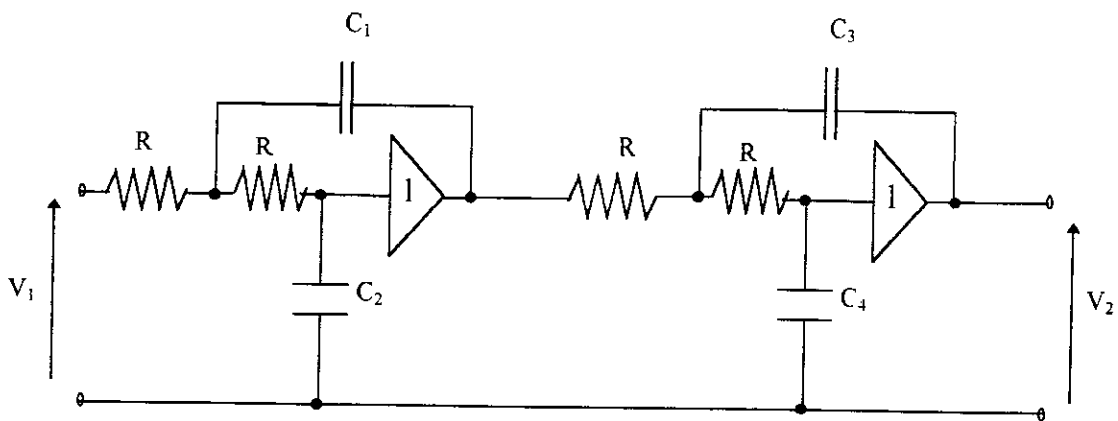


Fig (III.4) : Filtre passe bas de Tchebycheff d'ordre 4.

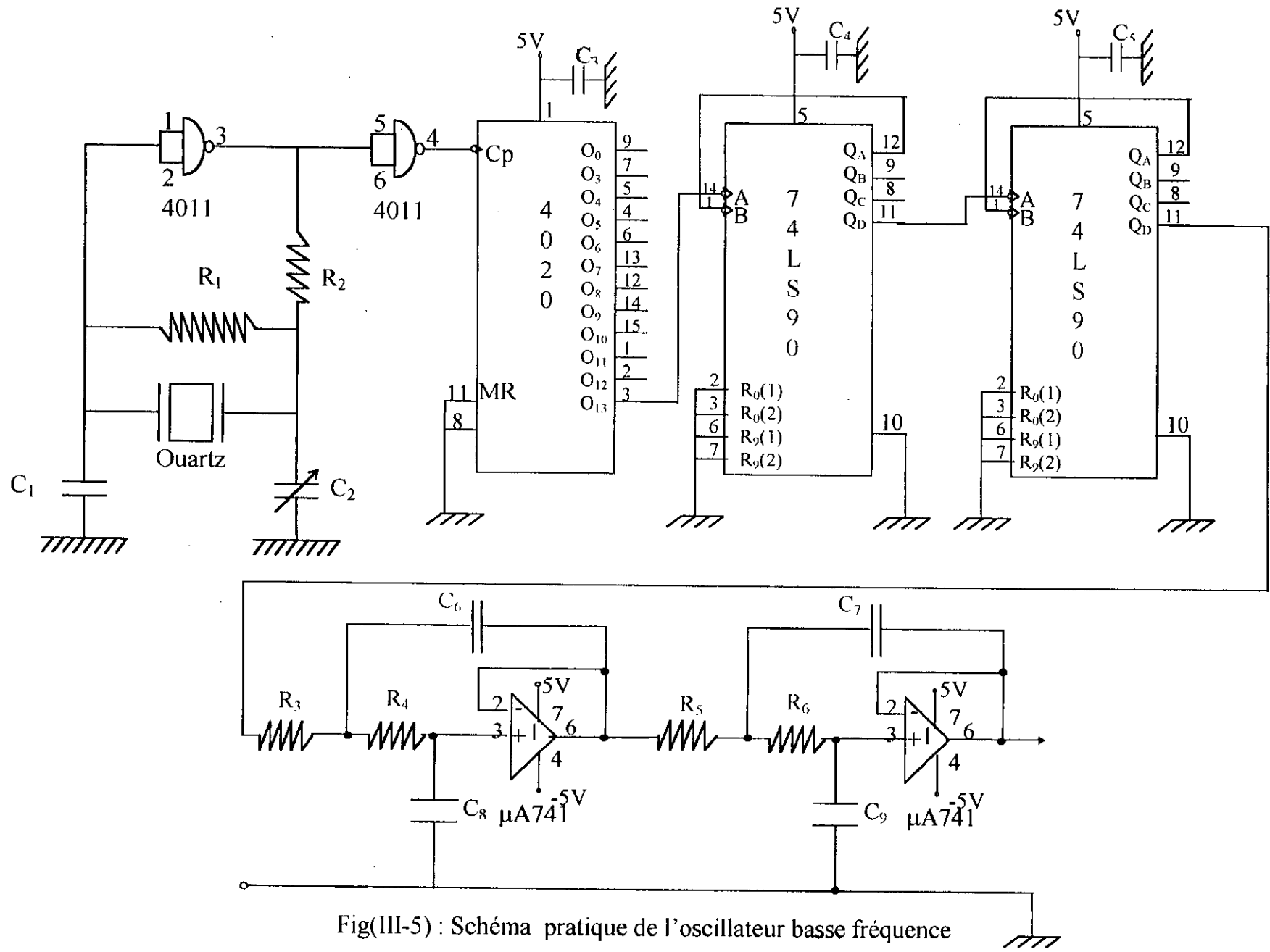
La fonction de transfert du filtre est :

$$f_T(p) = \frac{1}{(R^2 C_1 C_2 p^2 + 2RC_2 p + 1)(R^2 C_3 C_4 p^2 + 2RC_4 p + 1)} \quad (\text{III-4})$$

Par identification de (III-4) et (III-2) on peut déterminer les valeurs des capacités. En prenant  $R=10\text{K}\Omega$ , on trouve :

$$\begin{aligned} C_1 &= 33 \mu\text{F} & , & C_2 = 0.68 \mu\text{F} \\ C_3 &= 10 \mu\text{F} & , & C_4 = 4.7 \mu\text{F} \end{aligned}$$

La figure (III-5) de la page suivante représente le schéma pratique de cet oscillateur.



Fig(III-5) : Schéma pratique de l'oscillateur basse fréquence

**Chapitre (IV)****LA CONVERSION ANALOGIQUE NUMERIQUE****1. CONVERTISSEURS ANALOGIQUES-NUMERIQUES : [ 15 ]**

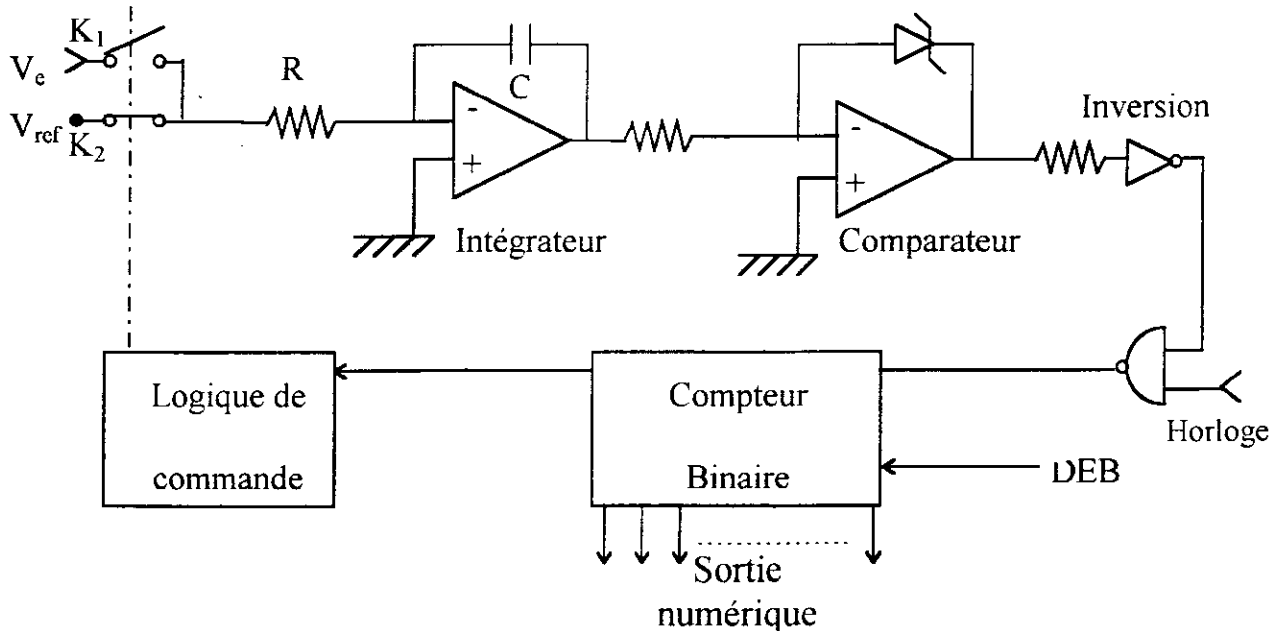
On va étudier les quatre techniques les plus utilisées sur le plan pratique. Chacune d'entre elles présente des caractéristiques intéressantes pour un type d'application donné.

**1.1 Convertisseur à double rampe :**

C'est un type de convertisseur très utilisé en instrumentation et dont les caractéristiques essentielles sont les suivantes :

- convertisseur lent, plusieurs dizaines de millisecondes ;
- faible sensibilité au bruit ;
- précision de conversion indépendante des composants utilisés et de la fréquence d'horloge ;
- résolution typique : 10, 12 et 13 bits.

La figure(IV-1) représente le schéma de principe de ce convertisseur. Les interrupteurs  $K_1$  et  $K_2$  fonctionnent en opposition de phase. La tension  $V_{ref}$  doit être de polarité opposée à  $V_e$ , ce qui suppose qu'un détecteur automatique de polarité fournisse  $V_{ref} > 0$  si  $V_e < 0$  et  $V_{ref} < 0$  si  $V_e > 0$ .  $V_e$  est supposée constante pendant toute la mesure.



Fig(IV-1) : Schéma de principe d'un convertisseur double rampe.

• **Fonctionnement :**

L'arrivée du front montant de début de conversion (DEB) provoque : la fermeture de  $k_1$  et la remise à zéro du compteur. Le comparateur étant supposé au niveau bas, l'horloge est appliquée au compteur jusqu'à ce que celui-ci soit plein, l'état 111...1 est détecté par la logique de commande des interrupteurs ; à cet instant  $T_1$ ,  $k_1$  s'ouvre et  $k_2$  se ferme. Arrêtons-nous à cet instant, la tension de sortie de l'intégrateur vaut :

$$V_s = \frac{-V_e}{RC}(T_1 - T_0) \quad \text{soit } V_s = kV_e .$$

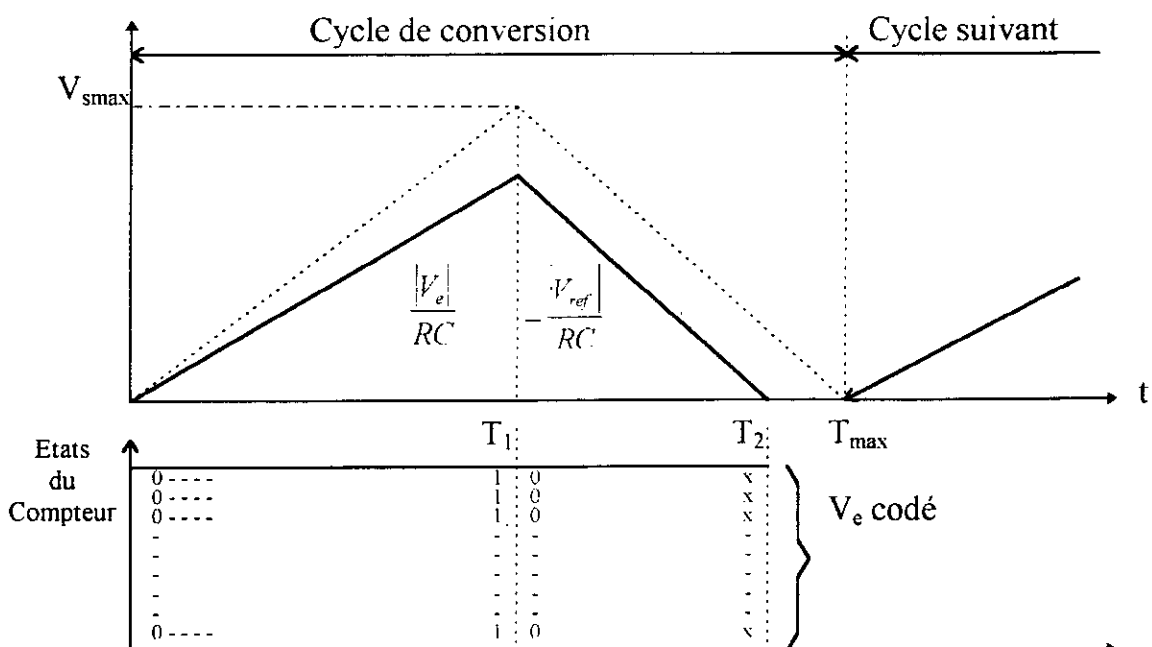
elle est proportionnelle à la tension d'entrée. L'intervalle d'intégration ( $T_1$ ,  $T_0$ ) est indépendant de la tension à mesurer, c'est le temps que met le compteur pour être plein. Dès la fermeture de  $k_2$ , le comparateur se remet à zéro, l'intégrateur se décharge linéairement avec une pente constante indépendante de la phase précédente qui vaut :  $V_{ref}/RC$ .

Dès que  $V_s$  passe par zéro, instant  $T_2$ , le comparateur bascule inhibant ainsi le passage de l'horloge, un ordre de fin de conversion est généré, et les sorties du compteur représentent alors le codage numérique de la tension  $V_e$ . Le cycle de conversion est représenté à la figure (IV-2).

Les paramètres suivants influent sur la précision de la conversion :

- courants et tensions de décalage de l'intégrateur ;
- dérivé de la tension de référence ;
- stabilité de la fréquence d'horloge.

Par contre, les composants du réseau RC de l'intégrateur n'interviennent pas.



Fig(VI-2) : Cycle de conversion .

## 1.2 Convertisseur à approximations successives :

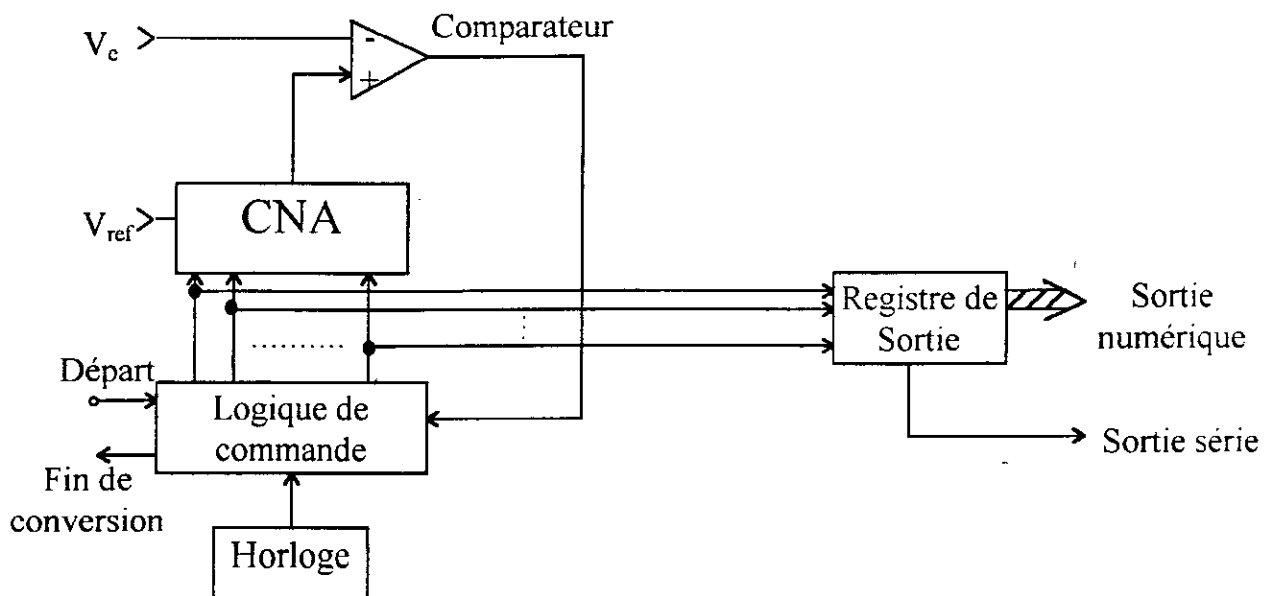
C'est un type de convertisseur très utilisé dans les interfaces microprocesseurs pour sa rapidité et sa bonne résolution (figure IV-3).

- **Principe :**

On compare en permanence le signal d'entrée  $V_e$ , à une tension analogique précise générée par un convertisseur numérique/analogique (CNA). Le processus de conversion est similaire à celui de la pesée à l'aide d'une balance à plateaux : on essaie le poids par valeurs décroissantes, tant que le poids de l'objet n'est pas équilibré, on garde tous les poids utilisés jusqu'à créer le déséquilibre. A ce moment, on retire le poids ayant provoqué le déséquilibre et l'on essaie celui immédiatement inférieur ... jusqu'à trouver l'équilibre.

- **Fonctionnement :**

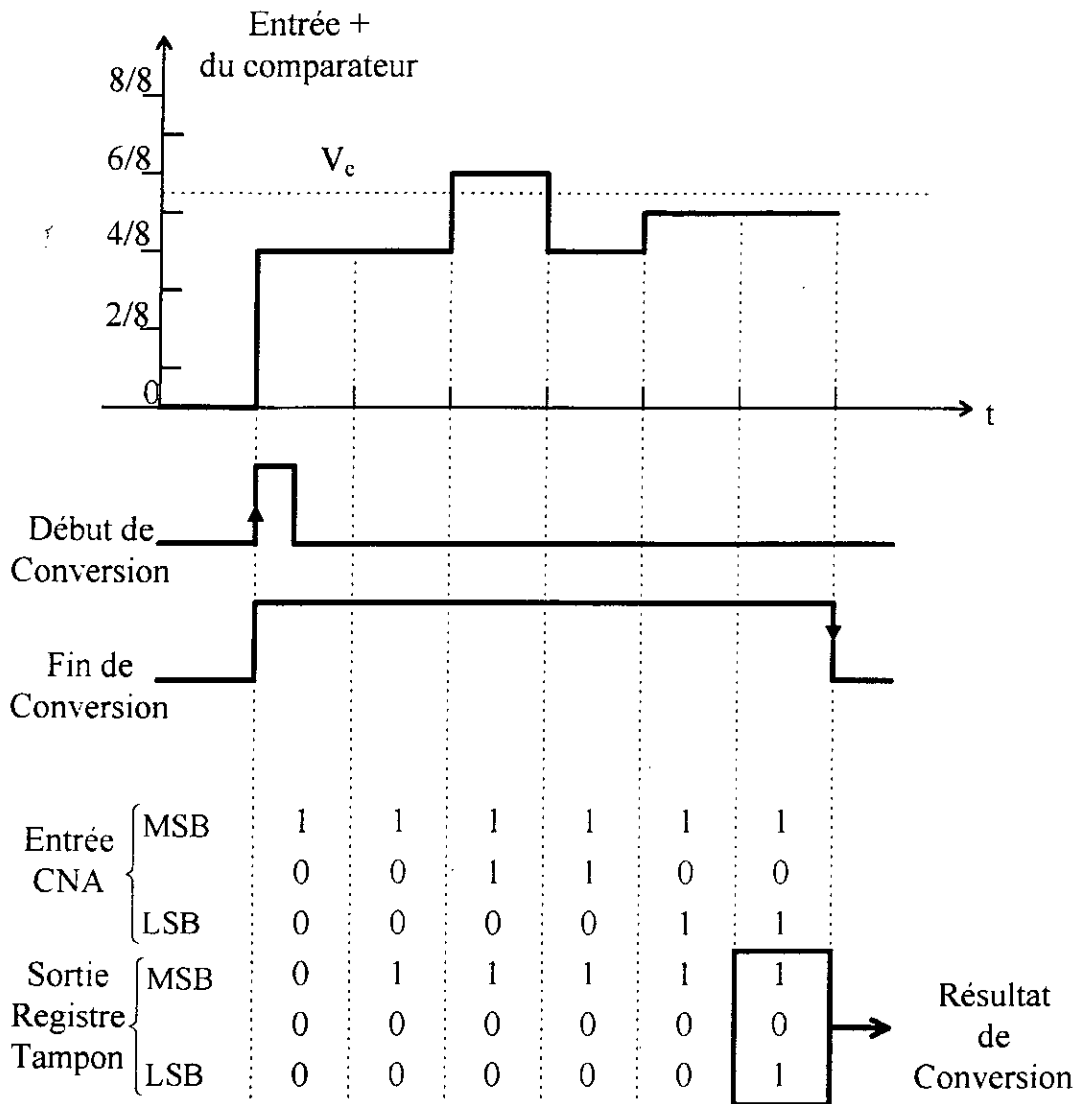
Après l'ordre de départ et la remise à zéro du convertisseur faite par la logique de commande, le CAN compare le bit le plus haut poids (MSB) à  $V_e$ , ce qui situe immédiatement  $V_e$  par rapport à la demi-échelle. Si  $V_e$  est supérieure à la tension analogique correspondant au MSB, on garde le MSB à 1 dans le registre de sortie, et le bit de rang immédiatement inférieur est essayé. Si l'entrée  $V_e$  était inférieure au MSB, celui-ci serait mis à zéro, on aurait 0 dans la position correspondante du registre de sortie et le bit suivant serait essayé. Si l'apport du second bit est insuffisant par rapport à  $V_e$ , on essaye le troisième bit, etc. Après avoir essayé tous les bits, le registre de sortie contient la présentation binaire d'un signal analogique le plus proche de  $V_e$ . Il est naturellement supposé que le signal d'entrée ne change quasiment pas durant la conversion, sous peine d'imprécision notable. Un exemple de fonctionnement d'un convertisseur 3 bits est donné en figure(IV-4).



Fig(IV-3) : Schéma de principe du convertisseur à approximations successives.



Les paramètres précision, linéarité, et vitesse dépendent essentiellement du C.N.A et du comparateur. Généralement, les temps d'établissement du convertisseur et du comparateur bien plus lents que le temps de commutation de la logique de commande. L'attention devra donc porter sur ces deux composants.



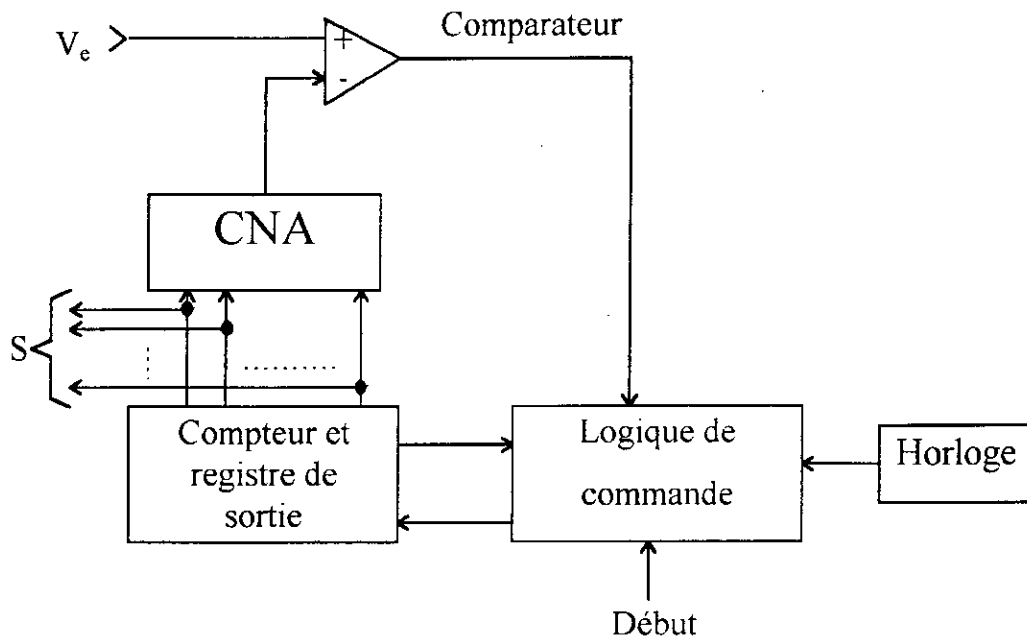
Fig(IV-4) : Fonctionnement d'un convertisseur 3 bits.

### 1.3 Convertisseur à comptage :

C'est un convertisseur très voisin de celui à rampe, de réalisation simple mais présentant le désavantage d'un temps de conversion long pour une résolution donnée. La figure(IV-5) représente le schéma de principe de ce convertisseur.

- **Fonctionnement :**

L'entrée analogique est comparée à la sortie d'un CNA recevant sur ces entrées digitales les sorties d'un compteur binaire. Le comptage commence dès l'arrivée de l'ordre DEBUT de conversion et continue jusqu'à ce que la tension délivrée par le CNA devienne égale à la tension d'entrée. A cet instant la conversion cesse, le contenu du compteur est transféré dans un registre de sortie, puis remis à zéro pour effectuer une nouvelle conversion. Le temps de conversion est indépendant de l'amplitude du signal d'entrée.



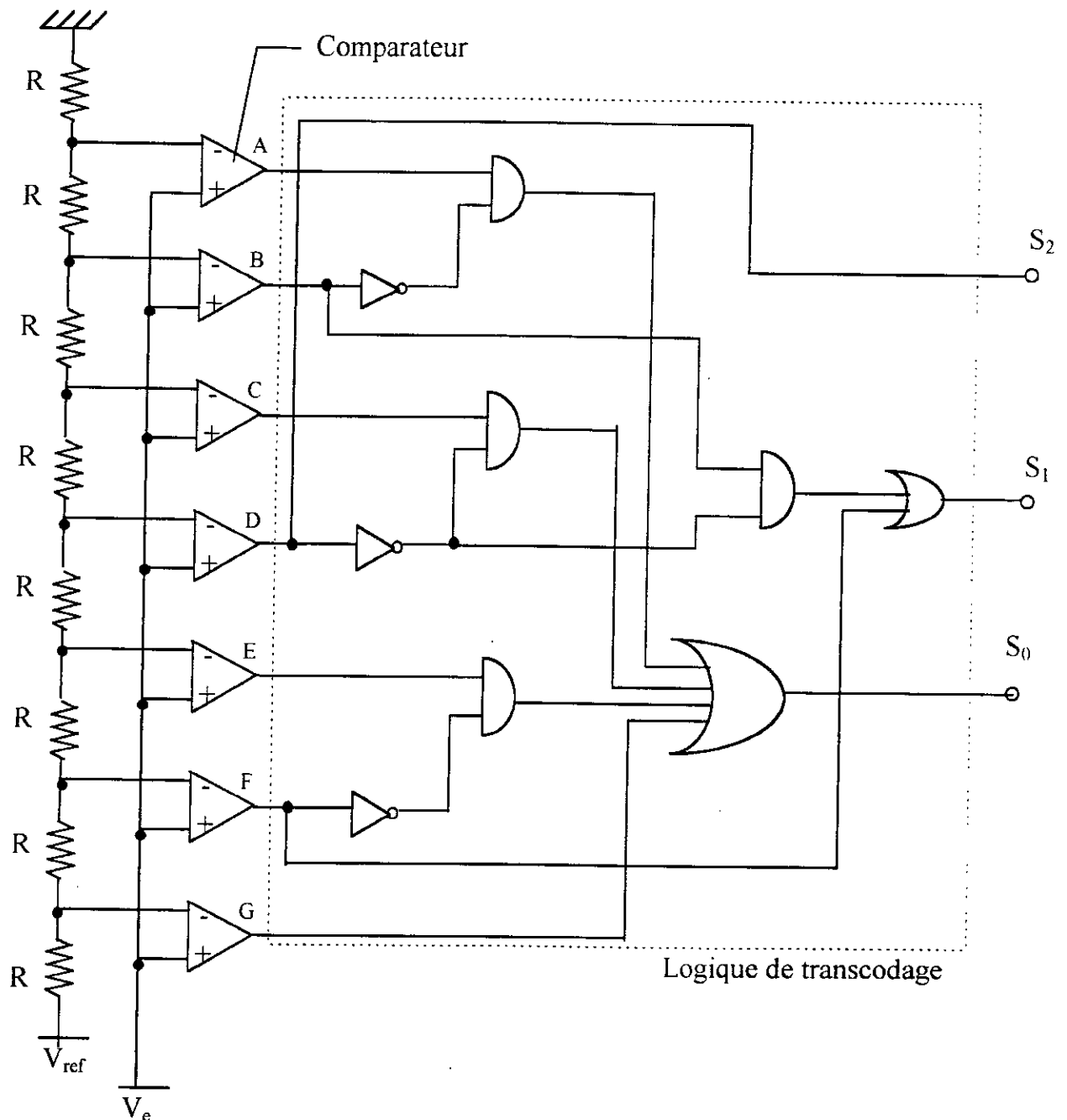
Fig(IV-5) : Schéma de principe d'un convertisseur à comptage.

#### 1.4 Convertisseur parallèle :

C'est le type de convertisseur le plus rapide, le temps de conversion se limitant aux temps de commutation des comparateurs et de propagation à travers les portes de transcodage.

Le temps de conversion est donc de l'ordre de quelques dizaines de nanosecondes. Le montage nécessite un comparateur par moment binaire: soit pour une conversion de  $n$  bits :  $2^n - 1$  comparateurs, ainsi pour 8 bits il faudra 255 comparateurs. La logique de transcodage peut être relativement complexe, car le nombre de couches des portes doit être le plus réduit possible afin de minimiser le temps de propagation ; elle peut fournir un code binaire naturel, ou un code de Gray.

Un exemple de réalisation d'un convertisseur 3 bits fournissant en sortie le code naturel est donné en figure(IV-6).



Fig(IV-6) : Convertisseur parallèle 3 bits.

## 2. L'ADC 0804 : [14]

Dans notre réalisation, on utilise le circuit intégré ADC 0804, qui est un convertisseur analogique numérique réalisé en technologie CMOS, fonctionnant d'après le principe de l'approximation successive et sa résolution est de 8 bits.

Les 8 lignes de sorties sont en logique trois états ; il peut être donc relié directement au bus de données d'un microprocesseur.

Afin de pouvoir employer le convertisseur analogique numérique dans un maximum de cas, le constructeur de ce CI a prévu une entrée de tension de référence réglable, qui permet de représenter des tensions analogiques relativement peu élevées avec une complète résolution 8 bits.

La tension de référence est réglée à la moitié de la gamme utilisable, car elle est amplifiée deux fois en interne.

La tension analogique à convertir est adressée sur les entrées différentielles analogiques  $V_{in}(+)$  et  $V_{in}(-)$ .

$V_{in}(-)$  : représente la plus faible valeur de la tension à convertir.

$V_{in}(+)$  : représente l'entrée de la tension à convertir.

Par exemple si la tension se situe entre 0.5 à 3.5 volt, la gamme sera alors 3 volt. Si l'on amène 0.5V sur  $V_{in}(-)$  pour compenser le décalage, la tension de référence peut être égale à la moitié de la gamme 3 volt, c'est-à-dire 1.5 volt.

Le cycle de conversion peut provenir du microprocesseur ou d'un réseau RC qui serait alors à prévoir pour mettre un cycle propre à disposition. L'entrée CLK à les caractéristiques d'un trigger de Schmitt comme le montre la figure (IV-8). De grandes capacités ou des charges de courant continu sur l'entrée CLK R sont à éviter, car elles dérangent le fonctionnement normal du convertisseur.

- **Principe de fonctionnement :**

le principe de fonctionnement de l'ADC 0804 est le suivant : le bit de la plus grande valeur est interrogé en premier, et après 8 comparaisons , un code binaire 8 bits est transmis au registre de sortie : par la suite est émise une interruption. qui signifie la fin de conversion. Le convertisseur peut travailler en mode individuel quand  $\overline{INTR}$  et  $\overline{WR}$  sont reliés,  $\overline{CS}$  étant alors 0.

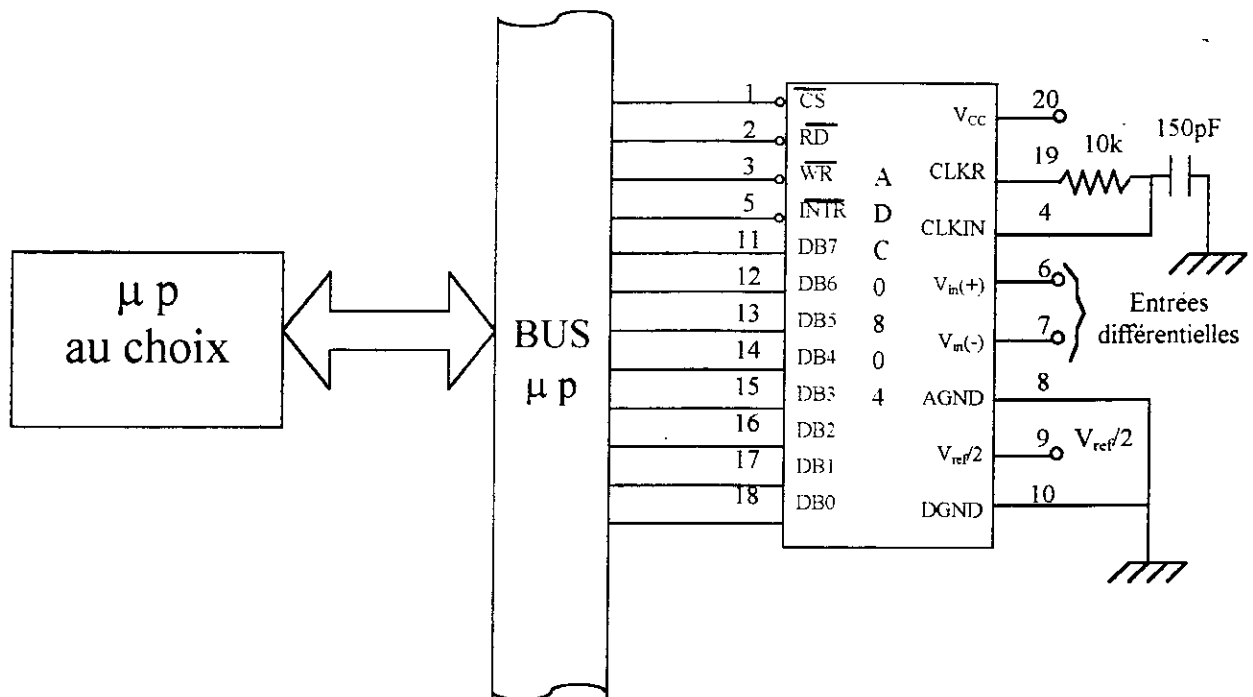
Lors de la transition <<high-low>> de  $\overline{WR}$ , le registre SAR interne est verrouillé et les différents niveaux de registre à décalage sont restaurés. Tant que les entrées  $\overline{CS}$  et  $\overline{WR}$  restent sur <<low>>, le convertisseur est maintenu à l'état initial (Reset). La conversion commence lorsque au moins une de ces entrées reconnaît une transition <<low-high>>.

Le convertisseur est mis en route quand  $\overline{CS}$  et  $\overline{WR}$  sont simultanément amenés sur <<low>>, ce qui déclenche la bascule de départ F/F<sub>1</sub>. Des signaux de cycle internes transmettent alors ce <<1>> à la sortie Q de F/F<sub>1</sub>. La porte ET(G<sub>1</sub>) effectue un chaînage entre ce <<1>> et un signal du cycle, pour créer ainsi une impulsion de mis à zéro de la bascule de démarrage F/F<sub>1</sub>.

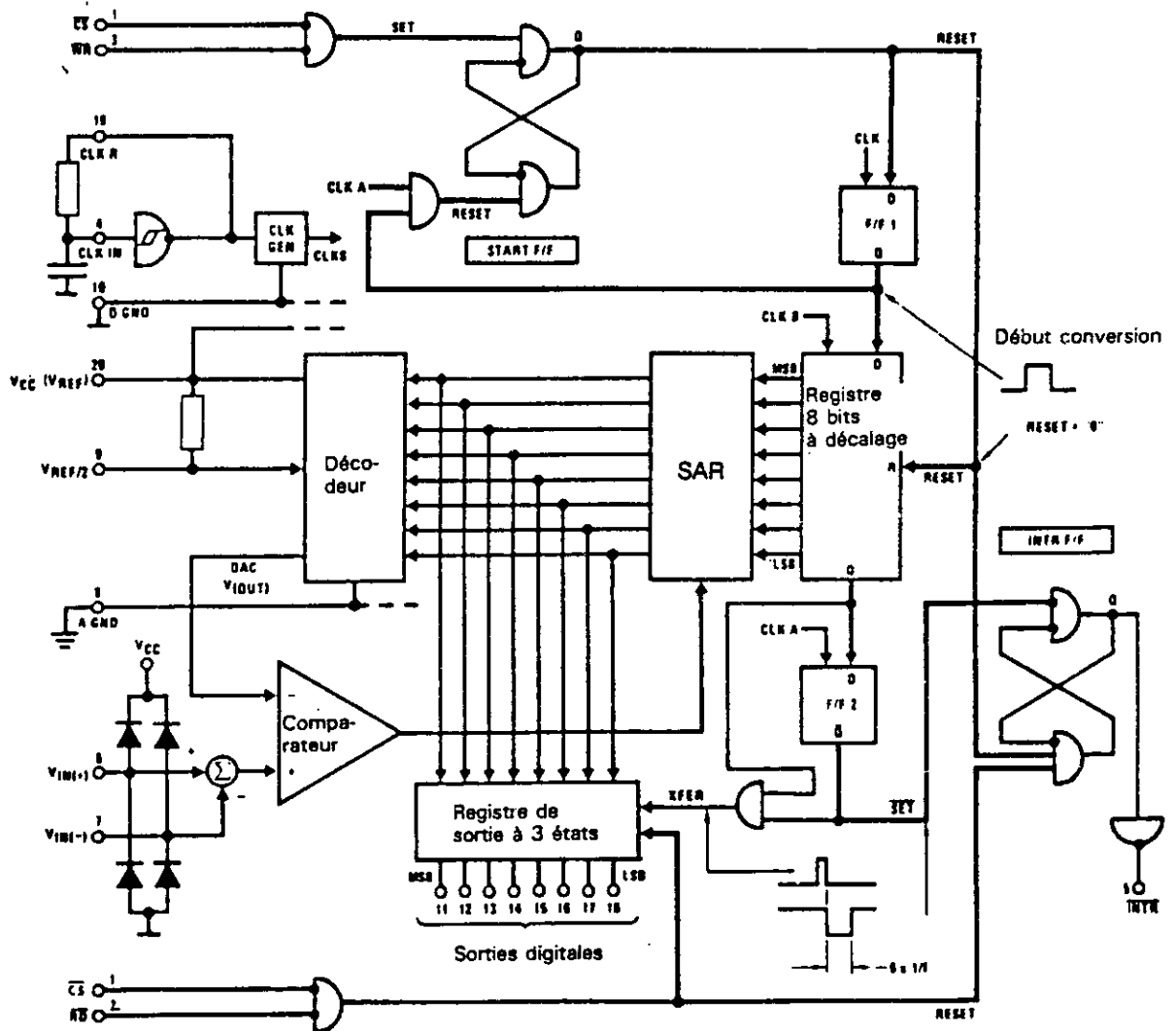
Le convertisseur se met en route, quand au moins un des signaux  $\overline{CS}$  et  $\overline{WR}$  devient <<high>> et quand le cycle interne fournit à nouveau un signal de mise à zéro de la bascule de démarrage.

Après avoir traversé le registre à décalage (l'approximation successive étant ainsi terminée), le <<1>> apparaît à l'entrée de la bascule D à verrou 1. Dès que ce <<1>> est mis par le registre à décalage la porte Et( $G_2$ ) fait le nécessaire pour que le nouveau code digital soit transmis au registre de sortie trois états, et dans ce cas le signal  $\overline{INTR}$  passe à l'état <<low>> indiquant la fin de conversion.

En mode individuel ou en continu ( $\overline{INTR}$  relié avec  $\overline{WR}$  et  $\overline{CS}$  sur 0) la bascule de démarrage est mise à <<1>> par la transition <<high-low>> du signal  $\overline{INTR}$ .



Fig(VI-7) : Brochage de l'ADC 0804 avec un système à microprocesseur.



Fig(IV.8) : Architecture interne de l'ADC 0804.

## Chapitre (V)

### BLOC DE CALCUL ET DE GESTION

#### 1. LE MICROCONTROLEUR 68705P3 : [ 1 ]

Le microcontrôleur 68705P3 est un circuit intégré réalisé en technologie HMOS. Il est programmable et effaçable par ultra violet. Son unité centrale est conforme à l'unité centrale d'un microprocesseur 8 bits 6800 à l'exception qu'il contient un seul accumulateur de travail.

##### 1.1 L'unité centrale :

C'est évidemment l'élément le plus important du circuit, la figure (V.1) présente les registres internes. On y trouve :

- Un accumulateur A sur 8 bits dans lequel sont réalisées toutes les opérations arithmétiques et logiques ;
  - Un registre d'index sur 8 bits, servant à l'adressage indexé mais pouvant aussi servir de registre à usage général lorsque l'adressage indexé n'est pas utilisé.
  - Un pointeur de pile SP sur 11 bits. Ce pointeur pointe initialement sur l'adresse la plus haute de la mémoire vive et descend dans celle-ci au fur et à mesure de son utilisation. La taille maximale de la pile est de 31 octets.
  - Le compteur ordinal, sur 11 bits, qui contient l'adresse de la prochaine instruction à exécuter. Son contenu est donc envoyé au registre adresse de la mémoire par l'intermédiaire du bus adresses.
  - Un registre d'état appelé registre de code condition. Ce registre 8 bits ne contient en fait que 5 bits significatifs : H, I, N, Z, et C. Ces bits sont positionnés automatiquement par l'unité centrale en fonction du résultat de l'instruction qui vient d'être exécutée.
- 
- \* Le bit C est le bit de retenue ;
  - \* Le bit Z est le bit de zéro ;
  - \* Le bit N est le bit de signe ;
  - \* Le bit I est le masque d'interruption. Ce bit lorsqu'il est mis à 1, et quelque soit le responsable de cette action, les interruptions masquables supportées par le circuit (interruption externe et interruption interne provenant des ressources internes) sont temporairement ignorées.
  - \* Le bit H et le bit de demi-retenu. Il est positionné à 1 lorsqu'une retenue a lieu entre les bits 3 et 4 durant une addition.

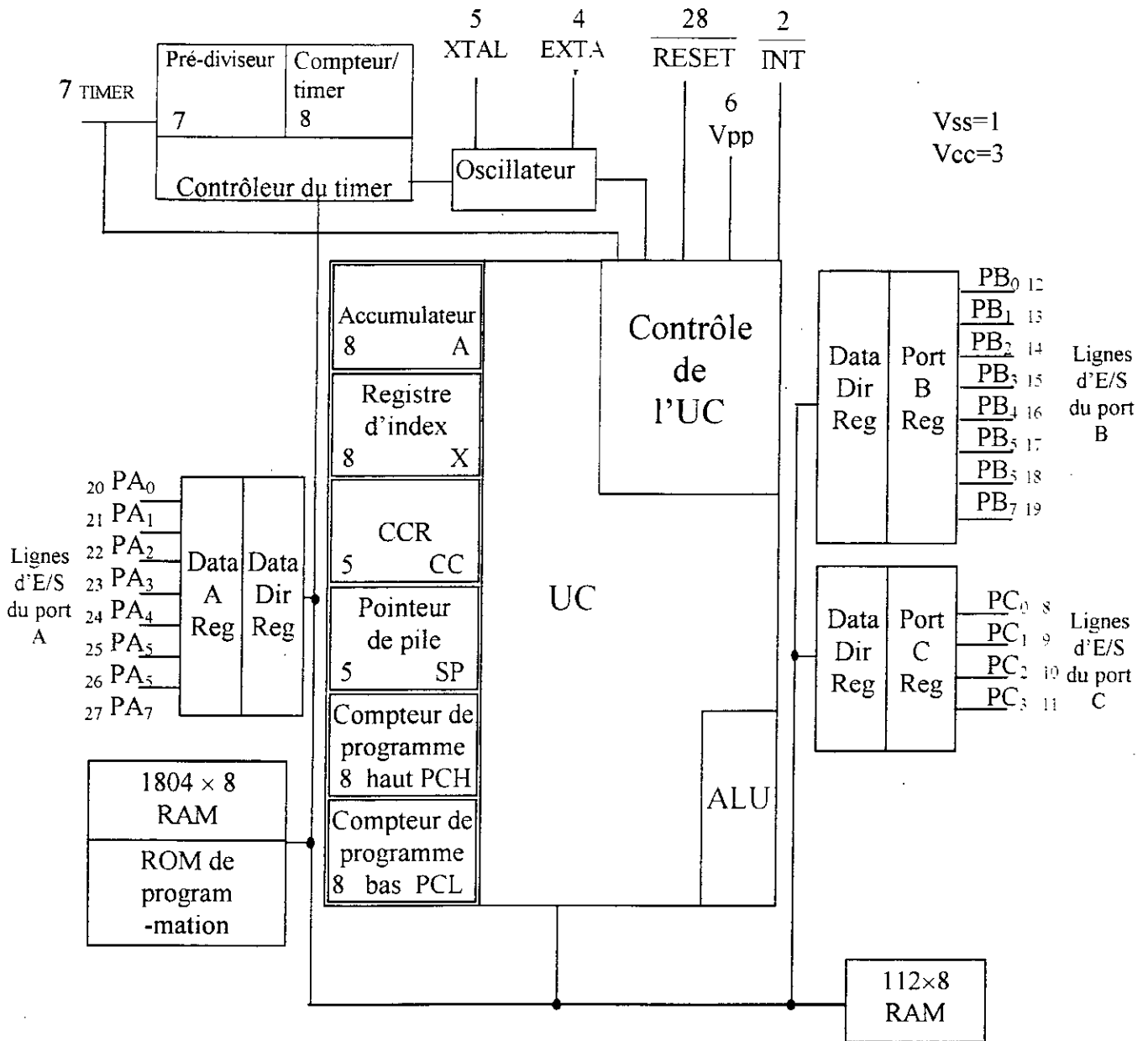


Fig (V.1) : Architecture interne et brochage du 68705P3.

### 1.2. Les entrées/sorties parallèles :

Le microcontrôleur 68705P3 contient deux ports de 8 bits appelés port A et B, et un port de 4 bits appelé port C. Toutes les connexions peuvent être configurées à tout instant et individuellement en entrées ou en sorties et ce aussi souvent que nécessaire dans le programme qui les gère. Les connexions utilisées en entrées sont en logique 3 états.



### 1.3 Organisation de la mémoire du 68705P3 :

Le microcontrôleur 68705P3 peut adresser 2 K octet de mémoire partagés en plusieurs segments dont la signification de chacun est la suivante :

- Les 16 premiers octets sont réservés pour les registres des ports d'entrées/sorties et les registres des ressources internes ;
- Une RAM de 112 octets où les programmes du système stock temporairement les données et réalisent les opérations arithmétiques ;
- Une EPROM utilisateur de 1668 octets programmables et effaçables par ultraviolet ;
- Une ROM de programmation de 115 octets qui contient un programme appelé boot strab qui est chargé de recopier le programme utilisateur résident dans une mémoire UVPROM dans l'EPROM utilisateur ;
- Dans l'espace adressable le plus haut réside une EPROM de 8 octets qui contient les vecteurs d'interruption des ressources internes et externes.

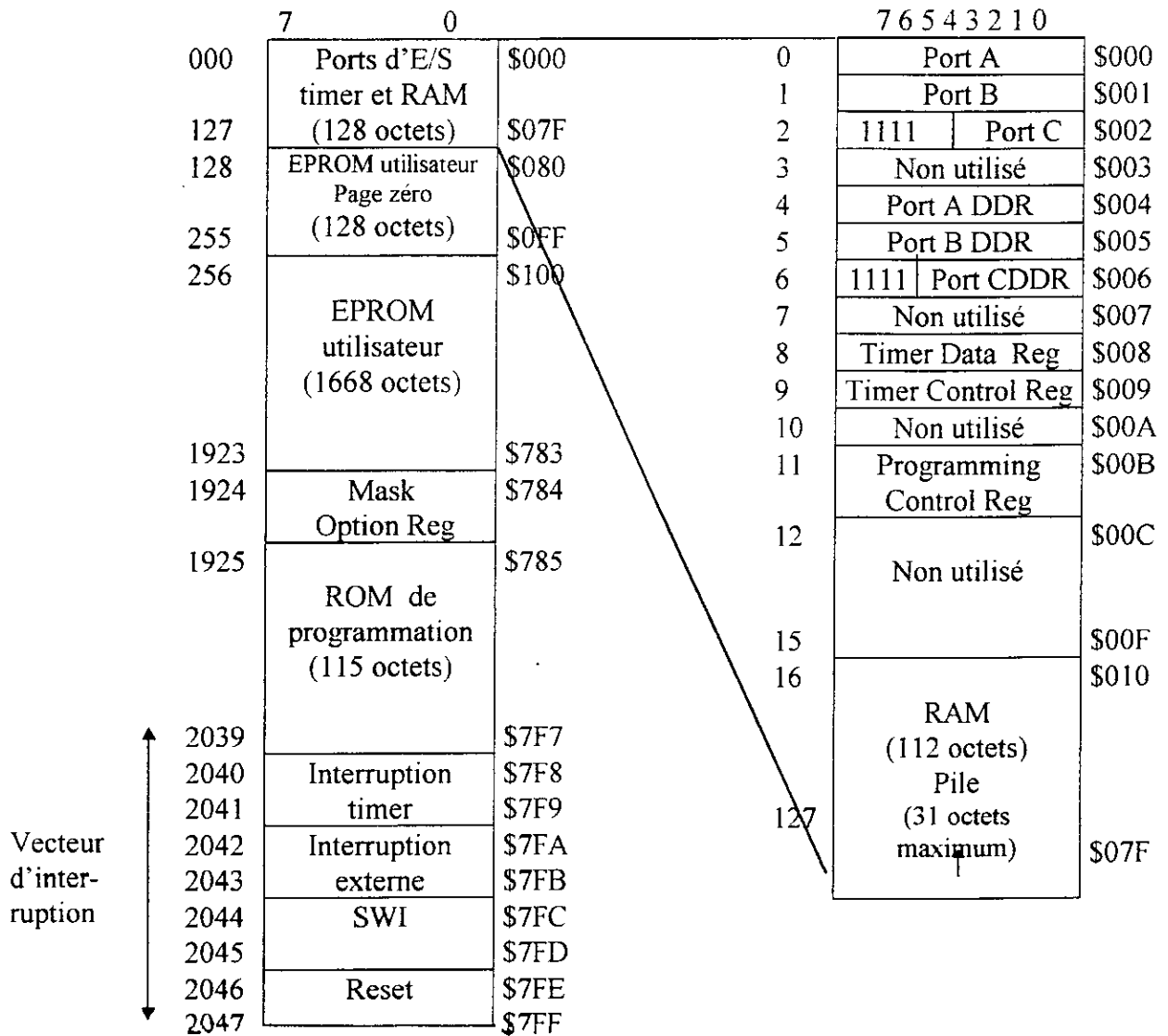


Fig (V.2) Memory map du 68705P3.

## 1.4 Mode d'adressage :

Le microcontrôleur 68705P3 dispose de sept modes d'adressages différents.

### 1.4.1 *L'adressage inhérent :*

Dans ce mode, il n'y a en effet pas d'adresse effective car l'instruction concernée se suffit à elle-même et n'a besoin d'accéder à aucune donnée externe. Par exemple TAX, CLC, CLKA, ... , Les instructions utilisant l'adressage inhérent sont toujours codées sur un octet qui est en fait le code de l'instruction proprement dite.

### 1.4.2 *L'adressage immédiat :*

Dans ce mode, l'adresse effective est égale à l'adresse suivant immédiatement celle contenant le code de l'instruction. En d'autres termes, une instruction utilisant l'adressage immédiat est immédiatement suivie par la donnée à manipuler.

La donnée manipulée par l'adressage immédiat doit impérativement être une donnée de 8 bits. Une instruction utilisant ce mode d'adressage occupe donc deux octets en mémoire de programme.

### 1.4.3 *L'adressage direct :*

Dans ce mode d'adressage l'instruction est en effet suivie d'un octet qui est l'adresse effective de la donnée manipulée. Comme cette adresse effective est codée sur un octet seulement, ce mode d'adressage ne fonctionne que de l'adresse \$00 à l'adresse \$FF.

Une instruction utilisant ce mode d'adressage occupe donc deux octets en mémoire de programme, l'octet du code de l'instruction et le code de l'adresse.

### 1.4.4 *L'adressage étendu :*

Ce mode est une évolution du précédent et permet d'atteindre n'importe quelle adresse mémoire car, il autorise un codage de l'adresse effective sur deux octets.

Une instruction utilisant ce mode d'adressage occupe donc trois octets en mémoire de programme, l'octet du code de l'instruction et les deux octets de l'adresse.

### 1.4.5 *L'adressage indexé :*

Ce mode d'adressage est particulièrement souple, puissant, et fait intervenir deux facteurs dans le calcul de l'adresse effective. Le contenu du registre X et une donnée

appelée déplacement. Il existe de ce fait trois sous-modes :

a) Adressage indexé avec déplacement nul :

Dans ce mode le contenu de l'index X est l'adresse effective, puisqu'il n'y pas de déplacement à prendre en compte. Comme l'index est une registre 8 bits, ce mode d'adressage ne permet d'accéder qu'aux adresses comprises entre 00 et \$FF.

Une instruction utilisant ce mode d'adressage occupe seulement un octet en mémoire de programme : l'octet de code de l'instruction.

b) Adressage indexé avec déplacement sur 8 bits :

Ce mode d'adressage est une évolution du précédent , en effet l'adresse effective est obtenue en ajoutant le contenu de l'index à l'octet qui suit le code de l'instruction. Ce mode d'adressage ne permet d'accéder qu'aux adresses comprises entre 00 et \$1FF.

Une instruction utilisant ce mode d'adressage occupe donc deux octets en mémoire de programme : l'octet de code de l'instruction et l'octet de déplacement.

c) Adressage indexé avec déplacement sur 16 bits :

Ce mode d'adressage est encore une évolution du précédent ; En effet l'adresse effective est maintenant obtenue en ajoutant le contenu de l'index X aux deux octets qui suivent le code de l'instruction dans l'ordre poids fort et poids faible.

En contre partie, une instruction utilisant ce mode d'adressage occupe trois octets en mémoire de programme.

#### 1.4.6 L'adressage relatif :

Ce mode d'adressage n'est utilisé que pour les instructions de saut et branchement. Il spécifie l'adresse effective sous forme d'un déplacement qui est donc ajouté à la valeur courante du PC pour déterminer l'adresse de poursuite du programme. Comme lors de l'exécution d'une instruction, le PC pointe toujours sur l'instruction suivante, l'adresse effective est en effet égale à la valeur courante du PC + 2 + le déplacement.

Le déplacement ici codé sur un octet mais, est une valeur signée. Il peut donc prendre toute valeur décimale comprise entre -128 et +127.

Une instruction utilise ce mode d'adressage occupe deux octets en mémoire de programme ; l'octet de code de l'instruction et l'octet de déplacement.

### 1.4.7 Adressage bit ou manipulation de bit :

Ce mode d'adressage comporte en fait deux sous modes ; le mode dit *bit set clear* que l'on peut appeler mode positionnement de bit, et le mode dit *test and branch* que l'on peut appeler mode test de bit et branchement.

#### a) Mode positionnement de bit :

Dans ce mode, l'instruction écrite avant assemblage est suivie de deux informations et se présente sous le format suivant :

INSTRUCTION N°, ADRESSE

où N° est le numéro de bit à manipuler étant entendu que le bit de poids le plus faible porte le numéro 0 et le bit de poids le plus fort le numéro 7 et où ADRESSE est l'adresse de la mémoire où du registre concerné, codée sur 8 bits.

Après assemblage toutefois cette instruction et l'information qui suit n'occupent plus que deux octets. Le numéro de bit est en effet compacté avec le code de l'instruction, et on trouve donc en mémoire de programme l'octet résultant de ce compactage qui est alors le code instruction suivi de l'octet d'adresse qui lui reste intact .

#### b) Mode test de bit et branchement :

Ce mode d'adressage permet en effet de tester n'importe quel bit contenu à une adresse spécifiée et, en fonction du résultat de ce test, de décider de la poursuite du programme en séquence ou du saut à une adresse déterminée. L'instruction avant l'assemblage, et donc suivie de trois informations et se présente de la façon suivante :

INSTRUCTION N°, ADRESSE, SAUT

où N° est le numéro de bit à manipuler étant entendu que le bit de poids le plus faible porte le numéro 0 et le bit de poids le plus fort le numéro 7 et où ADRESSE est l'adresse de la mémoire où du registre concerné, codée sur 8 bits, et où SAUT est le déplacement permettant de calculer l'endroit où doit se poursuivre le programme selon le résultat du test. Ce déplacement est également codé sur 8 bits, mais signé.

Après assemblage cette instruction et l'information qui suit n'occupent plus que trois octets.

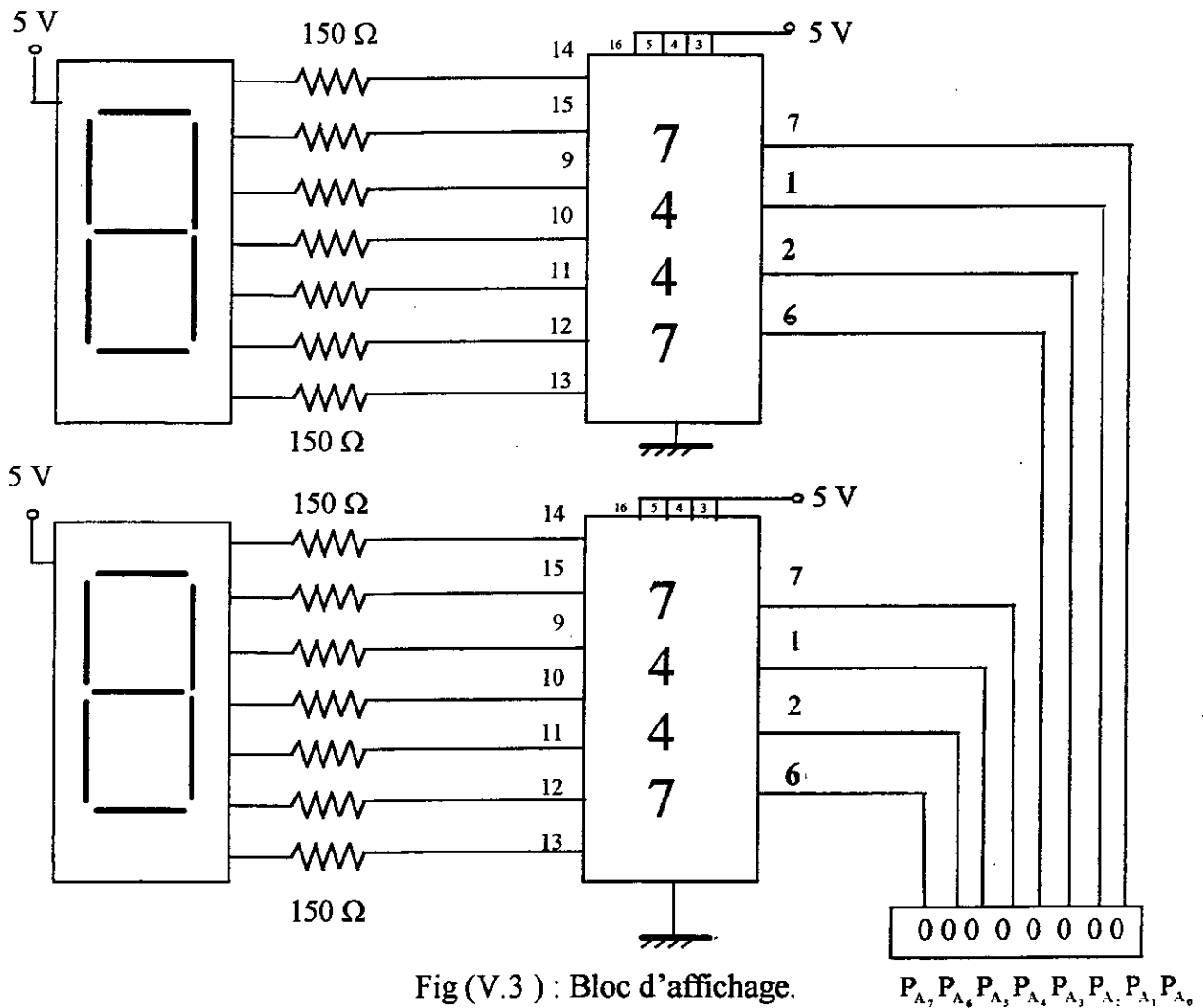
Comme dans le mode précédent le numéro du bit est en effet compacté avec le code de l'instruction, et on trouve donc en mémoire de programme l'octet résultant de ce compactage qui est alors le code instruction suivi de l'octet d'adresse, suivi à son tour du déplacement.

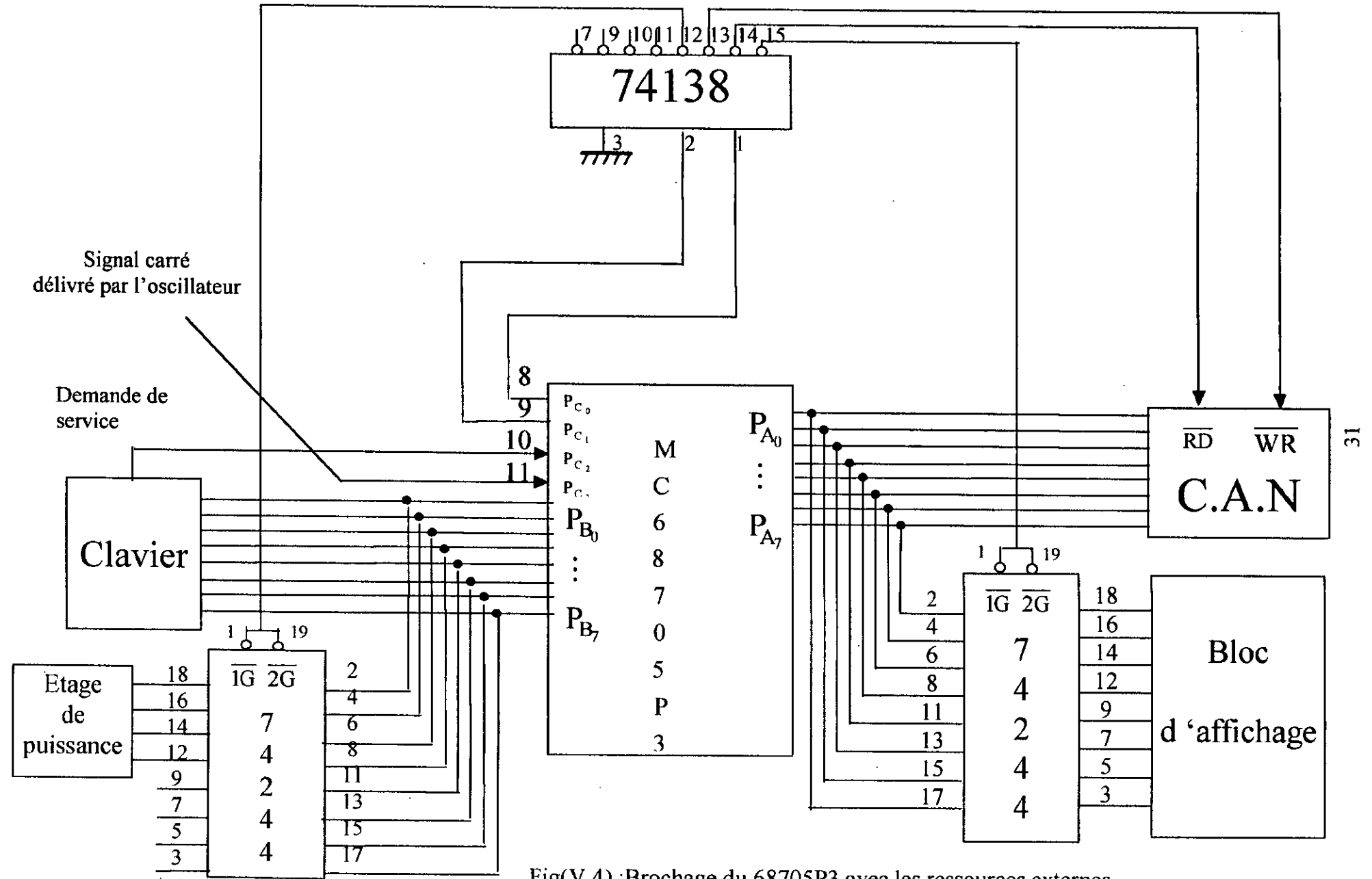
**2. DESCRIPTION TECHNIQUE :**

Le microcontrôleur 68705P3 comporte deux ports d'entrées/sorties de 8 bits appelés port A et port B, et un port d'E/S de 4 bits appelé port C. Ces ports d'entrées/sorties sont insuffisants pour commander les ressources externes du système (clavier, afficheurs, CAN, bloc de puissance ). Pour augmenter les entrées/sorties, on va utiliser des buffers validés par les lignes de commande du port C. Donc les informations disponibles sur le port A ou le port B sont dirigées vers le bloc approprié par la validation du buffer correspondant à l'aide du port C.

Deux lignes du port d'entrées/sorties C sont utilisées en entrée. Une ligne est utilisée pour tester le front montant du signal demande service généré par le clavier, et l'autre est utilisée pour la détection du passage par zéro de la tension sinusoïdale . Les deux lignes qui restent du port C sont insuffisants pour commander les buffers et le convertisseur analogique numérique. Comme le nombre des signaux de commandes est égal à quatre, la solution la plus simple consiste à utiliser un décodeur deux vers quatre. La figure (V.4) représente le brochage du 68705P3 avec les ressources externes.

Le bloc d'affichage est réalisé à l'aide de deux afficheurs sept segment a anode commune et de deux décodeurs BCD-sept segment comme représenter sur la figure (V-3).





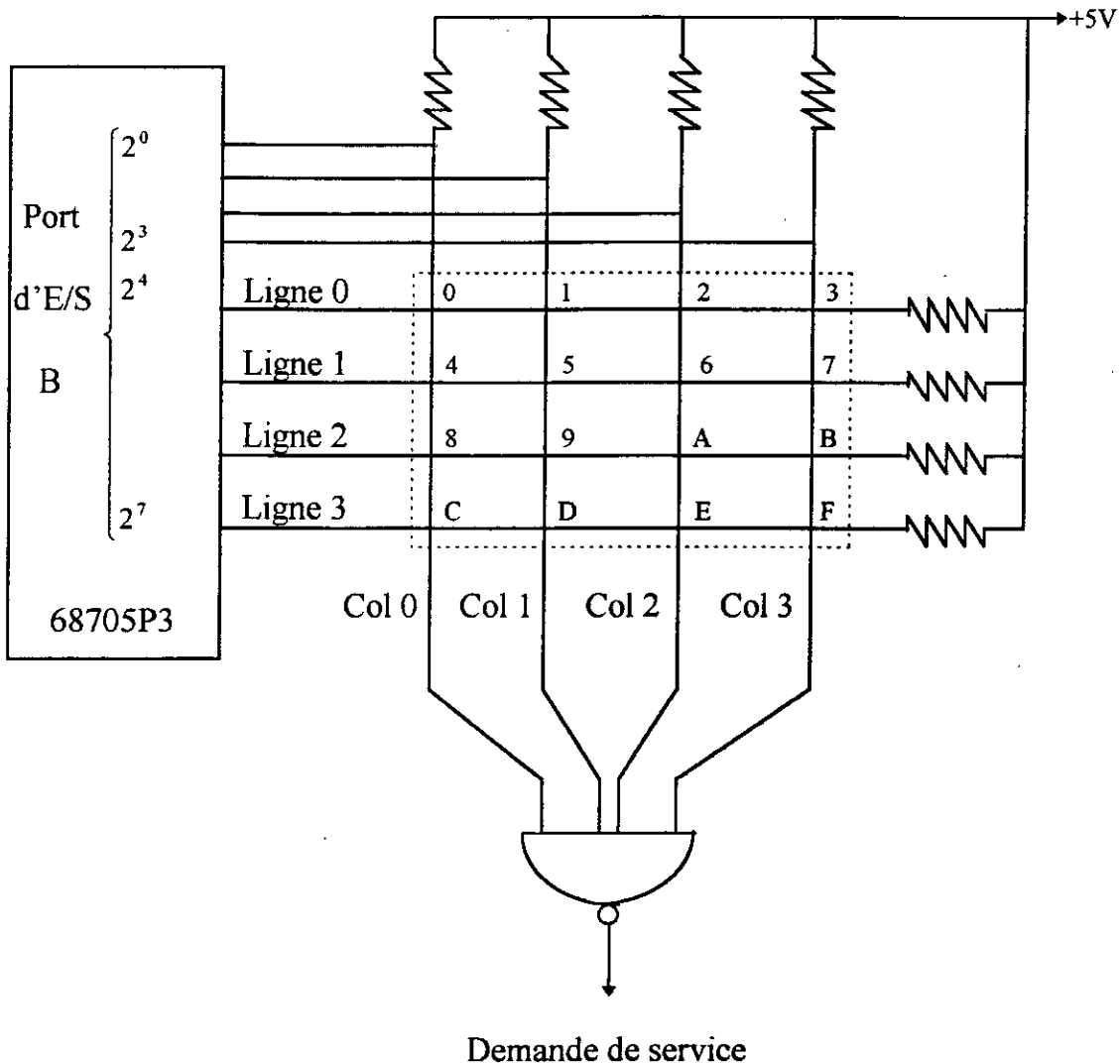
Fig(V.4) : Brochage du 68705P3 avec les ressources externes.

**Chapitre (VI)**

**LE CLAVIER HEXADÉCIMAL**

Comme la position du soleil est fonction de la longitude et de la date, il faut prévoir un outil qui nous permette d'indiquer ces valeurs au microcontrôleur. La solution la plus simple est d'utiliser un clavier numérique. Le microcontrôleur devra alors connaître la touche enfoncée et en déduire son code.

Pour les claviers numériques à 16 touches, l'appui sur une touche se traduit par un court-circuit entre la colonne et la ligne à l'intersection desquelles se trouve la touche enfoncée. Ainsi sur la figure (VI-1) l'appui sur la touche 8 entraîne un court-circuit entre la ligne 2 et la colonne 0 et uniquement entre cette ligne et cette colonne.



Fig(VI-1) : Clavier hexadécimal et son interfaçage.

Ainsi si nous envoyons un niveau zéro sur les lignes 0,1,2,3, ce niveau zéro ne se retrouve que sur la colonne 0 . Inversement si nous envoyons un niveau zéro sur les colonnes 0,1,2,3, ce niveau zéro ne se retrouve que sur la ligne 2. C'est sur ce principe qu'est réalisé le décodage d'un tel clavier. Ainsi dans notre exemple, nous avons récupéré le mot binaire 1110 sur les colonnes et 1011 sur les lignes.

L'envoi des niveaux zéro d'abord sur les lignes puis sur les colonnes s'effectue facilement par l'intermédiaire d'un Port d'E/S du microcontrôleur.

Dans une première phase, les bits du port d'E/S affectés aux lignes sont programmés en sortie et les bits affectés aux colonnes, sont programmés en entrée. Un niveau zéro est envoyé sur chacune des lignes : c'est l'état de repos du clavier ou d'attente. Tout appui sur une touche fait passer au niveau «1» le signal « demande de service » qui sert de bit d'état en mode programmé ou de demande d'interruption en mode interruptible.

La demande de service reçue par le microcontrôleur est alors traitée comme suit : le mot de quatre bits constitué par les quatre colonnes est lu et mémorisé. C'est la fin de la première phase.

Dans une deuxième phase les bits du port d'E/S affectés aux colonnes sont programmés en sortie et les bits affectés aux lignes sont programmés en entrée. L'envoi du mot binaire 0000 sur les quatre colonnes est suivi de la lecture du mot binaire recueilli sur les quatre lignes. Ce mot lu est ensuite mémorisé.

La technologie des boutons-poussoirs est telle qu'ils présentent des rebondissements qu'il faudra éliminer par logiciel : c'est-à-dire on crée une temporisation de quelques ms avant de lire la donnée sur le port E/S.

Chaque mot de quatre bits ne code que quatre nombres, les numéros 0,1,2,3, de la ligne ou de la colonne. Chaque mot de quatre bits peut donc être ramené à un mot de deux bits par une conversion décimale - binaire. En juxtaposant ces deux mots de deux bits, nous obtenons un mot de quatre bits qui est, l'adresse de position mémoire contenant le code approprié de la touche enfoncée. Mais cela impose que l'adresse la plus faible soit 0000 en hexadécimal ce qui n'est pas souvent possible. Pour remédier à cela, il suffit de considérer ce mot comme un déplacement par rapport au début de la table de décodage.

Quand à la conversion décimale - binaire il suffit de repérer par des décalages successifs dans le carry la position du « 0 » dans le mot de 4 bits affecté à la colonne ou à la ligne. Pour cela, nous utiliserons une position mémoire qui sera préalablement mise à zéro et incrémentée jusqu'à ce que le bit de l'accumulateur A décalé dans le carry soit zéro. L'organigramme de cette conversion décimale - binaire est représenté sur la figure (VI-2).



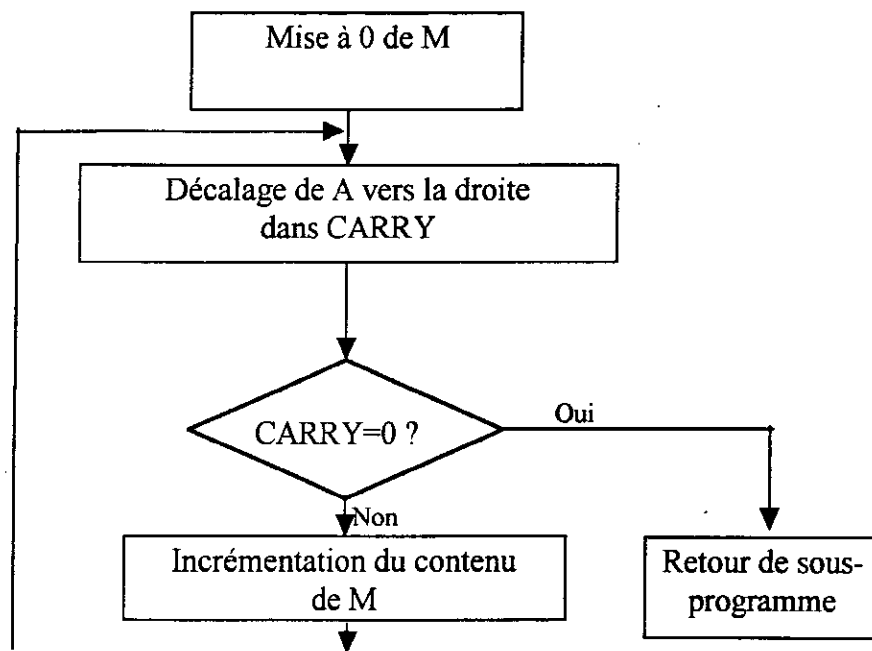
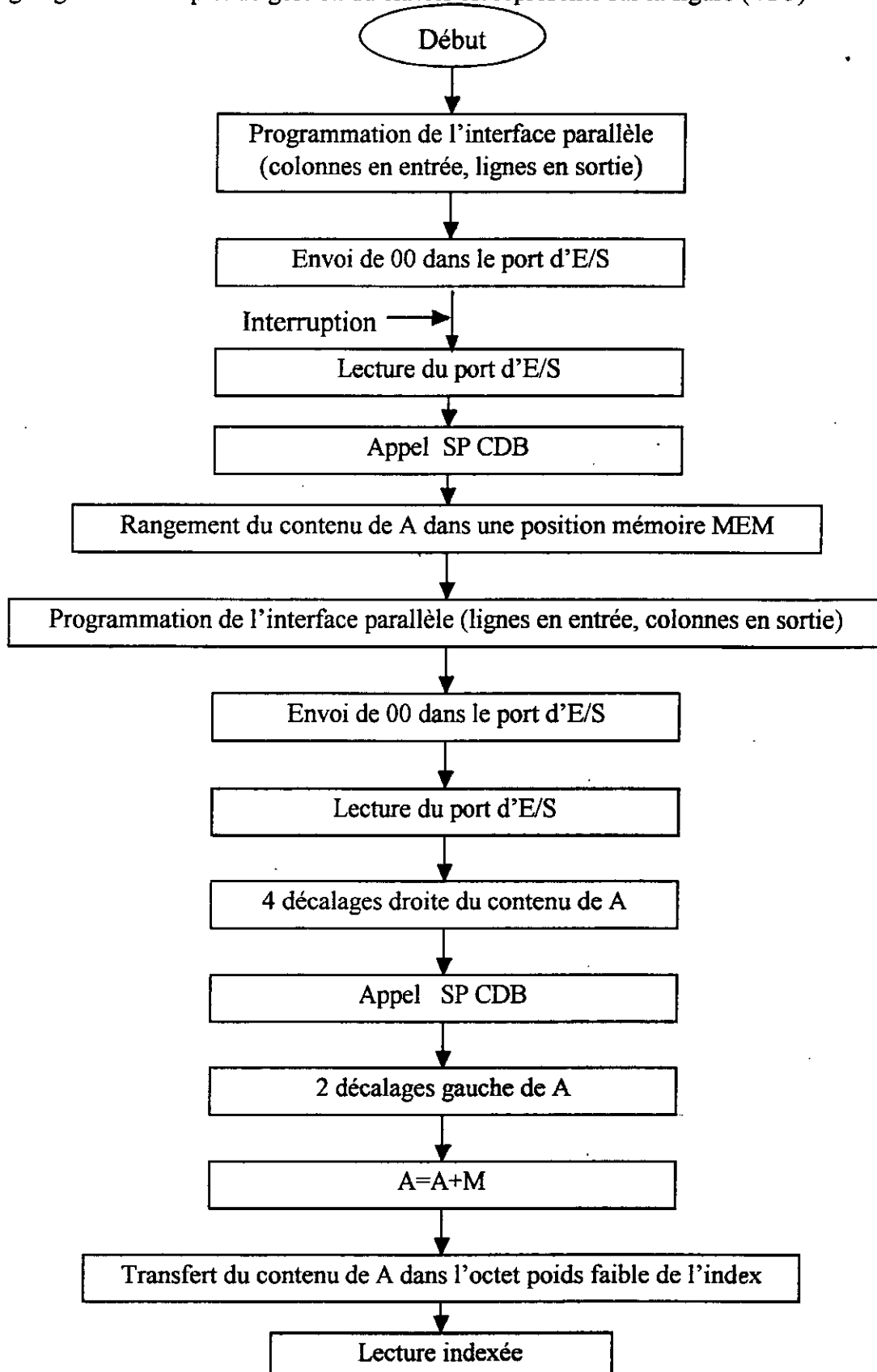


Fig (VI-2) : Sous-programme de conversion décimale - binaire.

L'organigramme complet de gestion du clavier est représenté sur la figure (VI-3)



Fig(VI-3) : Organigramme de gestion du clavier.

## **Chapitre (VII)**

### **BLOC DE PUISSANCE**

#### **1. MOTEURS PAS A PAS :**

Bien que l'utilisation des premiers moteurs pas à pas remonte au début des années 1930, ce n'est que depuis le début des années 1960 que ce type de moteur s'est véritablement développé. La cause principale de cette extension est le développement des composants de l'électronique numérique, et l'apparition des microprocesseurs.

Le moteur pas à pas est un moteur synchrone. En d'autres termes, à toute impulsion d'alimentation, il correspondra une avance élémentaire constante, dite pas. Un nombre déterminé d'impulsions entraîne un nombre correspondant de pas. De plus, la succession d'impulsions à une fréquence déterminée permet d'imposer une vitesse de rotation pratiquement constanté.

C'est le contrôle de ces deux fonctions « position et vitesse » qui permet d'assurer une conversion d'informations. Le moteur pas à pas assure cette conversion sans introduire d'erreurs cumulatives. Elle peut être réalisée sans asservissement.

#### **1.1 Définition : [ 9 ]**

Un moteur pas à pas est un transducteur permettant une conversion d'énergie et d'information de caractère électromécanique. Son alimentation est de type électrique digital ou impulsionnel. Son mouvement rotatif est de type incrémental continu.

La structure fixe « stator » est généralement de type réluctant variable et supporte les bobinage. La structure mobile « rotor » est également à réluctance variable (moteurs réluctants) ou à caractéristique de magnétisation périodique (moteurs à aimant permanent).

#### **1.2 Classification des moteurs pas à pas : [ 8 ]**

Les moteurs pas à pas sont principalement de trois types :

- Moteur pas à pas à réluctance variable ;
- Moteur pas à pas à aimant permanent ;
- Moteur pas à pas hybride.

### 1.2.1 Moteur pas à pas à réluctance variable :

La figure (VII.1) représente un moteur pas à pas à réluctance variable. Le stator porte six pôles ou dents tandis que le rotor a quatre pôles. Sur chaque pôle du stator est montée une bobine. Les bobines sur les pôles diamétralement opposés peuvent être reliées en parallèle ou en série et constituent ainsi une phase.

Le stator et le rotor sont réalisés en alliage de silicium. Ce matériau doit avoir une grande perméabilité pour permettre le passage d'un grand flux magnétique à travers lui quand une faible force électromotrice lui est appliquée.

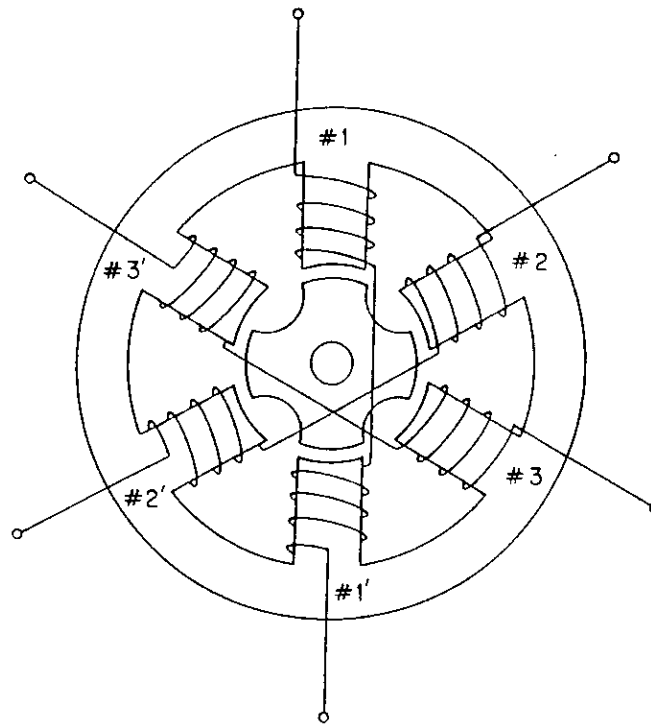


Fig (VII.1) : Moteur pas à pas à réluctance variable.

Au démarrage, quand la première impulsion arrive sur le circuit de commande, celui-ci va exciter les bobines statoriques 1 et 1'. Cette excitation engendre un flux magnétique dans l'entrefer de la dent 1 du rotor qui va s'aligner avec la dent 1 du stator et la dent 3 du rotor va s'aligner avec la dent 1' du stator comme indiqué sur la figure (VII.2.a).

Quand la deuxième impulsion arrive sur le circuit de commande, celui-ci va désactiver la première phase et simultanément la seconde phase est activée. Cette excitation engendre un champs magnétique qui oblige les dents 2 et 4 du rotor de s'aligner avec les dents 2 et 2' respectivement du stator comme indiqué sur la figure (VII.2.b). Dans ce cas le rotor tourne d'un angle  $30^\circ$  dans le sens anti-horaire, et le pas de ce moteur est égale à  $30^\circ$ . Quand l'impulsion suivante arrive sur le circuit de commande, celui-ci va désactivé la phase 2 et simultanément active la phase 3. Les dents du rotor 3 et 1 s'aligneront respectivement avec les dents 3 et 3' du stator, et le rotor tourne encore de  $30^\circ$  dans le sens anti-horaire, comme indiqué sur la figure (VII.2.c).

Donc, à chaque impulsion arrivant sur le circuit de commande, le rotor va tourner d'un pas de  $30^\circ$ .

Pour changer le sens de rotation du rotor, il faut changer l'ordre d'excitation des phases. En passant de la phase 1 - 1' à la phase 3 - 3', de la phase 3 - 3' à la phase 2 - 2' et de la phase 2 - 2' à la phase 1 - 1' le rotor du moteur pas à pas tourne dans le sens horaire.

Dans le moteur pas à pas traité dans ce paragraphe, le pas est égale à  $30^\circ$ , on peut le diminuer en augmentant le nombre des pôles du stator et du rotor.

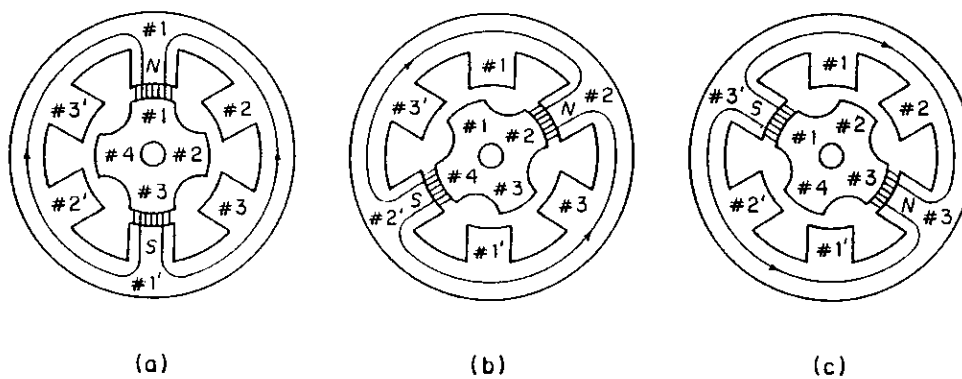


Fig (VII.2) : Position du rotor, (a) :phase (1) excitée,  
(b) : phase (2) excitée, (c) phase (3) excitée.

Si le rotor bouge d'un coté ou de l'autre de la position d'équilibre sous l'effet d'un couple externe, les lignes de forces du champ magnétique deviennent penchées comme indiqué sur la figue (VII.3). La réluctance magnétique est grande et les lignes du champ magnétique génèrent un couple pour aligner le rotor avec le stator.

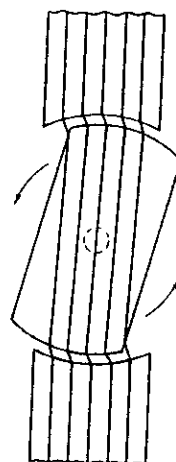


Fig (VII.3) : Les lignes de forces magnétique dans le rotor et le stator.

### 1.2.2 Moteur pas à pas à aimant permanent :

Dans les moteurs pas à pas à aimant permanent le rotor est réalisé avec un aimant permanent. La figure (VII.4) représente un moteur pas à pas à 4 phases. Il possède un rotor cylindrique magnétisé dans la direction radiale.

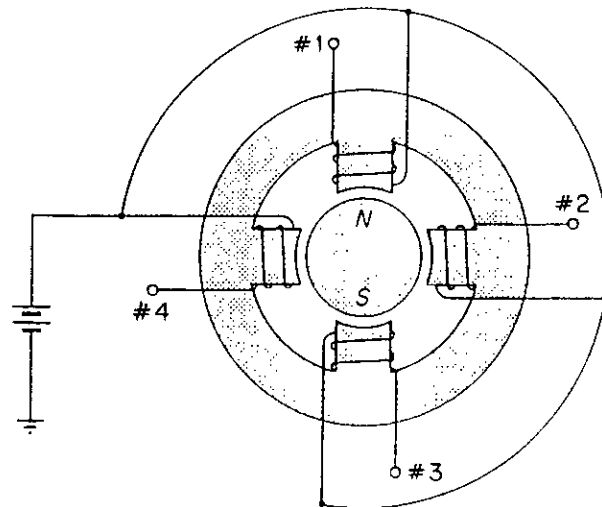


Fig (VII.4) : Moteur pas à pas à aimant permanent à 4 phases.

En se référant à la figure (VII.5), le courant d'excitation traverse la bobine (1) produit un champ magnétique, le pôle sud du rotor s'aligne avec le pôle nord du stator. Quant le courant d'excitation est appliqué à la bobine (2) le pôle sud du rotor tourne en sens horaire de  $90^\circ$  pour s'aligner avec le nouveau pôle nord du stator, et le pas dans ce cas est de  $90^\circ$ .

Quand le courant d'excitation est appliqué sur les bobines respectivement (1), (4), (3), ... le moteur tourne dans le sens anti-horaire avec un pas de  $90^\circ$ .

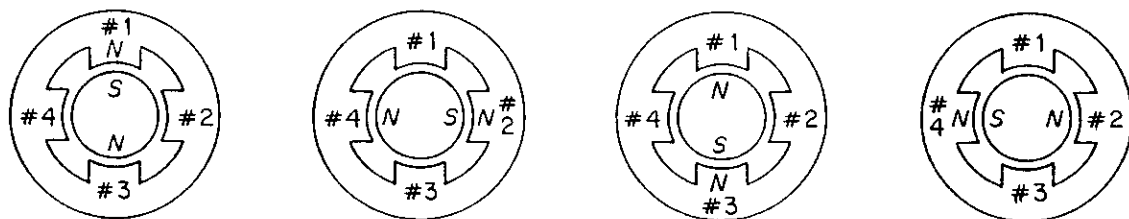


Fig (VII.5) : Les positions stables du moteur pas à pas à aimant permanent à 4 phases.

### 1.2.3 Moteur pas à pas hybride :

Le moteur pas à pas hybride ou magnéto-réductant associe le phénomène de réluctance variable à une excitation par aimant permanent. L'aimant permanent peut être

disposé dans le stator ou dans le rotor, c'est cette dernière disposition qui est la plus courante.

La figure (VII.6) montre une coupe schématique d'un tel moteur. Il comporte un stator portant huit pôles et un rotor comportant deux roues identiques de dix pôles, décalées entre elles d'un demi-pas.

La configuration de ce moteur fait que le flux créé par les bobines n'a pas d'influence de magnétisante sur le flux de l'aimant.

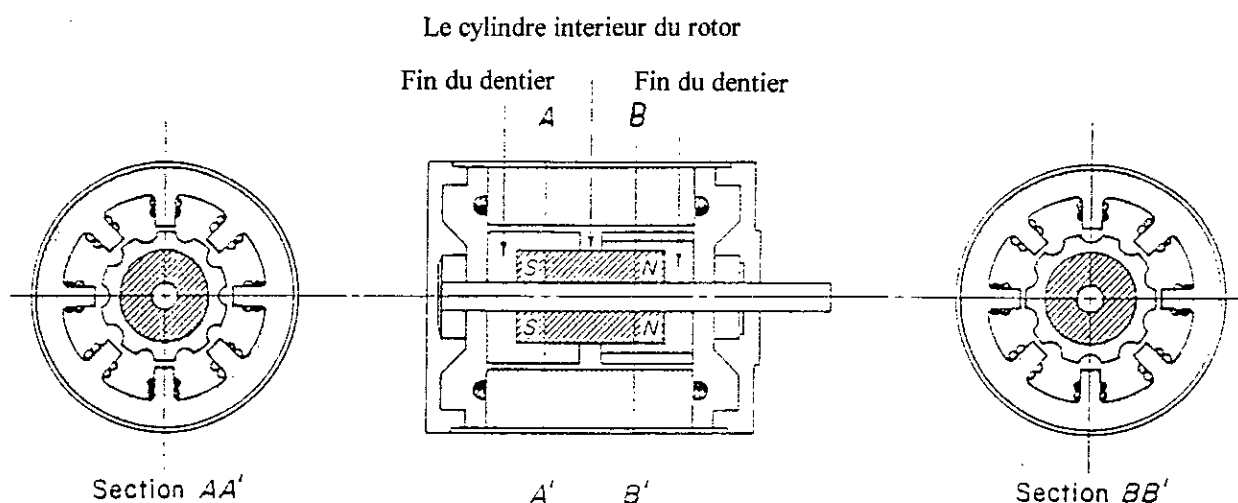


Fig (VII.6) : Moteur pas à pas hybride.

### 1.3 Alimentation : [ 9 ]

Le rôle de l'alimentation est d'assurer une amplification de la commande de façon à appliquer la tension successivement aux différentes phases, le couple étant proportionnel au courant ou au carré du courant. Cette alimentation doit s'approcher le plus possible d'une source de courant.

Certains moteurs atteignent des fréquences de l'ordre de 20000 pas par seconde. Ces enclenchements et déclenchements, successifs doivent alors s'effectuer à un rythme de plusieurs KHz.

La solution à la base de toute alimentation comprend des transistors fonctionnent alternativement en régime saturé (conduction) et bloqué. Ils sont commandés par un aiguillage logique attaquant les bases des transistors. ( fig (VII.7) )

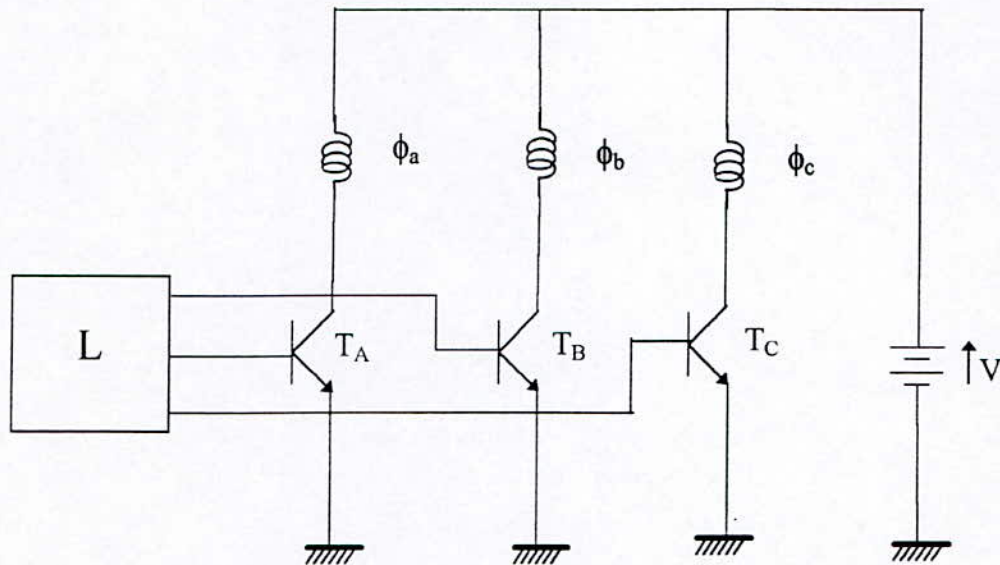
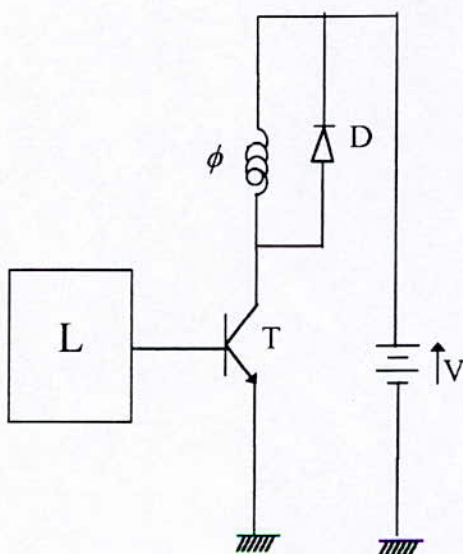


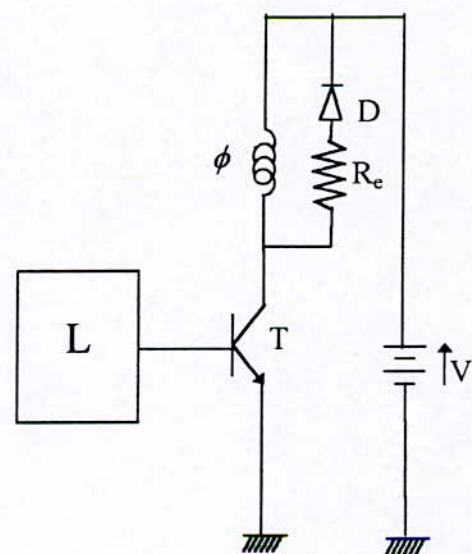
Fig (VII.7) : Alimentation de base d'un moteur triphasé.

**Circuit d'extinction :**

Pour le montage de la figure (VII.7), le blocage d'un transistor a pour conséquence l'apparition d'une surtension à ses bornes, qui peut alors produire un claquage de celui-ci. Afin d'éviter un tel défaut, un circuit d'extinction est indispensable. Sous sa forme la plus simple, il comprend un circuit en parallèle avec chaque bobinage, une diode assure le blocage du courant dans celui-ci lorsque la phase est alimentée (figure (VII.8)). Elle permet la décroissance progressive du courant après blocage du transistor correspondant.



Fig(VII.8)



Fig(VII.9)



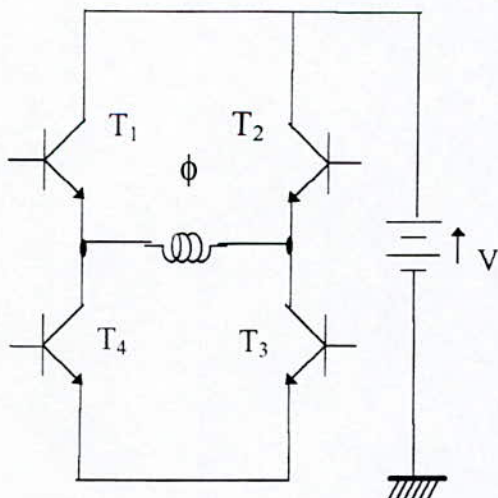
Lors d'une commutation de phase, le courant s'éteint avec une constante de temps dépendant de l'induction moyenne du bobinage et sa résistance. Selon la fréquence de fonctionnement, il peut être intéressant de diminuer cette constante de temps. Ceci peut être obtenu par l'introduction d'une résistance dans le circuit d'extinction.

La figure (VII.9) montre un exemple de montage utilise cette technique. Le choix de la résistance résulte d'un compromis entre la réduction de la constante de temps et la limitation de la surtension de blocage.

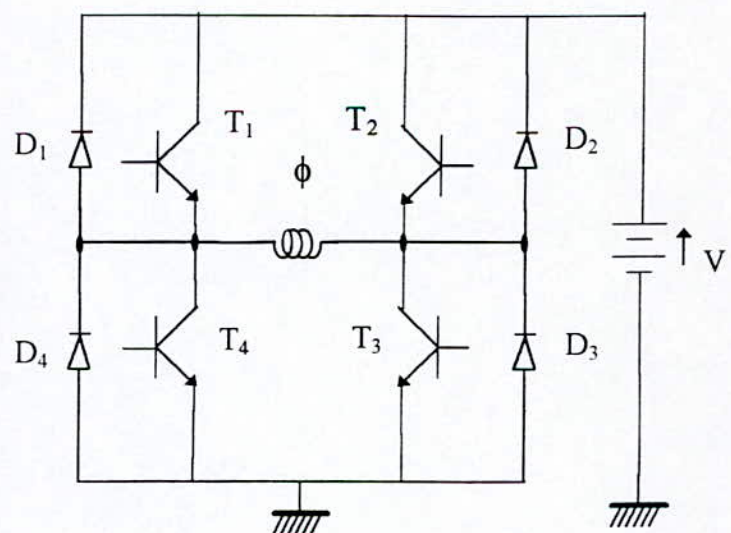
### Alimentation bipolaire :

Pour un moteur comportant un aimant permanent, une alimentation de caractère bipolaire est nécessaire. Deux solutions sont généralement utilisées dans ce cas :

- Chaque phase est dédoublée. L'un des bobinages est alimenté dans un sens et l'autre en sens inverse. Si cette solution n'est pas très rationnelle en ce qui concerne l'utilisation des matériaux, elle présente l'avantage de permettre des fréquences plus élevées.
- Une véritable alimentation bipolaire assure la circulation du courant dans les deux sens. On doit recourir à un montage en H selon le principe de la figure (VII.10). Les transistors 1 et 3 sont saturés ou bloqués simultanément, inversement les transistors 2 et 4 sont bloqués ou saturés simultanément. La figure (VII.11) montre la disposition des diodes des circuits d'extinction.



Fig(VII.10) : Alimentation bipolaire en H



Fig(VII.11) : Alimentation bipolaire en H avec circuit d'extinction.

### 1.4 Amélioration des performances au démarrage : [ 9 ]

Tout moteur pas à pas est caractérisé par une fréquence limite de démarrage, au delà de laquelle le moteur reste arrêté ou perd rapidement son synchronisme. Cette fréquence est fonction du couple résistant et de l'inertie entraînée. Tout accroissement de cette limite permet de réduire la durée de démarrage ou de freinage.

On constate que le seul degré de liberté permettant d'agir sur cette fréquence est le couple accélérateur, donc le couple moteur. Le pas et l'inertie sont généralement imposés par la charge. Toute solution visera donc à accroître le couple accélérateur sans changer le type de moteur. Deux solutions sont possibles :

- **Accroissement momentané de la tension :**

Selon le type de moteur, le couple est proportionnel au carré du courant. Tout accroissement de ce courant permet donc d'augmenter le couple. Pratiquement, cet effet est réalisé par un survoltage passager du moteur.

Deux phénomènes en limitent l'emploi :

- \* L'échauffement accru qui en résulte ;
- \* La saturation qui atténue progressivement l'accroissement possible.

- **Superposition de phases :**

En alimentant simultanément deux phases, le couple résultant est la somme (sans saturation) des couples partiels. La valeur moyenne est ainsi augmentée.

Toute combinaison intermédiaire (entre la non superposition et la superposition complète) dans la séquence d'alimentation des phases est possible.

### 1.5 Accélération et décélération :

Pour assurer un transfert de position en un temps minimum ou pour imposer une fréquence élevée déterminée. Une phase transitoire d'accélération est indispensable. Inversement, une période de décélération est nécessaire afin d'assurer un arrêt sur un seul pas. Malgré le caractère symétrique de ces deux problèmes, la décélération est plus délicate.

En circuit ouvert, l'accélération et la décélération sont assurée en imposant une rampe de fréquence en fonction du temps ou de la position. Voir la figure (VII.12).

La décélération doit intervenir avec une marge de sécurité suffisante quelque soit le régime établi. Au moyen d'un compteur - décompteur assurant la comparaison entre le nombre de pas parcourus et le nombre de pas à parcourir, il s'agit de déclencher le processus de décélération suffisamment tôt.

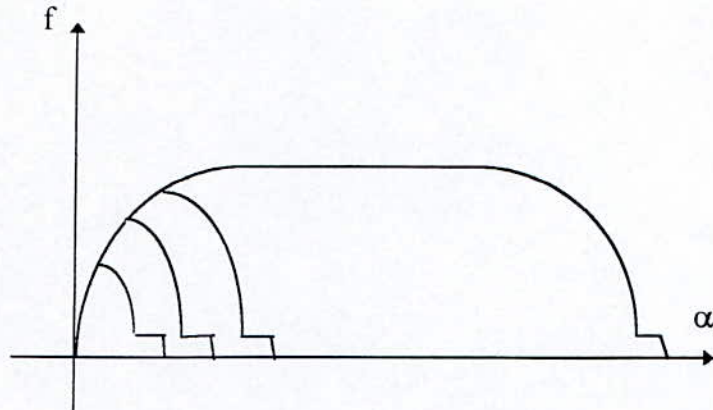


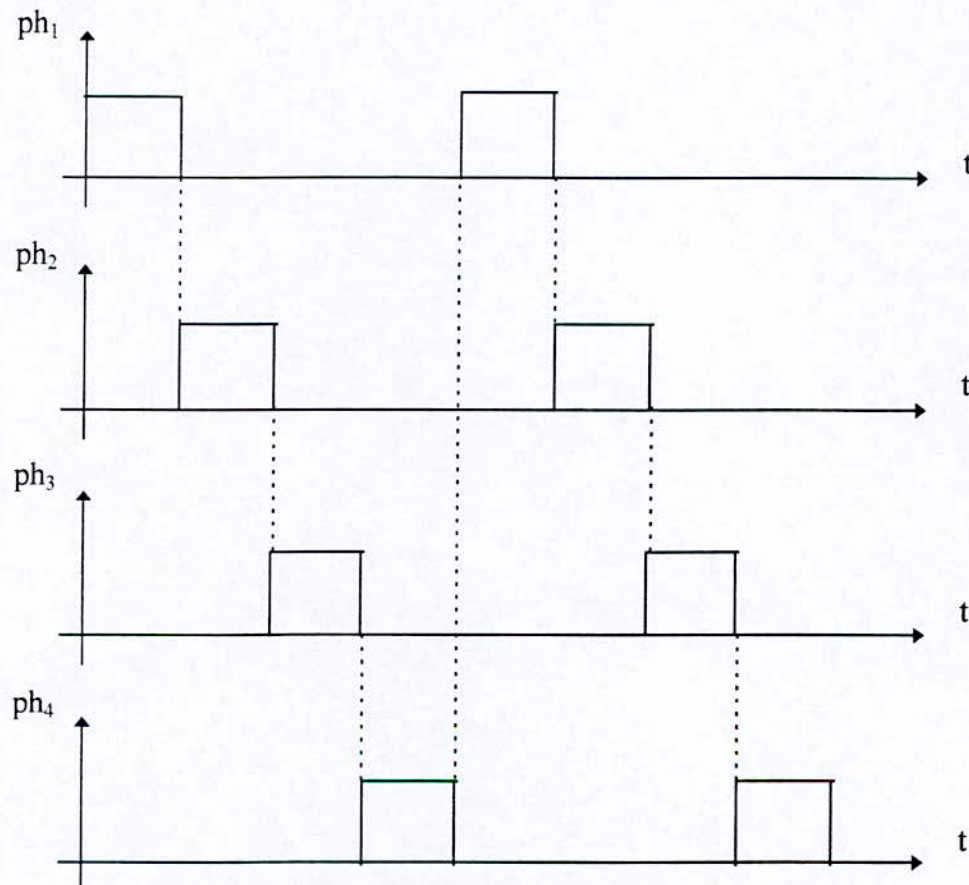
Fig (VII.12) : Séquences de transfert de position.

### 1.6 Modes d'excitation :

La rotation du moteur s'effectue par une séquence de permutation circulaire des configurations d'excitation dans un sens ou dans l'autre. Les excitations actuelles peuvent être classées selon les modes suivants :

#### 1.6.1 Mode d'excitation monophasé :

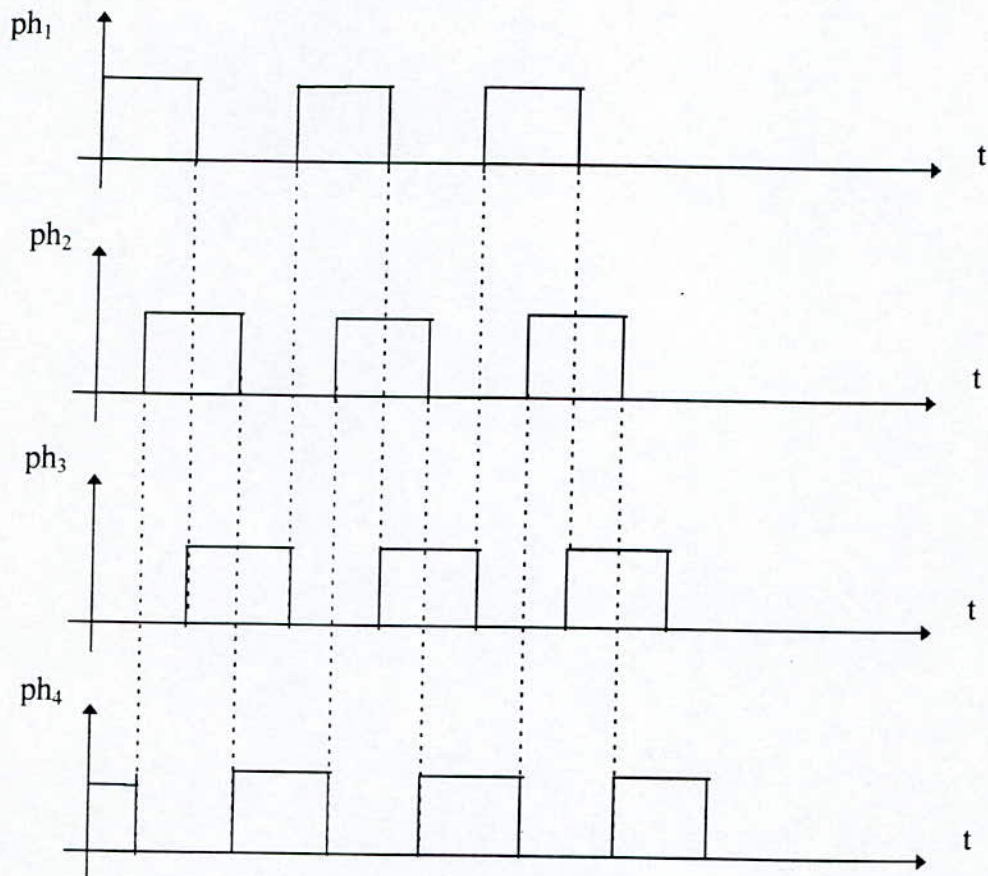
La figure (VII.13) montre le chronogramme des signaux d'excitation d'un moteur à quatre phases.



Fig(VII.13) : Chronogramme d'excitation monophasé.

### 1.6.2 Mode d'excitation double phase :

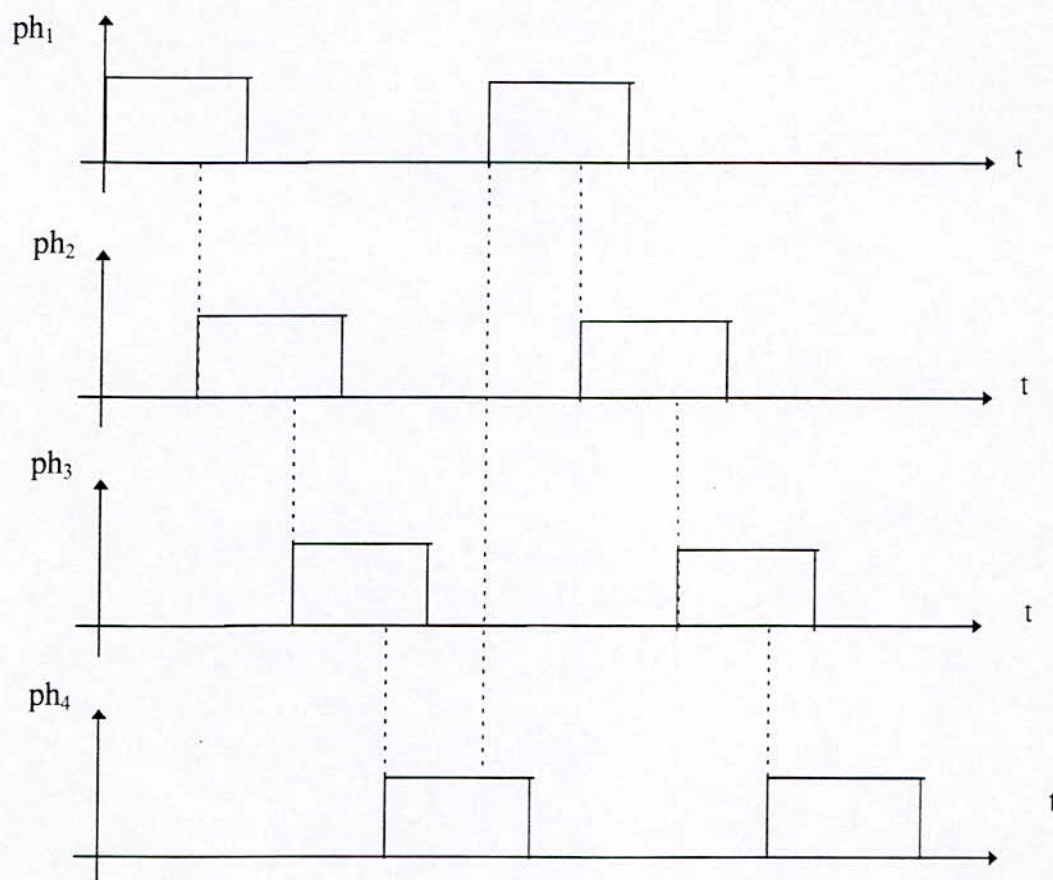
C'est le mode d'excitation le plus ancien et le plus répondu .La figure (VII-14) montre le chronogramme des signaux d'excitation d'un moteur à quatre phases. Dans ce mode deux phases du moteur sont excitées à la fois.



Fig(VII.14) : Chronogramme d'excitation double phase.

### 1.6.3 Mode d'excitation à demi-pas :

La combinaison en alternance des deux modes précédents permet un fonctionnement en demi-pas, la figure (VII-15)montre le chronogramme d'un tel mode.



Fig(VII.15) : Chronogramme d'excitation à demi-pas.

## 2. CONCEPTION DE L'ETAGE DE PUISSANCE : [ 7 ]

Pour ce qui est de notre application, notre choix s'est porté sur le moteur à aimant permanent. En effet, ce type de moteur est de conception simple et robuste, de plus, il présente un couple élevé.

En ce qui concerne le mode d'excitation, notre choix s'est posé sur le mode monophasé. Ce mode offre la plus faible consommation de courant pour des performances à peu près égales aux autres modes.

Les signaux fournis par le microcontrôleur n'excédant pas quelques milliampères, il est impératif d'insérer un étage de puissance capable de rehausser ces derniers afin de permettre la commande du moteur pas-à-pas dont la consommation en courant peut atteindre plusieurs ampères.

La figure (VII.16) illustre le schéma adopté pour cet étage. Dans la conception de cet étage de puissance, nous avons opté pour l'utilisation de transistors MOSFET.

Effectivement, ces derniers offrent de grande vitesse de commutation en plus d'une stabilité thermique accrue et sont aptes à faire passer des courants intenses ce qui les dispose tout naturellement pour des commutations de puissance telle que la commande du moteur pas-à-pas.

### 3. FONCTIONNEMENT DE L'ETAGE DE PUISSANCE :

Le blocage du transistor bipolaire provoque la saturation du MOSFET, ainsi l'enroulement sera alimenté. Une diode de roue libre est insérée en parallèle de l'enroulement pour éviter les surtensions.

La saturation du transistor bipolaire bloque le MOSFET déconnectant l'enroulement des moteurs pas-à-pas. Le port « B » attaque directement les transistors bipolaire à travers des résistances de limitation du courant. De plus la tension de 24 volts est abaissée à l'aide d'un pont diviseur pour éviter une tension sur le gate du MOSFET supérieure à 20 volts qui est la limite admise par ce dernier.

### 4. CALCUL PRATIQUE :

Une tension  $V_{GS} = 6$  volts est largement suffisante pour satisfaire les besoins de la commande, de ce fait, on fixe  $V_{cc} = 6$  volts.

$$\frac{V_{dd}}{V_{cc}} = 4 \Rightarrow \frac{R_1}{R_2} = 3$$

Comme le MOSFET est sensible à la tension et non au courant à son entrée, nous avons décidé de fixer le courant collecteur du transistor bipolaire à 2 mA pour des raisons d'économie.

On choisi :  $I = 10 I_c = 20$  mA

$$R_1 = (V_{dd} - V_{cc})/I = 900 \Omega \approx 1k\Omega$$

$$R_2 = R_1/3 = 300 \Omega$$

$$R_e = V_{cc}/I_c = 3k\Omega$$

$$R_b = \beta(V_e - V_{be})/I_c = 22k\Omega$$

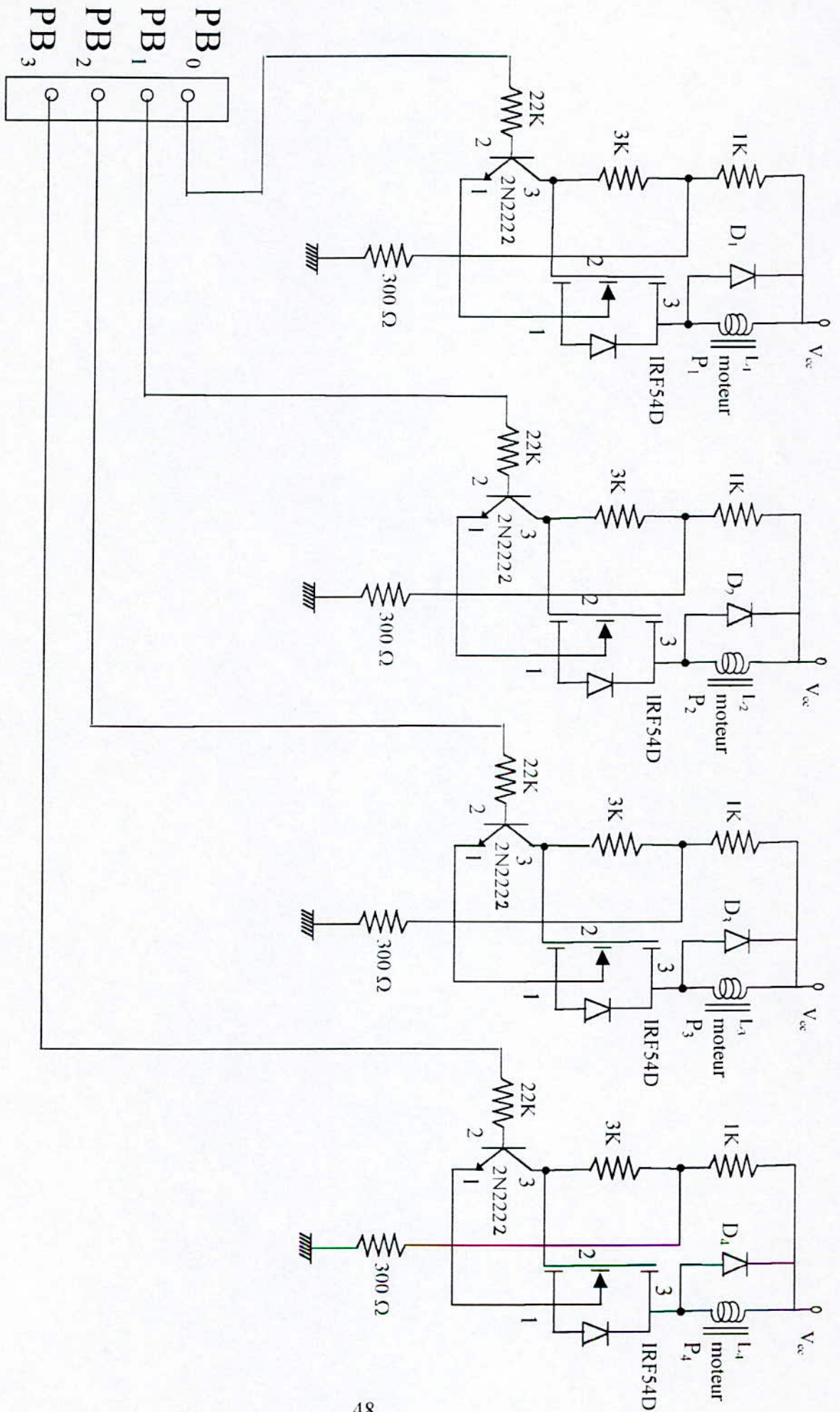


Fig (VII.16) : Schéma pratique de l'étage de puissance.

**Chapitre VIII****PROGRAMME DE CALCUL ET  
DE GESTION DU SYSTEME.**

Dans cette partie, on va développer les organigrammes nécessaires pour le fonctionnement du système. L'organigramme de la figure (VIII.1) donne les grandes lignes à suivre pour réaliser cette fonction.

A la mise sous tension, le microcontrôleur initialise le système et se met en attente. L'appui sur une touche du clavier va créer un front montant sur la ligne PB3, et la partie du programme de gestion de clavier est exécutée. Après cette étape, le microcontrôleur calcule la position optimale du panneau, détermine le nombre exact de pas et commande le moteur. La dernière étape consiste à mémoriser la dernière excitation puis à effectuer le retour au début du programme principal.

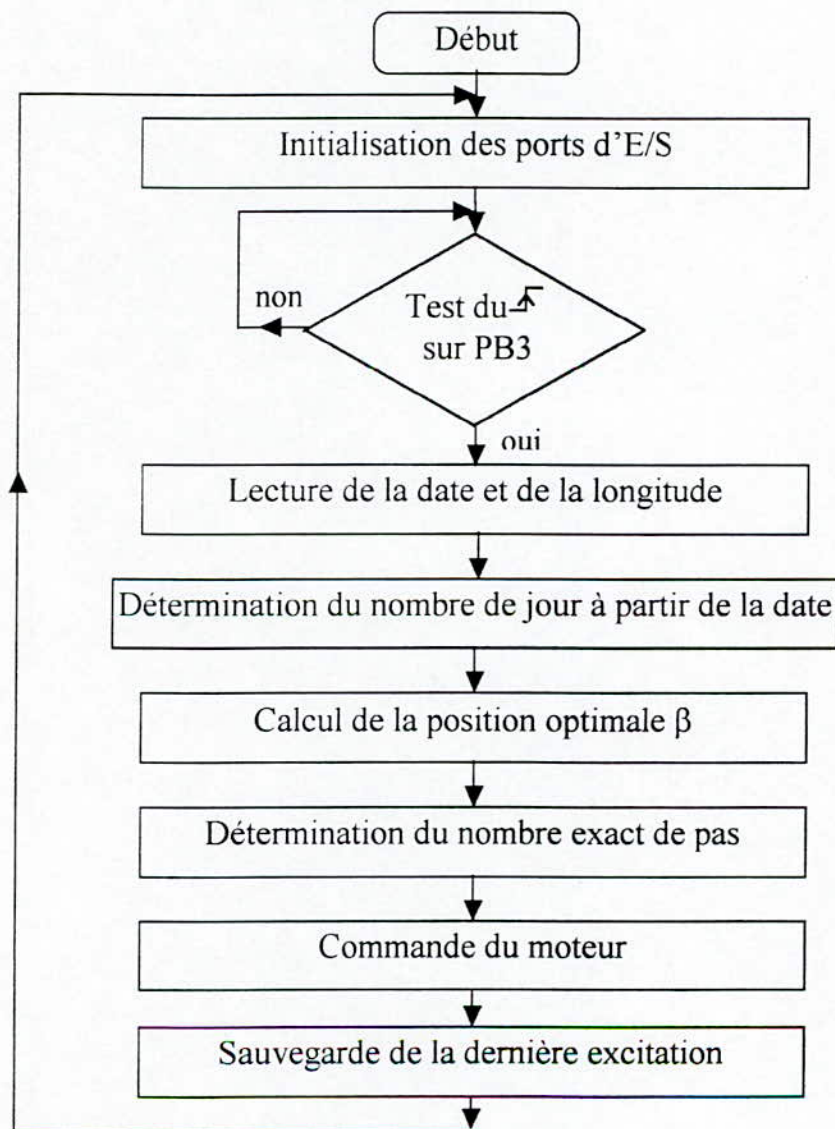


Figure (VIII.1) : Fonctionnement général du système.



## 1.CALCULE DE LA POSITION OPTIMALE :

Dans notre réalisation, le programme qui calcul la position optimale nécessite la réalisation de plusieurs opérations sur des nombres réels (addition, multiplication, division, cosinus, sinus,...,etc). Mais, l'unité centrale du microcontrôleur 8 bits ne réalise que les opérations arithmétiques suivantes : addition binaire 8 bits, soustraction binaire 8 bits et quelques fonctions de décalage et de rotation. Il nous faut donc réaliser ces fonctions par programme.

### 1.1 Représentation des nombres réels :

Il existe deux méthodes principales pour représenter les réels :

- La méthode de représentation en virgule flottante ;
- La méthode de représentation en virgule fixe.

La première méthode est un peu compliquée. Elle consiste à représenter un réel en deux parties. Une mantisse signée et un exposant signé.

La deuxième méthode est plus facile à utiliser et consiste à représenter un réel avec sa partie entière, et sa partie fractionnelle sur un nombre fixe de bits.

Dans les paragraphes qui suivent, on utilise la représentation des réels en virgule fixe avec deux octets pour la partie entière et deux octets pour la partie fractionnelle. Les nombres négatifs sont représentés en complément à deux, c'est-à-dire, le 32<sup>ème</sup> bit représente le signe. Donc toutes les opérations arithmétiques vont se réaliser sur quatre octets.

Dans toute la présentation qui va suivre, nous utiliserons la notation  $M_i$  pour désigner une position mémoire  $i$ .

### 1.2 L'addition des nombres réels :

L'organigramme suivant réalise l'addition binaire pour deux réels représentés chacun sur quatre octets. On a choisi la transmission d'argument par mémoire, les deux opérands constituent une table de données de 8 octets. On transmettra au sous-programme l'adresse de la première donnée à manipuler par l'intermédiaire du registre index X. Le principe de cette addition est illustré par l'organigramme suivant :

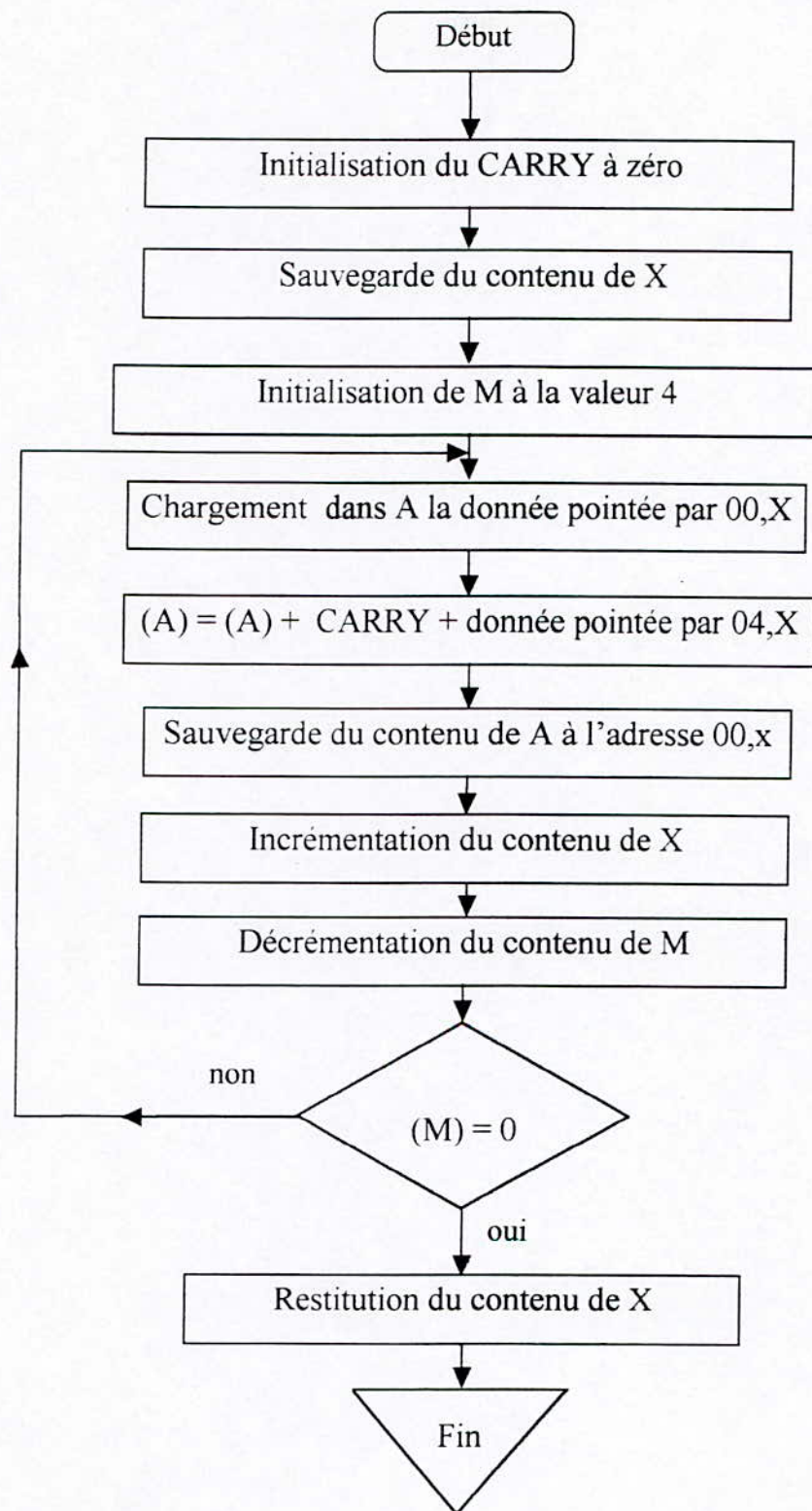
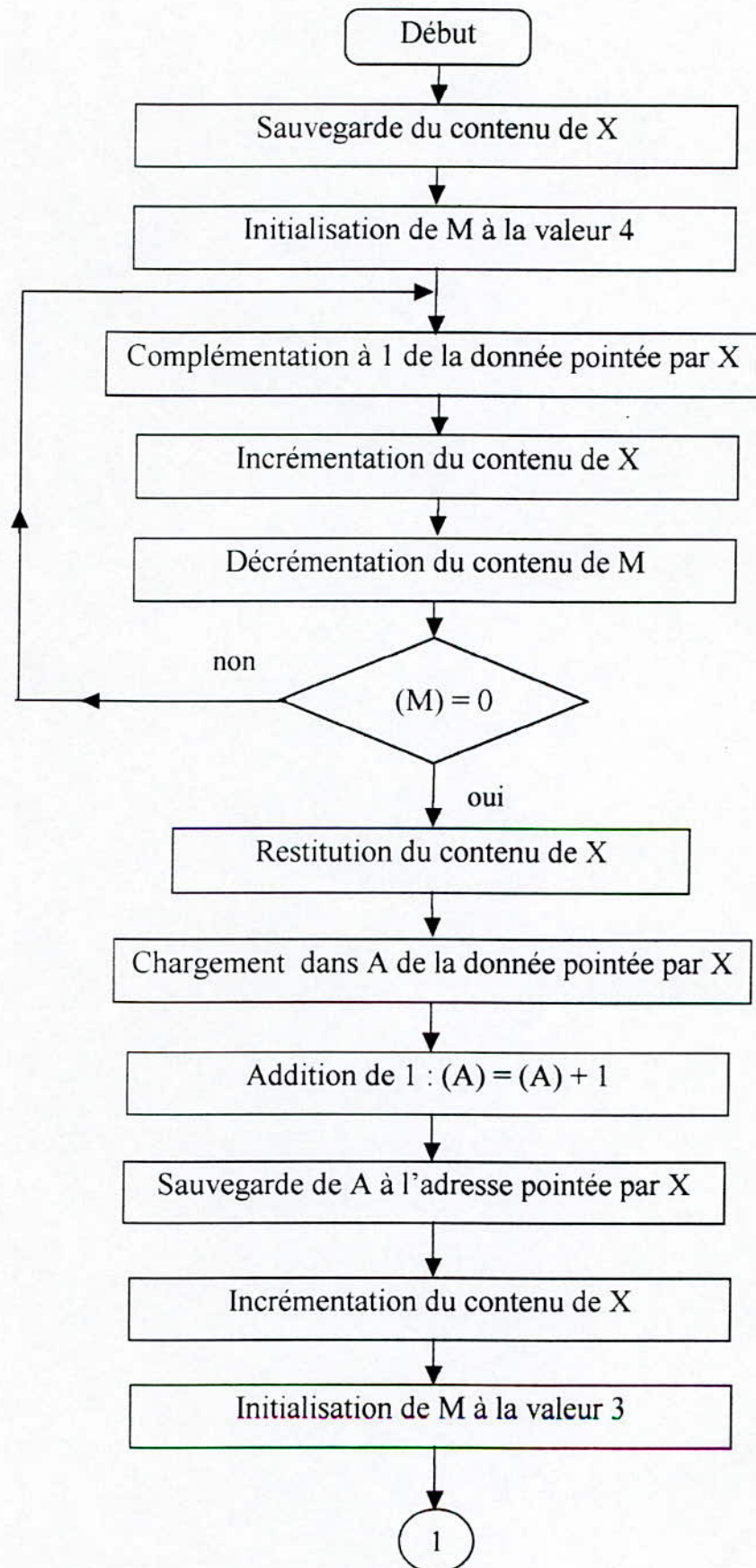


Fig (VIII.2) : Organigramme de l'addition binaire.

### 1.3 Le complément à deux :

Dans les sous-programmes, multiplication, division, et soustraction, on a besoin de réaliser le complément à deux sur quatre octets d'un opérande de donnée. L'organigramme suivant réalise cette fonction, la transmission d'argument et réalisée

comme précédemment et l'adresse de début de la donnée est transmis par le registre index X.



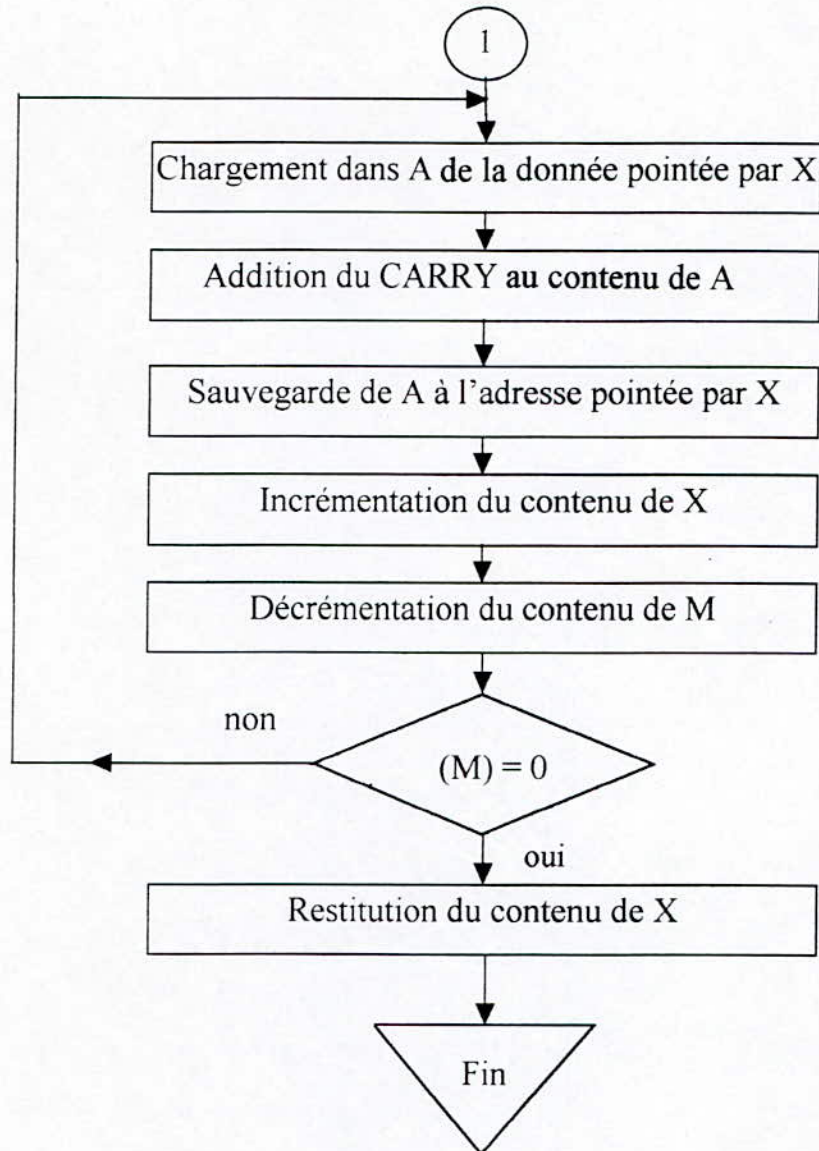


Fig (VIII.3) : Organigramme du complément à 2.

#### 1.4 La soustraction des nombres réels :

Au lieu d'écrire un programme qui réalise la soustraction de deux opérandes sur quatre octets, on peut réaliser cette opération par addition et complément à deux, c'est-à-dire qu'on fait le complément à 2 du deuxième opérande puis on fait appel au sous programme d'addition.

#### 1.5 La conversion BCD - Hexadécimal :

Le calcul de la position du soleil nécessite la connaissance de la date et de la longitude. Ces données sont fournies par un opérateur par l'intermédiaire d'un clavier binaire en BCD.

Comme dans notre application, les calculs à l'intérieur du microcontrôleur s'effectuent en binaire, il faut prévoir un sous-programme qui fait la conversion des données du BCD en Hexadécimal.

Pour écrire un nombre BCD en Hexadécimal, il faut retrancher à ce nombre autant de fois 6 qu'il comprend de dizaines. Cette procédure est illustrée par l'organigramme suivant :

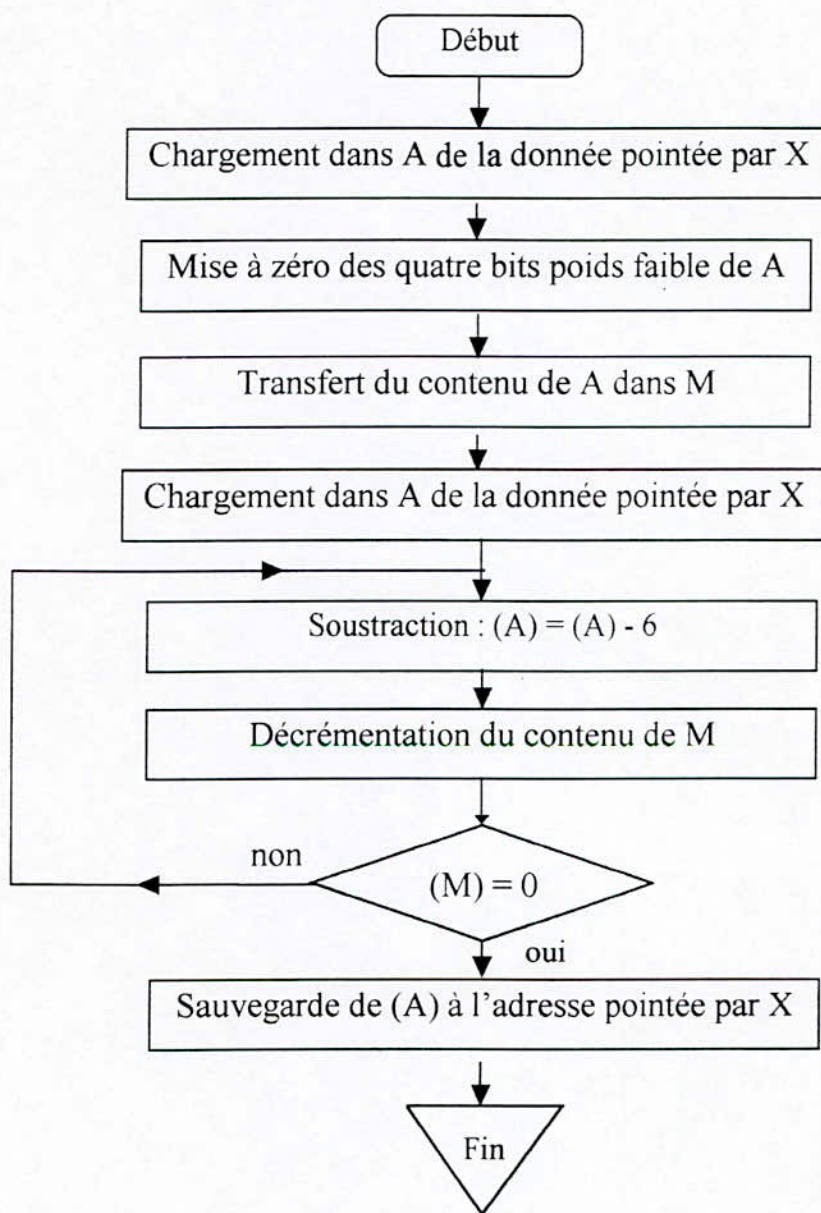


Fig (VIII.4) : Organigramme de la conversion BCD → Hexadécimal.

### 1.6 La conversion Hexadécimal - BCD :

L'angle optimal calculé par le microcontrôleur est exprimé en hexadécimal. Pour pouvoir afficher cet angle, il faut l'écrire en BCD. Le principe consiste à retrancher autant de fois que possible 10, dont l'écriture hexadécimale de l'équivalent binaire est 0AH, du nombre binaire. Le nombre de soustraction possible est bien entendu le nombre des dizaines de la valeur décimale à trouver. Le reste donne les unités.

L'adresse de la donnée à convertir est transmis au sous-programme par l'intermédiaire du registre index. L'organigramme suivant illustre ce principe.

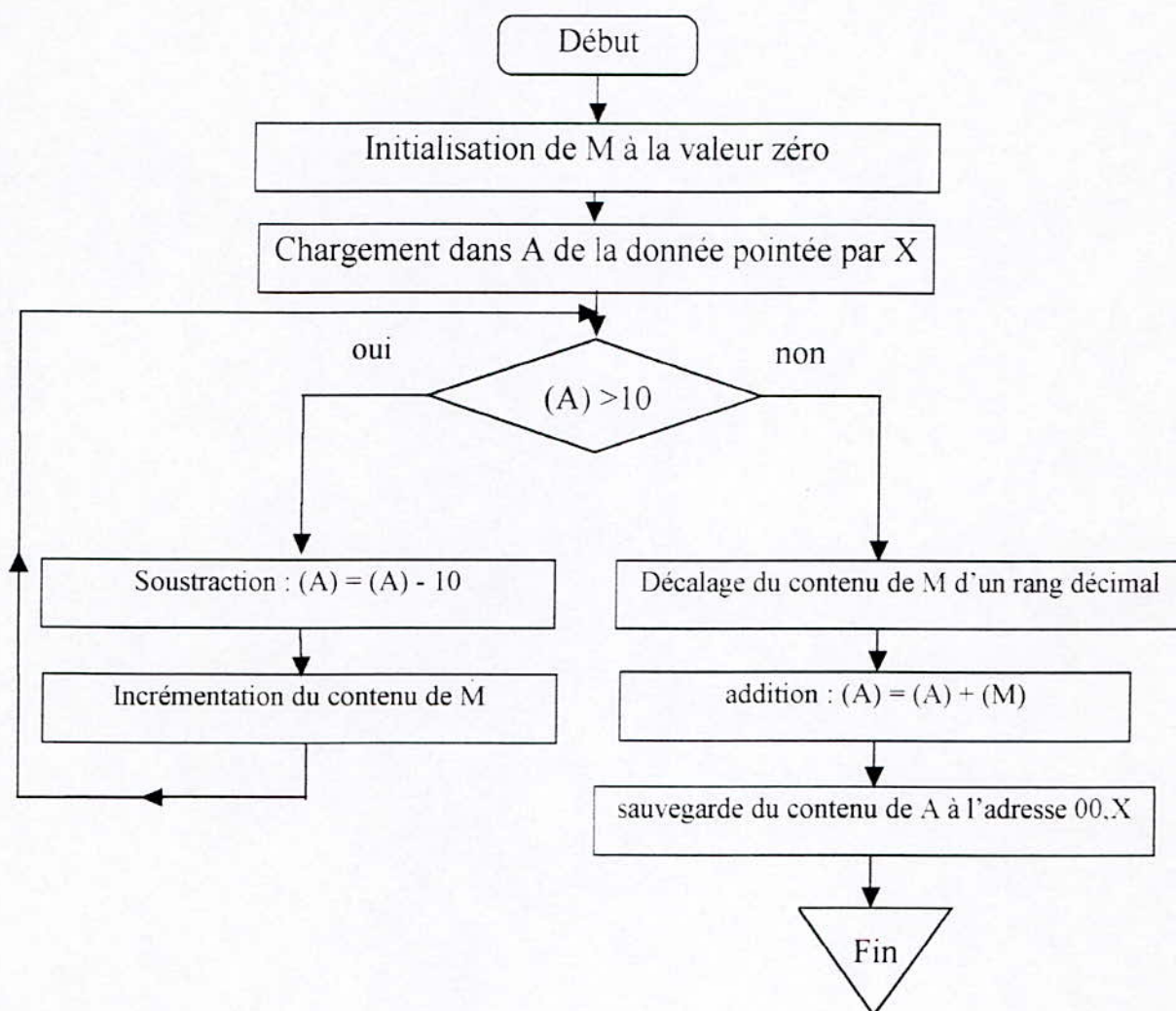


Fig (VIII.5) : Organigramme de la conversion Hexadécimale - BCD.

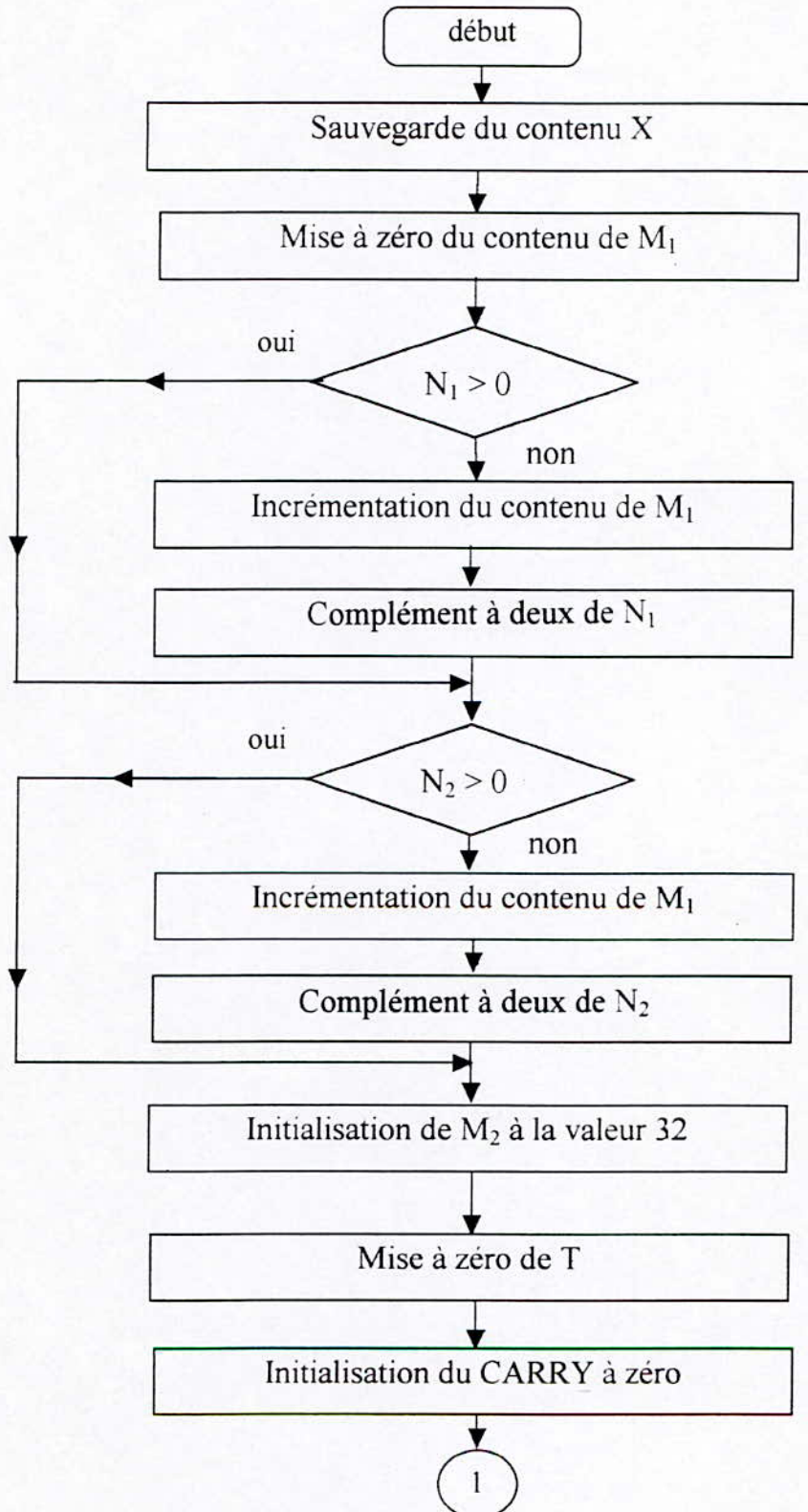
### 1.7 La multiplication des nombres réels signés :

Ce sous-programme réalise la multiplication de deux opérandes de 4 octets chacun. Les opérandes transmis au sous-programme doivent être placés dans une table comme représentée sur la figure (VIII.7). L'adresse de début de cette table est transmis au sous-programme par l'intermédiaire du registre d'index X. Le résultat de la multiplication est représenté sur les 8 octets du poids faible de la table de donnée.

Le processus de multiplication est alors le suivant :

1. Le bit le moins significatif du premier opérande est testé.
  - si ce bit vaut 1 le deuxième opérande est ajouté au contenu de T. ( T représente les quatre positions mémoires où le résultat de l'addition est sauvegardé) ;
  - si ce bit vaut 0 l'addition ci-dessus n'est pas faite ;

2. Le double contenu du premier opérande et T est décalé d'un rang binaire vers la droite, le LSB de T allant dans le MSB du premier opérande ;
3. Les opérations (1) et (2) sont répétées 32 fois ;  
Ce processus est illustré par l'organigramme suivant :



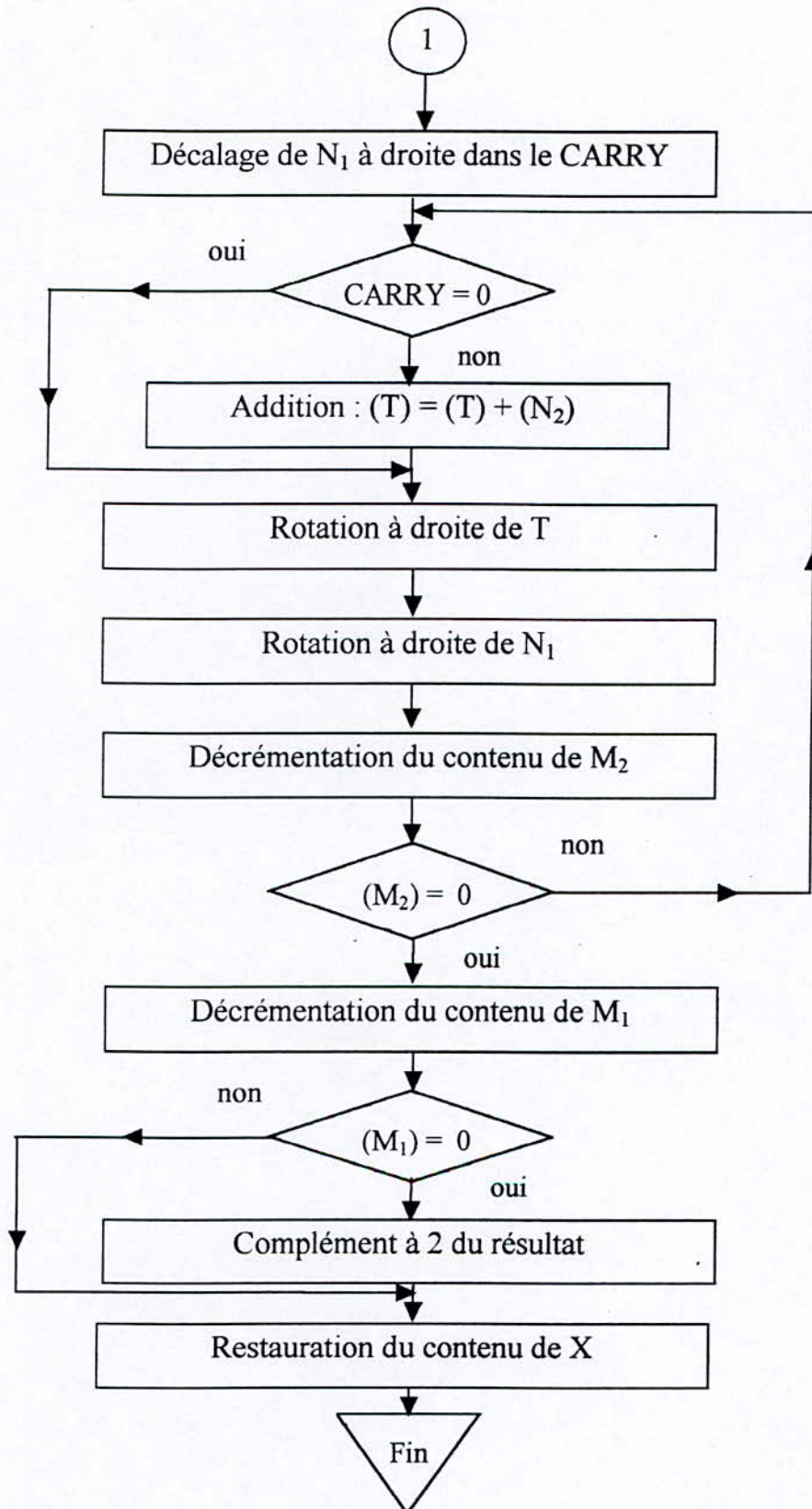


Fig (VIII.6) : Organigramme de la multiplication signée.



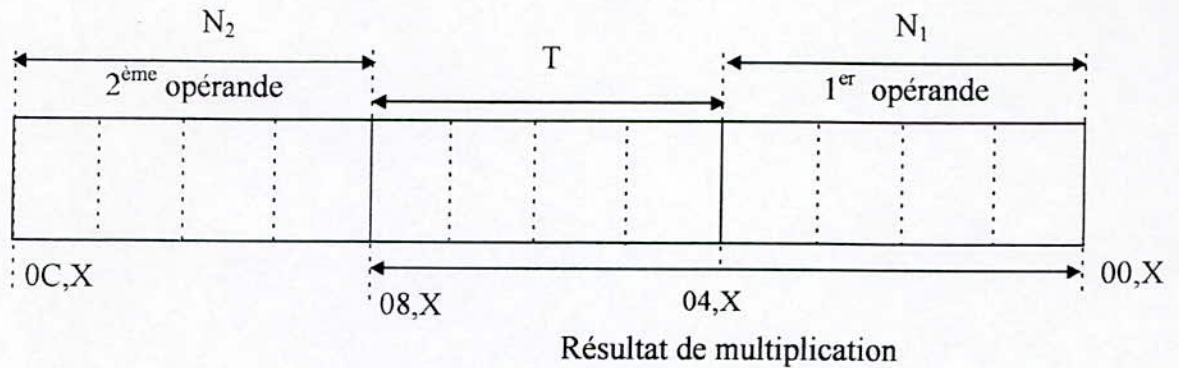


Fig (VIII.7) : La représentation de la table de donnée transmise au sous-programme.

### 1.8 La division des nombres réels signés :

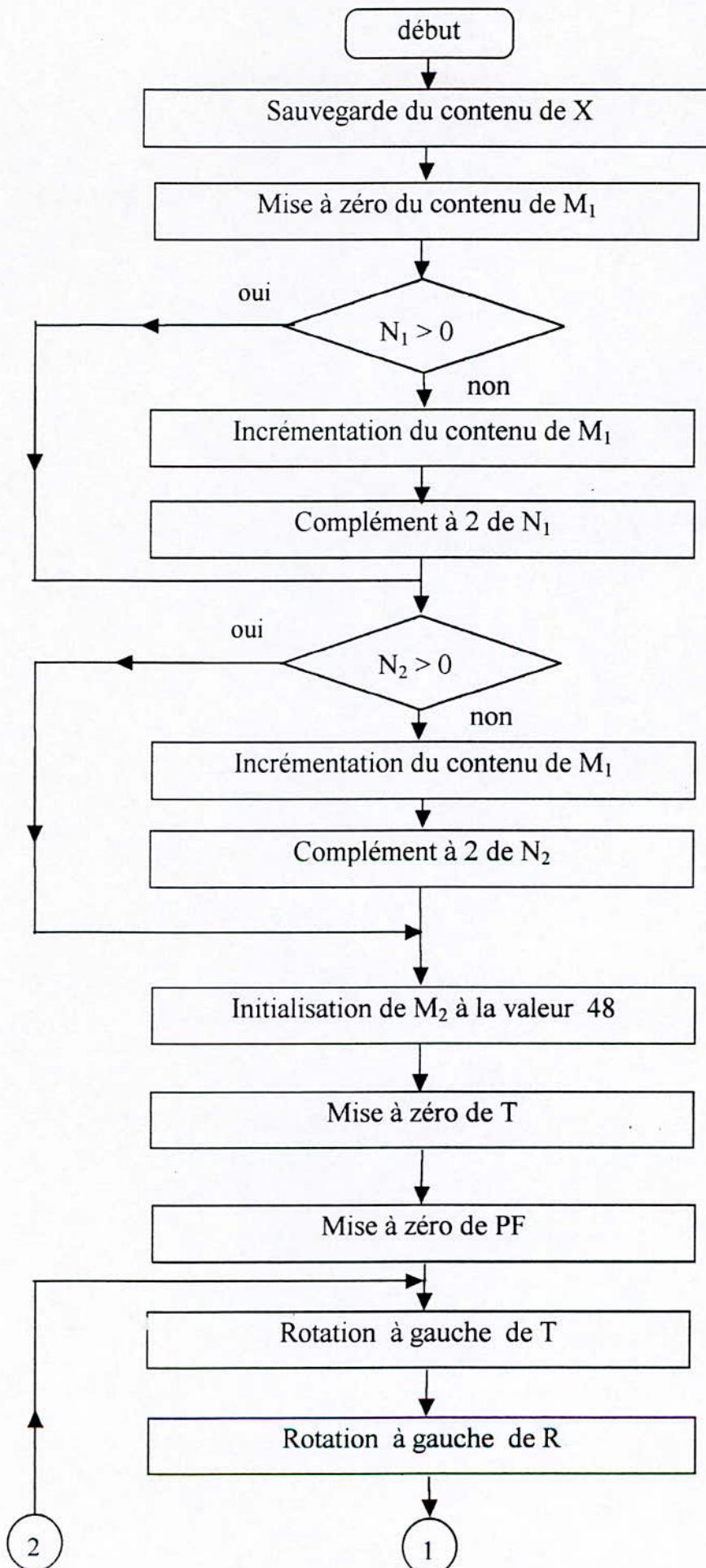
Ce sous-programme réalise la division binaire signée sur deux réels représentés chacun sur quatre octets. Le diviseur doit être installé dans les quatre octets poids forts de la table de données transmise au sous programme, voir figure (VIII-9). Le dividende doit être installé du troisième au sixième octet à partir du début de la table comme représenté sur la figure (VIII.9). Le résultat de la division est représenté sur les quatre octets de poids faibles de la table de données. L'adresse de début de la table est transmise au sous-programme par l'intermédiaire du registre index X.

Le quotient est élaboré bit après bit dans R. Chaque bit est en réalité le complément de la retenue pour chacune des soustractions (T) - diviseur. En réalité, nous élaborions bit après bit , le complément du quotient, lorsque la division sera terminée nous complimenterons ce quotient.

Le processus est le suivant :

1. Rotation du double contenu (R) et (T) d'un rang binaire à gauche.
2. Essai de soustraction : (T) - diviseur, si la soustraction est possible, la retenue est zéro ; si elle n'est pas possible, la retenue est 1 et le contenu de T avant la soustraction est restauré dans T par l'addition (T) = (T) + diviseur.
3. Les opérations 1) et 2) sont répétées 48 fois.
4. On fait le complément à 1 du résultat.

Le processus est illustré par l'organigramme suivant :



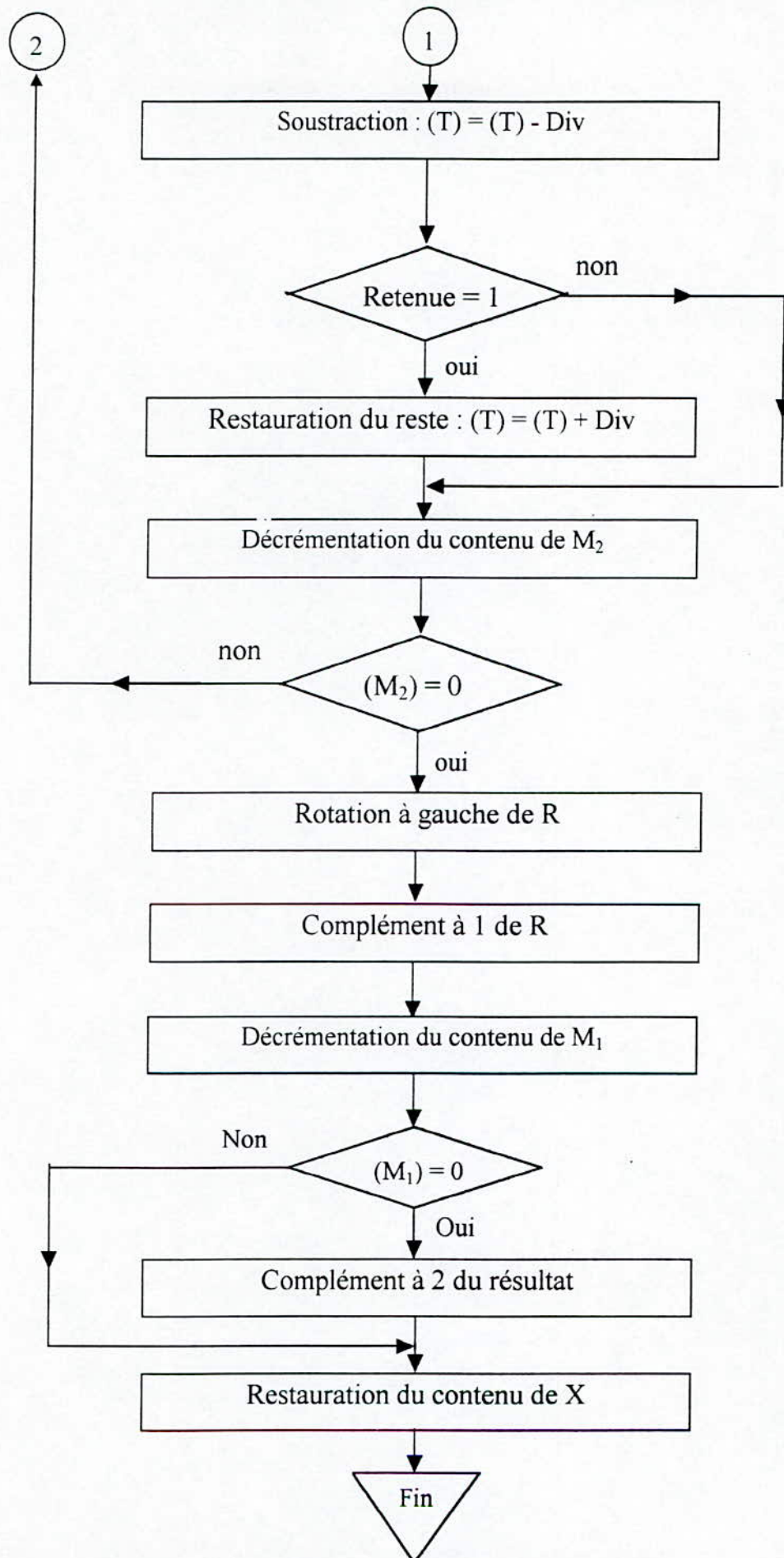
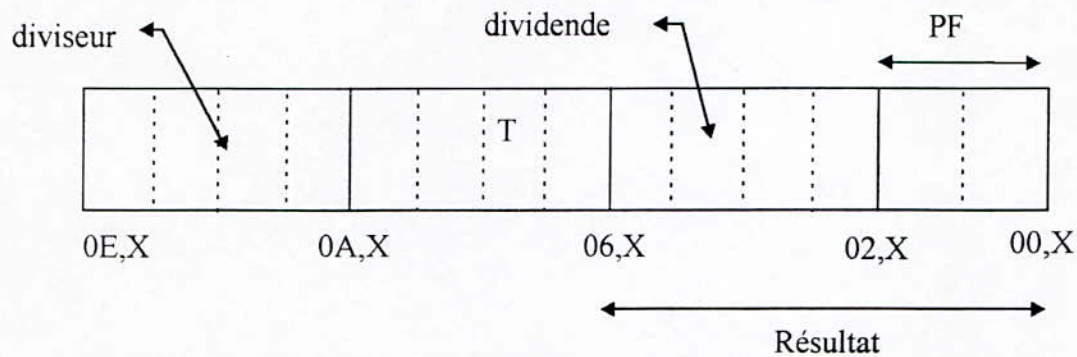


Fig (VIII.8) : Organigramme de la division binaire signée.



Fig(VIII.9) : La représentation de la table de données transmise au sous-programme.

### 1.9 La fonction trigonométrique sinus :

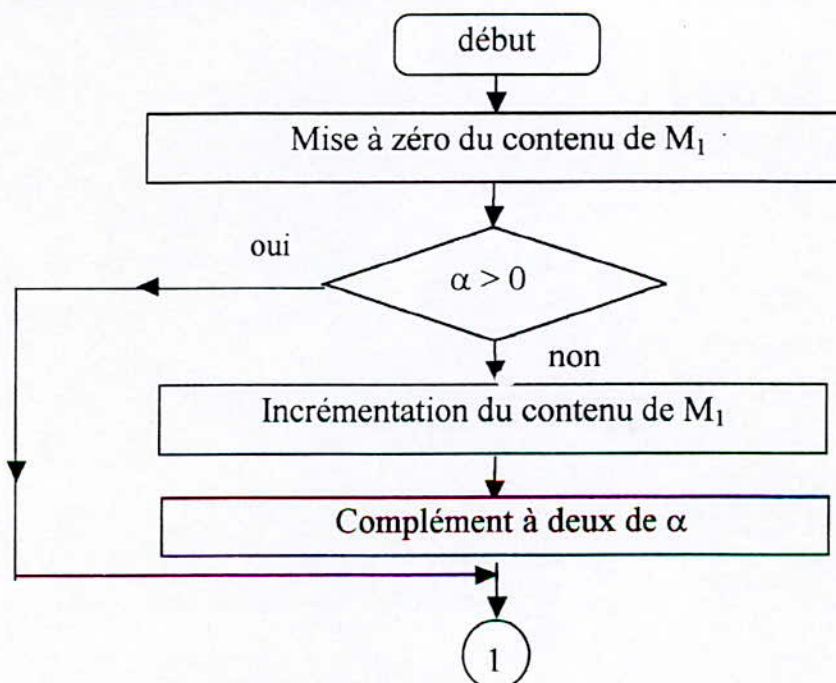
Ce sous-programme calcule la fonction trigonométrique sinus à l'aide d'un oscillateur sinusoïdal basse fréquence 2.7 Hz ( $T = 360\text{ms}$ ). On a choisi une période de 360ms pour faciliter la génération de la temporisation (360 ms équivalent à  $360^\circ$ ).

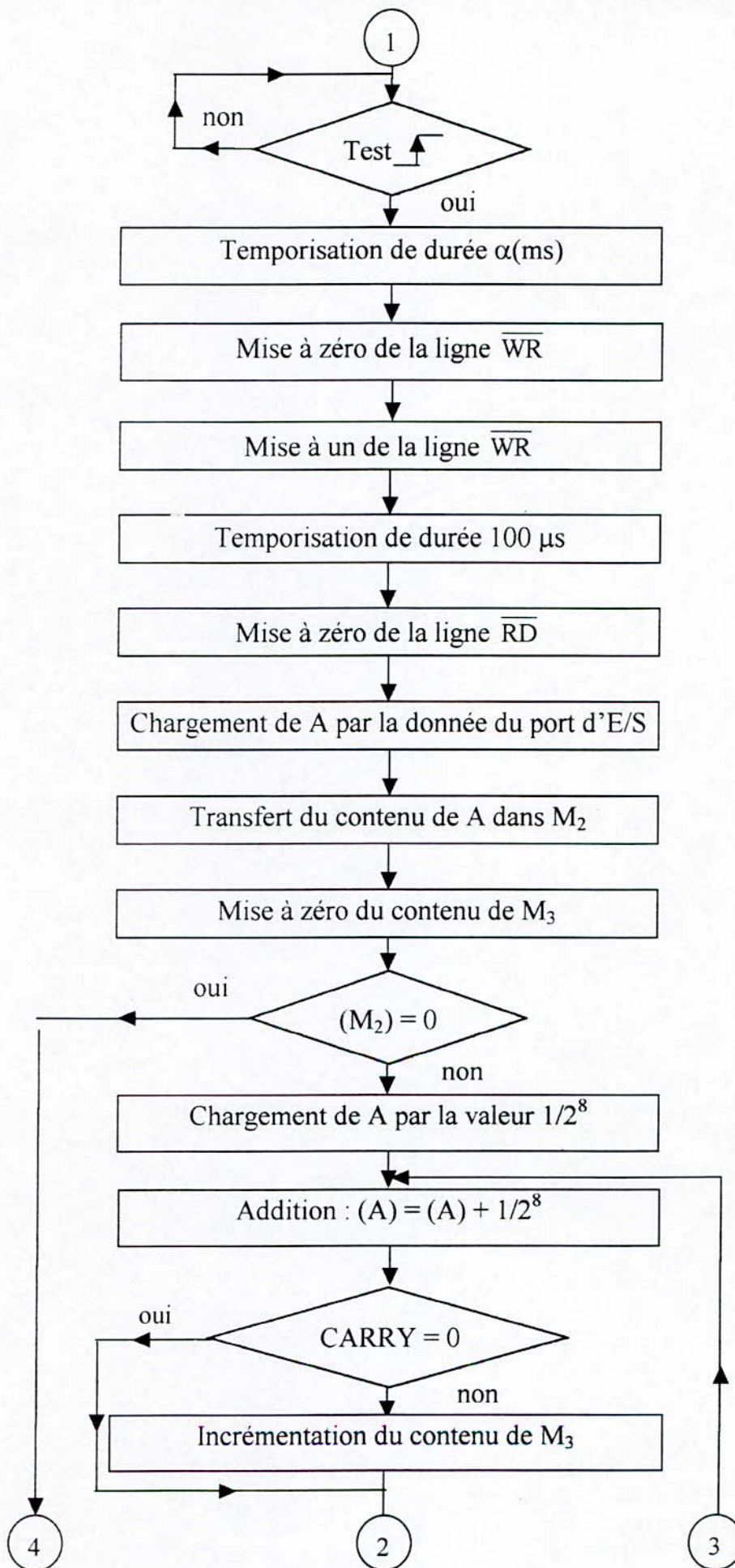
Le calcul du sinus nécessite la connaissance du temps de passage par zéro de la tension sinusoïdale avec précision. Pour faciliter cela, le signal sinusoïdal dirigé vers le CAN et au même temps dirigé vers un ampli-opérationnel qui travaille en commutation et fournit un signal carré de même fréquence et en phase. Dans ce cas le passage par zéro du signal sinusoïdal est connu par détection du front montant du signal carré.

La procédure de calcul de la fonction  $\sin(\alpha)$  est la suivante :

1. Détection du front montant du signal carré ;
2. temporisation de  $\alpha(\text{ms})$  ;
3. génération de l'ordre de conversion ;
4. lecture de la tension convertie ;
5. mise en forme de la donnée lue.

L'organigramme suivant illustre cette procédure.





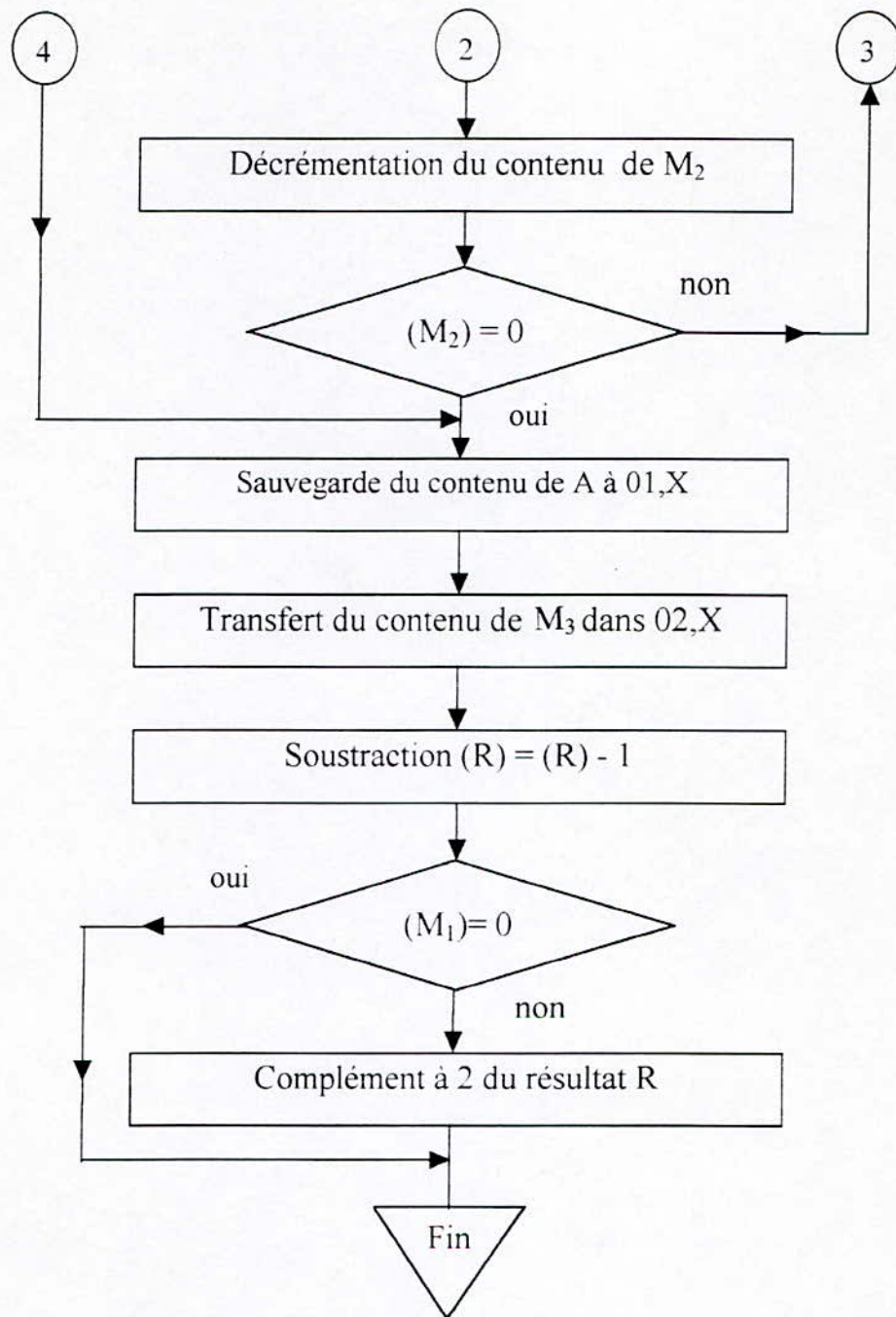
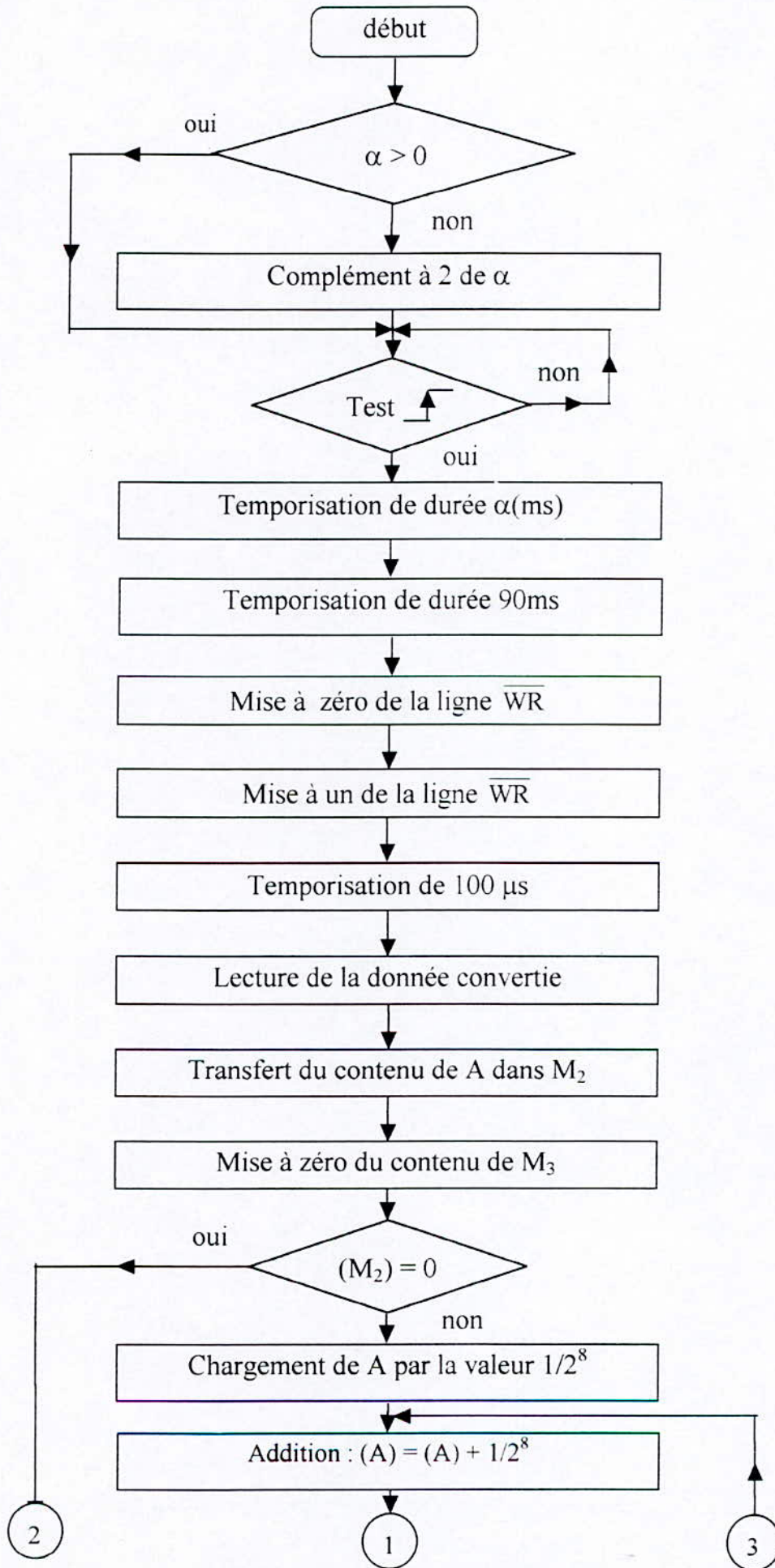


Fig (VIII.10) : Organigramme de calcul de sinus.

### 1.10 La fonction trigonométrique cosinus :

La procédure de calcul du cosinus est la même que le calcul de sinus à l'exception qu'il faut créer une temporisation supplémentaire de 90 ms, car le même oscillateur est utilisé pour la fonction sinus et cosinus.

Cette procédure est illustrée par l'organigramme suivant :



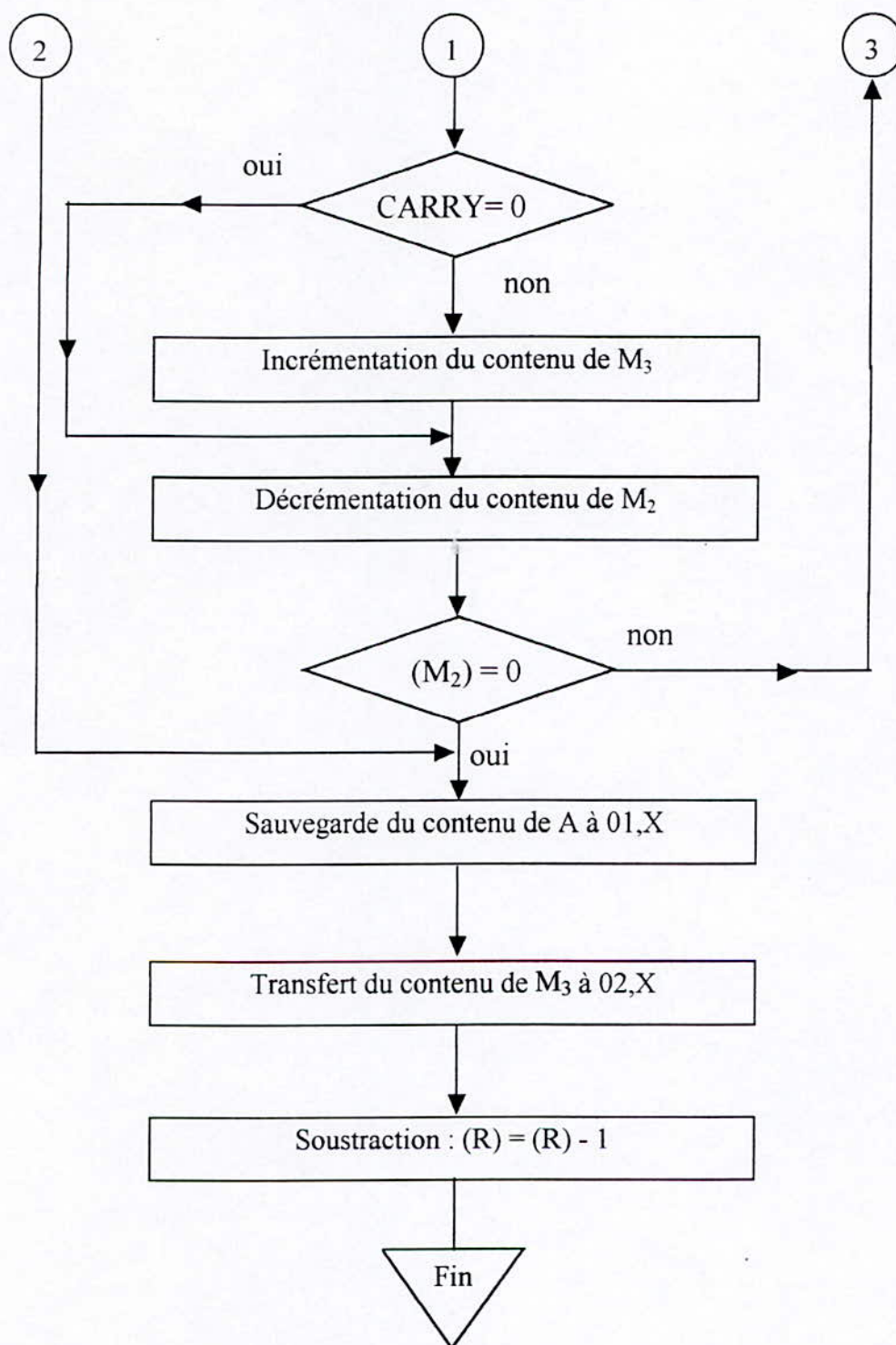


Fig (VIII.11) : Organigramme de calcul de cosinus.

### 1.11 La fonction trigonométrique arctangente :

La procédure de calcul de l'angle optimal nécessite l'utilisation de la fonction  $\text{Arctg}(x)$ . La solution la plus simple pour réaliser cette fonction est d'utiliser la méthode itérative de NEWTON-RAPHSON. Donnée par l'équation suivante :



$$x(n+1) = x(n) - \frac{f(x(n))}{f'(x(n))} \quad (n = 0, 1, 2, \dots, n_{\max})$$

On a :

$$x = \text{Arctag}(y) \Rightarrow y = \text{tg}(x) \Rightarrow \text{tg}(x) - y = 0$$

On pose  $f(x) = \text{tg}(x) - y = 0$

avec :

$$f'(x) = \frac{1}{\cos^2(x)}$$

Donc :

$$x(n+1) = x(n) - \frac{1}{2} \sin(2 \cdot x(n)) + y \cdot \cos^2(x(n)) \quad (\text{VIII-1})$$

Le procédure itérative est arrêtée quand  $|x_{n+1} - x_n| < \varepsilon$ ,

L'organigramme de la figure (VIII-12), illustre cette procédure.

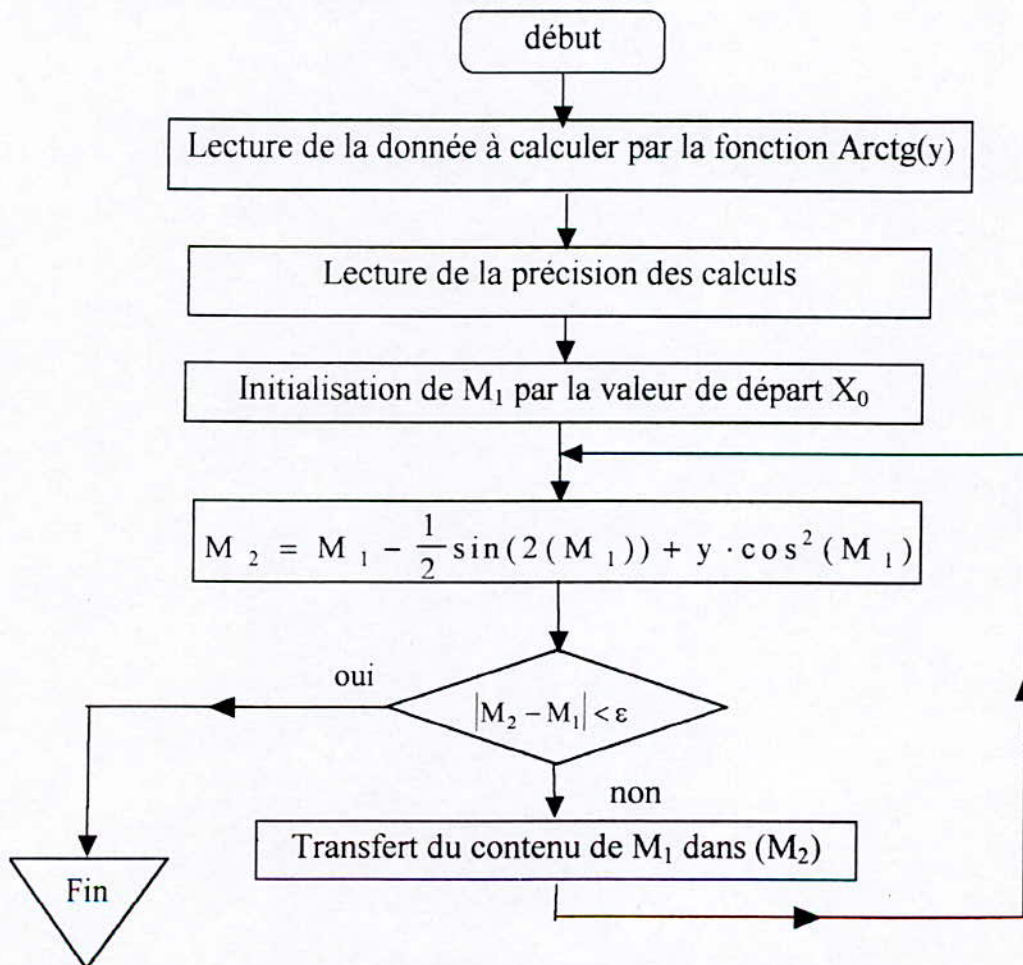


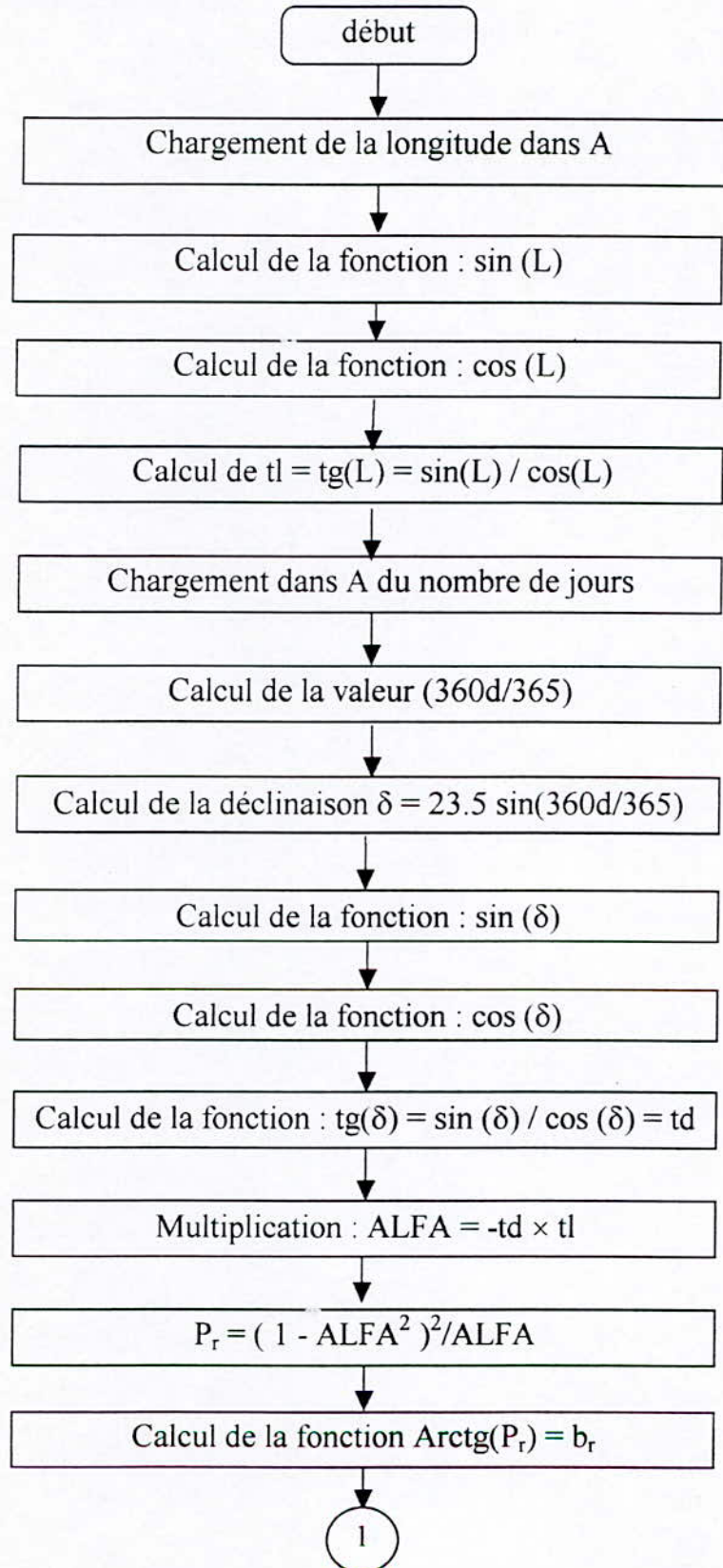
Fig (VIII.12) : Organigramme de calcul de l'Arctangente.

### 1.12 Algorithme de calcul de la position optimale :

L'organigramme de la figure (VIII.13) représente la procédure de calcul de la position optimale d'un panneau solaire.

L : représente la longitude ;

d : représente le nombre de jours.



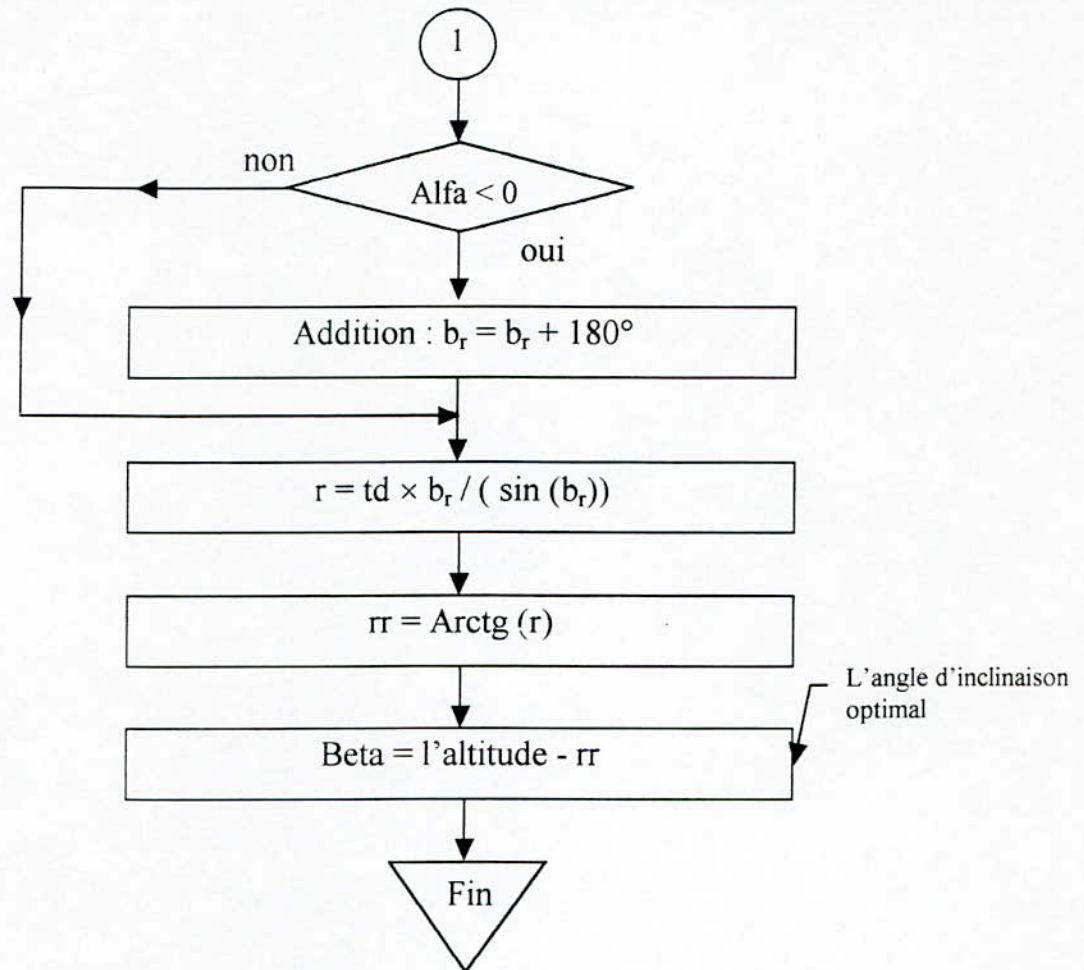


Fig (VIII.13) : Organigramme de calcul de la position optimale.

## 2. COMMANDE DU MOTEUR PAS A PAS :

Cette partie a pour objectif de matérialiser l'angle optimal calculé par le microcontrôleur en effectuant le positionnement du panneau :

### 2.1 Détermination du nombre exact de pas :

La première chose à faire serait le calcul de la différence entre l'angle optimal déterminé et l'angle de la dernière excitation. Le résultat de cette différence est convertie en deux informations ; l'angle avec lequel le moteur doit tourner, et le sens de rotation. Chaque information est sauvegardée sur une position mémoire bien définie. La deuxième étape consiste à reconstituer le nombre de pas exacte à exécuter en multipliant la donnée initiale par un facteur traduisant le rapport de démultiplication introduit par les engrenages du mécanisme de la monture.

L'organigramme de cette tâche est donnée sur la figure (VIII-14) :

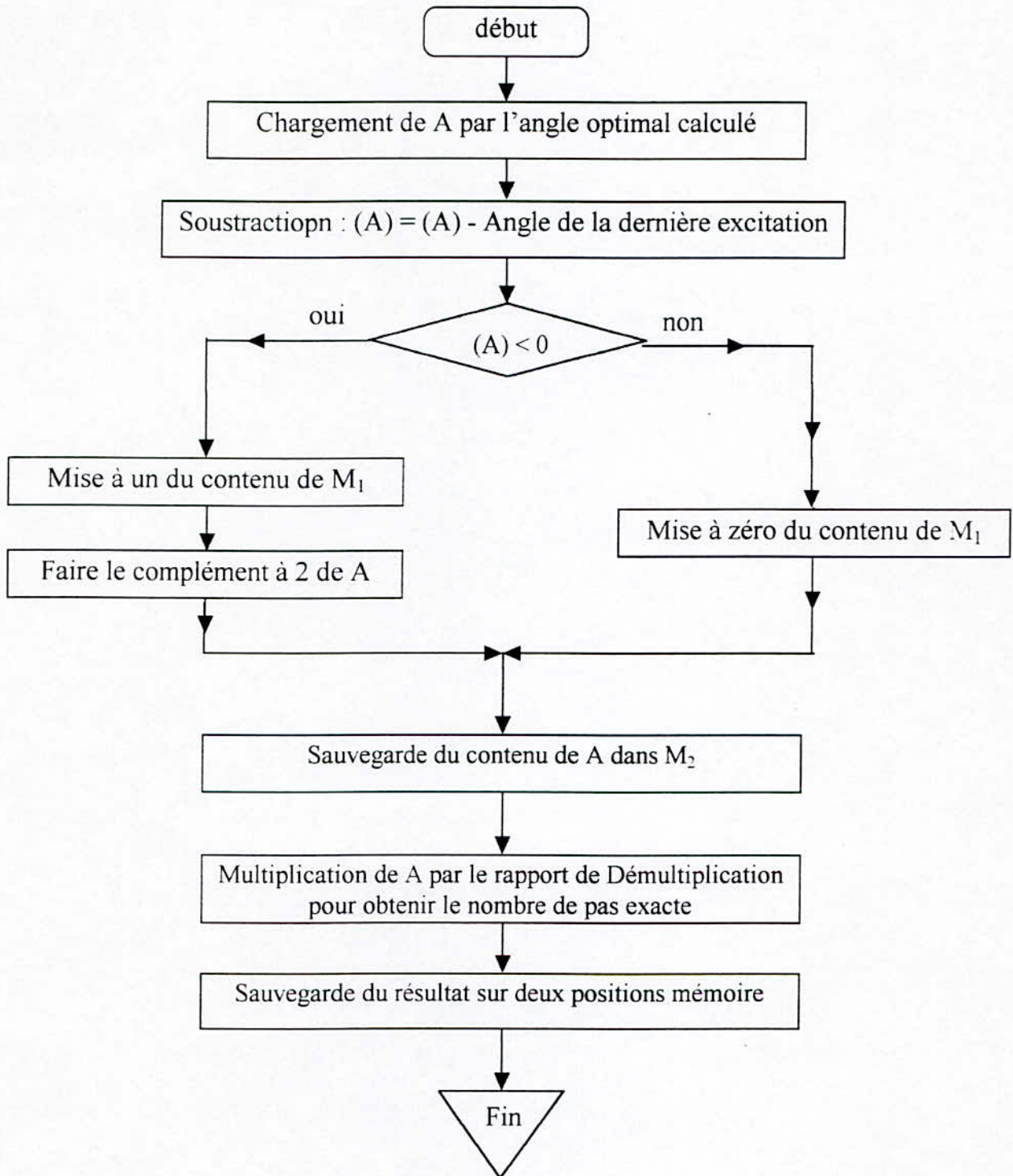
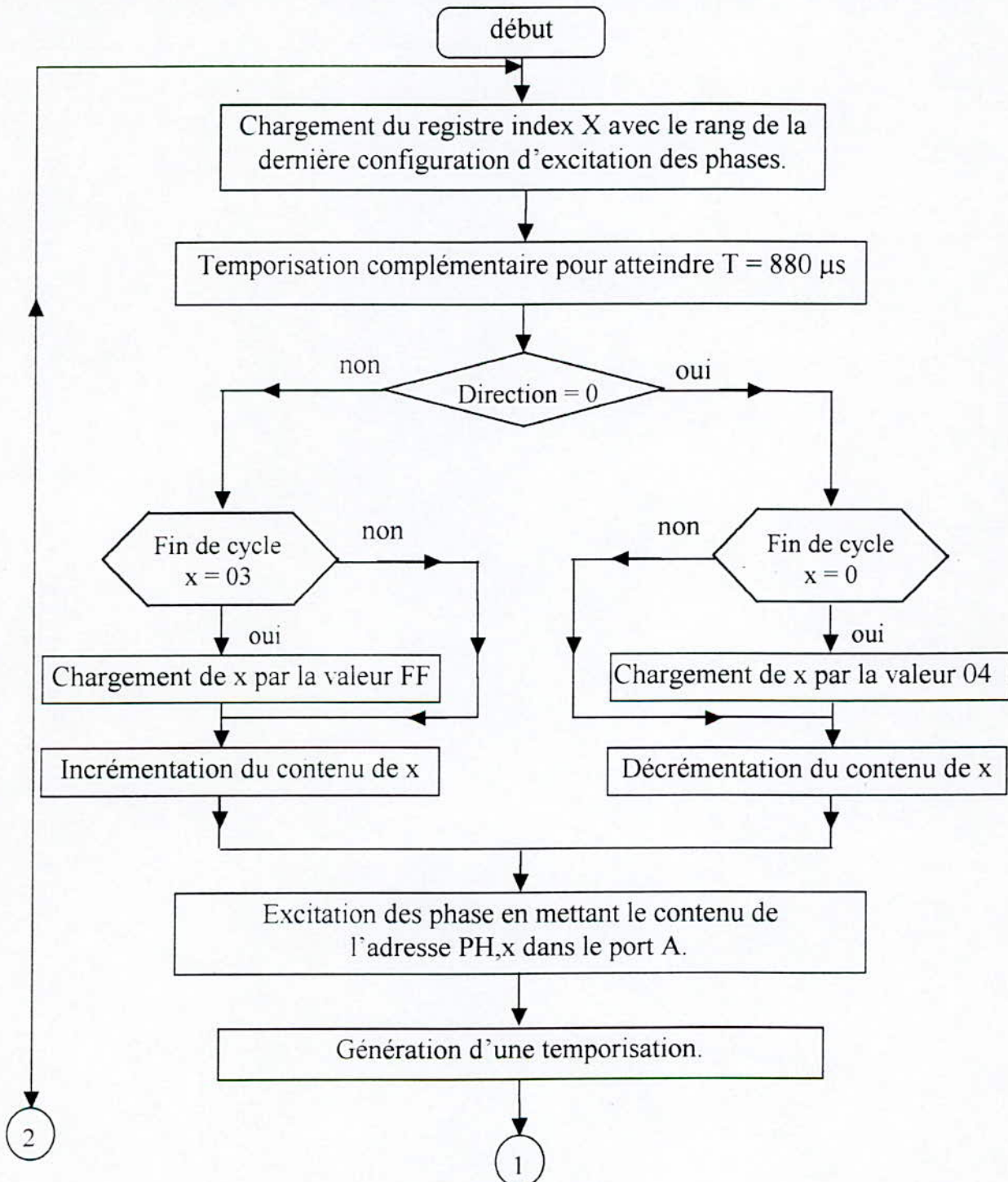


Fig (VIII.14) : Organigramme de la détermination du nombre de pas exact.

## 2.2 LA COMMANDE DU MOTEUR :

La commande du moteur commence par la récupération dans le registre index du rang de la dernière configuration d'excitation des phase.

La boucle d'excitation du moteur débute par une temporisation permettant d'atteindre la largeur d'impulsion de base correspondante à la vitesse maximale, la détection du sens de rotation permet d'incrémenter ou décrémente le registre index tout en testant la fin du cycle d'excitation, pour pouvoir exciter le moteur avec la configuration approprié. L'organigramme de commande est donnée en figure (VIII-15) :



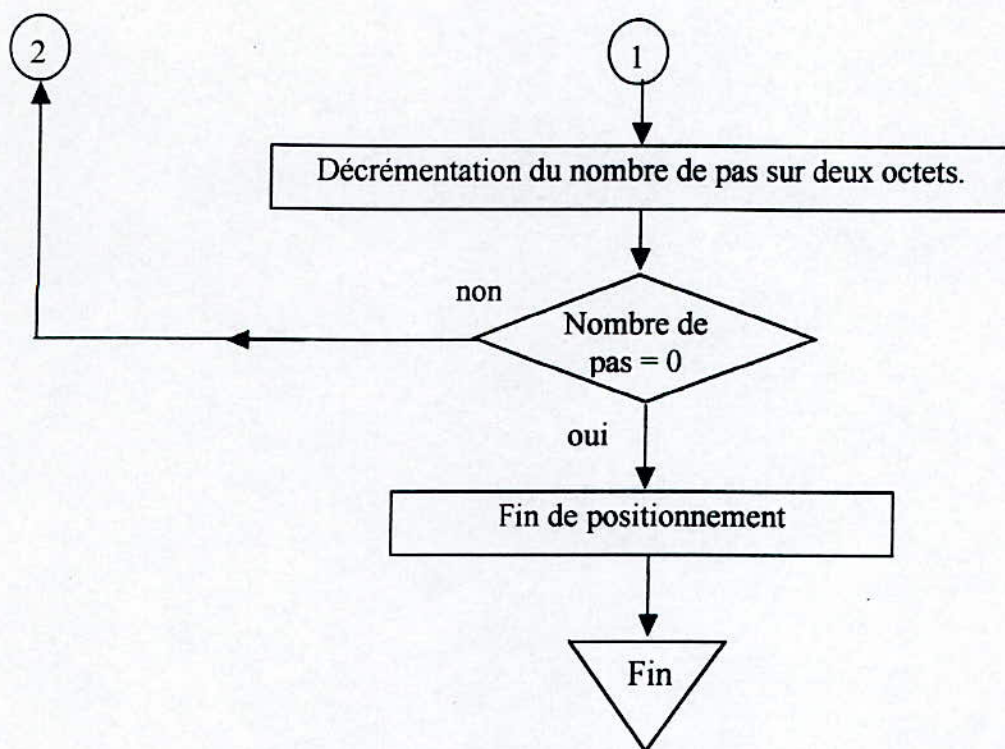


Fig (VIII.15) : Organigramme de commande du moteur.

## CONCLUSION

La recherche d'une énergie renouvelable et capable de concurrencer les autres énergies classiques est devenue une nécessité.

Cette recherche se heurte cependant à une double difficulté, d'ordre technique et d'ordre économique.

Au plan technique, il s'agira d'optimiser la génération photovoltaïque dont l'optimisation de la captation du rayonnement solaire est un élément assez déterminant.

Nous avons étudié un système semi-statique qui permet d'obtenir un flux maximal, en utilisant un minimum d'énergie pour son positionnement. Ceci est un compromis entre la consommation et la production d'énergie et vraisemblablement une étape dans la recherche de la rentabilité de ce procédé.

La commande de l'orientation du panneau solaire est basée sur la prédétermination des coordonnées de la position apparente du soleil à l'aide des équations cosmologiques.

Cependant la résolution de ces équations nécessite l'utilisation de plusieurs opérations et fonctions trigonométriques, qui sont difficilement réalisables à l'aide d'un microcontrôleur 68705P3 en raison de sa faible capacité mémoire.

Pour résoudre ce problème on a utilisé un oscillateur sinusoïdal pour calculer les fonctions trigonométriques, mais dans ce cas le problème qui se pose est que la précision des calculs de ces fonctions dépend de la résolution du convertisseur analogique numérique, la précision de la temporisation créée, et de la stabilité de l'oscillateur en amplitude et en fréquence.

Une meilleure solution pour éviter ces problèmes est l'utilisation d'un circuit de calcul arithmétique très performant appelé Am 9511, qui se connecte facilement à un microprocesseur 8 bits. Ce circuit effectue les opérations arithmétiques en virgule fixe et en virgule flottante sur 32 bits, ainsi que les fonctions trigonométriques et mathématiques réalisables sur un micro-ordinateur.

Au delà de ces considérations, il nous est apparu qu'un système réalisé à l'aide d'un microcontrôleur et un circuit arithmétique Am 9511 fournirait d'excellents résultats, et cela, avec un faible coût.

## BIBLIOGRAPHIE

- [1]. C.TAVERNIER, "Microcontrôleurs 6805 et 68HC05 et les versions UVPR0M 68705 et 68HC705", ed. DUNOD / TECH , 1993.
- [2]. J.M. AUMIAUX , "Système à microprocesseur", ed. MASSON , 1982.
- [3]. J.M. AUMIAUX , " Microprocesseur 8 bits", ed. MASSON , 1982.
- [4]. M. BOUMAHRAT, A. GOURDIN, "Méthodes numériques appliquées", ed. OPU. 1993.
- [5]. R .WALRAVEN, "Calculating the position of the sun ", Solar energy , vol . 20. N° 5, 1978.
- [6]. M. HADDADI , "Conception et optimisation de générateurs à conversion photovoltaïque", Thèse doctorat. ENP, 1994.
- [7]. N. DJIDI , "Commande d'un héliostat par microcontrôleur" , Thèse P.F.E. ,1995.
- [8]. K. OGATA," Systems dynamics", ed. PRENTICE -HALL -International . 1992.
- [9]. M. JUFER , " Transducteur électromécanique", ed. DUNOD, 1989.
- [10]. A. PELAT , " Pratique de l'amplificateur opérationnel ". T2. , ed. MASSON, 1983.
- [11]. E. J. PASAHOW, " L'électronique numérique pour la pratique", ed. MC-GRAW-HILL, 1983.
- [12]. J.C. CHAUVEAU, " Memotech électronique", ed. EDUCALIVRE, 1989.
- [13]. M. MESSUD , " La pratique du microprocesseur. Conception des applications". ed. BERTI, 1992.
- [14]. " Catalogue alphanumérique des principaux circuits intégrés", ed. WEKA.
- [15]. R.TOCCI," Circuit numérique. Théorie et application", ed. BORDAS, 1988.
- [16]. A. ZEROUAL, M. RAOUFI. M. ANKRIM, " Un système de commande automatique basé sur microprocesseur pour le suivi du soleil des capteurs solaires biaxiaux", Signaux, systèmes et automatismes, Premier Séminaire National, Blida, 1992.



# ANNEXE

## ANNEXE 1: PROGRAMME DE CALCUL ET DE GESTION

### 1. SOUS PROGRAMME D'ADDITION REEL SIGNEE:

	CLC	
	STX	\$ 58
	LDA	#\$ 04
	STA	\$ 57
BCL	LDA	00, X
	ADC	04, X
	STA	00, X
	INX	
	DEC	\$ 57
	BNE	BCL
	LDX	\$ 58
	RTS	

### 2. SOUS PROGRAMME DE COMPLEMENT A 2:

	STX	\$ 56
	LDA	#\$ 04
	STA	\$ 55
BCL1	COM	00, X
	INX	
	DEC	\$ 55
	BNE	BCL1
	LDX	\$ 56
	LDA	00, X
	ADD	#\$ 01
	STA	00, X
	LDA	01, X
	ADC	#\$ 00
	STA	01, X
	LDA	02, X
	ADC	#\$ 00
	STA	02, X
	LDA	03, X
	ADC	#\$ 00
	STA	03, X
	RTS	

### 3. SOUS PROGRAMME DE LA CONVERSION BCD - HEXADECIMALE:

	LDA	00, X
	AND	#\$ F0
	BEQ	ST

	LSR	
	LSR	
	LSR	
	LSR	
	STA	\$ 54
	LDA	00, X
BCL	ADD	#\$ FA
	DEC	\$ 54
	BNE	BCL
	STA	00, X
ST	RTS	

#### 4. SOUS PROGRAMME DE LA CONVERSION HEXADECIMALE - BCD:

	CLR	\$ 30
	LDA	00, X
BCL1	CMP	#\$ 0A
	BCS	ST
	SUB	#\$ 0A
	INC	\$ 30
	BRA	BCL1
ST	STA	\$ 36
	LDA	\$ 30
	LSLA	
	LSLA	
	LSLA	
	LSLA	
	ORA	\$ 36
	STA	01, X
	RST	

#### 5. SOUS PROGRAMME DE LA MULTIPLICATION REEL SIGNE:

	STX	\$ 53	
	CLR	\$ 52	
	TST	03, X	
	BPL	ST1	
	JSR	Com à 2	{ Appel du sous programme de complément à 2 }
	INC	\$ 52	
ST1	TST	0B, X	
	BPL	ST2	
	TXA		
	ADD	#\$ 08	
	TAX		
	JSR	Com à 2	{ Appel du sous programme de complément à 2 }
	INC	\$ 52	
ST2	LDA	#\$ 20	

	STA	\$ 51	
	LDX	\$ 53	
	CLR	04, X	
	CLR	05, X	
	CLR	06, X	
	CLR	07, X	
	CLC		
BCL	ROR	03, X	
	ROR	02, X	
	ROR	01, X	
	ROR	00, X	
	BCC	ST3	
	INX		
	INX		
	INX		
	INX		
	JSR	Add	{ Appel du sous programme d'addition }
ST3	LDX	\$ 53	
	ROR	07, X	
	ROR	06, X	
	ROR	05, X	
	ROR	04, X	
	DEC	\$ 51	
	BNE	BCL	
	LDX	\$ 53	
	ROR	03, X	
	ROR	02, X	
	ROR	01, X	
	ROR	00, X	
	DEC	\$ 52	
	BEQ	ST4	
	INX		
	INX		
	JSR	Com à 2	{ Appel du sous programme de complément à 2 }
	LDX	\$ 53	
	RTS		

#### 6.SOUS PROGRAMME DE LA DIVISION REEL SIGNEE:

	STX	\$ 50	
	INX		
	INX		
	CLR	\$ 4F	
	TST	03, X	
	BPL	ST1	
	JSR	Com à 2	{ Appel du sous programme de complément à 2 }
	INC	\$ 4F	

ST1	TST	0B, X	
	BPL	ST2	
	TXA		
	ADD	#\$ 08	
	TAX		
	JSR	Com à 2	{ Appel du sous programme de complément à 2 }
	INC	\$ 4F	
ST2	LDX	\$ 50	
	LDA	#\$ 30	
	STA	\$ 4E	
	CLR	00, X	
	CLR	01, X	
	CLR	06, X	
	CLR	07, X	
	CLR	08, X	
	CLR	09, X	
	CLC		
BCL2	ROL	00, X	
	ROL	01, X	
	ROL	02, X	
	ROL	03, X	
	ROL	04, X	
	ROL	05, X	
	ROL	06, X	
	ROL	07, X	
	ROL	08, X	
	ROL	09, X	
	LDA	#\$ 04	
	STA	\$ 4D	
	CLC		
BCL1	LDA	06, X	
	SBC	0A, X	
	STA	06, X	
	INX		
	DEC	\$ 4D	
	BNE	BCL1	
	BCC	ST	
	INX		
	INX		
	JSR	Add	{ Appel du sous programme d'addition }
	SEC		
ST	LDX	\$ 50	
	DEC	\$ 4E	
	BNE	BCL2	
	ROL	00, X	
	ROL	01, X	
	ROL	02, X	

	ROL	03, X	
	ROL	04, X	
	ROL	05, X	
	COM	00, X	
	COM	01, X	
	COM	02, X	
	COM	03, X	
	COM	04, X	
	COM	05, X	
	DEC	\$ 4F	
	BNE	ST4	
	JSR	Com à 2	{ Appel du sous programme de complément à 2 }
ST4	RTS		

**7. SOUS PROGRAMME DE GENERATION DE LA FONCTION SINUS:**

	CLR	\$ 30	
	TST	03, X	
	BPL	ST1	
	JSR	Com à 2	{ Appel du sous programme de complément à 2 }
	INC	\$ 30	
ST1	NOP		Test du front montant
	BRSET	3,02, ST1	
ST2	NOP		
	BRSET	3,02, ST2	
	TST	02, X	
	BEQ	ST3	
BCL2	LDA	#\$ E1	Temporisation de durée $\alpha$ ms
	STA	\$ 3F	
BCL1	DEC	\$ 3F	
	BNE	BCL1	
	DEC	02, X	
	BNE	BCL2	
ST3	LDA	#\$ 01	Ordre de conversion
	STA	\$ 03	
	CLR	\$ 03	
	LDA	#\$ 80	
BCL4	DECA		Temporisation de durée 100 $\mu$ s
	BNE	BCL4	

	LDA	\$ 00	
	STA	08, X	
	LDA	#\$ 02	
	STA	05, X	
	CLR	06, X	
	CLR	07, X	
	CLR	00, X	
	CLR	01, X	
	CLR	02, X	
	CLR	03, X	
	TST	08, X	
	BEQ	ST4	
BCL5	JSR	Add	{ Appel du sous programme d'addition }
	DEC	08, X	
	BNE	BCL5	
ST4	LDA	#\$ FF	
	STA	06, X	
	STA	07, X	
	CLR	04, X	
	CLR	05, X	
	JSR	Add	{ Appel du sous programme d'addition }
	TST	\$ 30	
	BEQ	ST5	
	JSR	Com à 2	{ Appel du sous programme de complément à 2 }
ST5	RTS		

#### 8. SOUS PROGRAMME DE GENERATION DE LA FONCTION COSINUS:

	TST	03, X	
	BPL	ST1	
	JSR	Com à 2	{Appel du sous programme de complément à 2 }
ST1	NOP		Test du front montant
	BRSET	3,02, ST1	
ST2	NOP		
	BRCLR	3,02, ST2	
	LDA	#\$ 2D	Temporisation de durée 90 ms
	STA	\$ 3D	
BCL3	LDA	#\$ E1	
	STA	\$ 3C	
BCL2	DEC	\$ 3C	
	BNE	BCL2	
	DEC	\$ 3D	
	BNE	BCL3	

	TST	02, X	
	BEQ	ST3	
BCL5	LDA	#\$ E1	
	STA	\$ 3F	
BCL4	DEC	\$ 3F	
	BNE	BCL4	
	DEC	02, X	
	BNE	BCL5	
ST3	LDA	#\$ 01	Ordre de conversion
	STA	\$ 03	
	CLR	\$ 03	
BCL6	LDA	#\$ 80	Temporisation de durée 100 $\mu$ s
	DECA		
	BNE	BCL6	
	LDA	\$ 00	
	STA	08, X	
	LDA	#\$ 02	
	STA	05, X	
	CLR	04, X	
	CLR	05, X	
	CLR	06, X	
	CLR	07, X	
	CLR	00, X	
	CLR	01, X	
	CLR	02, X	
	CLR	03, X	
	TST	08, X	
	DEQ	ST4	
BCL7	JSR	Add	{ Appel du sous programme d'addition }
	DEC	08, X	
	BNE	BCL7	
	LDA	#\$ FF	
	STA	06, X	
	STA	07, X	
	CLR	04, X	
	CLR	05, X	
	JSR	Add	{ Appel du sous programme d'addition }
	RTS		

## 9. SOUS PROGRAMME DE GENERATION DE LA FONCTION ARCTG:

STX	\$ 4D
LDA	02, X
STA	\$ 50



	LDA	#\$ 01	
	STA	\$ 4F	
DEBUT	LDA	\$ 4F	
	STA	\$ 32	
	CLR	\$ 31	
	LDX	#\$ 30	
	JSR	Cos	{ Appel du sous programme de la fonction cosinus }
	LDA	\$ 32	
	STA	\$ 3A	
	LDA	\$ 31	
	STA	\$ 39	
	CLR	\$ 38	
	CLR	\$ 3B	
	JSR	Mul	{ Appel du sous programme de multiplication }
	LDA	\$ 50	
	STA	\$ 3C	
	CLR	\$ 3A	
	CLR	\$ 3B	
	CLR	\$ 3D	
	INX		
	INX		
	JSR	Mul	{ Appel du sous programme de multiplication }
	LDA	\$ 4F	
	STA	\$ 32	
	CLR	\$ 30	
	CLR	\$ 31	
	CLR	\$ 33	
	CLC		
	ROL	\$ 30	
	ROL	\$ 31	
	ROL	\$ 33	
	LDX	#\$ 30	
	JSR	Sin	{ Appel du sous programme de la fonction sinus }
	CLC		
	ROR	\$ 33	
	ROR	\$ 32	
	ROR	\$ 31	
	ROR	\$ 30	
	JSR	Com à 2	{ Appel du sous programme de complément à 2 }
	JSR	Add	{ Appel du sous programme d'addition }
	LDA	\$ 4F	
	STA	\$ 36	
	CLR	\$ 34	
	CLR	\$ 35	
	CLR	\$ 37	
	JSR	Add	{ Appel du sous programme d'addition }
	LDA	\$ 32	

	STA	\$ 4F	
	LDX	#\$ 34	
	JSR	Com à 2	{ Appel du sous programme de complément à 2 }
	LDX	#\$ 30	
	JSR	Add	{ Appel du sous programme d'addition }
	BPL	ST1	
	JSR	Com à 2	{ Appel du sous programme de complément à 2 }
ST1	LDA	\$ 32	
	CMPA	#\$ 01	
	BCC	DEBUT	
	LDX	\$ 4D	
	LDA	\$ 4F	
	STA	02, X	
	RTS		

### LE PROGRAMME PRINCIPAL :

{ La partie gestion du clavier }

	LDX	#\$ 59	
	CLR	\$ 05	
Début	LDA	#\$ F0	
	STA	\$ 05	
ST1	NOP		
	BRSET3	\$ 02 ST1	Test du front montant du signal demande de service
ST2	NOP		
	BRCLR3	\$ 02 ST2	
	CLR	00, X	
	LDA	#\$ 0A	
	STA	\$ 10	Temporisation pour élimination de rebondissement
BCL2	LDA	#\$ 64	
	STA	\$ 11	
BCL1	DEC	\$ 11	
	BNE	BCL1	
	DEC	\$ 10	
	BNE	BCL2	
BCL3	LDA	\$ 01	
	LSR		
	BCC	ST3	
	INX	00, X	
	BRA	BCL4	
	LDA	#\$ 0F	
	STA	\$ 05	

	CLR	\$01	
	LSRA		
	LSRA		
	LSRA		
	LSRA		
BCL4	CLR	01, X	
	LSRA		
	BCC	ST4	
	INC	01; X	
	BRA	BCL4	
	LDA	01, X	
	LSLA		
	LSLA		
	ADD	00, X	
	ADD	#\$ 80	
	STA	\$ 2F	
	STX	\$ 2C	
	LDX	\$ 2F	
	LDA	00, X	
	LDX	\$ 2C	
	STA	00, X	
	LDA	\$ 40	
	ADD	00, X	
	STA	\$ 00	
	ASLA		
	ASLA		
	ASLA		
	ASLA		
	STA	\$ 40	
	LDA	#\$ FF	Temporisation
BCL6	STA	\$ 10	
	LDA	#\$ 64	
BCL7	STA	\$ 11	
	DEC	\$ 11	
	BNE	BCL7	
	DEC	\$ 10	
	BNE	BCL6	
	INX		
	CMX	#\$ 61	
	BNE	DEBUT	
BCL8	LDX	#\$ 59	
	LDA	00, X	
	LSLA		
	LSLA		
	LSLA		

LSLA	
ADD	01, X
STA	00, X
JSR	conv BCD-HEX {Appel du sous programme de conversion BCD- Hexadecimale }
INX	
INX	
CMX	#\$ 61
BNE	BCL8

{ La partie calcul de la position optimale }

	CLR	\$ 4B	
	LDX	#\$ 80	
	DEC	\$ 5B	
	BEQ	ST1	
	CLRA		
BCL1	ADD	00, X	
	BCC	ST2	
	INC	\$ 4B	
ST2	INX		
	DEC	\$ 5B	
	BNE	BCL1	
ST1	ADD	\$ 59	
	BCC	ST3	
	INC	\$ 4B	
ST3	CLR	\$ 10	
	CLR	\$ 11	
	STA	\$ 12	
	LDA	\$ 4B	
	STA	\$ 13	
	LDA	#\$ 7E	
	STA	\$ 18	
	LDA	#\$ FC	
	STA	\$ 19	
	CLR	\$ 1A	
	CLR	\$ 1B	
	LDX	#\$ 10	
	JSR	Mul	{ Appel du sous programme de multiplication }
	INX		
	INX		
	JSR	Sin	{ Appel du sous programme de la fonction sinus }
	LDA	#\$ 17	
	STA	\$ 1C	
	LDA	#\$ 80	
	STA	\$ 1B	
	CLR	\$ 1A	

CLR	\$ 1D	
JSR	Mul	{ Appel du sous programme de multiplication }
LDA	\$ 14	
STA	\$ 1C	
LDA	\$ 15	
STA	\$ 1D	
LDA	\$ 16	
STA	\$ 1E	
LDA	\$ 17	
STA	\$ 1F	
INX		
INX		
JSR	Sin	{ Appel du sous programme de la fonction sinus }
LDX	#\$ 1A	
JSR	Cos	{ Appel du sous programme de la fonction cosinus }
LDX	#\$ 12	
JSR	Div	{ Appel du sous programme de division }
LDA	\$ 12	
STA	\$ 38	
LDA	\$ 13	
STA	\$ 39	
LDA	\$ 14	
STA	\$ 3A	
LDA	\$ 15	
STA	\$ 3B	
LDA	\$ 5F	
STA	\$ 1E	
STA	\$ 26	
LDX	#\$ 1C	
JSR	Sin	{ Appel du sous programme de la fonction sinus }
LDX	#\$ 24	
JSR	Cos	{ Appel du sous programme de la fonction cosinus }
LDX	#\$ 1A	
JSR	Div	{ Appel du sous programme de division }
JSR	Com à 2	{ Appel du sous programme de complément à 2 }
LDX	#\$ 12	
JSR	Mul	{ Appel du sous programme de multiplication }
LDA	\$ 14	
STA	\$ 3C	
STA	\$ 1C	
LDA	\$ 15	
STA	\$ 3D	
STA	\$ 1D	
LDA	\$ 16	
STA	\$ 3E	
STA	\$ 1E	
LDA	\$ 17	

STA	\$ 3F	
STA	\$ 1F	
LDX	#\$ 14	
JSR	Com à 2	{ Appel du sous programme de complément à 2 }
JSR	Mul	{ Appel du sous programme de multiplication }
CLR	\$ 1A	
CLR	\$ 1B	
CLR	\$ 1D	
LDA	#\$ 01	
STA	\$ 1C	
INX		
INX		
JSR	Add	{ Appel du sous programme d'addition }
LDA	\$ 16	
STA	\$ 1E	
LDA	\$ 17	
LDA	\$ 18	
STA	\$ 20	
LDA	\$ 19	
STA	\$ 21	
JSR	Mul	{ Appel du sous programme de multiplication }
LDA	\$ 3C	
STA	\$ 20	
LDA	\$ 3D	
STA	\$ 21	
LDA	\$ 3E	
STA	\$ 22	
LDA	\$ 3F	
STA	\$ 23	
JSR	Div	{ Appel du sous programme de division }
JSR	Arctg	{ Appel du sous programme de l'arctg }
LDA	\$ 16	
STA	\$ 24	
LDA	\$ 17	
STA	\$ 25	
LDA	\$ 18	
STA	\$ 26	
LDA	\$ 19	
STA	\$ 27	
TST	\$ 3F	
BPL	ST1	
CLR	\$ 1A	
CLR	\$ 1B	
LDA	#\$ B4	{ Charger A par la valeur 180 " }
STA	\$ 1C	
CLR	\$ 1D	
JSR	Add	{ appel du sous programme d'addition }

ST1	LDA	\$ 38	
	STA	\$ 1E	
	LDA	\$ 39	
	STA	\$ 1F	
	LDA	\$ 3A	
	STA	\$ 20	
	LDA	\$ 3B	
	STA	\$ 21	
	JSR	Mul	{ Appel du sous programme de multiplication }
	LDA	\$ 24	
	STA	\$ 20	
	LDA	\$ 25	
	STA	\$ 21	
	LDA	\$ 26	
	STA	\$ 22	
	LDA	\$ 27	
	STA	\$ 23	
	JSR	Div	{ Appel du sous prpgramme de division }
	JSR	Arctg	{ Appel du sous programme de l'arctg }
	JSR	Com à 2	{ Appel du sous programme de complément à 2 }
	LDA	\$ 5F	
	STA	\$ 24	
	CLR	\$ 12	
	CLR	\$ 13	
	CLR	\$ 14	
	JSR	Add	{ Appel du sous programme d'addition, le résultat est l' angle optimale }.
	LDA	#\$ 02	Affichage de l'angle optimale
	STA	\$ 02	
	LDA	\$ 14	
	STA	\$ 01	

**{ La partie commande du moteur }**

**{ A. La mise en forme de la donnée }**

	LDA	#\$ 01	chargement du rapport de démultiplication
	STA	\$ 42	
	LDA	\$ 14	
	CLR	\$ 46	
	BRCLR	7.\$ 46	BCL1
	COM	\$ 46	
BCL1	BCLR	7.\$ 46	
	CLC		
	LDA	#\$ 08	

	STA	\$ 40
	LDX	\$ 42
	CLRA	
	LSRX	
BCL	BCC	BL
	ADDA	41
BL	RORA	
	RORX	
	DEC	\$ 40
	BNE	BCL

	STA	\$ 45
	STX	\$ 44

Stockage de la donnée  
résultante sur deux octets.

{B. L'excitation du moteur}

	LDA	#\$ 01	Mise en sortie du port A
	STA	\$ 00	
	LDA	#\$ FF	

	STA	\$ 04
	STA	\$ 00

	LDA	#\$ FE	Chargement des différentes configurations d'excitation des phases du moteur.
	STA	\$ 40	
	LDA	#\$ FD	
	STA	\$ 41	
	LDA	#\$ FB	
	STA	\$ 42	
	LDA	#\$ F7	
	STA	\$ 43	

BC2	LDA	\$ 46	Test du sens de rotation.
	CMPA	#\$ 00	
	BEQ	AV	

	CMPX	#\$ 03	Marche arrière.
	BNE	BCL4	
	LDX	#\$ FF	
BCL4	INX		
	BRA	EXE	

AV	CMPX	#\$ 00	Marche avant.
	BNE	BC1	
	LDX	#\$ 04	
BC1	DEX		
EXE	LDA	40,X	

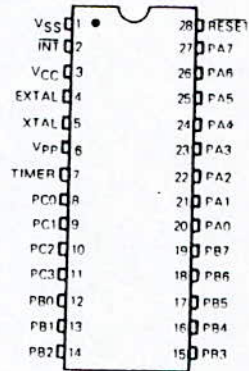


	STA	\$ 00	
	LDA	#\$ 5D	Temporitation
	STA	\$ 3D	
B2	LDA	#\$ E1	
	STA	\$ 3C	
B1	DEC	\$ 3C	
	BNE	B1	
	DEC	\$ 3D	
	BNE	B2	
	DEC	\$ 44	
	BNE	BCL3	
	LDA	\$ 45	
	BEQ	SUI	
	BRA	BC2	
BCL3	LDA	\$ 44	
	CMPA	#\$ FF	
	BNE	BC2	
	DEC	\$ 45	
	BRA	BC2	
SUI	NOP		Fin de positionnement.

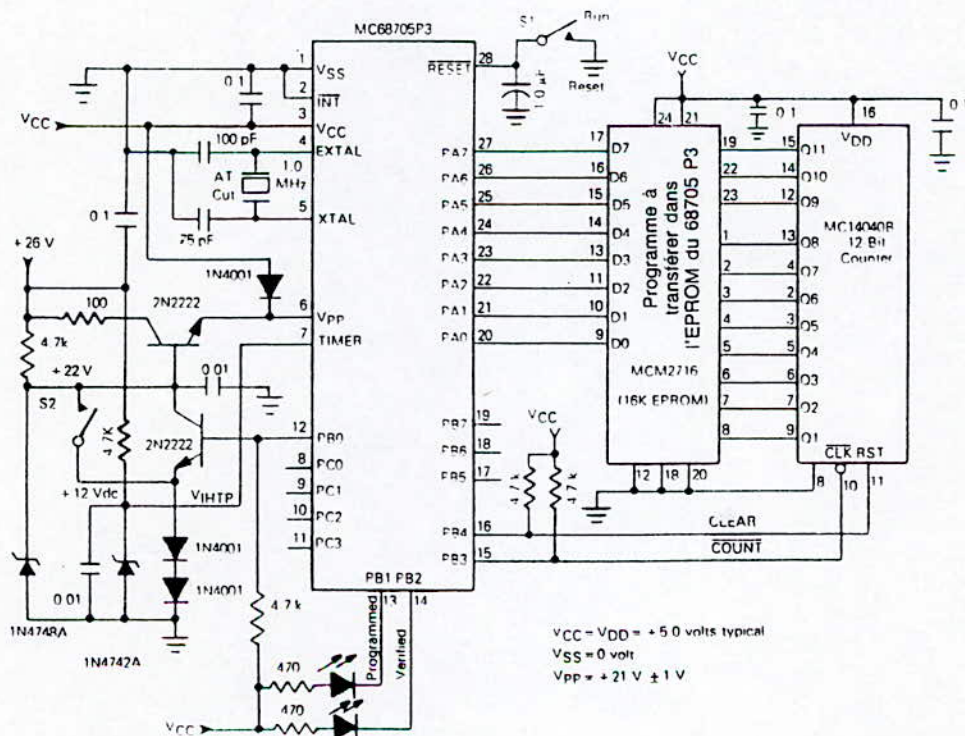
## ANNEXE 2:

### 1. LE MICROCONTROLEUR 68705P3:

#### 1.1. brochage:

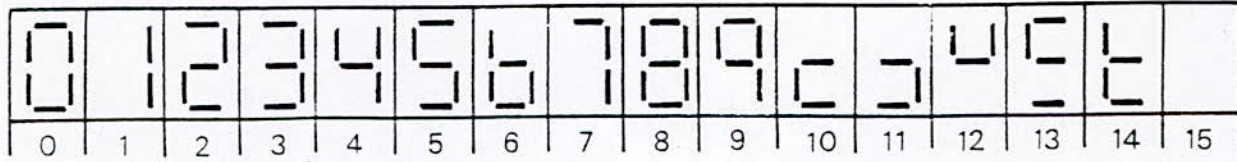
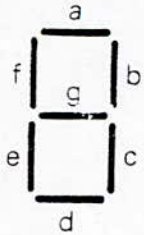


#### 1.2. Schema pour la programmation de l'EPROM du 68705P3:

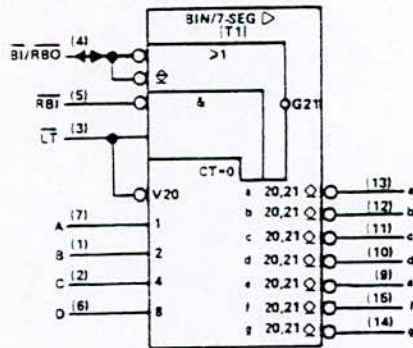


## 2. LE DECODEUR BCD - 7 SEGMENTS 74LS47 DE COMMANDE D-AFFICHEUR :

### 2.1. Identification des segment et visualisation sur les afficheur :



### 2.2. brochage :

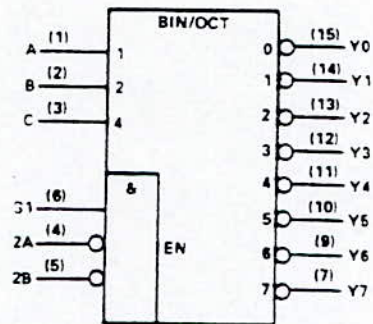


### 2.3. La table de veritéé :

DECIMAL OU FONCTION	Entrees						BI/RBC'	Sorties						
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g
0	H	H	L	L	L	L	H	ON	ON	ON	ON	ON	ON	OFF
1	H	X	L	L	L	H	H	OFF	ON	ON	OFF	OFF	OFF	OFF
2	H	X	L	L	H	L	H	ON	ON	OFF	ON	ON	OFF	ON
3	H	X	L	L	H	H	H	ON	ON	ON	ON	OFF	OFF	ON
4	H	X	L	H	L	L	H	OFF	ON	ON	OFF	OFF	ON	ON
5	H	X	L	H	L	H	H	ON	OFF	ON	ON	OFF	ON	ON
6	H	X	L	H	H	L	H	OFF	OFF	ON	ON	ON	ON	ON
7	H	X	L	H	H	H	H	ON	ON	ON	OFF	OFF	OFF	OFF
8	H	X	H	L	L	L	H	ON	ON	ON	ON	ON	ON	ON
9	H	X	H	L	L	H	H	ON	ON	ON	OFF	OFF	ON	ON
10	H	X	H	L	H	L	H	OFF	OFF	OFF	ON	ON	OFF	ON
11	H	X	H	L	H	H	H	OFF	OFF	ON	ON	OFF	OFF	ON
12	H	X	H	H	L	L	H	OFF	ON	OFF	OFF	OFF	ON	ON
13	H	X	H	H	L	H	H	ON	OFF	OFF	ON	OFF	ON	ON
14	H	X	H	H	H	L	H	OFF	OFF	OFF	ON	ON	ON	ON
15	H	X	H	H	H	H	H	OFF	OFF	OFF	OFF	OFF	OFF	OFF
BI	X	X	X	X	X	X	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF
RBI	H	L	L	L	L	L	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF
LT	L	X	X	X	X	X	H	ON	ON	ON	ON	ON	ON	ON

### 3. LE DECODEUR 3 VERS 8 74LS138:

#### 3.1. brochage :

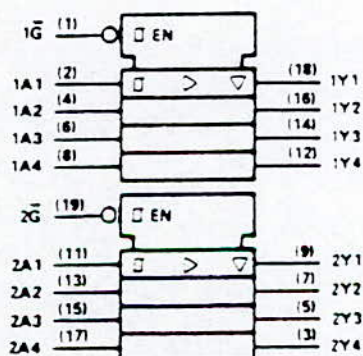


#### 3.2. La table de veritéé :

Entrees			Sorties									
Valic		SELECT										
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	L	H	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

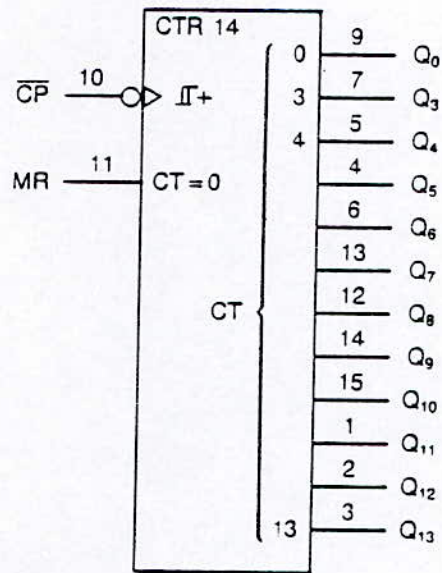
\* G2 = G2A - G2B

### 4. LE BUFFER 74LS244:



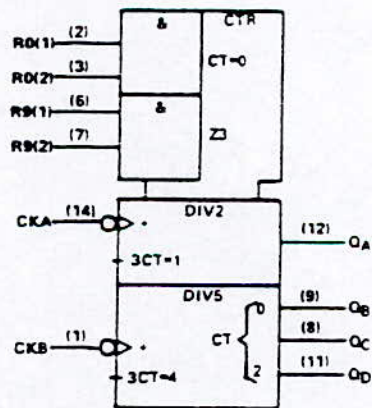
Boîtier 20 - V<sub>CC</sub> 20 - GND 10

## 5. LE COMPTEUR BINAIRE DIVISEUR 4020:



Boîtier DIL 16  
 $V_{DD}$  16 -  $V_{SS}$  8

## 6. LE COMPTEUR DECIMAL 74LS90:



Boîtier DIL 16  
 $V_{CC}$  5 - GND 10

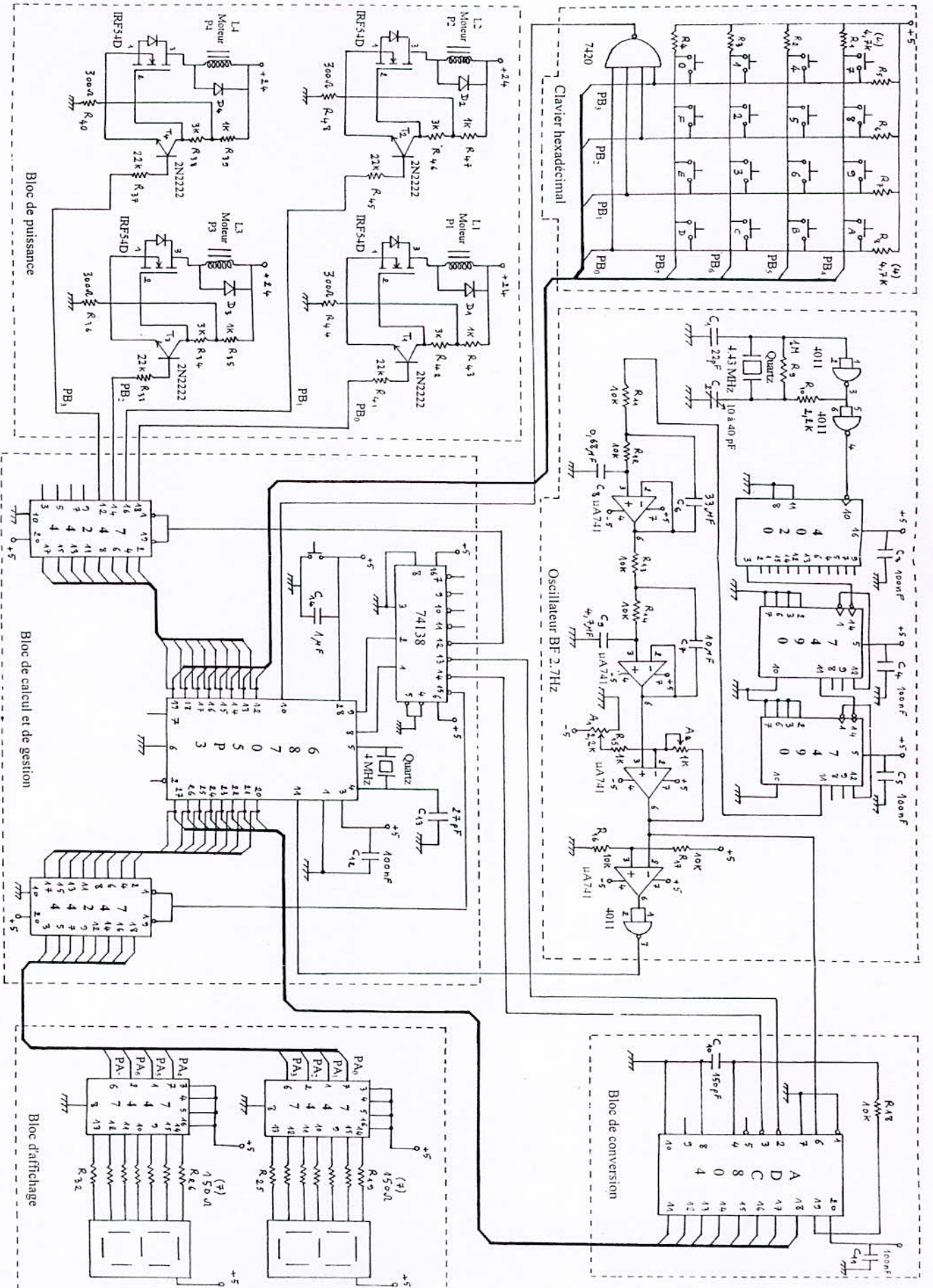


Schéma général du système