

8/01
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

DEPARTEMENT D'ELECTRONIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Projet de Fin d'Etudes

Pour l'obtention de diplôme d'ingénieur d'état en électronique

Thème

*Génération de signaux MLI en temps réel à base du
Microcontrôleur 68HC11*

Proposé et dirigé par :

Mr .C.LARBES

Mr.M.S.AIT CHEIKH

Etudier par :

Mr Belaïd DJOUADI

Année universitaire 2000/2001

E.N.P.10, Avenue Hassen-Badi, EL-HARRACH, ALGER.

Résumé :

Dans notre thèse on a essayé de trouver un algorithme optimal afin d'approximer le modèle de solution des angles de la MLI « modulation en largeur d'impulsion » pour une implémentation sur Microcontrôleur 68HC11.

En addition, on a fait une description détaillée du 68HC11 ainsi que sa méthodologie de programmation.

Le signal MLI obtenu sert à commander un onduleur triphasé qui gère le fonctionnement d'une pompe photovoltaïque.

Mots clés:

Implémentation, Algorithme, MLI, Microcontrôleur, 68HC11

Abstract :

In our thesis we tried to find an optimal algorithm in order to approximate the PWM « pulse width modulation » model of angles solution for implementation using Microcontroller 68hc11.

In addition we made a large description of 68HC11 and its ways of programming.

The PWM signal is used to command a three-phase DC-AC converter to manage the working of photo-voltaic pump.

Key words

Implementation, algorithm, PWM, Microcontroller, 68HC11.

DEDICACE

A la mémoire de ma mère, ma sœur et ma grand-mère

*A mon cher père ALI pour sa présence et son dévouement tout au
long de cette recherche*

*A mes sœurs ASSIA et AMEL et mon frère Lyes pour leur
encouragement*

A mes tantes, mon oncle, mes cousines et cousins surtout DALIL

A mes grands-parents

*A tous mes AMIS particulièrement Abderrahman, Fares,
Riad et Amine*

Remerciements



Je tiens à remercier :

Messieurs C.LARBES et M.s.Ait CHEIKH pour leur encadrement et leurs orientations ;

Les membres du jury pour avoir accepté d'examiner mon travail ;

Mr HADDADI pour son aide précieuse ;

Mon père pour son aide précieuse, sa contribution et son encouragement.

Liste des figures**CHAPITRE I**

Figure I.1 : Schéma principe de l'onduleur de tension

Figure I.2 : Les formes d'ondes des tensions et des courants lorsque le récepteur est
a) inductif ou b) capacitif

CHAPITRE II

Figure II.1 : Onduleur monophasé en pont

Figure II.2 : Intersection de l'onde de modulation avec l'onde modulée

Figure II.3 : ($m=14$) : tracé des formes d'ondes des courants

Figure II.4 : Onduleur en pont triphasé deux niveaux

Tableau 1 : Etablissement des expressions des tensions et des courants

Figure II.5 : Formes d'ondes avec références sinusoïdales et courants de sortie supposés sinusoïdaux pour $m=6$

Figure II.6 : a) La référence. b) Terme fondamental

Figure II.7 : Technique de la modulation partielle

CHAPITRE IV

Figure 1 : Synoptique générale du microcontrôleur 68HC11

Tableau 1 : Modes de fonctionnement du microcontrôleur 68HC11

Figure 2 : Interfaces parallèles

Figure 3 : Principe de connexion de divers circuits via l'interface série 62

Figure 4 : Synoptique interne de la SPI

CHAPITRE V

Figure V.1 : Organigramme Type de développement d'une application

CHAPITRE VI

Figure VI.1 : Forme d'onde de sortie

Figure VI.2 : Trajectoire des angles en fonction de M pour N pair. ($N=4$; $N=6$; $N=8$; $N=10$)

Figure VI.3 : Erreur sur P1

Figure VI.4 : Organigramme de génération de la MLI cas pair

Figure VI.5 : Erreur sur les angles entre les valeurs théoriques et les valeurs calculées par le MC pour $N=4$.

Figure VI.6 : Erreur sur les angles entre les valeurs théoriques et les valeurs calculées par le MC pour $N=6$.

Figure VI.7 : Trajectoire des angles α , en fonction de l'indice de modulation M (ASHE) ; N impair. (a- $N=5$; b- $N=7$)

Figure VI.8 : Erreur sur P1

Figure VI.9 : Erreur sur P2

Figure VI.10 : Organigramme de génération de la MLI cas impair

Figure VI.12 : Erreur sur les angles entre les valeurs théoriques et les valeurs calculées par le MC pour $N=3$.

Figure VI.11 : Erreur sur les angles entre les valeurs théoriques et les valeurs calculées par le MC pour $N=7$.

Figure VI.12 : Organigramme de génération des niveaux bas et haut de la MLI cas pair

Figure VI.13 : Organigramme de génération des niveaux bas et haut de la MLI cas impair

Tableau 1 : Configuration du adapt11

Sommaire*Introduction générale***CHAPITRE I Généralités**

Introduction.....	3
I.1 Les principales applications.....	3
I.1.1 La variation de vitesse des moteurs a courant alternatif	3
I.1.2 Entraînement avec moteur asynchrone	4
I.1.2.1 Alimentation par un onduleur de tension	4
I.2 Effet des harmoniques.....	5
I.3 Les onduleurs (autonomes) de tensions.....	6

CHAPITRE II Modulation en Largeur d'Impulsions dans les onduleurs de tension

Introduction.....	9
II.1 Onduleur monophasé en pont complet.....	11
II.1.1 Principe de la modulation sinus-triangle.....	11
II.1.1.1 Caractérisation de la modulation.....	11
II.1.1.2 Commande des interrupteurs.....	12
II.1.1.3 Tracé des formes d'ondes.....	13
II.1.2 Etude de la tension de sortie.....	15
II.1.2.1 Fondamental. Déchet de tension.....	15
II.1.2.2 Harmoniques.....	16
II.2 Onduleur triphasé en pont (commande phase par phase).....	16
II.2.1 Les deux types de commande.....	17
II.2.2 Commande des interrupteurs «forme d'ondes».....	17
II.2.3 Référence sinusoïdale.....	19
II.2.3.1 Amélioration des tensions de sortie.....	20
II.2.3.2 Amélioration des courants de sortie.....	20
II.2.4 Technique MLI par Injection d'harmonique trois dans la référence.....	23
II.2.5 Autres modification de la loi de commande (Modulation partielle).....	24
Conclusion.....	25

CHAPITRE III Eléments d'analyse numérique

Introduction.....	26
III.1 Méthode de RAPHSON-NEWTON.....	26
III.1.1 Formulation développée.....	26
III.1.2 Formulation matricielle.....	29
III.1.3 Variante de la méthode de Raphson-Newton.....	29
III.1.4 Conditions de convergence.....	30
III.1.5 Avantages et inconvénients de la méthode de Newton.....	30
III.1.6 Critère d'arrêt.....	31
III.2 Interpolation polynomiale.....	32
III.2.1 Polynôme d'interpolation de Lagrange.....	32
III.2.1.1 Approximation.....	33

III.2.2 Différences divisées d'une fonction.....	34
III.2.2.1 Définition.....	34
III.2.2.2 Formule des différences divisées.....	34
III.2.2.3 Polynôme d'interpolation de NEWTON.....	35
III.2.2.4 Approximation.....	36
III.2.3 Convergence de l'interpolation polynomiale.....	36
III.2.4 Avantages et inconvénients des méthodes d'interpolation.....	37
III.3 Interpolation par la méthode des moindres carrés.....	37
III.3.1 Définition.....	37
III.3.2 Méthode des moindres carrés généralisée.....	38
III.3.3 Avantages et inconvénients de la méthode des moindres carrés.....	39
III.3.4 Application de la méthode des moindres carrés pour les fonctions non polynomiales.....	39
 CHAPITRE IV Description du 68HC11	
IV.1 Présentation générale.....	40
IV.2 Les signaux disponibles.....	42
IV.2.1 Les registres de configuration.....	45
IV.3 L'unité centrale.....	47
IV.3.1 Le modèle du programmeur.....	48
IV.3.2 comportement lors d'un Reset.....	50
IV.3.2.1 Les différentes sources de Reset.....	51
IV.3.3 Les interruptions.....	51
IV.3.3.1 Priorité des interruptions.....	52
IV.4 Les mémoires.....	53
IV.5 Les entrées sorties parallèles.....	56
IV.5.1 Le port A.....	56
IV.5.2 Les ports B et C.....	58
IV.5.2.1 Les modes "dialogue" des ports B et C.....	59
IV.5.3 Le port D.....	60
IV.5.4 Le port E.....	61
IV.6 Interface série.....	61
IV.6.1 L'interface série synchrone ou SPI.....	61
IV.6.2 L'interface série asynchrone ou SCI.....	64
IV.7 Le Timer.....	65
IV.7.1 Le Timer à usage général.....	65
IV.7.1.1 Concept général.....	65
IV.7.2 Générateur d'interruptions temps réel.....	67
IV.7.3 Le chien de garde ou COP.....	67
IV.7.4 L'accumulateur d'impulsions.....	68
IV.8 Convertisseur Analogique Numérique CAN.....	69
IV.8.1 Le procédé de conversion et ses implications.....	70
 CHAPITRE V : Développement d'une application	
V.1 Emulateurs, simulateurs et maquettes de test.....	72
V.2 Méthode général de travail.....	73

V.3 Règles d'écriture des programmes.....	73
V.3.1 Les Modes d'Adressage.....	73
V.3.1.1 L'adressage inhérent.....	74
V.3.1.2 L'adressage immédiat.....	74
V.3.1.3 L'adressage direct.....	74
V.3.1.4 L'adressage étendu.....	75
V.3.1.5 L'adressage indexé.....	75
V.3.1.6 L'adressage relatif.....	75
V.3.2 L'adressage «bit » ou «manipulation de bit ».....	76
V.3.2.1 Mode positionnement de bit.....	76
V.3.2.2 Mode test de bit et branchement.....	76
V.3.3 Evaluation des expressions.....	78
V.3.4 Directives d'assemblage.....	78
V.4 Utilisation de l'assembleur.....	80
V.4.1 Le format S1-S9.....	81

CHAPITRE VI Modulation calculée et implémentation

Introduction.....	82
VI.1 PPWM avec élimination sélective d'harmoniques (SHE).....	82
VI.1.1 Elimination sélective d'harmonique pure (PSHE).....	85
VI.1.2 Elimination Sélective d'Harmoniques avec Asservissement de fondamental ASHE.....	85
VI.1.2.1 Cas Des Systèmes D'ordre Pair.....	85
VI.1.2.2 Cas Des Systèmes D'ordre impair.....	92
VI.1.3 Organigramme de génération des niveaux haut et bas.....	99
VI.1.4 Implémentation sur microcontrôleur.....	101
Conclusion.....	102

Conclusion générale

Annexes (A,B,C)

Bibliographie

المدرسة الوطنية المتقدمة للتكنولوجيا
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

INTRODUCTION

INTRODUCTION GENERALE

La variation de vitesse des moteurs est d'une très grande importance dans le domaine industriel. Elle a fait l'objet de nombreux travaux de recherche durant ces dernières années. Ces travaux sont basés essentiellement sur les stratégies de commande et de leur implémentation.

Les progrès apparus ces dernières années dans les domaines de la micro-électronique et la micro-informatique ont permis aux machines à courant alternatif et en particulier le moteur asynchrone à cage de concurrencer les moteurs à courant continu.

Le moteur asynchrone est non seulement robuste mais aussi d'un faible coût ; son poids est faible, sa taille est réduite et il possède une bonne fiabilité de fonctionnement. Cependant, la variation de sa vitesse nécessite une source d'alimentation alternative variable en fréquence et en amplitude.

L'onduleur de tension est le mieux adapté à cette fonction. L'un des principaux problèmes de l'onduleur de tension est la présence d'harmoniques dans le signal de sortie. Elles provoquent un échauffement excessif et des couples pulsés.

La modulation en largeur d'impulsion permet de contrôler simultanément l'amplitude et la fréquence du fondamental de la tension de sortie et surtout de repousser les harmoniques vers les fréquences élevées, ce qui facilite le filtrage et réduit considérablement la taille du filtre.

L'évolution du microcontrôleur a permis à celui-ci de s'imposer dans les réalisations électroniques et principalement dans les fonctions de commande grâce à son faible coût, à la réduction de l'encombrement et à sa fiabilité; son principal avantage par rapport aux techniques analogiques est que la modification des paramètres de la commande se font par logiciel.

Le but de notre PFE est de trouver meilleur algorithme pour approcher les angles de commutation afin de générer un signal MLI en temps réel.

Ce mémoire est structuré de la manière suivante :

- Dans le premier chapitre on introduit des généralités sur les onduleurs et les machines asynchrones.
- Au deuxième chapitre on a présenté les différentes techniques MLI, leurs avantages et leurs inconvénients et enfin quelques techniques de minimisation d'harmoniques.
- Le troisième chapitre traite des méthodes numériques à savoir : la méthode de NEWTON-RAPHSON pour la résolution des systèmes d'équations non linéaires, les méthodes d'interpolation linéaires qu'on compare avec la méthodes des moindres carrés.

- Le quatrième chapitre est entièrement consacré à la description du microcontrôleur 68HC11.
- Le cinquième chapitre traite des différentes méthodes pour tester les programmes écrits en assembleur et la méthode à suivre pour une bonne programmation du microcontrôleur 68HC11 ainsi que des modes d'adressages.
- Dans le dernier chapitre on traite de la modulation calculée et de l'implémentation de certaines de ces méthodes. On y fait également une comparaison avec les résultats théoriques.
- Dans la conclusion on trouvera un résumé du travail fait, les résultats obtenus ainsi que les performances du microcontrôleur.

CHAPITRE I

Généralité sur les onduleurs et les moteurs

Introduction :

Un onduleur est un convertisseur statique assurant la conversion continu-alternatif. Par exemple, si on dispose à l'entrée d'une tension continue et si, grâce à des semi-conducteurs, on relie chacune des bornes du récepteur tantôt à l'une tantôt à l'autre des bornes d'entrée, on peut obtenir entre les bornes du récepteur une tension tantôt positive, tantôt négative.

Par une séquence adéquate de commande des semi-conducteurs, il est donc possible de produire à la sortie du convertisseur une tension alternative de valeur moyenne nulle. Cette tension peut comporter un ou plusieurs créneaux par alternance suivant qu'il s'agit d'une commande à un créneau par alternance ou d'une commande par Modulation de Largeur d'Impulsions (Pulse Width Modulation, en anglo-saxon).

Si l'établissement, le maintien puis l'ouverture des connexions entre l'entrée et la sortie ne dépendent que de la commande des semi-conducteurs, l'onduleur est dit autonome.

On distingue deux types d'onduleurs : [SEG.95]

- Les onduleurs de tension : qui sont alimentés par une source de tension continue.
- Les onduleurs de courant : qui sont alimentés par une source de courant continue.

I.1 Les principales applications :

On peut, en ce qui concerne leurs applications, regrouper la majorité des onduleurs en trois familles :

- Les onduleurs alimentant à tension et fréquence constantes un équipement ou un réseau local,
- Les onduleurs alimentant à tension et fréquence variables des moteurs à courant alternatif,
- Les onduleurs à résonance utilisés soit directement, soit comme étapes dans une chaîne de conversion de l'énergie.

I.1.1 La variation de vitesse des moteurs à courant alternatif :

Le moteur à courant continu reste très utilisé pour la réalisation des entraînements à vitesse variable. Mais l'emploi du moteur à courant alternatif pour réaliser de tels entraînements et de plus en plus fréquent.

Pour faire varier la vitesse du moteur synchrone, il faut faire varier la fréquence de ses tensions d'alimentation : il faut donc l'alimenter par un onduleur. Ce procédé est également le plus utilisé pour le moteur asynchrone.

En même temps que la fréquence, il faut faire varier la valeur efficace des tensions d'alimentation. d'ordinaire on adopte une commande à flux constant qui permet de développer le même couple à toutes les vitesses ; la valeur des tensions croît alors à peu près proportionnellement à leur fréquence l'onduleur doit donc être à fréquence et à tension variable.

Le plus souvent le moteur est triphasé. les trois phases étant équilibrées on peut utiliser un onduleur triphasé dont les trois phases ne sont pas indépendantes.

On demande presque toujours à l'équipement de permettre des freinages rapides par récupération, donc d'assurer la réversibilité fonctionnelle. A noter que l'onduleur est d'ordinaire alimenté par le réseau industriel à travers un redresseur.

La fréquence de sortie de l'onduleur est variable depuis une valeur très faible (éventuellement nulle) correspondant du démarrage du moteur jusqu'à la fréquence nominale qui correspond à la marche à pleine vitesse, voir à une fréquence correspondant à une vitesse sensiblement plus élevée. Il est pratiquement impossible de filtrer les tensions de sortie de l'onduleur car il faudrait faire varier des éléments du filtre au fur et à mesure que la fréquence varie.

I.1.2 Entraînement avec moteur asynchrone :

Le moteur asynchrone à cage offre l'intérêt d'une grande robustesse de son rotor. Il consomme nécessairement de la puissance réactive et constitue une charge inductive : L'onduleur qu'il l'alimente, qu'il soit de tension ou de courant, fonctionne en commutation forcée. En première approximation, on peut assimiler chaque phase à une *f.e.m.*, de même pulsation que le fondamental de la tension d'alimentation, en série avec une résistance et une inductance.

I.1.2.1 Alimentation par un onduleur de tension :

Lorsqu'on alimente un moteur par un onduleur de tension, son inductance n'intervient pas dans les commutations et ne produit pas de pointe de tension ; Si la source continue est bien filtrée et si on utilise des semi-conducteurs rapides, on peut fonctionner à fréquence relativement élevée.

Tant que la fréquence d'alimentation du moteur est sensiblement inférieure à celle à laquelle peuvent fonctionner les éléments semi-conducteurs on utilise la commande par Modulation de Largeur d'Impulsion. Cela permet de réduire les pertes du moteur dû aux harmoniques et, surtout, d'atténuer les pulsations du couple de la machine.

Le principal inconvénient de l'alimentation de l'onduleur de tension apparaît lorsqu'on alimente celui-ci par un redresseur : la réalisation de la réversibilité fonctionnelle est onéreuse.

On utilise l'alimentation du moteur asynchrone par onduleur de tension

- De manière quasi systématique dans le cas des machines de faibles puissances pour lesquelles l'emploi d'onduleurs à transistors fonctionnant en M.L.I est particulièrement bien adapté.
- Dans le cas de machines moyennes à fortes puissances alimentées par un réseau de tension continue, On utilise alors d'ordinaire des onduleurs à GTO fonctionnant tantôt en M.L.I, tantôt en pleine onde suivant que la fréquence de sortie est basse ou élevée.

- Dans le cas de machines moyenne à fortes puissances alimentées par un réseau via un redresseur, lorsqu'on ne doit pas assurer la réversibilité fonctionnelle (marche de la machine en génératrice) ou qu'on ne doit l'assurer que de manière brève et peu fréquente (on peut alors se contenter de dissiper dans des résistances l'énergie renvoyée par la machine à travers l'onduleur).

I.2 Effet des harmoniques :

Dans la commande à vitesse contrôlée, les machines sont alimentées par des convertisseurs dont les signaux de sorties sont riches en harmoniques.

L'effet des harmoniques sur les moteurs apparaît sous deux formes, l'échauffement et les couples pulsés.

- **L'échauffement**

Quand le signal est riche en harmoniques, la valeur efficace du courant augmente et les pertes dans le noyau et dans le cuivre augmentent aussi.

- **Les couples pulsés**

Un harmonique de couple pulsé est produit par l'interaction d'un harmonique de flux dans l'entrefer avec un harmonique de courant dans le rotor. L'importance du flux dans l'entrefer aux fréquences harmoniques est négligeable, et les couples dominants sont ceux dus à l'interaction du flux fondamental dans l'entrefer avec les harmoniques de courant du rotor.

Ainsi le $k^{\text{ième}}$ harmonique de courant I_k réagit avec le flux fondamental Φ_1 donnant une amplitude proportionnelle à :

$$T_{k+1} = \Phi_1 I_k$$

Le couple est d'ordre $(k+1)$ pour les séquences négatives du courant, et d'ordre $(k-1)$ pour les séquences positives.

On voit qu'une composante d'ordre k est commune à deux harmoniques de courant (ou de tension) d'ordre $(k+1)$ et $(k-1)$.

Ces couples causent une instabilité de la vitesse de la machine. L'effet des composantes du couple de hautes fréquences peut être affaibli par l'inertie mécanique du système. La fluctuation de vitesse peut s'aggraver si les vitesses de travail sont réduites ou que le système possède une faible inertie.

La fréquence d'un couple pulsé peut être proche de la fréquence de résonance mécanique ce qui provoque de fortes vibrations d'axe, une usure des dents de l'engrenage et d'insatisfaisantes performances dans les systèmes asservis.

I.3 Les onduleurs (autonomes) de tensions :

On appelle onduleur de tension un onduleur qui est *alimenté par une source de tension*, c'est à dire par une source d'impédance interne négligeable ; sa tension u n'est pas affectée par les variations du courant i qui la traverse. La source continue impose la tension à l'entrée de l'onduleur et donc à sa sortie.

Si u est constant à U , quel soit i ,

Pour $0 < t < T/2$, $u' = +U$

Pour $T/2 < t < T$, $u' = -U$.

- Le courant à la sortie i' et donc le courant à l'entrée i dépendent de la charge placée du côté alternatif. Cette charge peut être quelconque à la seule condition qu'il ne s'agisse pas d'une autre source de tension (capacité ou f.e.m alternative) *directement* branché entre les bornes de sortie.
- L'interrupteur K_1 , quelle que soit la charge alternative, doit relier la borne O à la borne M de $t=0$ à $t=T/2$. Il doit donc pouvoir écouler un courant aussi bien positif que négatif. Son ouverture à l'instant $t=T/2$ doit être commandée si i' est positif (passage de $i_{K1} > 0$ à $v_{K1} > 0$). Si i' est négatif, cette ouverture est spontanée (passage de $v_{K1} > 0$ à $i_{K1} < 0$) et résulte de la fermeture commandée de K'_1 ($v_{K'1} > 0$ à $i_{K'1} > 0$). L'interrupteur K_1 est donc formé par la mise en parallèle d'un semi-conducteur commandé à l'ouverture et à la fermeture TC_1 et d'une diode D_1 .

On arrive ainsi au schéma du principe de l'onduleur autonome de tension de la **figure I.1**.

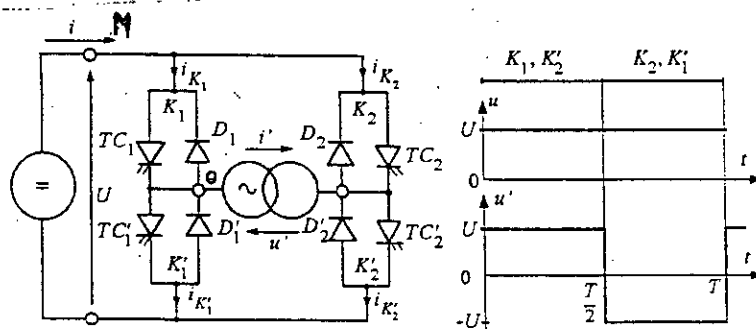


Figure I.1 Schéma principe de l'onduleur de tension

- Pour montrer l'influence de la charge sur le travail demandé aux semi-conducteurs, on suppose le récepteur de courant alternatif parfait, c'est à dire absorbant un courant i' sinusoïdal dans le cas d'une charge inductive et celui de la charge capacitive.

• La (figure I.2.a) donne les formes d'ondes de u', i', i, v_{K1} et i_{K1} lorsque le récepteur est inductif (i' déphasé en arrière du fondamental de la tension u').

En ce qui concerne l'interrupteur K_1 ,

- de $t=0$ à $t=t_1$, la diode D_1 permet le passage de i' alors négatif ;
- de $t=t_1$ à $t=T/2$, c'est TC_1 qui assure le passage de i' ;
- à l'instant $t=T/2$, le blocage de TC_1 provoque le transfert de i' de TC_1 à D'_1 .

Ce fonctionnement appelle trois remarques :

- L'entrée en conduction de TC_1 pour $t=t_1$, s'effectue à courant nul ; elle n'entraîne aucun à-coup de courant dans ce semi-conducteur ou de la tension à ses bornes.
- Il faut utiliser une commande 'longue' des semi-conducteurs commandés, c'est à dire commander la fermeture de TC_1 et TC'_2 pendant tout l'intervalle $[0, T/2]$, de TC_2 et TC'_1 pendant tout l'intervalle $[T/2, T]$ car la valeur de t_1 dépend de la charge. Bien que commandée à la fermeture depuis $t=0$, TC_1 ne sera conducteur qu'à partir de $t=t_1$.
- On doit laisser un petit intervalle entre la commande à la fermeture de TC'_1 (ou TC'_2) et la commande à l'ouverture de TC_1 (ou TC_2) et inversement, afin de ne pas court-circuiter la source de la tension continue. A cause des diodes, il n'y a pas de risque d'ouverture du circuit du récepteur de courant alternatif.

• La (figure I.2.b) donne les formes d'ondes de u', i', i_{VKI} et i_{KI} lors du débit sur un récepteur capacitif (i' déphasé en avant du fondamental de la tension u').

A l'instant $t=0$, l'amorçage de TC_1 assure le transfert du courant i' de la diode D'_1 à TC_1 . Celui-ci écoule i' jusqu'à $t=t_1$.

Pour $t=t_1$, le courant i' s'annule et devient négatif, TC_1 se bloque naturellement et D_1 devient conducteur. D_1 conduit jusqu'à l'instant $t=T/2$ ou son blocage est provoqué par l'amorçage de TC'_1 .

On trouve, comme dans le cas du débit sur charge inductive, une commutation sans à-coup de courant i de tension.

L'autre commutation est assurée par la commande à la fermeture d'un semi-conducteur commandé. L'amplitude au blocage est maintenant inutilisée.

• Les commutations du courant de K_1 à K'_1 (K'_1 à K_1) font donc appel

- à la capacité de TC_1 (TC'_1) d'être commandé à l'ouverture si le courant i' est positif (négatif) au moment de la commutation,
- à la capacité de TC'_1 (TC_1) d'être commandé à la fermeture si i' est négatif (positif) au moment de la commutation.

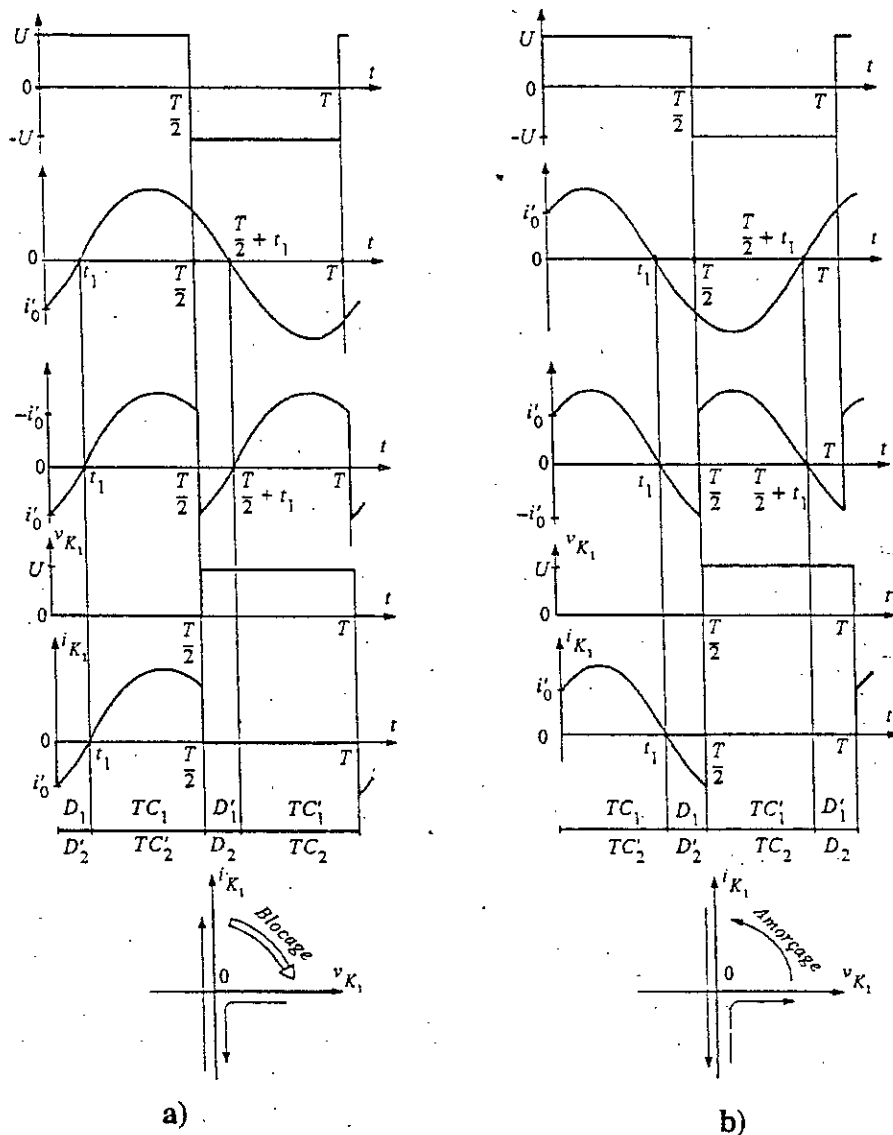


Figure I.2 : Les formes d'ondes des tensions et des courants lorsque de récepteur est
 a) inductif ou b) capacitif

Si la charge peut être inductive ou capacitive, la condition précédente n'est plus garantie : on fera appel tantôt à la capacité des interrupteurs d'être commandés au blocage, tantôt à leur capacité d'être commandés à l'amorçage. Il en va de même si on choisit une commande par Modulation de largeur d'Impulsions. En effet, dans ce cas, on introduit à l'intérieur de chaque alternance un hachage de la tension en effectuant plusieurs commutations supplémentaires entre K_1 et K'_1 (et/ou entre K_2 et K'_2). La nature de charge ne permet plus de garantir que le courant i' ait toujours la même polarité lors des commutations de K_1 et K'_1 et la polarité inverse lors des commutations de K'_1 et K_1 .

CHAPITRE II

Introduction

La modulation de largeur d'impulsion (Pulse Width Modulation) consiste à adopter une fréquence de commutation supérieure à la fréquence de grandeurs de sortie et à former chaque alternance d'une tension de sortie, d'une succession de créneaux de largeurs convenables.

L'essor de la modulation de largeur d'impulsions est lié aux progrès sur les semi-conducteurs de puissance ; l'augmentation du nombre des commutations entraînait des pertes excessives si on n'avait pas réduit les pertes à chacune des commutations.

La multiplication du nombre des «impulsions» formant chacune des alternances d'une tension de sortie offre la possibilité de moduler la forme de cette tension de manière à satisfaire un critère déterminé.

Les onduleurs de tensions servent principalement à alimenter, à fréquence fixe ou variable, des charges alternatives, le critère qui sera considéré dans ce chapitre est l'obtention pour chaque tension de sortie d'une forme d'onde qui approche au mieux la sinusoïde.

Nous montrerons donc comment la Modulation de Largeur d'impulsion permet

- de repousser vers des fréquences élevées les harmoniques de la tension de sortie, ce qui facilite le filtrage.
- de faire varier la valeur du fondamental de la tension de sortie même avec les onduleurs à deux interrupteurs par phase.

Le plus souvent on détermine en temps réel les instants de fermeture et d'ouverture des «interrupteurs» à l'aide d'une électronique de commande analogique ou numérique ou en faisant simultanément appel à ces deux techniques.

Pour assurer cette détermination, lorsque l'objectif est d'approcher au mieux une tension sinusoïdale,

➤ La solution la plus largement employée consiste à utiliser les intersections d'une onde de référence ou modulante, généralement sinusoïdale, avec une onde de modulation ou porteuse, généralement triangulaire, d'où son appellation de «*modulation sinus-triangle*».

➤ On peut également utiliser une bascule à hystérésis commandée par la différence entre une onde (ou son intégrale) et l'intégrale de tension en créneaux à la sortie de l'onduleur ; c'est la «*modulation en delta* » (ou en sigma-delta). [SEG.95].

➤ On peut également générer la séquence des signaux de commande des interruptions de façon à suivre au mieux le vecteur défini par les composants de Clarke du système de tensions qu'on veut produire : c'est la «*modulation vectorielle* ». [SEG.95].

➤ Dans certaines applications, on calcule au préalable, sur la base d'un critère d'optimisation, les instants de commande. Les valeurs calculées sont introduites sous forme de table dans une mémoire morte ; la lecture de celle-ci par un système à microprocesseur assure la génération des signaux de commande des interrupteurs ; c'est la «*modulation calculée* ».

➤ En monophasé, on utilise principalement l'onduleur monophasé en pont complet. La M.L.I. permet de former l'alternance positive de la tension de sortie de créneaux positifs séparés par des intervalles à tension nulle. La même stratégie peut être suivie avec l'onduleur en demi-pont à trois niveaux. [SEG.95].

En triphasé, le schéma le plus utilisé est le pont à six interrupteurs, sauf en forte puissance ou on emploie de plus en plus fréquemment le montage trois niveaux, surtout lorsqu'il faut recourir à la mise en série de semi-conducteurs pour supporter la tension d'alimentation.

Dans ce chapitre on examinera la commande MLI des onduleurs monophasés en pont, puis les onduleurs triphasés en pont à deux niveaux et quelques variantes de cette commande.

II.1 Onduleur monophasé en pont complet :

II.1.1 Principe de la modulation sinus-triangle :

Dans la modulation en largeur d'impulsion, on essaie de chercher à approximer une tension sinusoïdale en faisant varier sa « valeur moyenne ».

L'onduleur de tension monophasé, dont la figure II.1 rappelle le schéma, alimenté par une tension continue U permet d'obtenir à la sortie une tension u' égale à $+U$, à $-U$ ou zéro. Cette possibilité d'obtention d'intervalles à u' nul, qui ne présente pas l'onduleur monophasé en demi-pont à deux niveaux, permet de mieux approximer la tension de sortie sinusoïdale désirée.

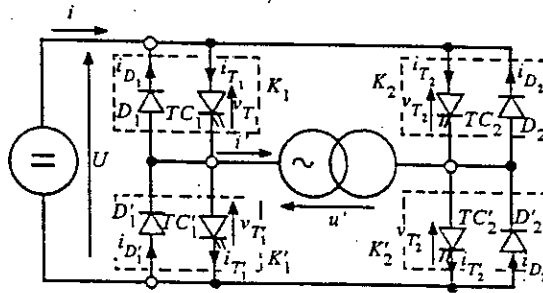


figure II.1 : Onduleur monophasé en pont

II.1.1.1 Caractérisation de la modulation :

- Si la référence est sinusoïdale, deux paramètres caractérisent la commande :
 - L'indice de modulation m , égal au rapport f'/f des fréquences de la modulation et de la référence.
 - Le coefficient de réglage en tension r , égal au rapport de l'amplitude de la tension de référence à la valeur de crête $U/2$ de l'onde de modulation.
- D'ordinaire la modulation est *synchrone*, c'est à dire que f' est un multiple de f .

La tension u' est périodique avec une période T égale à $1/f$.

Mais dans certain cas la modulation est *asynchrone*, notamment quand la fréquence de modulation f' donnée fait varier de façon continue la fréquence de la référence.

En modulation synchrone, si m est impair, l'alternance négative de u' reproduit au signe près son alternance positive, c'est à dire qu'on a une symétrie par rapport à la demi-période. Le développement en série de cosinus de u' ne comporte que des harmoniques impaires.

Au contraire si m est pair, on trouve dans le développement en série de u' une composante continue, des harmoniques pairs et impairs.

On traitera toutefois le cas de m pair à cause de l'onduleur en pont triphasé qu'on examinera ensuite, car alors la composante continue et certains harmoniques sont supprimés.

Les instants de fermeture des "interrupteurs" complémentaires K_1 et K'_1 sont déterminés par les intersections de l'onde de référence u'_w , représentant la tension de sortie désirée de fréquence f , avec l'onde de modulation ou porteuse, M .

L'onde de modulation de fréquence f' nettement supérieure à f , est une onde triangulaire d'amplitude U .

II.1.1.2 Commande des interrupteurs

Pour former l'alternance positive de u' , on prend des créneaux d'amplitude $+U$ séparés par des intervalles à u' nul. Pour former l'alternance négative de u' on utilise des créneaux négatifs séparés par des intervalles à u' nul.

- Lors de l'alternance positive on peut, par exemple, utiliser K'_2 comme interrupteur d'aiguillage en le fermant en permanence et prendre K_1 et K'_1 comme interrupteurs de hachage :

K_1 fermé, u' égale $+U$; K'_1 fermé, u' égale zéro.

Pour que la "valeur moyenne" de u' suive les variations de la tension de référence u'_w , on utilise une onde de modulation M^+ triangulaire unidirectionnelle d'amplitude U (figure II.2a). L'intersection de la partie croissante de M^+ avec u'_w donne la fermeture de K'_1 et le début de l'intervalle à u' nul ; l'intersection de M^+ descendant avec u'_w donne la fermeture de K_1 et le début l'intervalle à u' égal à $+U$.

L'examen de la figure II.2.a montre que

$$u'_{\text{moy}} = U \frac{t_1}{T'/2} = U \frac{u'_w}{U} = u'_w \quad (\text{II.1})$$

- De la même façon pour l'alternance négative de u' , on peut commander la fermeture de K_2 et hacher en commandant tantôt la fermeture de K_1 ($u' = 0$), tantôt celle de K'_1 ($u' = -U$).

On utilise maintenant une onde de modulation M formée de triangles négatifs (figure II.2.b). L'intersection de M descendant avec u'_w commande la fermeture de K_1 et rend u' nul, celle de M descendant avec u'_w commande la fermeture de K'_1 et rend u' égal à $-U$.

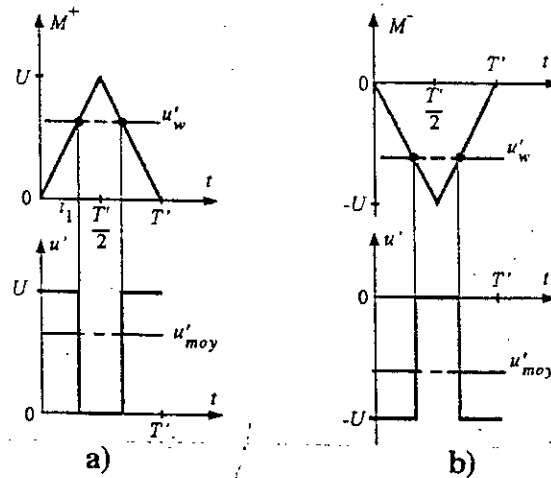


figure II.2 : Intersection de l'onde de modulation avec l'onde modulé

Si u'_w ne varie pas pendant la période T' de la modulation, on a encore :

$$u'_{moy} = u'_w$$

La tension de modulation doit être inversée à chaque changement d'alternance de u'_w .

II.1.1.3 Tracé des formes d'ondes

En modulation synchrone, pour que les deux alternances de u' soient identiques au signe près il faut que la fréquence de modulation f' soit un multiple pair de la fréquence f du u' .

On caractérise la commande par

- le coefficient de réglage en tension de sortie désirée

$$u'_w = rU \sin(\omega t) \tag{II.2}$$

- et l'indice de modulation m , rapport de fréquence de modulation f' à la fréquence f de la tension de sortie, ou le nombre d'impulsions p par alternance.

Entre p et m on la relation
$$p = \frac{m}{2} - 1 \tag{II.3}$$

La figure II.3, tracée pour $m = 14$, donc $p = 6$, et pour $r = 0,8$, montre successivement :

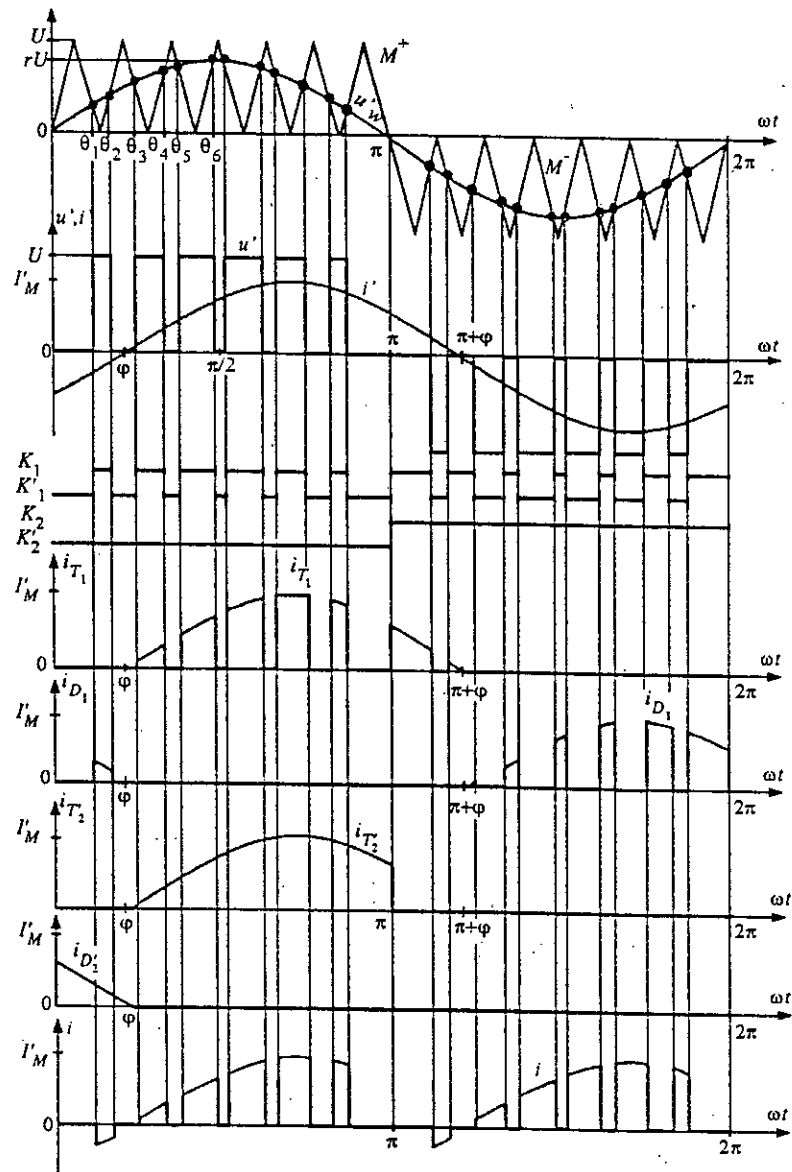


figure II.3 (m=14) : tracé des formes d'ondes des courants

- la détermination des angles de commutation ; vu les symétries, il suffit de déterminer six angles $\theta_1, \theta_2, \dots, \theta_6$.
- le tracé de la tension de sortie u' . On a indiqué le courant de sortie i' supposé sinusoïdal et déphasé d'un angle φ en arrière du fondamental de u' ;

- les intervalles de conduction des quatre interrupteurs dans le cas où K_2 et K'_2 servent d'interrupteurs d'aiguillage, K_1 et K'_1 d'interrupteurs de hachage:
 - pour $0 < \omega t < \pi$, K_2 est fermé : la fermeture de K_1 donne $u' = U$, celle de K'_1 donne $u' = 0$;
 - pour $\pi < \omega t < 2\pi$, K_2 est fermé : la fermeture de K_1 donne $u' = 0$, celle de K'_1 donne $u' = -U$;
- la forme d'onde du courant dans le semi-conducteur commandé TC_1 de l'interrupteur K_1 .
Le courant i_{T1} égale i' quand i' positif et K_1 fermé,
- la forme d'onde du courant dans la diode D_1 montée en parallèle inverse avec TC_1 . le courant i_{D1} égale $-i'$ quand i' est négatif et K_1 fermé,
- la forme d'onde du courant i_{T2} dans le semi-conducteur commandé TC_2 de l'interrupteur K_2 . Ce courant égale i' quand i' est positif et K_1 fermé;
- la forme d'onde du courant i_{D2} dans la diode de l'interrupteur K_2 . Ce courant égale $-i'$ quand i' est négatif et K_2 fermé ;
- enfin, la forme d'onde du courant i pris à la source de tension continue. Le courant i égale $-i'$ quand K_1 et K_2 sont fermés ; il égale $-i'$ quand K_2 et K'_1 sont fermés ; il est nul pendant les autres intervalles.

II.1.2 ETUDE DE LA TENSION DE SORTIE

Le développement en série de la tension de sortie comporte le fondamental et des harmoniques impaires.

Les valeurs efficaces de ces termes sont données par

$$U'_n = \frac{2\sqrt{2}U}{n\pi} |\cos n\theta_1 - \cos n\theta_2 + \dots + \cos n\theta_{p-1} - \cos n\theta_p|$$

II.1.2.1 Fondamental. Déchet de tension

- La valeur efficace du fondamental est, déjà pour p égal à 2, très peu différente de celle de la tension désirée. Dès que p est égal ou supérieur à 4, l'écart est négligeable et on peut prendre

$$U'_1 = \frac{1}{\sqrt{2}} rU \tag{II.4}$$

- Pour les valeurs impaires de p , le coefficient de réglage en tension r peut dépasser légèrement l'unité, mais le dépassement possible diminue quand p croît. Pour p pair, le maximum théorique de r est toujours égal à l'unité, ce qui correspond à

$$U'_{1max} = \frac{U}{\sqrt{2}} \quad (\text{II.5})$$

Il en résulte un déchet de tension, c'est à dire que U'_{1max} est inférieur à la valeur efficace du fondamental que donnerai l'onduleur en pont délivrant des signaux rectangulaires de largeur égale à π , soit $(2\sqrt{2}/\pi)U$.

Pour p pair ou p impair suffisamment élevé,

$$\left(\frac{U'_1}{2\sqrt{2}U/\pi} \right)_{\max} = \frac{\pi}{\sqrt{4}};$$

le déchet de tension théorique est de 21.46%.

II.1.2.2 Harmoniques

- le taux d'harmoniques de la tension, de sortie est beaucoup plus réduit que celui que donne l'onduleur monophasé en demi-pont. C'est nettement visible lorsqu'on observe les formes d'ondes.

Si p est impair, la tension u' a pour valeur efficace

$$U' = U \left(\frac{-\theta_1 + \theta_2 - \theta_3 + \dots - \theta_{p-1} + \theta_p}{\pi/2} \right)^{1/2} \quad (\text{II.6})$$

Si p pair, la valeur efficace de u' est donnée par :

$$U' = U \left(\frac{-\theta_1 + \theta_2 - \theta_3 + \dots - \theta_p + \pi/2}{\pi/2} \right)^{1/2} \quad (\text{II.7})$$

On en déduit la somme quadratique des harmoniques :

$$\left(\sum_{n=3}^{\infty} U_n'^2 \right)^{1/2} = \sqrt{U'^2 - U_1'^2} \quad (\text{II.8})$$

II.2 Onduleur triphasé en pont : (commande phase par phase)

En groupant trois demi-ponts monophasé à deux niveaux, on obtient l'onduleur triphasé à six interrupteurs de figure II.4. Cet onduleur fonctionne en MLI est très utilisé pour alimenter à tension et fréquence variables des récepteurs triphasés équilibrés.

En groupant trois demi-ponts monophasé à trois niveaux, on obtient l'onduleur triphasé à douze interrupteurs. Cet onduleur fonctionne en MLI est de plus en plus utilisé en forte puissance.

II.2.1 Les deux types de commande :

- On peut faire découler la commande MLI des onduleurs triphasés en pont, à deux ou trois niveaux, de celle des onduleurs monophasés, en générant de manière indépendante les signaux de commandes des interrupteurs de chacun des trois demi-ponts.

L'étude de l'onduleur triphasé à deux niveaux ainsi commandé phase par phase fait l'objet du prochain point.

- On peut aussi assurer de manière globale la commande de l'ensemble des interrupteurs constituant l'onduleur. Dans ce cas on utilise la Modulation vectorielle. [SEG.95].

• Possibilités offertes par l'onduleur triphasé à commander phase par phase

Avec l'onduleur triphasé en pont avec commande séparée de chaque demi-pont, si on conserve une fréquence sinusoïdale, on retrouve les résultats effectués en monophasé ;

• Possibilité de faire varier le fondamental des tensions de sortie depuis zéro jusqu'à 78,54% de la valeur qu'on obtiendrait avec une commande pleine onde, si m est grand.

• rejet vers les fréquences élevées des harmoniques de ces tensions,

Mais de plus, si le récepteur est en étoile sans conducteur neutre ou en triangle, les harmoniques de rang 3 ou multiple de 3 qui étaient présents en monophasé sont absents des tensions de sortie, car ils forment des systèmes homo polaires.

Cette propriété est souvent mise à profit pour modifier la forme d'onde des tensions de références, afin de réduire le déchet de tension et diminuer certains harmoniques restent présents dans les tensions de sortie.

II.2.2 Commande des interrupteurs « Forme d'ondes » :

La figure II.4 rappelle le schéma de l'onduleur en pont triphasé deux niveaux.

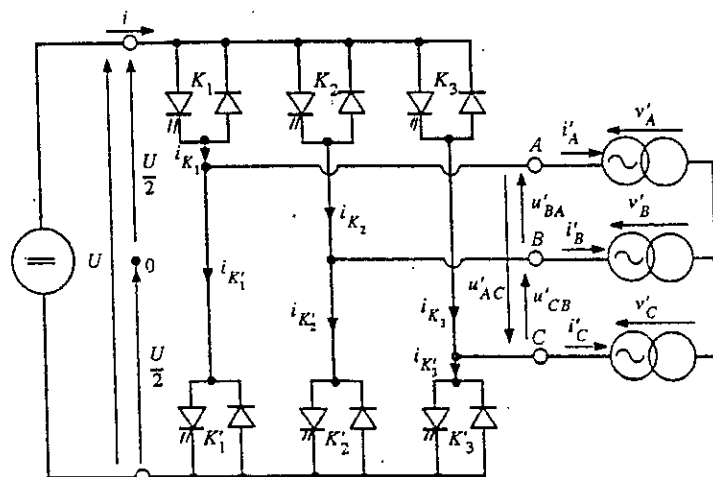


Figure II.4 : Onduleur en pont triphasé deux niveaux

• Pour déterminer les instants de fermetures des interrupteurs complémentaires deux à deux, K_1 et K'_1 , K_2 et K'_2 , K_3 et K'_3 , on procède comme on l'a indiqué pour l'onduleur en demi-pont monophasé :

- On trace les trois ondes de référence représentant les trois tensions de sortie désirées entre les trois bornes de sortie A, B, C et le point milieu (fictif) O de la source de tension continue;
- Les intersections avec l'onde de modulation triangulaire M donnent les instants de fermeture des divers interrupteurs.

• Les tensions fournies par les demi-ponts sont :

$$v_A - v_O = \frac{U}{2}, \text{ quand } K_1 \text{ est fermé ;}$$

$$v_A - v_O = -\frac{U}{2}, \text{ quand } K'_1 \text{ est fermé ;}$$

$$v_B - v_O = \frac{U}{2}, \text{ quand } K_2 \text{ est fermé, ...}$$

De ces tensions on déduit les tensions composées entre les bornes de sortie :

$$u'_{BA} = (v_A - v_O) - (v_B - v_O)$$

$$u'_{CB} = (v_B - v_O) - (v_C - v_O)$$

$$u'_{AC} = (v_C - v_O) - (v_A - v_O)$$

Si la charge est équilibrée et couplée en étoile, on détermine les tensions v'_A, v'_B, v'_C aux bornes des trois phases à l'aide des relations suivantes :

$$\left. \begin{aligned} v'_A &= \frac{2}{3}(v_A - v_O) - \frac{1}{3}(v_B - v_O) - \frac{1}{3}(v_C - v_O) \\ v'_B &= -\frac{1}{3}(v_A - v_O) + \frac{2}{3}(v_B - v_O) - \frac{1}{3}(v_C - v_O) \\ v'_C &= -\frac{1}{3}(v_A - v_O) - \frac{1}{3}(v_B - v_O) + \frac{2}{3}(v_C - v_O) \end{aligned} \right\} \quad (\text{II.9})$$

• Les courants dans les interrupteurs se réduisent des courants de sortie i'_A, i'_B, i'_C comme pour l'onduleur en demi-pont monophasé :

$$i_{K_1} = i'_A ; i_{K'_1} = 0, \text{ quand } K_1 \text{ est fermé ;}$$

$$i_{K_1} = 0 ; i_{K'_1} = -i'_A, \text{ quand } K'_1 \text{ est fermé ;}$$

$$i_{K_2} = i'_B ; i_{K'_2} = 0, \text{ quand } K_2 \text{ est fermé ;}$$

Le courant fourni par la source i est donné par

$$i = i_{K_1} + i_{K_2} + i_{K_3}$$

•Le tableau montre les expressions que prennent $u'_{BA}, v'_A, i_{K_1}, i_{K_2}, i_{K_3}$ et i en fonction de l'état ouvert ou fermé des interrupteurs K_1, K_2, K_3 .(Rappelons que les états de K_1, K_2, K_3 sont respectivement complémentaires de ceux de K_1, K_2, K_3).

Tableau 1 –Etablissement des expressions des tensions et des courants

K_1	K_2	K_3	$v_A - v_O$	$v_B - v_O$	$v_C - v_O$	U'_{BA}	v'_A	i_{K_1}	i_{K_2}	i_{K_3}	i
F	F	F	U/2	U/2	U/2	0	0	i'_A	i'_B	i'_C	0
F	0	F	U/2	-U/2	U/2	U	U/3	i'_A	0	i'_C	$-i'_B$
F	F	0	U/2	U/2	-U/2	0	U/3	i'_A	i'_B	0	$-i'_C$
F	0	0	U/2	-U/2	-U/2	U	2U/3	i'_A	0	0	i'_A
0	F	F	-U/2	U/2	U/2	-U	-2U/3	0	i'_B	i'_C	$-i'_A$
0	0	F	-U/2	-U/2	U/2	0	-U/3	0	0	i'_C	i'_C
0	F	0	-U/2	U/2	-U/2	-U	-U/3	0	i'_B	0	i'_B
0	0	0	-U/2	-U/2	-U/2	0	0	0	0	0	0

En modulation synchrone, on adopte une valeur de l'indice de modulation multiple de 3 pour que les trois tensions $v_A - v_O, v_B - v_O, v_C - v_O$ soient identiques à un tiers de leur période près ; Cette identité se trouve alors avec les tensions $u'_{BA}, u'_{CB}, u'_{AC}$ ou avec les tensions v'_A, v'_B, v'_C .

La **figure II.5** donne un exemple de tracé des formes d'ondes avec références sinusoïdales et courants de sortie supposés sinusoïdaux. Ce tracé correspond à

$$M = 6 ; r = 0.75 , \varphi = \frac{\pi}{6} .$$

II.2.3 Référence sinusoïdale :

Si les tensions de références sont sinusoïdales, les tensions $v_A - v_O, v_B - v_O, v_C - v_O$ ont, pour des valeurs données de m et de r , la même forme d'onde que celle trouvée en monophasé. La valeur et la phase du fondamental et des harmoniques de ces tensions sont celles trouvées en monophasé. Les contraintes sur les semi-conducteurs se déduisent de la tension U et du courant débité par chaque demi-pont de la même façon qu'en monophasé.

Si l'indice de modulation m est multiple de 3, les trois tensions $v_A - v_O$, $v_B - v_O$ et $v_C - v_O$ sont identiques d'un tiers de période près. Cela entraîne deux effets bénéfiques : il y a réduction des harmoniques des tensions de sortie v'_A, v'_B, v'_C et du courant d'entrée i .

II.2.3.1 Amélioration des tensions de sortie :

Les tensions de sortie v'_A, v'_B, v'_C se déduisent de $v_A - v_O$, $v_B - v_O$, $v_C - v_O$ par la suppression des harmoniques de rang 3 ou multiple de 3.

Pour m multiple de 3, on supprime notamment [SEG.95]

- l'harmonique le plus important, celui de fréquence mf situé au centre de la première famille,
- les harmoniques de fréquences $2mf-3f$ et $2mf+3f$ de la deuxième famille,...
- l'harmonique de fréquence $3mf$ situé au centre de la troisième famille,...

Des harmoniques importants des deux premières familles il ne reste que ceux de rang

$$\begin{array}{c} m-2, m+2 \\ 2m-1, 2m+1 \end{array} \quad (\text{II.10})$$

Ceux de rang $m, m-6, m+6$, et $2m-3, 2m+3$ ont été supprimés.

Pour le fondamental et les harmoniques restant, les caractéristiques tracées en monophasé sont utilisables.

II.2.3.2 Amélioration des courants de sortie : [SEG.96]

Si les courants de sortie i'_A, i'_B, i'_C sont équilibrés, le courant d'entrée i est la somme de trois courants, $i_{K_1}, i_{K_2}, i_{K_3}$ identiques à $T/3$ ou $2T/3$ près.

Lorsqu'on fait la somme de ces trois courants,

- Les termes moyens s'ajoutent ;
- Les harmoniques de rang 3 ou multiple de 3, formant des systèmes homopolaires, s'ajoutent également ;
- Les autres harmoniques, formant des systèmes équilibrés directs ou inverses, donnent des sommes nulles.

Il y a donc suppression d'un grand nombre d'harmoniques, notamment le plus gênant, celui ayant la fréquence $2f$.

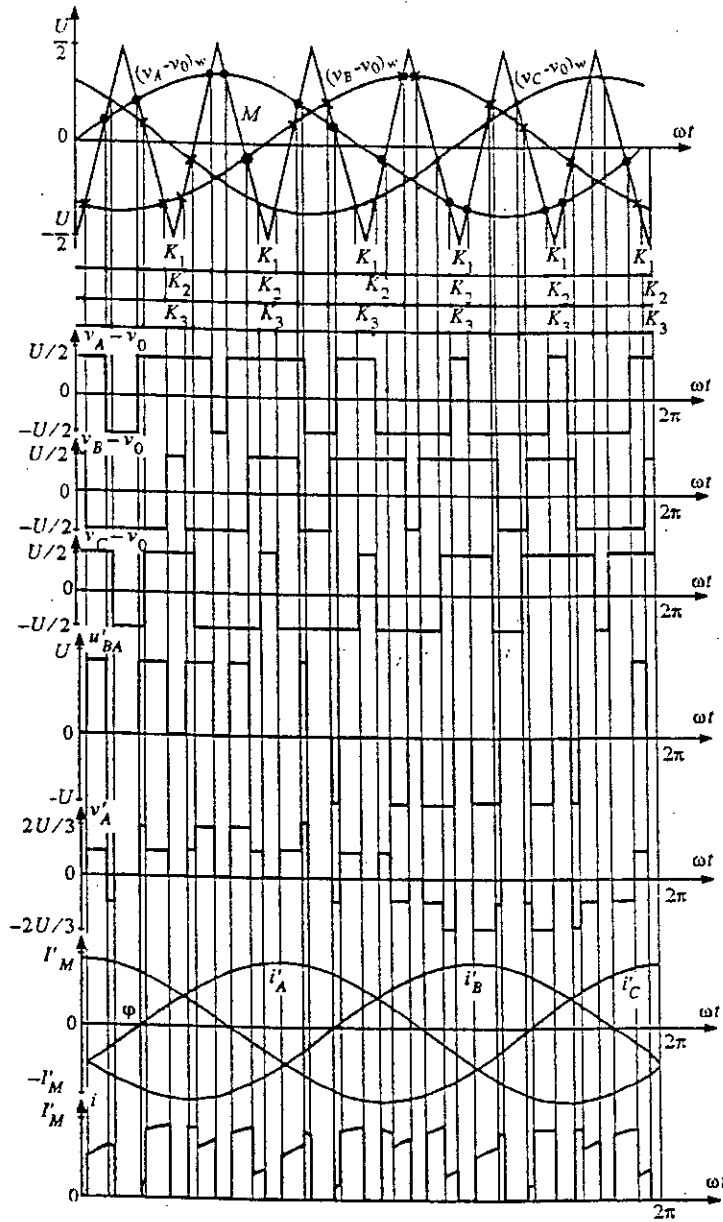


Figure II-5: Formes d'ondes avec références sinusoïdales et courants de sortie supposés sinusoïdaux pour $m=6$.

Pour trouver la valeur et le rang des harmoniques restants, on peut utiliser la conservation de la puissance instantanée

$$U_i = v'_A i'_A + v'_B i'_B + v'_C i'_C \tag{II.11}$$

avec

$$i'_A = I' \sqrt{2} \sin(\omega t - \varphi)$$

$$\begin{aligned}
 i'_B &= I' \sqrt{2} \sin\left(\omega t - \varphi - \frac{2\pi}{3}\right) \\
 i'_C &= I' \sqrt{2} \sin\left(\omega t - \varphi - \frac{4\pi}{3}\right)
 \end{aligned} \tag{II.12}$$

Si m est grand, le début du développement en série de v'_A, v'_B, v'_C est, compte-tenu des harmoniques restant dans ces tensions (encadré II.10) :

$$\begin{aligned}
 v'_A &= V'_1 \sqrt{2} \sin \omega t + V'_{m\pm 2} \sqrt{2} [\sin(m-2)\omega t + \sin(m+2)\omega t] \\
 &+ V'_{2m\pm 1} \sqrt{2} [-\sin(2m-1)\omega t + \sin(2m+1)\omega t] \\
 v'_B &= V'_1 \sqrt{2} \sin\left(\omega t - \frac{2\pi}{3}\right) + V'_{m\pm 2} \sqrt{2} \left[\sin(m-2)\left(\omega t - \frac{2\pi}{3}\right) + \sin(m+2)\left(\omega t - \frac{2\pi}{3}\right) \right] \\
 &+ V'_{2m\pm 1} \sqrt{2} \left[-\sin(2m-1)\left(\omega t - \frac{2\pi}{3}\right) + \sin(2m+1)\left(\omega t - \frac{2\pi}{3}\right) \right] \\
 v'_C &= V'_1 \sqrt{2} \sin\left(\omega t - \frac{4\pi}{3}\right) + V'_{m\pm 2} \sqrt{2} \left[\sin(m-2)\left(\omega t - \frac{4\pi}{3}\right) + \sin(m+2)\left(\omega t - \frac{4\pi}{3}\right) \right] \\
 &+ V'_{2m\pm 1} \sqrt{2} \left[-\sin(2m-1)\left(\omega t - \frac{4\pi}{3}\right) + \sin(2m+1)\left(\omega t - \frac{4\pi}{3}\right) \right]
 \end{aligned} \tag{II.13}$$

$V'_1, V'_{m\pm 2}$ et $V'_{2m\pm 1}$ ayant les valeurs indiquées pour $U'_1, U'_{m\pm 2}$ et $U'_{2m\pm 1}$.

En reportant les expressions de v'_A, v'_B, v'_C , de i'_A, i'_B, i'_C dans (II.11) et en tenant compte du fait que $m-3, m+3$ et $2m$ sont multiples de 3, il vient :

$$\begin{aligned}
 U_i &= 3V'_1 I' \cos \varphi + 3V'_{m\pm 2} I' \{ \cos[(m-3)\omega t + \varphi] - \cos[(m+3)\omega t - \varphi] \} \\
 &+ 3V'_{2m\pm 1} I' [\cos(2m\omega t - \varphi) + \cos(2m\omega t + \varphi)]
 \end{aligned}$$

Comme harmoniques importants, il ne reste que ceux de rang

$$\boxed{m-3, m+3, 2m} \tag{II.14}$$

On voit que l'augmentation de m facilite le filtrage ou le lissage du courant d'entrée.

La valeur moyenne de ce courant et la valeur efficace de ses premiers harmoniques importants sont données par :

$$\left. \begin{aligned}
 I &= \frac{3V'_1 I'}{U} \cos \varphi = \frac{3}{2\sqrt{2}} r I' \cos \varphi \\
 I_{m-3} &= I_{m+3} = \frac{3}{U\sqrt{2}} V'_{m\pm 2} I' \\
 I_m &= \frac{6}{U\sqrt{2}} V'_{2m\pm 1} I' \cos \varphi^2
 \end{aligned} \right\} \tag{II.15}$$

II.2.4 Technique MLI par Injection d'harmonique trois dans la référence : [SEG.95]

Cette technique a été déduite de la technique sinus-triangle par injection de l'harmonique trois à la référence. La charge étant équilibrée. Cette injection n'aura pas d'effet nuisible car les harmoniques d'ordre trois et multiple de trois n'apparaissent pas.

L'addition d'harmonique 3 permet d'augmenter l'amplitude maximale du fondamental dans la référence et, par-là, dans les tensions de sortie.

La commande utilise une tension de référence de la forme :

$$v_A - v_O = \frac{U}{2}(r \sin \omega t + k \sin 3\omega t) \tag{II.16}$$

est appelée *commande suboptimale*.

Si la référence est sinusoïdale, $(v_A - v_O)_w$ égale à v'_{1w} , ne peut excéder $U/2$, du moins si m est grand (**figure II.6.a**).

Si la référence $(v_A - v_O)_w$ est la somme d'un terme fondamental v'_{1w} et d'un harmonique trois v'_{3w} de valeur suffisante, les maxima de la référence ne se situent plus au milieu de ses alternances. Sans que le tension $(v_A - v_O)_w$ excède $U/2$, on peut obtenir une valeur de v'_{1w} supérieur à $U/2$ (**figure II.6.b**).

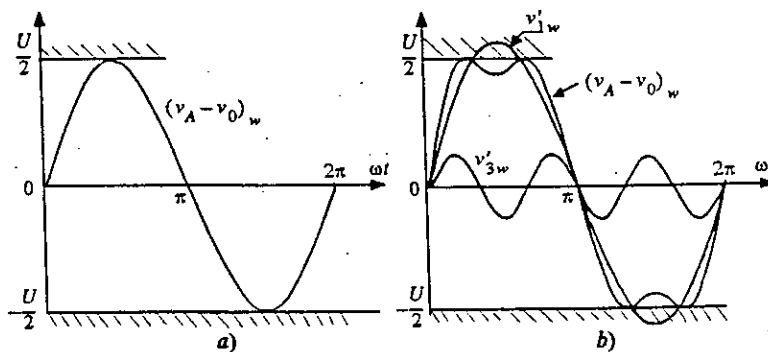


Figure II.6.a) La référence .b) Terme fondamental

• On trouve d'autres lois de commande avec injection d'harmonique 3 dans la référence.

Certaines utilisent une autre valeur relative de cet harmonique ou le font varier en même temps que le fondamental. Par exemple, on trouve

$$(v_A - v_O)_w = r \frac{U}{2} (\sin \omega t + \frac{1}{4} \sin 3\omega t) \tag{II.17}$$

- Parfois en plus de l'harmonique 3, on injecte de l'harmonique 9 avec une référence de la forme

$$(v_A - v_O)_W = \frac{U}{2}(r \sin \omega t + k_3 \sin 3\omega t - k_9 \sin 9\omega t) \tag{II.18}$$

ou

$$(v_A - v_O)_W = r \frac{U}{2}(\sin \omega t + k'_3 \sin 3\omega t - k'_9 \sin 9\omega t)$$

Sans diminuer sensiblement r_{max} on peut, si m est grand, réduire ainsi très fortement les harmoniques de rang $m \pm 3$ du courant d'entrée.

- Le choix de référence dépend de l'importance attachée au résidu de tension, aux harmoniques des tensions de sortie, aux harmoniques du courant d'entrée.

II.2.5 Autres modification de la loi de commande : (Modulation partielle) :

• Le fondamental V'_1 des tensions de sortie ne doit pas varier dans le cas où l'onduleur joue le rôle d'une source de tension alternative constante ; il suffit pour cela de compenser les effets des variations de la charge et de la valeur de la tension continue. Il est alors intéressant d'utiliser la *Modulation de Largeur d'Impulsions partielle*. [SEG.95].

On supprime les ordres de commutations pendant la partie centrale de chaque alternance de la tension de référence, par exemple pendant le tiers de celle-ci (figure II.7). Cela nécessite une commande un peu plus compliquée et limite la plage des variations des tensions de sortie mais présente deux avantages :

- le nombre de commutations est réduit, ce qui entraîne une diminution des pertes par commutation ;
- le déchet de tension est réduit car les intervalles de très faibles durées situés au milieu des alternances sont supprimés.

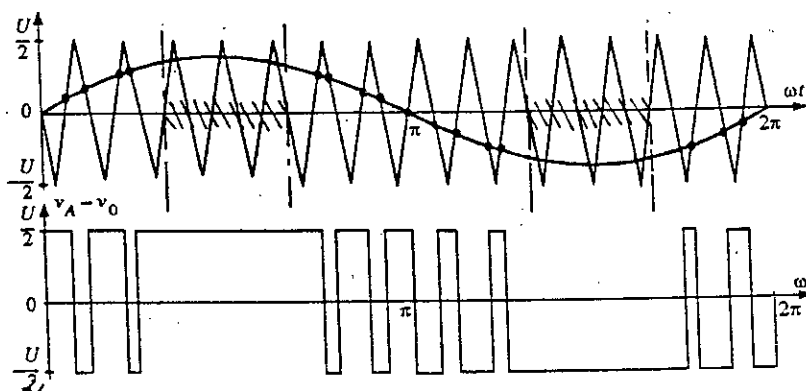


Figure II.7 : Technique de la modulation partielle

Conclusion

Dans ce chapitre, on a présenté différents principes de la technique PWM. Leurs avantages et leurs inconvénients qui est surtout la présence d'harmoniques.

Pour cela, on a abordé quelques techniques pour la minimisation de ces derniers. Des méthodes classiques comme la SPWM (modulation sinus-triangle) sont réalisées à partir de circuits analogiques relativement simples (bascule à hystérésis), mais en point de vue pratique, on essaye d'utiliser la modulation calculée PPWM afin de déterminer les angles de commutations, pour les programmer sur microprocesseur.

CHAPITRE III

INTRODUCTION:

dans ce chapitre on s'intéressera à la résolution des systèmes d'équations non linéaire de la forme suivante :

$$\left\{ \begin{array}{l} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \dots\dots\dots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{array} \right.$$

Pour ça nous allons introduire dans ce chapitre en premier une méthode de résolution des systèmes d'équation non linéaire, méthode de NEWTON-RAPHSON.

Pour l'implémentation deux cas se présente:

- Stocké les données dans une mémoire table LUT (look up table).
- Génére les données à partir de formules approximative. (approximer les solutions)

Dans la deuxième partie du chapitre on parlera sur les méthodes d'interpolation polynomiale leur avantages et leurs inconvénients.

RESOLUTION DES SYSTEME NON LINEAIRE F(X)=0**III.1 Méthode de RAPHSON-NEWTON :**

Principe :

III.1.1 Formulation développée :

Soit $X^{(0)} = (x_1^{(0)}, \dots, x_m^{(0)})$ un point initial donné de R^m de composantes $x_i^{(0)}, i = 1, \dots, m$.

$X^{(0)}$ est considéré comme une approximation de la solution $X^* = (x_1^*, \dots, x_m^*) \in R^m$ du système non linéaire $F(X)=0$.

$f_i^{(n)}$ désigne $f_i(X^{(n)})$

II.1.2 Formulation matricielle :

Considérons la matrice carrée d'ordre m définie par :

$$g(X) \left[\frac{\partial f_i}{\partial x_j}(X) \right] \quad i = 1, \dots, m; \quad j = 1, \dots, m$$

dont l'élément en position i, j est $\frac{\partial f_i}{\partial x_j}(X)$.

Cette matrice $g(X)$ est appelée matrice jacobienne des fonctions, $i=1, \dots, m$ évaluée au point $X \in R^m$.

Le déterminant de cette matrice " $\det(g(X))$ " est appelé le Jacobien des fonctions $f_i, i=1, \dots, m$ au point $X \in R^m$.

A l'aide de ces notations, nous pouvons écrire le système linéaire (1.5) sous la forme :

$$g(X^{(n)})\Delta X^{(n)} + F(X^{(n)}) = 0 \quad (1.6)$$

dont la solution, lorsque le Jacobien n'est pas nul au point, s'exprime par :

$$\Delta X^{(n)} = -g^{-1}(X^{(n)})F(X^{(n)}) \quad (1.7)$$

d'où l'expression de la formule itérative récurrente sous la forme :

$$X^{(n+1)} = X^{(n)} - g^{-1}(X^{(n)})F(X^{(n)}) \quad (1.8)$$

III.1.3 Variante de la méthode de Raphson-Newton

La méthode de Newton décrite ci-dessus, implique à chaque itération, soit la résolution d'un système linéaire du type (I-5), soit l'inversion de la matrice jacobienne $g(X^{(n)})$. Afin d'éviter une telle répétition de calculs, le processus simplifié suivant peut être utilisé :

Le calcul de l'inverse de la matrice jacobienne n'est effectué qu'à la première itération. Cela veut dire que seule $g^{-1}(X^{(0)})$ est calculée. C'est cette matrice qui sera utilisée dans toutes les itérations suivantes. Ainsi donc, la formule itérative récurrente dans cette variante est donnée par :

$$X^{(n+1)} = X^{(n)} - g^{-1}(X^{(0)})F(X^{(n)}) \quad (1.9)$$

Il est à remarquer que cette variante diminue notablement le nombre de calculs à effectuer ; toutefois, même sous les conditions de convergence dont nous allons parler, la rapidité de convergence est très ralentie par rapport à la méthode classique.

III.1.4 Conditions de convergence :

repetés

→ La suite des itérés de Newton converge sous les conditions suffisantes suivantes:

→ Si les fonctions $f_i(X)$ admettent des dérivées partielles premières continues dans un canevas contenant la solution X^* .

Si la matrice jacobienne est non singulière dans un voisinage de X^* .

Si le vecteur initial $X^{(0)}$ est suffisamment proche de X^* .

Alors : la suite des itérés de Newton, définie à partir de $X^{(0)}$, converge vers X^* .

repetés

La méthode de Newton, lorsqu'elle est convergente, a une convergence quadratique.

Cependant, la présence d'une racine multiple ralentit la convergence.

III.1.5 Avantages et inconvénients de la méthode de Newton :

• Avantages

- Son principal avantage est la convergence quadratique.
- En plus il suffit d'un seul vecteur de départ pour lancer le processus d'itération.

• Inconvénients

Ils sont essentiellement :

- le grand nombre de calculs effectué à chaque itération.

➤ la possibilité de non-convergence de la suite des itérés de Newton est l'inconvénient majeur de la méthode; lorsque l'approximation $X^{(0)}$ n'est pas suffisamment proche de la solution cherchée X^* ou lorsque des irrégularités dans la fonction se présentent au voisinage de cette solution, la méthode ne peut être utilisée.

Les divers cas de non-convergence sont:

- Non-convergence due à des irrégularités de la matrice jacobienne en X^* .
- Interruption de la suite itérative par rejet à l'infini, lorsque l'itéré $X^{(n)}$ est très éloigné de X^* .
- Attraction répulsion par des points X de R^m vérifiant $\text{Det}(g(X)) = 0$.

III.1.6 Critère d'arrêt :

Le processus itératif peut être arrêté :

- soit sur un critère d'erreur absolue c'est-à-dire, étant donnée une erreur absolue tolérée ε_a fixée, si l'une des inégalités suivantes est vérifiée :

$$\max_{j=1, \dots, m} |x_j^{(n)} - x_j^{(n-1)}| \leq \varepsilon_a \quad (1.10)$$

ou :

$$\sum_{j=1}^m (x_j^{(n)} - x_j^{(n-1)})^2 \leq \varepsilon_a \quad (1.11)$$

- soit sur un critère d'erreur relative c'est-à-dire, étant donnée une erreur relative tolérée ε_r fixée, si l'une des inégalités suivantes est vérifiée:

$$\max_{j=1, \dots, m} |x_j^{(n)} - x_j^{(n-1)}| \leq \varepsilon_r \max_j |x_j^{(n)}| \quad (1.12)$$

ou :

$$\sum_{j=1}^m (x_j^{(n)} - x_j^{(n-1)})^2 \leq \varepsilon_r \sum_{j=1}^m (x_j^{(n)})^2 \quad (1.13)$$

ou :

$$|x_j^{(n)} - x_j^{(n-1)}| \leq \frac{\varepsilon_r}{2} |x_j^{(n)} + x_j^{(n-1)}| \quad j=1, \dots, m \quad (1.14)$$

Il est à remarquer que les inégalités (I-10) à (I-14), lorsqu'elles sont satisfaites, ne signifient pas que $X^{(n)}$ est une bonne approximation de X^* . D'autre part elles ne fournissent aucune estimation de la précision sur $X^{(n)}$.

III.2 INTERPOLATION POLYNOMIALE

→ Il s'agit du cas particulier $\varphi_0(x) = 1, \varphi_1(x) = x, \dots, \varphi_n(x) = x^n$. La fonction d'interpolation devient alors le polynôme d'interpolation p_n , de degré $\leq n$, dont les coefficients a_0, \dots, a_n sont solution du système linéaire

$$\left| \begin{array}{l} \sum_{k=0}^n a_k x_i^k = f(x_i) \\ i = 0, \dots, n. \end{array} \right. \quad (2.1)$$

Le déterminant (du type vandermonde) de ce système vaut $\Delta = \prod_{r > m} (x_r - x_m)$; il est par conséquent non nul si les points x_i sont distincts (ce que nous supposons). D'où :

THEOREME. Le polynôme d'interpolation p_n , de degré $\leq n$, de la fonction f c'est à dire le polynôme vérifiant les conditions
 $P_n(x_i) = f(x_i)$ pour $i = 0, \dots, n$
 existe et est unique.

Il n'est pas indiqué de déterminer p_n en résolvant le système linéaire qui donne ses coefficients ; il est classique de l'exprimer sous deux formes : polynôme d'interpolation de Lagrange et polynôme d'interpolation de Newton.

III.2.1 Polynôme d'interpolation de Lagrange

Pour faire apparaître explicitement les valeurs $f(x_i)$ dans le polynôme d'interpolation p_n , considérons les polynômes de Lagrange L_i , définis par

$$L_i(x) = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}, \quad (2.2)$$

Qui sont de degré $\leq n$ et qui vérifient les conditions :

$$\begin{aligned} L_i(x_i) &= 1 \\ L_i(x_j) &= 0 \quad \text{pour toutes les valeurs de } j \text{ différentes de } i. \end{aligned} \quad (2.3)$$

Il est alors évident que le polynôme

$$\sum_{i=0}^n L_i(x) f(x_i) \quad (2.4)$$

est le polynôme d'interpolation P_n de la fonction f :

$$P_n(x) = \sum_{i=0}^n L_i(x) f(x_i)$$

REMARQUES :

1) En posant $f(x) = x^k$, $k \leq n$, les polynômes de Lagrange sont tels que

$$\sum_{i=0}^n x_i^k L_i(x) = x^k ;$$

2) En écrivant $\pi(x) = (x - x_0) \dots (x - x_n)$, le polynôme de Lagrange L_i s'exprime sous la forme

$$L_i(x) = \frac{1}{x - x_i} \frac{\pi(x)}{\pi'(x_i)} ;$$

3) Le polynôme d'interpolation mis sous forme de Lagrange se prête mal à la modification du nombre des points x_i .

III.2.1.1 APPROXIMATION : CAS PARTICULIER

Les points x_i sont équidistants et rangés en ordre croissant, $x_i = x_0 + ih$ (avec h le pas de la table numérique). En faisant le changement de variable $x = x_0 + uh$ il vient

$$L_i = \frac{(-1)^{n-i}}{n!} C_n^{n-i} \frac{u(u-1)\dots(u-n)}{u-i} \quad (2.5)$$

Cette expression est indépendante du pas h et peut être tabulée pour différentes valeurs de u : tables des coefficients de Lagrange.

III.2.2 Différences divisées d'une fonction

III.2.2.1 DEFINITION

Etant donnés des points x_0, x_1, \dots, x_n distincts et quelconques et les valeurs correspondantes $f(x_0), f(x_1), \dots, f(x_n)$ d'une fonction f , on appelle différence divisée d'ordre 0, 1, ..., n de la fonction f les expressions suivantes :

$$\delta[x_i]f = f(x_i)$$

$$\delta[x_i, x_j]f = \frac{f(x_i) - f(x_j)}{x_i - x_j} \quad (i \neq j)$$

$$\delta[x_i, x_j, x_k]f = \frac{\delta[x_i, x_j]f - \delta[x_i, x_k]f}{x_j - x_k} \quad (j \neq k)$$

$$\delta[x_0, \dots, x_{n-2}, x_{n-1}, x_n]f = \frac{\delta[x_0, \dots, x_{n-2}, x_{n-1}]f - \delta[x_0, \dots, x_{n-2}, x_n]f}{x_{n-1} - x_n} \quad (2.6)$$

En fait, ces expressions sont indépendantes de la numérotation des points x_i (donc de l'ordre dans lequel ils interviennent); en effet, il est facile de montrer par récurrence que

$$\delta[x_0, \dots, x_n]f = \sum_{i=0}^n \frac{f(x_i)}{\pi(x_i)} \quad (2.7)$$

Où $\pi(x) = (x - x_0) \dots (x - x_n)$. Cette relation prouve alors que les points x_i et les valeurs $f(x_i)$ interviennent symétriquement.

Il est donc simple de définir la différence divisée d'ordre n par la formule récurrente

$$\delta[\sigma, a, b]f = \frac{\delta[\sigma, a]f - \delta[\sigma, b]f}{a - b}, \quad (2.8)$$

Où σ désigne un ensemble de points x_i et a, b deux points supplémentaires.

III.2.2.2 Formule des différences divisées

D'après ce qui vient d'être dit, nous pouvons écrire successivement

$$\delta[x_0, x]f = \frac{f(x) - f(x_0)}{x - x_0},$$

D'où
$$f(x) = f(x_0) + (x - x_0)\delta[x_0, x]f,$$

Puis
$$\delta[x_0, x_1, x] f = \frac{\delta[x_0, x] f - \delta[x_0, x_1] f}{x - x_1},$$

D'où
$$\delta[x_0, x] f = \delta[x_0, x_1] f + (x - x_1) \delta[x_0, x_1, x] f, \tag{2.9}$$

C'est à dire, en portant dans l'expression précédente de $f(x)$:

$$f(x) = f(x_0) + (x - x_0) \delta[x_0, x_1] f + (x - x_0)(x - x_1) \delta[x_0, x_1, x] f. \tag{2.10}$$

En continuant ainsi on obtient la formule des différences divisées :

$$f(x) = f(x_0) + (x - x_0) \delta[x_0, x_1] f + (x - x_0)(x - x_1) \delta[x_0, x_1, x_2] f + \dots$$

$$\dots + (x - x_0) \dots + (x - x_{n-1}) \delta[x_0, \dots, x_n] f$$

$$+ (x - x_0) \dots (x - x_n) \delta[x_0, \dots, x_n, x] f.$$

CONSEQUENCE :

La différence divisée d'ordre q+1 d'un polynôme p , de degré q, est nulle.

En effet, les identités

$$p(x) = p(x_0) + (x - x_0) \delta[x_0, x] p$$

$$p(x) = p(x_0) + (x - x_0) \delta[x_0, x_1] p + (x - x_0)(x - x_1) \delta[x_0, x_1, x] p$$

montrent que les différences divisées

$$\delta[x_0, x] p$$

$$\delta[x_0, x_1, x] p$$

sont des polynômes de degré q-1, q-2,

III.2.2.3 Polynôme d'interpolation de NEWTON

Considérons les polynômes de base

$$\pi_0(x) = 1,$$

$$\pi_1(x) = (x - x_0),$$

$$\pi_2(x) = (x - x_0)(x - x_1), \dots, \pi_n(x) = (x - x_0) \dots (x - x_{n-1}),$$

et écrivons le polynôme d'interpolation sous la forme

$$p_n(x) = \sum_{s=0}^n \lambda_s \pi_s(x). \quad (2.11)$$

Les coefficients λ_s sont déterminés par la résolution du système linéaire triangulaire

$$\begin{aligned} f(x_0) &= \lambda_0 \\ f(x_1) &= \lambda_0 + (x_1 - x_0)\lambda_1 \\ &\vdots \\ f(x_n) &= \lambda_0 + (x_n - x_0)\lambda_1 + \dots + (x_n - x_0)\dots(x_n - x_{n-1})\lambda_n. \end{aligned}$$

III.2.2.4 APPROXIMATIONS

Il apparaît alors que λ_s ne dépend que des points x_i et des valeurs $f(x_i)$ pour lesquels $i \leq s$. L'expression du polynôme d'interpolation sous forme de Newton se prête donc à la modification du nombre des points x_i (il suffit d'ajouter ou de retrancher des équations au système) ; elle est la base de l'obtention des formules usuelles d'interpolation.

L'expression des λ_s est obtenue en comparant les équations du système linéaire avec la formule des différences divisées appliquée au polynôme p_n et dans laquelle on fait $x = x_0, x_1, \dots, x_n$. On en déduit immédiatement

$$\lambda_s = \delta[x_0, x_1, \dots, x_s] f. \quad (2.12)$$

D'où le polynôme d'interpolation de Newton

$$\begin{aligned} p_n(x) &= f(x_0) + (x - x_0)\delta[x_0, x_1]f + (x - x_0)(x - x_1)\delta[x_0, x_1, x_2]f \\ &\quad \dots + (x - x_0)\dots(x - x_{n-1})\delta[x_0, \dots, x_n]f. \end{aligned}$$

Ou, rappelons-le, les points x_i sont quelconques (mais tous distincts) et interviennent dans un ordre arbitraire.

III.2.3 Convergence de l'interpolation polynômiale

Le problème qui se pose est le suivant : étant donné le polynôme d'interpolation p_n d'une fonction f construit sur $n+1$ points x_i appartenant à un intervalle (a, b) , l'erreur d'interpolation $\varepsilon(x) = f(x) - p_n(x)$ devient-elle nulle lorsque n augmente indéfiniment, autrement dit, l'interpolation est-elle de plus en plus précise lorsque le nombre de points x_i utilisés dans (a, b) est de plus en plus grand ? Ce problème n'a pas de réponse positive dans tous les cas, même lorsque la fonction f et ses dérivées de tous ordres sont continues.

THEOREME. Si la fonction f est analytique dans l'intervalle (a, b) , c'est-à-dire possède un développement en série entière

$$f(x) = a_0 + a_1(x - c) + a_2(x - c)^2 + \dots$$

qui converge pour tout $x \in (a, b)$, alors

$$\lim_{n \rightarrow \infty} \max_{x \in (a, b)} |f(x) - p_n(x)| = 0.$$

II.2.4 Avantages et inconvénients des méthodes d'interpolation

L'avantage que présente ces méthodes est que les points à approcher sont récupérés à partir du polynôme interpolateur sans erreurs, et on a une bonne approximation pour les points intermédiaires.

L'inconvénient est que le degré du polynôme est d'autant plus grand que le nombre de points à approximer.

III.3 INTERPOLATION PAR LA METHODE DES MOINDRES CARRES

III.3.1 Définition:

Cette méthode consiste à déterminer les coefficients a et b de la "meilleure" droite $y=ax+b$ "passant" Par les points à approximer.

Les coefficients a et b sont déterminés tels que la somme des carrés des résidus soit minimale. On veut donc minimiser

$$S = \sum_{i=1}^m r_i^2 = \sum_{i=1}^m (y_i - a - bx_i)^2 \quad (3.1)$$

Où m est le nombre total de points et r le résidu.

Géométriquement les résidus sont les distances entre les points et la droite d'approximation

$$y=ax+b. \quad (3.2)$$

Pour que la somme soit minimale, sa dérivée par rapport à a et b doit être nulle.

$$\begin{cases} \frac{\partial S}{\partial a} = -2 \sum_{i=1}^m (y_i - a - bx_i) = 0 \\ \frac{\partial S}{\partial b} = -2 \sum_{i=1}^m (y_i - a - bx_i)x_i = 0 \end{cases} \quad (3.3)$$

Ce système d'équation linéaire peut s'écrire de la manière suivante :

$$\begin{cases} \alpha_{11} a + \alpha_{12} b = c_1 \\ \alpha_{21} a + \alpha_{22} b = c_2 \end{cases} \quad (3.4)$$

avec

$$\alpha_{11} = m, \quad \alpha_{12} = \alpha_{21} = \sum_{i=1}^m x_i$$

$$\alpha_{22} = \sum_{i=1}^m x_i^2, \quad c_1 = \sum_{i=1}^m y_i, \quad c_2 = \sum_{i=1}^m y_i x_i.$$

Pour trouver les coefficients a et b il faut résoudre le système ci-dessus soit par substitution ou par les méthodes matricielles.

En notation matricielle, les équations deviennent:

$$\alpha x = c \text{ tel que}$$

$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}, \quad x = \begin{bmatrix} a \\ b \end{bmatrix}, \quad c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

Et la solution est donnée par :

$$x = \alpha^{-1} c$$

III.3.2 Méthode des moindres carrés généralisée

Dans le cas où la droite de lissage présente une grande erreur, on généralise la méthode des moindres carrés pour un polynôme.

$$P(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n \quad (3.5)$$

Pour trouver ce polynôme on doit minimiser la somme suivante :

$$S = \sum_{i=0}^M [y_i - a_0 - a_1 x_i - \dots - a_n x_i^n]^2 \quad (3.6)$$

Les données (x_i, y_i) étant spécifiées et $m < N$.

Nous opérons comme dans le cas le plus simple de la droite. En égalant à 0, les dérivées relatives à a_0, a_1, \dots, a_n , nous obtenons les $n+1$ équations.

$$\frac{\partial S}{\partial a_k} = -2 \sum_{i=0}^M x_i^k [y_i - a_0 - a_1 x_i - \dots - a_n x_i^n] = 0 \quad (3.7)$$

CHAPITRE IV

IV.1 Présentation générale

Le 68HC11, est un micro contrôleur 8 bits de Motorola. La majorité des 68HC11 adopte l'architecture interne qu'on peut découvrir dans son schéma synoptique présenté dans la figure1.

Autour de l'unité centrale du 68HC11propement dite, on trouve tout d'abord la mémoire.

Elle est subdivisée au maximum en trois blocs distincts dont la taille et la présence varient selon les références exactes du circuit.

La ROM qui peut être présente ou absente. Lorsqu'elle est présente elle peut être de l'un des types suivants:

- ROM programmable par masque (lors de la fabrication du circuit)
- EPROM : ROM programmable électriquement. Dans ce cas la ROM peut être non effaçable ; elle s'appelle alors OTPROM c'est-à-dire ROM programmable une seule fois. Elle peut aussi être du type UVPROM c'est-à-dire programmable électriquement et effaçable aux ultraviolets.

En plus de la ROM on trouve une EEPROM (mémoire programmable et effaçable électriquement) qui est destinée aux données sauf dans des cas rares tels que le test ou la mise au point de courts programmes.

Enfin une RAM est intégrée avec le système. Elle est réservée aux divers échanges de données.

L'unité centrale est entourée d'un certain nombre de ports parallèles baptisés portA à portE qui peuvent être bidirectionnels ou unidirectionnels selon le cas. Certaines lignes de ces ports sont également partagées avec d'autres ressources internes et ne sont donc pas nécessairement accessibles directement en permanence. Ainsi par exemple le port E est commun avec le convertisseur analogique numérique.

Des entrées/sorties séries sont aussi disponibles et peuvent fonctionner en mode synchrone ou asynchrone selon que l'on utilise la SPI ou la SCI.

Un timer est également disponible. Il comporte plusieurs timers très évolués ainsi qu'un accumulateur d'impulsions, une horloge temps réel et un 'chien de garde destiné à surveiller le fonctionnement du microcontrôleur.

Un convertisseur analogique numérique à 8 entrées complète cette panoplie de ressources internes.

Toute la logique nécessaire, tant au traitement des interruptions qu'à la génération de l'horloge, est intégrée dans le 68HC11 dont la mise en œuvre matérielle est ainsi fort simple.

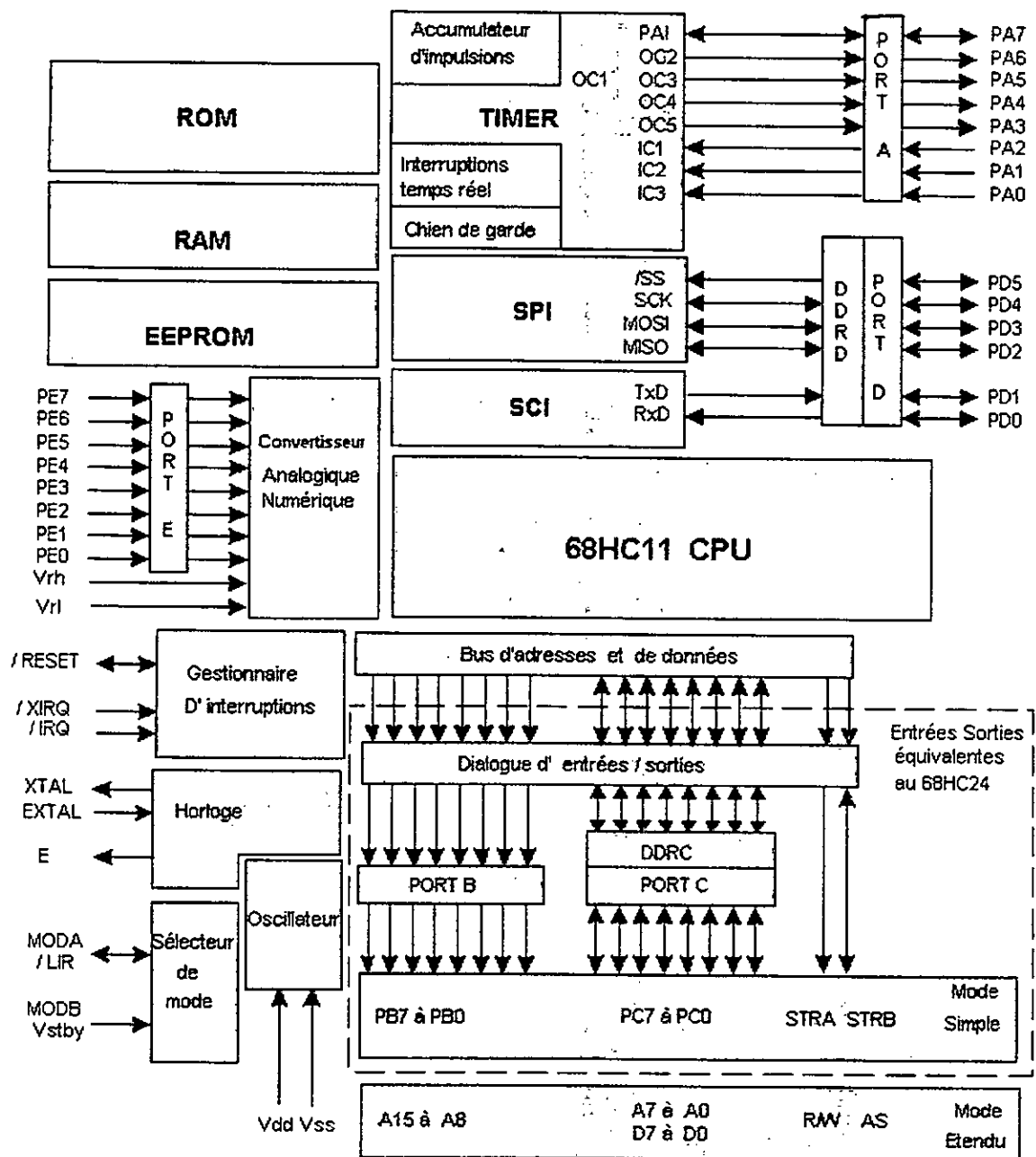


figure 1 : synoptique générale du micro-contrôleur 68HC11

IV.2 Les signaux disponibles

Parmi les broches présentes sur les boîtiers des circuits de la famille HC11 on trouve :

- **VDD (+V) et VSS (masse)** : sont les pattes d'alimentation du boîtier. Bien qu'il soit réalisé en technologie HCMOS, le 68HC11 s'alimente typiquement sous cinq volts.
- **MODA/LIRbarre et MODB/Vstby** : sont des pattes à double fonction. Hormis la phase de Reset :
 - La patte MODA devient une sortie baptisée LIRbarre.
 - La patte MODB devient une entrée d'alimentation de sauvegarde pour la RAM interne.

Lors d'un reset du circuit, les pattes sont à considérer comme des entrées appelées MODA et MODB qui permettent de sélectionner un des quatre modes de fonctionnement suivants :

- mode normal circuit seul
- mode normal circuit étendu
- mode spécial bootstrap
- mode spécial test

Tableau.1-Modes de fonctionnement du microcontrôleur 68HC11

MODA	MODB	Mode sélectionné
0	0	BOOTSTRAP- SPÉCIAL
0	1	CIRCUIT SEUL – NORMAL
1	0	TEST – SPÉCIAL
1	1	ETENDU – NORMAL

D'après le tableau (Tableau.1) les deux premiers modes sont considérés comme principaux, le troisième comme accessoire et le dernier comme mode de test, d'usage exceptionnel.

➤ LE MODE CIRCUIT SEUL (SINGLE SHIP)

C'est le mode microcontrôleur proprement dit, le circuit est configuré de façon à disposer de ses ports B et C sans exploiter le bus externe.

➤ LE MODE ETENDU OU MULTIPLEXE (EXTENDED)

Le microcontrôleur utilise des ports d'entrées/sorties parallèles comme des bus d'adresses et de données, cela lui permet ainsi d'adresser 64 ko. Le bus d'adresses est constitué du port B pour les bits de poids fort et du port C pour les bits de poids faible, ce dernier étant multiplexé pour obtenir aussi le bus de données.

Le signal (read \ write) est obtenu à partir de la ligne STRB et le signal de commande de multiplexage est obtenu à partir de la ligne STRA.

➤ LE MODE SPECIAL TEST

Ce mode est réservé au fabricant pour tester le fonctionnement du microcontrôleur en fin de chaîne de fabrication.

➤ LE MODE SPECIAL BOOTSTRAP

Ce mode charge en RAM un programme par la liaison série et lance son exécution.

Il permet à l'utilisateur de brancher le microcontrôleur avec la liaison série asynchrone de celui-ci sur un ordinateur et ainsi grâce au dialogue établi, de mettre au point et de charger un logiciel dans le microcontrôleur. Dans celui-ci, le fabricant a placé un logiciel en ROM (de \$BF40 à \$BFFF) appelé chargeur (boot loader) et son vecteur RESET.

Au moment du RESET, le microcontrôleur lit les deux connexions MODA et MODB, si celles-ci sont toutes les deux à zéro, il exécute le programme boot loader. Celui-ci initialise la liaison série pour établir la liaison avec l'ordinateur. Une fois cette liaison établie, l'ordinateur envoie le logiciel à tester dans la RAM du microcontrôleur, puis lance celui-ci à partir de l'adresse zéro. Bien entendu, pour pouvoir fonctionner avec des branchements et des interruptions il est nécessaire de prévoir aussi dans la RAM un emplacement pour la PILE et un emplacement pour les vecteurs d'interruption.

- **EXTAL, XTAL et E** : sont les broches relatives à l'horloge. EXTAL et XTAL permettent la connexion d'un quartz ou d'une horloge externe. La broche E est une sortie d'horloge bus dont l'état, haut ou bas, permet de savoir si des données ou des adresses valides sont présentes sur le bus du 68HC11.

- **RESET** : est la broche de remise à l'état initial mais, contrairement à de nombreux circuits, c'est une ligne bidirectionnelle. Utilisée en entrée et forcée à la masse, elle provoque une réinitialisation du circuit. Selon les cas le Reset peut être automatique ou manuel.
- **L'IRQ et XIRQ** : sont des entrées d'interruptions externes. IRQ est une entrée d'interruption masquable alors que XIRQ est non masquable.
- **VREFL et VREFH** : sont respectivement les entrées de référence basse et haute du convertisseur analogique/digital.
- **PE0 à PE7** : sont les entrées du port parallèle E. Ces lignes sont unidirectionnelles et ne fonctionnent qu'en entrée ; elles sont partagées avec les entrées du convertisseur analogique/digital.
- **PDO à PD5** : sont les entrées/sorties parallèles du port D qui sont communes avec les lignes des ports série de la SPI et de la SCI qui comptent :
 - La ligne de transmission de donnée série synchrone TxD.
 - La ligne de réception de donnée série RxD.
- **La ligne MOSI** : lorsque l'utilisation de l'interface SPI (Serial Peripheral interface) a été programmée, cette broche est configurée soit comme une sortie en mode maître, soit comme une entrée en mode esclave.
- **La ligne MISO** : lorsque l'utilisation de l'interface SPI (Serial Peripheral interface) a été programmée, cette broche est configurée soit comme une entrée en mode maître, soit comme une sortie en mode esclave. La broche MISO d'un esclave est placée en haute impédance quand il n'est pas sélectionné.
- **PC0 à PC7** : sont des entrées/sorties parallèles à usage général en mode circuit seul. En mode étendu, elles deviennent les lignes où circulent les données et les huit bits de poids faibles des adresses.
- **PB0 à PB7** : sont des sorties parallèles en mode circuit seul et véhiculent les 8 bits de poids forts des lignes d'adresses en mode étendu.
- **STRA et STRB** : sont respectivement AS et R/Wbarre en mode étendu. En mode circuit seul, ces lignes peuvent être utilisées pour divers protocoles de dialogue via les ports B et C.
- **PA0 à PA7** : sont les entrées (PA0, PA1, PA2), les sorties (PA3, PA4, PA5, PA6) et l'entrée/sortie (PA7) du port A. Ces lignes sont partagées avec celles du timer ce qui justifie le fait que certaines d'entre elles soient unidirectionnelles.

IV.2.1 Les registres de configuration

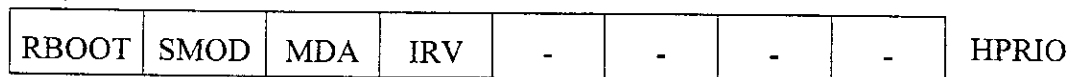
Compte tenu des différentes possibilités de configuration du 68HC11 un certain nombre de registres spéciaux ou de configuration sont présents.

• Le registre HPRI0

Le registre HPRI0 contient 4 bits qui entrent dans le cadre de cette section Les autres baptisés PSEL0 à PSEL3 sont relatifs aux interruptions.

Ce registre nous permet :

- De sélectionner un mode de fonctionnement
- De valider la lecture de la ROM de bootstrap.



• Le registre CONFIG

Le registre CONFIG est un registre un peu particulier en ce sens qu'il est en fait composé de deux registres distincts. Le premier est réalisé sous forme d'un registre EEPROM qui conserve en permanence les informations qui y sont contenues. Le second étant un registre de travail. Il est une copie du premier et peut être modifié car il est recopié après chaque reset. Il représente le seul accès possible en lecture.

L'effacement et la programmation de la version EEPROM de ce registre se font comme celle de la mémoire EEPROM sans pour autant que ce registre ne soit implanté dans l'espace mémoire de l'EEPROM.

Ce registre nous permet :

- De valider l'EEPROM
- De valider la ROM de programme en mode étendu
- De valider le COP
- De valider la sécurité du HC11 ce qui signifie que le circuit peut être lancé uniquement dans le mode circuit seul ou bootstrap.
- De déterminer le poids fort de l'adresse de base de l'EEPROM.

0	0	0	0	NOSEC	NOCOP	ROMON	EEON	CONFIG
0	0	0	0	ETAT DE CONFIG EN EEPROM				RESET

• **Le registre INIT**

Ce registre est typiquement un registre de configuration du circuit et tous ses bits sont donc protégés, c'est à dire qu'ils ne peuvent être écrits qu'une et une seule fois pendant les 64 cycles d'horloge E qui suivent un Reset.

Ce registre contient les informations qui permettent de positionner la RAM et les 64 registres internes du 68HC11 dans l'espace adressable.

Ce registre permet aussi :

- De définir les chiffres poids fort de l'adresse du bloc de 64 registres internes du 68HC11.
- De définir le chiffre poids fort de l'adresse de base de la RAM.

	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
RESET	0	0	0	0	0	0	0	0	

• **Le registre TMSK2**

Deux bits seulement sont protégés dans ce registre et permettent de fixer le taux de prédivision du timer libre contenu dans le 68HC11 par rapport à l'horloge E.

	-	-	-	-	0	0	PR1	PR0	TMSK2
RESET	0	0	0	0	0	0	0	0	

• **Le registre OPTION**

Quatre bits de ce registre sont protégés sur les sept bits significatifs qu'il contient. Il permet

De fixer le taux de prédivision du COP par rapport à $E/2^{15}$.

- De forcer un temps d'attente à l'unité centrale à la sortie du mode STOP.
- De programmer l'entrée IRQbarre à être sensible à un front descendant mémorisé ou bien à un niveau bas maintenu jusqu'à ce que l'interruption prise en compte.

	-	-	IRQE	DLY	-	0	CR1	CR0	OPTION
RESET	0	0	0	0	0	0	0	0	

VI.3 L'unité centrale

L'unité centrale du 68HC11 est un modèle 8 bits disposant de fonctionnalités particulières lui permettant d'exécuter avec un maximum de souplesse certaines instructions, y compris arithmétiques et logiques, sur 16 bits. Elle est issue de l'unité centrale, qui équipait les désormais désuets 6801, mais a été enrichie de nouvelles instructions.

L'unité centrale des 68HC11 est capable d'exécuter, au niveau code objet donc, tout programme écrit pour un 6800 ou un 6801 ce qui peut être intéressant pour faire évoluer rapidement une application existante. De ce fait, les programmes sources 6800 et 6801 peuvent évidemment être soumis eux aussi à un assembleur 68HC11 pour donner du code exécutable. Quarante-deux nouvelles instructions étant cependant disponibles par rapport aux jeux d'instructions initiaux de ces deux microprocesseurs, il est toutefois évident qu'il sera préférable de réécrire tout ou partie d'une application existante pour bénéficier pleinement de la puissance du 68HC11.

IV.3.1 Le modèle du programmeur :

Il est d'une architecture classique avec des registres aux fonctions bien définies contrairement à certaines familles de microcontrôleurs où les registres sont parfois polyvalents. On distingue :

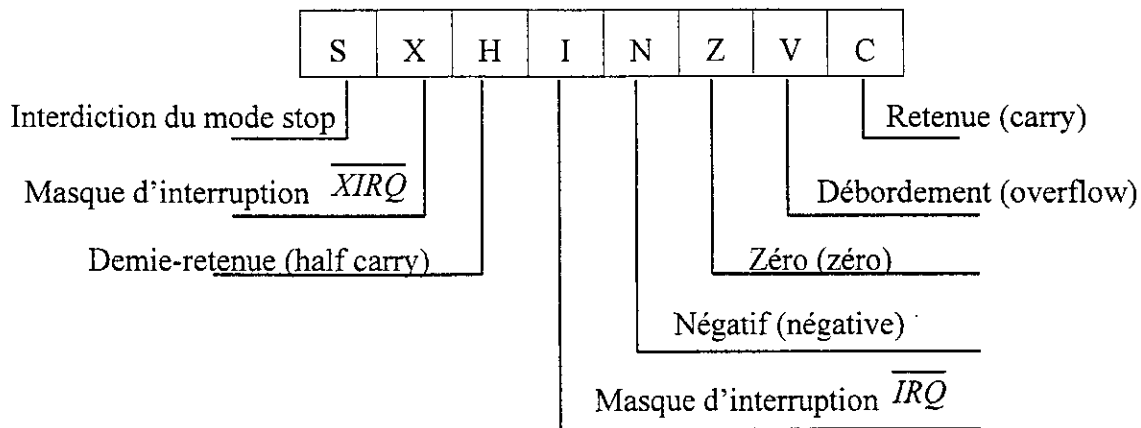
7	accumulateur A	0	7	accumulateur B	0
15	accumulateur D				0

15	Registre d'index X	0
Registre d'index Y		
Registre pointeur de pile S		
Compteur ordinal		

Registre des codes condition CC

- Deux accumulateurs A et B qui sont des registres 8 bits, leur concaténation forme un accumulateur 16 bits D.
- Deux registres d'index 16 bits utilisés principalement pour l'adressage indexé.
- Un pointeur de pile S ou SP (Stack pointer). Ce registre sur 16 bits pointe sur la première adresse libre de la zone mémoire définie pour ranger la pile. Le 68HC11 gère automatiquement ce registre ; en cas d'appel de sous programme ou d'interruptions il range sur la pile le contenu des registres internes et décrémente le registre S d'autant d'unités que nécessaires de façon à ce qu'il pointe toujours à la prochaine adresse de pile disponible. Au retour de sous programme ou d'interruptions il recharge les registres internes avec les informations reprises sur la pile et incrémente le registre S d'autant d'unités que nécessaires.
- Le registre PC (Program Counter) est un compteur ordinal de 16 bits. Il pointe sur la prochaine instruction à exécuter.

- Le registre d'état ou CCR (Condition Code Register). C'est un registre 8 bits qui nous informe sur l'état de l'unité centrale en permanence.



ce registre contient 8 bits signifiant:

- Le bit C pour Carry. Mise à 1 lorsque qu'une opération arithmétique génère une retenue.
- Le bit V pour overflow ou débordement est positionné à 1 lorsqu'une opération arithmétique a généré un débordement de l'accumulateur.
- Le bit Z pour Zéro est positionné à 1 lorsque le résultat de l'instruction exécutée est nul.
- Le bit N pour Négative est positionné à 1 si le résultat de la dernière opération arithmétique réalisée est négatif c'est-à-dire si le bit de poids fort du résultat est à 1.
- Le bit H pour Half carry ou demi-reteneue est positionnée à 1 lors d'une retenue entre les bits 3 et 4 d'une opération arithmétique.
- Le bit I pour Interrupt Mask ou masque d'interruption interdit toute interruption masquable lorsqu'il est mis à 1. Suite à un Reset, ce bit est automatiquement mis à 1 et doit être mis à 0 volontairement par logiciel pour valider les interruptions.
- Le bit X pour XIRQ interrupt mask fonctionne comme le bit I vu auparavant mais pour les interruptions susceptibles d'être appliquées sur l'entrée XIRQbarre.
- Le bit S pour Stop disable permet, lorsqu'il est mis à 1 d'interdire l'exécution de l'instruction STOP.

IV.3.2 comportement lors d'un Reset :

Opération très importante parce qu'elle conditionne l'exécution de tout programme.

Nous allons voir son comportement comme suit :

- L'unité centrale tout d'abord va chercher à l'adresse \$FFFF et \$FFFE le vecteur de Reset et charge celui-ci dans le PC.
- Toute interruption masquable et non masquable ainsi que le mode STOP sont interdits.
- Initialisation du pointeur de pile SP.
- La RAM est placée à partir de l'adresse \$0000 et les registres internes à partir de l'adresse \$1000.
- En mode circuit seul toutes les interruptions pouvant provenir des ports B et C sont interdites.
- Le port C est initialisé en entrée.
- Le port B qui est toujours en sortie voit tous ses bits mis à 0.
- Les bits 0, 1, 2 et 7 du port A, les bits 0 à 5 du port D et le port E sont configurés en entrée. Les bits de ces mêmes ports qui sont obligatoirement des sorties (port A, bits 3 à 6) seront mis à 0.
- Le timer est initialisé avec une valeur de comptage de \$0000. Tous les bits du prédiviseur sont mis à 0. Toute interruption issue du timer est interdite quelle qu'en soit la cause.
- L'accumulateur d'impulsions est invalidé, de ce fait son entrée PAI devient une entrée parallèle à usage général.
- Le COP ou chien de garde est validé selon le bit NOCOP du registre CONFIG. La vitesse de fonctionnement du COP est programmée à la plus grande valeur possible.
- L'interface série synchrone ou SPI est invalidé par un Reset. Les pattes habituellement réservées à cette ressource deviennent donc elles aussi des entrées/sorties parallèles à usage général.
- Le convertisseur analogique/digital est dans un état indéterminé après un Reset.
- Seule la lecture de l'EEPROM est possible.
- La plus haute priorité est conférée à IRQbarre programmée pour être sensible à un niveau bas.

IV.3.2.1 Les différentes sources de Reset :

Le 68HC11 dispose de trois sources internes susceptible de provoquer également un Reset indépendamment du Reset externe, chacun de ces trois Reset dispose d'un vecteur permettant ainsi une parfaite adaptation du comportement du 68HC11 à la cause réelle du Reset.

Ces Reset internes provoquent le passage au niveau bas de la ligne Reset du boîtier qui se comporte comme une sortie.

Ces trois Reset sont les suivants :

➤ Reset suite à la mise sous tension ou POR (*Power On Reset*) maintient la patte Reset au niveau bas pendant 4 096 cycles d'horloge E, ce Reset n'est réarmé que si la tension d'alimentation VDD ne tombe à 0.

➤ Le COP ou chien de garde est un compteur qui décompte sans arrêt. Lorsqu'il n'est pas rechargé par le logiciel un Reset est généré lorsque le Cop atteint 0.

➤ Le circuit de surveillance de fréquence d'horloge. Il fonctionne avec une cellule R-C interne au 68HC11 qui génère un Reset lorsqu'il ne détecte pas de transition du signal d'horloge dans une période inférieure à sa constante de temps. Cette fonction peut être invalidée grâce au bit CME du registre OPTION.

IV.3.3 Les interruptions :

Le 68HC11 dispose de multiples possibilités d'interruptions :

Deux entrées externes IRQbarre et XIRQbarre, un certain nombre de sources internes liées aux différentes interfaces qui équipent le circuit. Une interruption par logiciel grâce à l'instruction SWI.

Le comportement en présence d'une interruption est le suivant :

- L'instruction en cours d'exécution est terminée.
- Le contenu de tous les registres de l'unité centrale est poussé sur la pile dans un ordre précis.

- Le masque d'interruption I du CCR est positionné à 1 dans le cas de l'IRQ. Les bits X et I du CCR sont mis à 1 pour une interruption du type XIRQ étant donnée qu'elle soit plus prioritaire.

- Le vecteur d'interruptions est chargé dans le PC et l'exécution du programme d'interruption est lancée.

En fin d'exécution de ce programme les opérations inverses ont lieu.

La XIRQbarre est plus prioritaire que la IRQbarre.

IV.3.3.1 Priorité des interruptions

Les priorités des interruptions sont fixes sauf pour l'une d'entre elles que l'on peut définir grâce au registre HPRIO (*Highest PRIOrity interrupt*). Si l'on considère le Reset comme une source d'interruption, l'ordre s'établit comme suit :

- Reset de mise sous tension.
- Reset par action sur l'entrée Reset.
- Reset par la circuiterie de surveillance d'horloge.
- Reset par le COP.
- Interruption par code d'instruction illégal.
- Interruption XIRQbarre.
- Interruption définie par le registre HPRIO.
- Autres interruptions.

Les autres interruptions sont :

- Débordement timer.
- Débordement de l'accumulateur d'impulsions.
- Entrée de l'accumulateur d'impulsions.
- L'interface série synchrone.
- L'interface série asynchrone.
- IRQbarre.
- Interruption temps réel.
- Interruptions des registres de capture : TIC de 1 à 3.
- Interruptions de registres de comparaison : TOC de 1 à 5

IV.4 LES MÉMOIRES :

Dans un microcontrôleur 68HC11 il cohabite trois mémoires différentes ROM, RAM et EEPROM.

- Un 68HC11 de base dispose normalement de :
- Une zone de mémoire vive ou RAM dont la taille varie de 256 octets (68HC811E2) à 1,25 KO (68HC11M2).
- Une zone mémoire ROM qui varie en espace et en technologie selon la version du microcontrôleur.
- Une zone mémoire EEPROM elle a pour rôle de mémoriser des paramètres de configuration sélectionnés par l'utilisateur ; et à conserver même en l'absence d'alimentation.

a) La RAM :

Elle est du type statique, ne nécessite aucune circuiterie externe de rafraîchissement et n'impose aucune fréquence minimale d'horloge.

Pour sauvegarder le contenu de la RAM d'une période de fonctionnement il existe deux méthodes :

- Utilisation de l'instruction stop, elle arrête l'horloge et suspend toute activité interne ce qui préserve la RAM et les registres.

Le 68HC11 étant réalisé en technologie CMOS l'essentiel de sa consommation se produit lors des transitions des signaux logiques, l'horloge arrêtée, les transitions sont absentes et le boîtier ne consomme que quelques micros-ampère.

- Pour les circuits gourmands en énergie qui demandent d'arrêter l'alimentation on utilise la patte MODB/VSTby et appliquer sur cette patte une tension qui ne sera utilisée que pour maintenir la RAM.

Une circuiterie interne au 68HC11 commute l'alimentation de la RAM de VDD à VSTby lorsque VDD devient inférieur d'environ 0,7 volts à VSTby.

b) La ROM

Elle peut être non présente dans le 68HC11 selon le boîtier retenu. Dans ce cas il est impératif au microcontrôleur de fonctionner en mode étendu avec une ROM en externe afin de stocker des vecteurs de Reset et d'interruption par exemple. Tout en étant présente la ROM pourrait être désactivée en usine ou par l'utilisateur au moyen du registre CONFIG.

On pourrait avoir :

- ROM programmable par masque dans le cas de reproduction à grande série programmable par masque elle est programmable par masque sur les versions 68HC11 sans chiffre entre le C et le 11.
- OTPROM – EPROM : ROM programmable électriquement mais non effaçable. Par exemple le 68HC711.
- EEPROM : programmable et effaçable électriquement par exemple 68HC811.

Les deux dernières ROM sont onéreuses et donc réservées à la phase de développement. Elles sont programmables par des programmeurs universels supportant le 68HC11.

Cette ROM est toujours complétée par une autre ROM supplémentaire de 192 octets implantés de \$BF40 à \$BFFF. Elle contient le bootloader.

c) l'EEPROM :

L'EEPROM de donnée équipe la majorité des versions du 68HC11 elle peut être lue sans aucune restriction ni précaution. Pour programmer écrire ou effacer cette mémoire on doit effectuer des procédures de configuration à l'aide de deux registres :

- Le registre OPTION pour piloter la génération des hautes tensions.
- Le registre PROG pour la gestion des fonctions de l'EEPROM.
- Pour effacer ou programmer l'EEPROM la génération de la haute tension se fait par un circuit interne à pompe de charge à partir de l'alimentation normale VDD. Ce circuit est piloté par deux sources d'horloge distinctes :
 - Horloge bus système E, lorsque sa fréquence est supérieure à 1,5-2 Mhz.

- Horloge générée localement par une cellule R-C interne au circuit et ce lorsque la fréquence de l'horloge E est inférieure à 1,5 Mhz. La fréquence typique de la cellule est de 2,5 Mhz.

La sélection de l'horloge se fait au moyen du bit CSEL (bit 6) du registre OPTION. Lorsque ce bit est à 0, l'horloge E est utilisée et sa mise à 1 provoque l'utilisation de l'horloge R-C.

La gestion des fonctions particulières de l'EEPROM est assurée par le registre PROG.

Ainsi on pourra contrôler :

- La génération de la tension de programmation VPP.
- Le mode de fonctionnement de cette mémoire soit en lecture soit en écriture et effacement, ce qui est une sécurité pour celle-ci.
- La sélection de l'effacement et de son mode parmi :
 - L'effacement complet.
 - L'effacement complet.
 - L'effacement ligne(16 octets).
 - L'effacement d'un octet.
- Sélectionner le côté pair ou impair de la mémoire.

Ainsi la programmation et l'effacement de l'EEPROM suivent des procédures bien précises, à respecter absolument.

IV.5 LES ENTREES SORTIES PARALLELES

Le 68HC11 dans sa version de base A8, dispose de 40 lignes d'entrées/sorties parallèles. La majorité d'entre elles reposent sur le principe utilisé sur tous les circuits d'interface parallèle et microcontrôleurs 8 bits de Motorola à savoir que chaque ligne est programmable en entrée ou en sortie, sous le contrôle d'un bit de direction de fonctionnement lui correspondant, dans un registre appelé DDR pour (*Data Direction Register*).

Cette situation simple et pratique que l'on rencontre sur le PIA 6821 ou sur les microcontrôleurs 6801, 6805 ou 68HC05 est quelque peu différente dans le cas du HC11 par le fait que certaines lignes d'entrées/sorties sont partagées avec d'autres ressources. De ce fait :

- soit leur sens est fixé définitivement (cas des ports B et E par exemple).
- soit elles sont bidirectionnelles mais l'utilisation de la ressource qui partage la patte impose temporairement la fonction et le sens de celle-ci.

Lorsque le sens de fonctionnement d'une patte est sous le contrôle d'un bit DDR, on convient qu'une ligne de port est en sortie lorsqu'il est à 1 alors qu'un 0 la programme en entrée.

Lors d'un Reset tous les bits DDR sont initialisés à 0.

Voir figure 2

IV.5.1 Le port A

Le port A est un port parallèle à usage général partagé avec le timer et l'accumulateur d'impulsions. De ce fait :

- **PA0 à PA2** : sont figées en entrées, ce sont aussi les entrées de capture IC3 à IC1 du timer.
- **PA3 à PA6** sont figées en sorties, ce sont aussi les sorties de comparaison OC1 à OC5 du timer.
- **PA7** est bidirectionnelle : elle peut être configurée comme entrée de capture ou sortie de comparaison. Si le timer à accumulation d'impulsions est utilisé, c'est également PA7 qui lui sert d'entrée.

Le sens de fonctionnement de PA7 est programmé par le bit DDRA7 contenu dans le bit 7 du registre de contrôle de l'accumulateur d'impulsions PACTL.

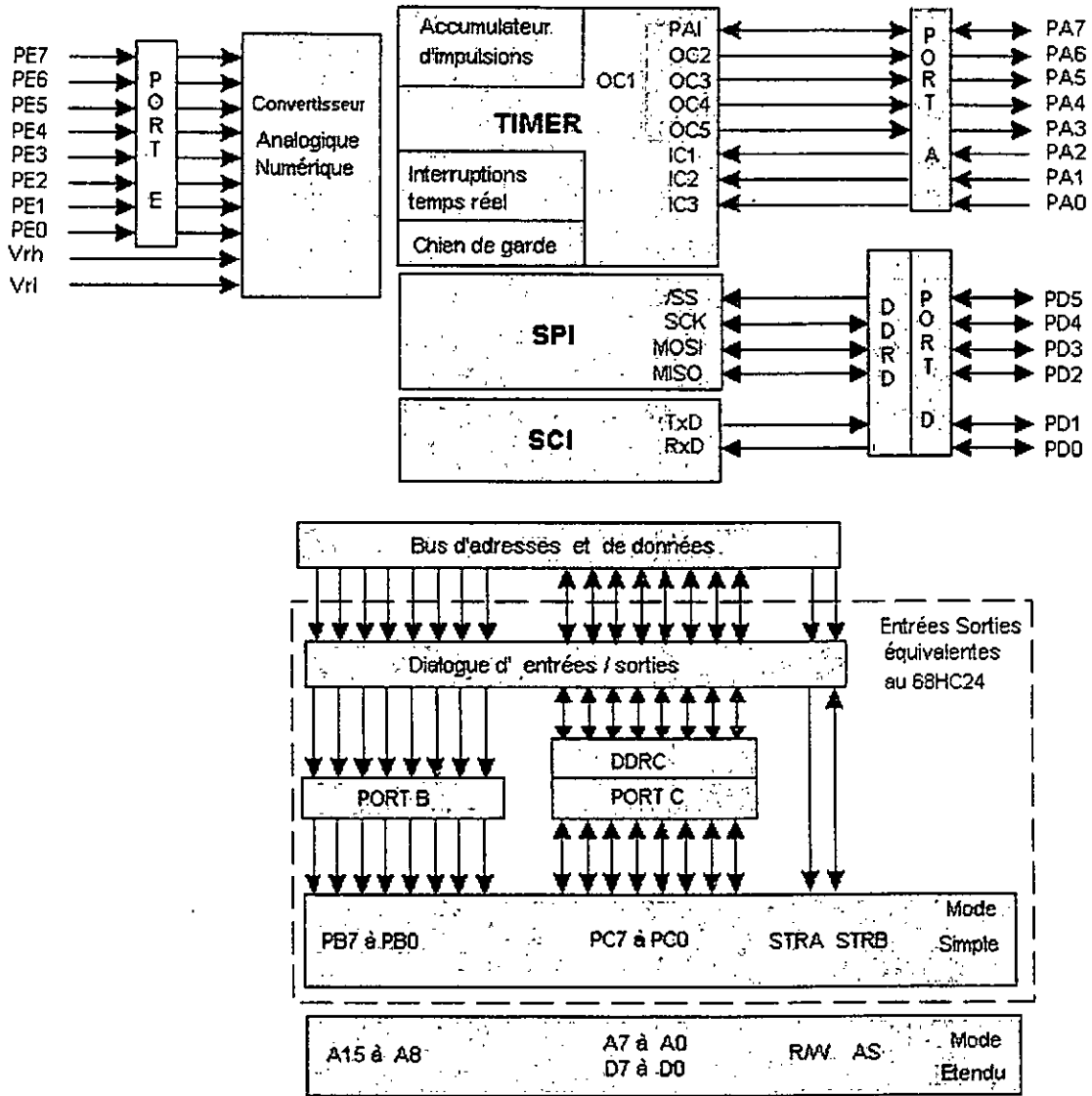


figure2: interfaces parallèles

L'état des lignes de sortie du port A est défini en écrivant dans le registre PORTA tandis que l'état des lignes d'entrées de ce même port est lu dans ce même registre.

L'état des lignes peut toujours être lu par la lecture du port A même si elles sont configurées pour être utilisées par le timer.

Lorsque les lignes sont utilisées par le timer, il est impossible de modifier leur état par une écriture dans le PORTA. Le résultat de cette écriture est cependant mémorisé dans un latch et, lorsque la ou les pattes de sortie correspondantes sont libérées par le timer, les données contenues dans ce latch sont transférées sur ces pattes.

IV.5.2 Les ports B et C :

Même si on peut parfois les considérer comme indépendants, ces ports partagent un certain nombre de mécanismes de fonctionnement communs.

En mode étendu :

- Le port C est bidirectionnel et véhicule le bus de données (dans les deux sens) et les 8 bits de poids faibles du bus d'adresses.

La ligne STRA qui s'appelle alors AS pour *Address Strobe* en mode étendu permet la distinction entre l'une et l'autre de ces informations.

- Le port B est unidirectionnel et véhicule les 8 bits de poids fort du bus d'adresses.

La ligne STRB véhicule le signal R/W barre ou lecture écriture en mode étendu. Un état haut signifie une lecture donc une entrée de données dans le 68HC11 via le port C et un état bas une écriture c'est-à-dire une sortie de données par le port C.

En mode circuit seul :

les ports B et C redeviennent des ports parallèles plus classiques mais bénéficient en outre de possibilités de dialogue (handshake) grâce aux lignes STRA et STRB.

Le port C : Le sens de chaque ligne de ce port bidirectionnel est programmé grâce aux bits correspondants du registre DDRC.

La lecture ou l'écriture des données vers le port C fait par contre appel à deux registres : le registre PORTC et le registre PORTCL qui est en fait un latch ou verrou.

En entrée, la lecture du registre PORTC est le reflet de l'état courant des lignes de ce port. La lecture de PORTCL par contre est le reflet de l'état de ces lignes au moment de la transition valide de la ligne STRA. lorsque la transition sur STRA est une impulsion on aura configuré le mode impulsif du PORT C.

En sortie, l'écriture dans PORTC n'agit pas sur les mécanismes de dialogue complets alors que l'écriture dans PORTCL y agit.

Les données tentées d'être écrites dans PORTC ou dans PORTCL, lorsque le port est en entrée, sont mémorisées localement dans un LATCH et seront présentées en sortie dès que les pattes correspondantes sont en sortie.

Le port C peut être configuré pour fonctionner en OU câblé c'est-à-dire qu'il ne dispose plus alors en interne de transistors de rappel au VDD. On peut donc relier ses lignes de sorties à d'autres sorties de circuits logiques sans craindre de destruction.

Le port B : est quant à lui un port de sortie à usage général. Il ne dispose pas de DDR car son sens de fonctionnement est figé.

Le positionnement des lignes de sortie du port B se fait en écrivant dans le registre PORTB dont les données sont mémorisées (Latchées). Le fait de lire PORTB permet de relire les données qui y ont été précédemment écrites mais sont sans rapport avec l'état réel des pattes du port.

IV.5.2.1 Les modes "dialogue" des ports B et C :

Les ports B et C supportent divers modes de dialogue qui font intervenir les lignes STRA et STRB.

STRA : entrée sensible à un front qui est le signal de saisie de données.

STRB : sortie de dialogue qui génère un signal prêt.

Ces modes se programment grâce à divers bits contenus dans le registre PIOC et sont :

	STAF	STAI	CWOM	HDNS	OIN	PLS	EGA	INVB	PIOC
REEST	0	0	0	0	0	-	1	1	

Le mode dialogue simple : il est sélectionné déjà par le Reset (mise à 0 du bit HDNS de PIOC) c'est donc le mode par défaut.

Dans ce mode :

- L'écriture dans le registre PORTB place les données correspondantes en sortie et génère une impulsion d'une durée de deux cycles d'horloge E sur la ligne STRB. La polarité de cette impulsion peut être configurée.
- Le fait de programmer ce mode dialogue simple agit également sur le principe de fonctionnement du port C. Ce dernier mémorise en effet dans son registre PORTCL les données présentes sur ses pattes au moment de la transition valide du signal STRA, il peut alors être lu.

Le mode dialogue complet en entrée : il est sélectionné par mise à 1 du bit HDNS du PIOC et par mise à 0 du bit OIN du même registre.

Dans ce mode :

- Le système connecté au 68HC11 attends une condition "prêt" matérialisée par le passage à l'état valide de STRB.
- On peut alors lire le port C et le signal STRA s'active pour assurer mémorisation des données dans PORTCL et provoquer la remise à l'état "non prêt" du signal STRB.

En variante on pourrait configurer le signal STRB en une impulsion de deux périodes d'horloge après lecture du PORTCL indiquant que le microcontrôleur est prêt pour la prochaine lecture.

Le mode dialogue complet en sortie :

- Dans ce mode, des données sont écrites sur le port C qui est en sortie. La ligne STRB passe alors à un état valide pour le signaler à un système externe
- Le périphérique lit ces données et signale à travers STRA lui une transition valide de reconnaissance *Acknowledge*.

En variante des deux derniers modes de dialogue on pourrait configurer le signal STRB en une impulsion, valide pendant deux périodes d'horloge E suite à une lecture du PORTCL indiquant que le microcontrôleur est prêt pour la prochaine lecture.

Le sens de STRB, le sens de la transition de STRA et le mode dialogue ou impulsionnel de STRB peuvent être programmés à travers les registres correspondants.

Le registre PIOC sert à sélectionner le mode de dialogue utilisé mais aussi les états actifs des divers signaux ou transitions.

IV.5.3 Le port D :

Le port D est un port d'entrées/sorties à usage général partagé avec les interfaces séries SPI et SCI.

Lorsque ces interfaces ne sont pas utilisées, le sens de travail des lignes du port est programmé par les bits correspondants de DDRD.

L'émission de données en sortie se fait par écriture dans le registre PORTD et c'est dans ce même registre que l'on vient lire l'état des pattes qui sont positionnées en entrées.

Comme pour le port C, il est possible de programmer le port D pour fonctionner en OU câblé.

Si l'une ou l'autre des interfaces séries SPI et SCI est utilisée, le sens des lignes correspondantes du port D est fixé indépendamment du DDRD.

Un autre cas particulier doit être signalé à propos du port D et de l'interface série SPI. Le sens de fonctionnement des lignes MOST et MISO est influencé par les bits correspondants du DDRD, aussi SSbarre peut être utilisée comme sortie lorsque la SPI fonctionne sous réserve que l'interface fonctionne en maître.

IV.5.4 Le port E :

Ce port, est un port d'entrée parallèle il partage ses lignes avec les entrées du convertisseur analogique/digital, mais peut tout de même fonctionner simultanément dans les deux modes. Il ne dispose donc pas de DDR et les données appliquées sur les lignes de ce port peuvent être lues à tout instant dans le registre PORTE.

IV.6 INTERFACE SERIE :

Le 68HC11 dispose de types d'entrées/sorties séries différentes :

- La SCI (Serial Communication Interface) interface série de type asynchrone.
- La SPI (Serial Peripheral Interface) interface série de type synchrone.

La SCI peut travailler comme une liaison série asynchrone micro-informatique classique et permet donc de relier le 68HC11 à n'importe quel équipement informatique disposant d'une telle liaison.

La SPI quant à elle permet de relier le 68HC11 à une multitude de dispositifs disposant d'une interface série synchrone tels que : afficheurs LCD intelligents, claviers codés, circuits pour bus "2 fils" divers, etc.

Ces deux interfaces disposent de multiples possibilités de programmation grâce à divers registre de contrôle et d'état.

IV.6.1 L'interface série synchrone ou SPI:

Cette interface est de type série synchrone et permet la connexion sous forme série maître esclave de plusieurs microcontrôleurs de la famille 68HC11 ou de familles disposant d'interfaces compatibles (famille 68HC05, par exemple). Les figures3 présentent le principe d'interconnexion retenu.

On peut relier la SPI à de très nombreux autres circuits : registres à décalage, afficheurs disposant de leur propre logique, etc.

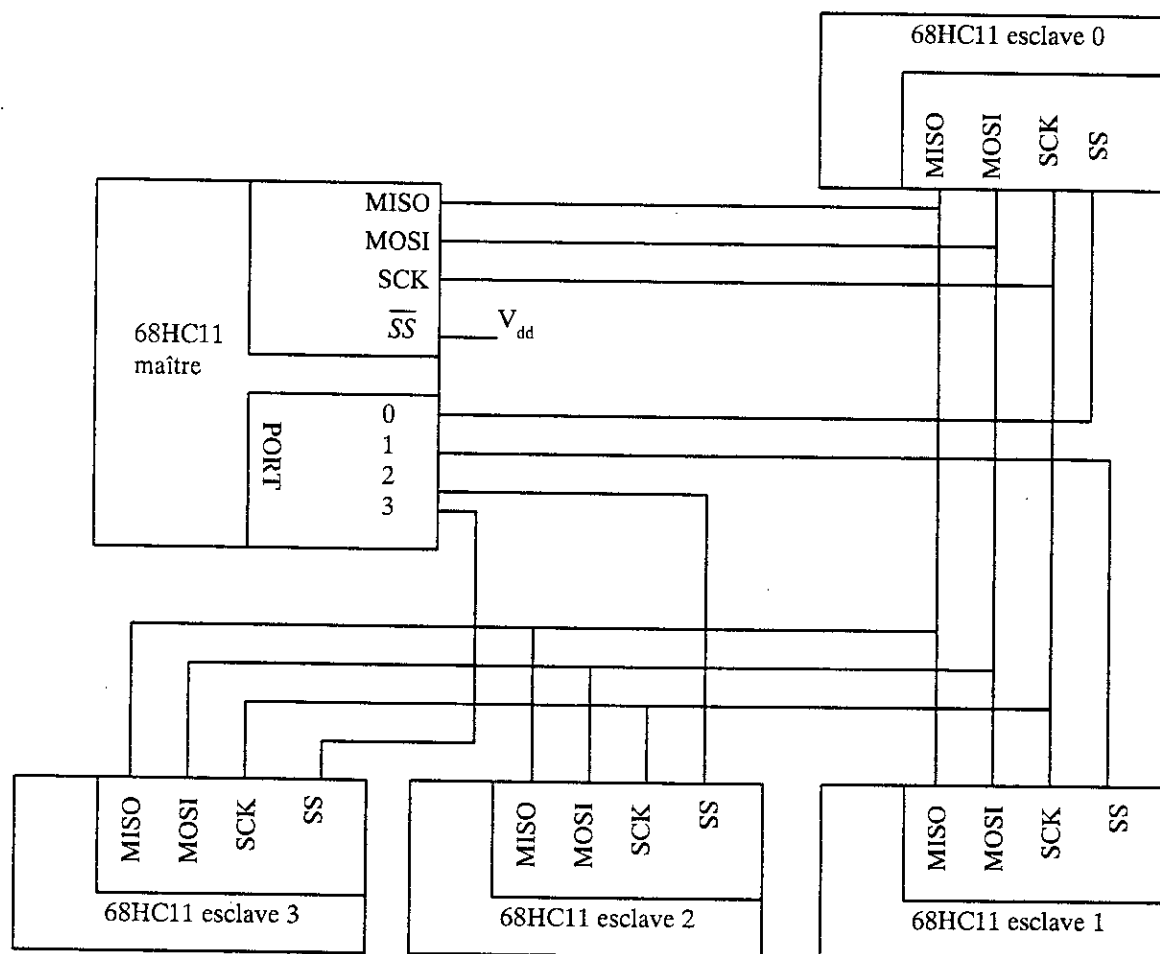


Figure 3 : principe de connexion de divers circuits via l'interface série

On discerne dans la figure, trois lignes utiles :

- MISO.
- MOSI.
- SCK.

Une ligne de sélection du mode de fonctionnement du circuit concerné SS barre.

- *MISO (Master In Slave Out)* : est configurée en entrée sur un maître ou en sortie sur un esclave. C'est donc elle qui reçoit les données qui arrivent sur le maître et c'est elle qui envoie les données qui sortent de l'esclave. Si ce dernier n'est pas autorisé à émettre, cette ligne reste en haute impédance.

- **MOSI (Master Out Slave In) :** C'est une sortie sur le maître et une entrée sur les esclaves. C'est donc par-là que le maître envoie des données et que les esclaves les reçoivent.
- **SCK (Serial Clock):** est utilisée pour synchroniser les échanges de données puisque nous sommes ici en mode série synchrone. Cette ligne est une sortie sur le maître et une entrée sur les esclaves. Cette horloge est programmée grâce aux bits CPOL et CPHA du registre SPCR.
- **SS barre :** cette ligne permet la sélection du mode maître ou esclave .Elle est au niveau haut sur un maître et au niveau bas sur un esclave. Pour éviter toute erreur de transmission, cette ligne doit être au niveau bas sur le ou les esclaves avant toute transmission et y rester pendant toute la durée de celle-ci.

La figure 4 présente le synoptique interne détaillé de l'interface SPI.

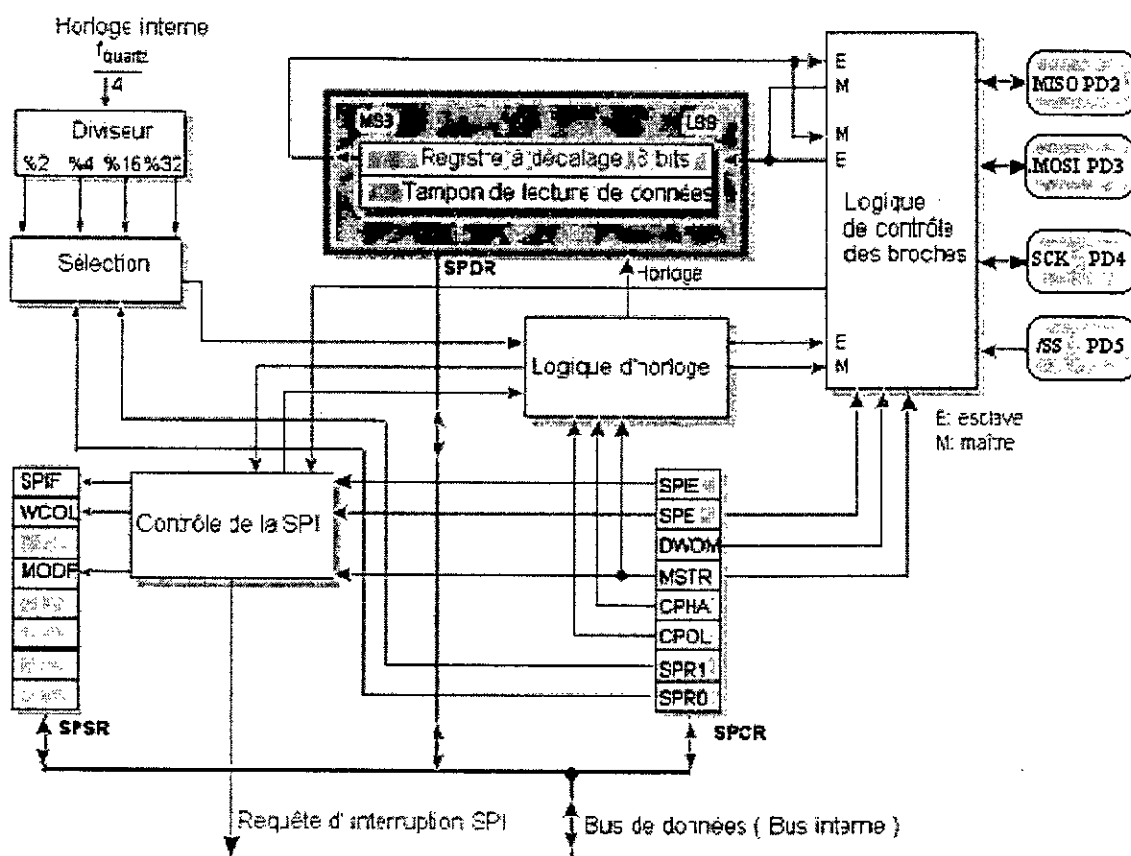


Fig 4 : synoptique interne de la SPI

IV.6.2 L'interface série asynchrone ou SCI:

L'interface série asynchrone du 68HC11 est comparable aux UART (universal asynchron receive transmit) classiques que l'on rencontre en micro-informatique mais, du fait de la possibilité de programmer la majorité de ses fonctions au moyen de registres de contrôle, elle offre quelques fonctionnalités supplémentaires.

Même si elle porte le même nom que les anciennes interfaces du même type qui équipaient ou équipent encore les 6801, 6805 et 68HC05, la logique implantée dans le 68HC11 est plus performante, particulièrement au niveau de la réception des données séries de façon à s'affranchir au maximum des problèmes de bruit et de variation de vitesse de transmission. En effet, contrairement aux UART classiques qui se resynchronisent au début de chaque caractère reçu, la SCI du 68HC11 se resynchronise sur chaque transition de 1 à 0 du signal reçu ce qui minimise l'influence d'éventuels écarts de vitesse entre émetteur et récepteur.

De plus, alors que quasiment tous les UART échantillonnent chaque bit du caractère reçu au centre de sa durée théorique, la SCI réalise la même opération trois fois de suite au voisinage de ce centre et confie à une logique à décision majoritaire la charge de déterminer si le bit reçu est à 1 ou à 0.

On s'affranchit ainsi au maximum des problèmes de bruit ou des parasites pouvant affecter la liaison.

Outre les deux particularités vues auparavant, cette interface dispose des fonctionnalités plus "classiques" suivantes

- Fonctionnement en mode full duplex sans restriction.
- Transmission de mots au format asynchrone classique (start, données utiles, stop).
- Transmission de mots de 8 ou 9 bits autorisant ainsi l'émission de parité si nécessaire.
- Vitesse de transmission programmable sur de très nombreuses valeurs, normalisées ou non, par logiciel.
- Indication d'erreur de réception analogue à celle permise par les UART : erreur de format, erreur de débordement, erreur due au bruit.
- Mode réveil automatique lors de la réception de signaux valides.
- Fonctionnement sous interruption possible avec plusieurs conditions de génération d'interruptions différentes.

Cette interface est très complète. Elle se programme comme un UART programmable quelconque.

deux registres de contrôle ou de commande fixent les modes de fonctionnement

- un registre de vitesse fixe la vitesse de transmission.
- un registre d'état indique l'état de l'interface.
- un registre d'émission reçoit l'octet à envoyer.
- un registre de réception qui fournit la donnée reçue.

Précisons que cette interface fonctionne indépendamment des timers internes des 68HC11.

IV.7 LE TIMER

Le timer du 68HC11 est un sous-ensemble très complet offrant à l'utilisateur de multiples fonctions. Même si certains signaux d'horloge générés par ses chaînes de diviseurs sont utilisés par d'autres ressources internes, il est totalement indépendant de ces dernières et aucune interaction n'est donc à craindre quels que soient ses modes de fonctionnement.

On considère que le timer du 68HC11 est divisé en quatre blocs fonctionnels :

- le timer à usage général
- le COP ou chien de garde
- le générateur d'interruptions temps réel
- l'accumulateur d'impulsions

Ces quatre blocs utilisent une source d'horloge et une chaîne de diviseurs en partie commune mais ils sont néanmoins indépendants les uns des autres et disposent tous de leurs propres registres de données, de contrôle et d'état.

IV.7.1 Le timer à usage général :

IV.7.1.1 Concept général

Le cœur de ce timer est un compteur 16 bits qui compte sans arrêt dès que l'horloge du 68HC11 fonctionne.

il positionne un bit de débordement et continue à compter; il est impossible de l'arrêter ou de modifier son contenu sauf à arrêter l'horloge ou à faire un Reset.

Ce compteur est précédé d'un prédiviseur programmable alimenté par l'horloge E. Le choix du taux de pré division permet tout à la fois de fixer la résolution du compteur mais aussi sa période de débordement qui est évidemment égale à 65536 fois cette résolution.

Parallèlement à cela ce timer dispose de trois registres 16 bits dits de capture en entrée. Ces registres sont prévus pour pouvoir être lus lorsque vous le désirez. Lorsque la patte de capture en entrée qui leur correspond passe à un niveau valide, que vous pouvez librement définir, la valeur du compteur 16 bits à cet instant est copiée automatiquement dans le registre correspondant, un bit d'état est positionné et une interruption peut être générée.

De la sorte, il est facile de dater un événement externe puisqu'il suffit qu'il provoque une transition valide sur une des pattes de capture en entrée pour que la valeur du timer à cet instant vous soit alors accessible grâce au registre de capture en entrée correspondant.

Comme le 68HC11 dispose de trois entrées de ce type, la mesure de largeur d'impulsion, de fréquence, ou d'écart entre événements différents deviennent des applications facilement réalisables. L'une des entrées ICI à IC3 positionne le bit IC1F à IC3F lors d'une transition valide et agit sur l'un des latch qui suivent, latch dont le contenu est lisible par l'intermédiaire des registres TIC1 à TIC3. On voit également que chaque bit IC1F à IC3F peut générer une interruption sous réserve d'y être autorisé par l'un des bits IC11 à IC31.

Chacune de ces interruptions dispose de son propre vecteur, ce qui facilite énormément leur gestion et diminue leur temps de prise en compte.

La deuxième fonction importante de ce timer s'appelle la comparaison en sortie et fait appel à l'un des cinq registres 16 bits dits de comparaison en sortie. Ces registres, dont on peut programmer le contenu à tout instant sont en permanence comparés avec le compteur 16 bits du timer. Dès qu'il y a égalité, un bit d'état est positionné, une interruption peut être générée et la patte de sortie correspondante peut changer d'état de la façon dont vous le désirez. Il est ainsi très facile de faire de la génération de signal de forme et de durée programmable et ce d'autant que le comportement de la patte de sortie pilotée par les registres de comparaison est programmable de manière très souple.

Quatre de ces registres agissent directement sur la patte qui leur est affectée OC2 à OC5 tandis que le cinquième agit sur la patte OC1 qui peut être en fait la patte PA7 du port A mais qui peut aussi être rendue commune avec l'une quelconque des pattes OC2 à OC5.

Les registres 16 bits de comparaison en sortie TOC1 à TOC5 agissent en effet sur les bits d'état OC1F à OC5F qui peuvent déclencher une interruption s'ils y sont autorisés par les

bits OC11 à OC51. Comme pour la fonction capture en entrée, chaque interruption de comparaison en sortie dispose de son propre vecteur.

La sortie de comparaison OC1 bénéficie du fait que son rôle peut être tenu par une quelconque des pattes OC2 à OC5 ou par la patte PA7.

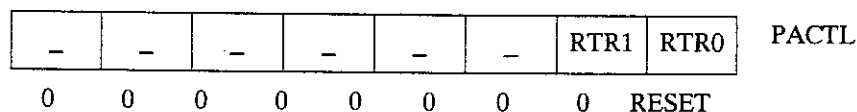
IV.7.2 Générateur d'interruptions temps réel

La circuiterie de génération d'interruptions temps réel du timer, ou RTI pour *Real Time Interrupt*, peut être utilisée pour produire, à des intervalles de temps réguliers et rigoureusement contrôlés, des interruptions.

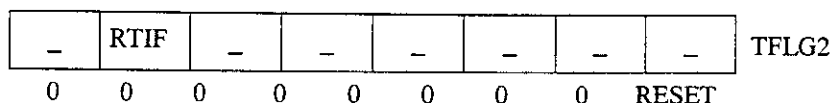
La source d'horloge utilisée par cette fonction est pilotée par la circuiterie générant l'horloge E et ne peut être ralentie ou interrompue pendant le fonctionnement du circuit. On est assuré d'un écart de temps rigoureusement constant entre deux interruptions successives.

Cette circuiterie de génération d'interruptions temps réel fait appel à quelques bits localisés dans trois registres distincts afin de :

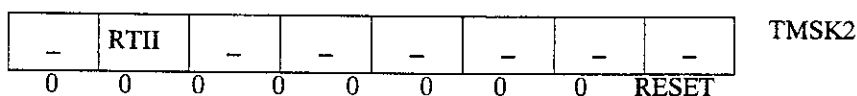
- fixer le rythme de génération des interruptions temps réel. A travers le registre PACTL



déterminer l'état de la RTI par un registre d'état TFLAG



- valider la source d'interruption à travers le registre TMSK2



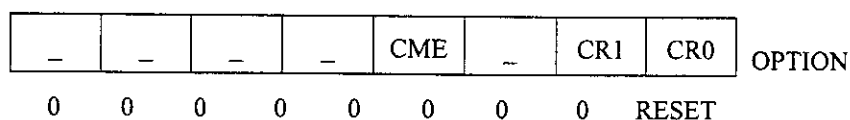
IV.7.3 Le chien de garde ou COP

C'est un compteur qui décompte en permanence et qui doit être réarmé avant qu'il n'atteigne 00. Dans le cas contraire, un Reset de type COP est généré.

Le COP peut être validé ou non par registre CONFIG mais, une fois qu'il est validé, il doit impérativement être géré par le logiciel.

La procédure de réarmement du COP est simple mais cependant assez efficace pour éviter tout réarmement involontaire par un programme fantaisiste. Il faut en effet écrire \$55 dans le registre COPRST puis ensuite venir écrire \$AA dans ce même registre pour que le réarmement ait lieu.

Le registre OPTION fixe l'horloge de décomptage utilisée par le COP.



L'horloge qui pilote le COP ne peut être arrêtée sauf si l'horloge système s'arrête elle-même. Pour cette raison, un mécanisme de surveillance de la fréquence de cette horloge existe dans le 68HC11. Ce mécanisme est contrôlé par le bit CME du registre OPTION. Si ce bit est mis à 1, la surveillance d'horloge est validée alors qu'elle est inactive s'il est à 0.

IV.7.4 L'accumulateur d'impulsions

L'accumulateur d'impulsions est en fait un timer 8 bits relativement simple et comparable à celui qui existait sur les premières versions de 6805 du même constructeur.

Il dispose de deux modes de fonctionnement différents.

- Dans le premier mode, le compteur compte les impulsions appliquées sur son entrée PAI (commune avec PA7).

- Dans le deuxième mode, l'état de PAI sert en fait à ouvrir une porte qui autorise alors le comptage d'un signal d'horloge interne à la fréquence de l'horloge E divisée par 64.

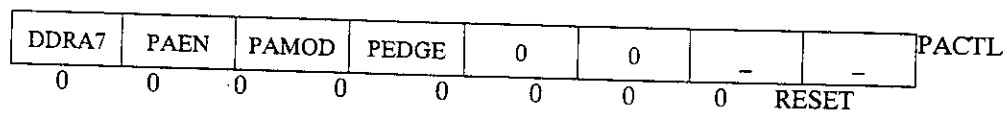
Contrairement au timer général le compteur de l'accumulateur peut être lu ou écrit à tout moment ; ceci permet de faire un comptage à partir de n'importe quelle valeur de notre choix. le registre PACNT rend le compteur accessible à tout instant

Deux sources d'interruptions sont associées à cet accumulateur

- lorsque le compteur déborde de \$FF à \$00.
- lorsque une transition valide est détectée sur PAI.

L'une comme l'autre peut être masquée.

Le registre PACTL programme les modes de fonctionnement du compteur comme suit:



- permettre à PAI de réagir soit à des fronts descendants ou ouvre la porte de comptage à un niveau bas (selon mode de fonctionnement choisi) soit à des fronts montants ou ouvre la porte de comptage pour un niveau haut.
- fixer le mode de fonctionnement de l'accumulateur d'impulsions f.
- valider l'accumulateur
- configurer la patte PA7 soit en entrée soit en sortie et utilisée par OC1 du Timer général ; elle peut être à la fois utilisée par l'accumulateur et OC1 afin de compter les transitions actives de OC1.

IV.8 CONVERTISSEUR ANALOGIQUE NUMERIQUE

Les 68HC11 sont équipés d'un convertisseur analogique/digital précédé d'un multiplexeur à 8 entrées pour la version étudiée.

Sa précision est garantie meilleure que $\frac{1}{2}$ bit le moins significatif sur toute la plage de température fonctionnement du circuit.

IV.8.1 Le procédé de conversion et ses implications

Le convertisseur qui équipe les 68HC11 est du type à approximations successives obtenues par répartition de charge dans des capacités. Ce transfert de charge se fait grâce à des commutateurs analogiques. Pour qu'ils puissent commuter des tensions de 5 volts il est nécessaire de les alimenter avec une tension d'au moins 8 volts. Ceci est rendu possible par la mise en œuvre d'une pompe de charge.

La consommation de courant du convertisseur est très réduite tant sur ses entrées de tensions de référence que sur ses entrées de mesure.

Le fonctionnement du convertisseur est basé sur :

- L'échantillonnage
- Le maintien.
- La conversion qui consiste à comparer avec le comparateur la tension emmagasinée maintenue à une tension de référence.

V_{rh} et V_{rl} sont les entrées de référence qui permettent la décision du convertisseur.

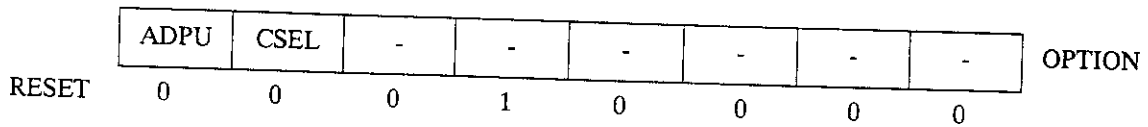
Le processus de transfert de charge utilisé doit se dérouler assez rapidement par le fait qu'il repose sur des charges emmagasinées dans des condensateurs.

L'action sur les différents commutateurs internes est pilotée et synchronisée par l'horloge E lorsque celle-ci est supérieure à 750 Khz car les phénomènes de décharge parasite sont négligeables.

A des fréquences plus basses il est possible d'utiliser une autre source d'horloge. Une horloge à cellule R-C intégrée est prévue sur la puce pour ce faire.

Le registre OPTION permet :

- de sélectionner l'horloge à utiliser.
- de valider le convertisseur.



Le module CAN contient aussi un registre de contrôle propre à lui. C'est le registre ADCTL (*Analog Digital Control*).

Il permet de fixer le comportement du convertisseur :

- Choix de la voie en service
- Mode répétitif ou non
- Mode multiple ou non

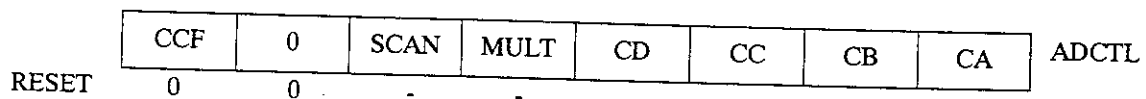
Ainsi on définit deux modes de fonctionnement :

- Quatre conversions successives sur une même entrée et récupération d'un résultat sur quatre bits.
- Une conversion de quatre entrées simultanées.

On discerne à travers le registre précédent si les conversions demandées ont lieu en continu où les registres de résultat s'actualisent régulièrement ou bien en une seule fois.

Le tableau ci-joint nous permet de représenter la sélection des entrées et des registres de résultat au moyen des bits CA à CD du registre ADCTL.

Ce registre contient aussi un bit d'état qui indique l'accomplissement des conversions demandées. En mode conversion continue il suffit de mettre ce bit à 0 pour interrompre une conversion car il reste à 1.



Afin de diminuer le bruit qui peut éventuellement affecter les signaux à convertir, il est courant de leur faire subir un filtrage passe-bas, plus ou moins léger selon leur vitesse de variation supposée.

Les entrées sont protégées contre les tensions excessives et les tensions négatives par rapport à la masse par une diode Zener.

CHAPITRE V

V.1 EMULATEURS, SIMULATEURS ET MAQUETTES DE TEST

La première solution pour valider une application, consiste à utiliser un émulateur.

L'émulateur est en fait un montage particulier, qui peut être très complexe, et qui se comporte exactement comme le microcontrôleur qu'il remplace. Mais c'est une version «éclatée» de ce dernier, c'est à dire, une version dans laquelle tous les constituants internes du microcontrôleur ont été séparés.

Par ailleurs, l'émulateur est relié généralement par une liaison série RS232, au système de développement de façon à pouvoir être téléchargé par le logiciel de l'application.

La deuxième solution reste relativement confortable à l'usage mais peut donner des résultats très variables selon le type d'application à développer. Elle consiste à faire appel à un simulateur. Ce simulateur est un programme, écrit spécialement pour le microcontrôleur qu'il est sensé simuler. On lui fournit en entrée le code objet à exécuter et il se comporte alors comme se comporterait le microcontrôleur qu'il simule.

Comme un microcontrôleur comporte un nombre non négligeable d'entrées-sorties et que cela ne peut être le cas de la machine sur laquelle tourne le simulateur, celui-ci simule les entrées-sorties sous forme de mémoires particulières. Ainsi si l'on dispose d'un port parallèle huit bits, il va être représenté par un octet mémoire particulier. Il suffira donc d'aller le lire pour savoir à tout instant de l'exécution du programme dans quel état sont les lignes de sortie. L'écriture en mémoire dans cet octet, quant à elle, simulera une entrée de données.

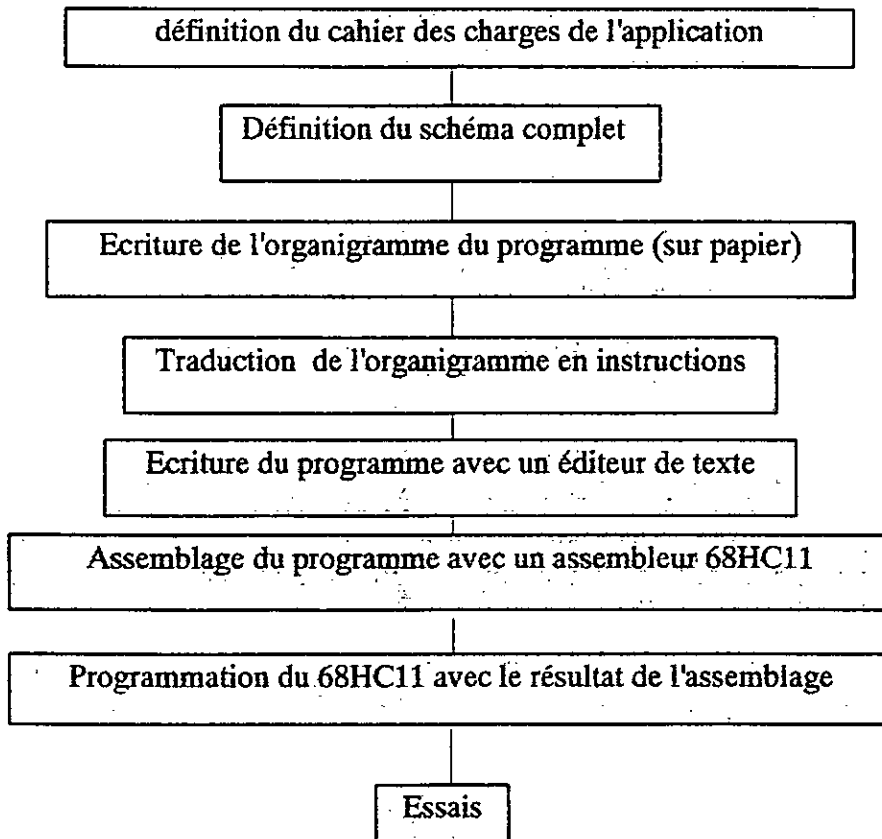
Il est évident que cette phase de simulation des entrées-sorties est un peu longue et lourde mais qu'elle permet, si elle est bien conduite, de vérifier 80% des fonctionnalités d'un programme.

La troisième solution consiste à faire appel à une maquette sur laquelle on va monter une version UVPRM du microcontrôleur. Toutes les erreurs seront ainsi permises grâce à la possibilité d'effacement du circuit.

Une solution intermédiaire existe toutefois si l'application n'est pas très gourmande en entrées-sorties parallèles. Elle consiste à faire la mise au point du programme en utilisant le 68HC11 en mode étendu.

V.2 METHODE GENERALE DE TRAVAIL

Voici l'organigramme qui présente la meilleure façon de travailler pour développer une application à base du 68HC11.



Organigramme Type de développement d'une application

Avant tout il faut bien définir le cahier des charges, à partir de ça on dessine le schéma complet. Ensuite, il faut écrire les organigrammes du logiciel. Une fois que c'est fait, ceux-ci sont traduits en instructions qui sont à leur tour traduites en code machine par le cross-assembleur. Ces codes sont alors programmés dans le 68HC11.

V.3 REGLES D'ECRITURE DES PROGRAMMES

V.3.1 Les Modes d'Adressage :

Le 68HC11 dispose de six modes d'adressage différents. Nous avons conservé les appellations adoptées par Motorola ; dans un souci de cohérence avec les documentations techniques de ce fabricant.

L'espace adressable :

La structure d'entrée sortie par instruction mémoire est adoptée à l'adressage des mémoires et périphériques.

Les registres des ressources internes sont donc considérés comme des emplacements mémoires et sont accessibles par leur adresse propre.

Dans ce qui suit nous utiliserons la notion d'adresse effective ; c'est l'adresse réelle de la donnée manipulée par l'instruction, ou alors l'adresse physique.

Nous utiliserons aussi les notations qui respectent les conventions suivantes :

- l'absence de tout symbole signifie qu'il est en décimal
- le symbole (\$) signifie que le nombre est hexadécimal
- le symbole (@) signifie que le nombre est octal
- le symbole (%) signifie que le nombre est binaire
- le symbole (') signifie que le caractère doit se faire remplacer par son code ASCII.
- le symbole (#) signifie que c'est un adressage immédiat.

V.3.1.1 L'adressage inhérent :

Dans ce mode il n'y a pas d'adresse effective, l'instruction concernée se suffit à elle-même et n'a besoin d'accéder à aucune donnée externe.

Exemple :

ABA : instruction qui ajoute A à B et place le résultat dans B.

V.3.1.2 L'adressage immédiat :

Dans ce mode l'instruction est immédiatement suivie par la donnée à manipuler ; qui est précédée par le symbole dièse(#).

La taille de la donnée dépend du registre concerné par l'instruction.

Exemple :

LDX #\$1234

V.3.1.3 L'adressage direct :

L'instruction est suivie d'un octet qui est l'adresse effective de la donnée manipulée. Ce mode d'adressage ne peut fonctionner que de l'adresse 00 à l'adresse \$FF ; sur les 256 premiers octets de l'espace adressable du microcontrôleur.

Selon que le programme réalise de nombreux accès à la RAM ou aux registres; il serait intéressant de déplacer l'adresse de base de l'un ou l'autre de façon à ce qu'elle soit à \$0000 autorisant l'utilisation intensive de ce mode d'adressage.

Exemple :

LDAA \$10

V.3.1.4 L'adressage étendu :

Ce mode permet d'atteindre n'importe quelle adresse mémoire ; de l'adresse \$0000 à l'adresse \$FFFF ; car il autorise un codage de l'adresse effective sur deux octets soit 16 bits.

Exemple :

```
LDAA $10000
```

V.3.1.5 L'adressage indexé :

Ce mode d'adressage est particulièrement souple et puissant ; il fait intervenir deux facteurs dans le calcul l'adresse effective :

- le contenu d'un des registres d'index X ou Y
- un offset ou déplacement.

Il existe de ce fait deux sous modes :

➤ Adressage indexé avec déplacement nul :

Le contenu de l'index X ou Y est l'adresse effective puisqu'il n'y a pas de déplacement à prendre en compte.

Exemple :

```
LDD 0, X
```

➤ Adressage indexé avec déplacement non nul :

Dans ce mode, l'adresse effective est obtenue en ajoutant le contenu de l'index X ou Y au déplacement qui est toujours un octet non signé suivant le code instruction.

Exemple :

```
    LDD 4, X
    ADDD $25, X
DATA EQU $FE
    LDY DATA, X
```

V.3.1.6 L'adressage relatif :

Ce mode d'adressage est utilisé pour les instructions de saut et de branchement. Il spécifie l'adresse effective sous forme d'un déplacement qui est donc ajouté à la valeur courante du PC, pour spécifier l'adresse de poursuite du programme.

Adresse effective = valeur courante du PC + deux + déplacement.

Le déplacement est codé sur un octet, mais c'est une valeur signée comprise entre -128 et 127 en décimal, ce qui autorise une amplitude globale de saut de 255 octets. En effet ces considérations seront prises en compte par l'assembleur car le programmeur indique dans l'instruction l'étiquette de destination.

V.3.2 L'adressage «bit » ou «manipulation de bit » :

Ce mode d'adressage est extrêmement intéressant dans le cadre d'un microcontrôleur qui fait très souvent des accès à des ressources internes, et qui est amené à manipuler des bits individuels dans des registres.

Il comporte deux sous-modes qui sont les suivants :

V.3.2.1 Mode positionnement de bit :

L'instruction est présentée comme suit :

INSTRUCTION OPERANDE MASQUE

Ou MASQUE indique le ou les bits de OPERANDE à manipuler par mise à 1 des bits correspondants.

La définition de l'adresse de OPERANDE peut être faite en adressage direct ou indexé tandis que le masque est toujours considéré comme codé en immédiat sans que le symbole dièse ne soit utilisé.

Exemple :

BCLR PORTB %00110001 : positionnement à zéro les bits 0,4,5 du registre PORTB

V3.2.2 Mode test de bit et branchement :

Ce mode est une évolution du précédent et confère au 68HC11 une souplesse de test de bits individuels. Il permet 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 conséquence ou du saut à une adresse déterminée.

L'instruction se présente donc ainsi :

INSTRUCTION OPERANDE MASQUE SAUT

Ou MASQUE indique le ou les bits de OPERANDE à tester par mise à 1 des bits correspondants

SAUT indique le déplacement permettant de calculer l'endroit où doit se poursuivre le programme selon le résultat du test.

L'adresse de OPERANDE peut être faite en adressage direct ou indexé, tandis que le masque est codé en immédiat.

Exemple :

BRCLR TFLG1 \$80 BOUCLE : Cette instruction teste le bit 7 du registre TFLG1 si ce bit est nul il cause un branchement à l'étiquette BOUCLE.

Maintenant que nous avons vu les modes d'adressage, voyons quelles sont les règles de syntaxe de l'assembleur pour 68HC11 utilisé étant entendu que la majorité d'entre elles est commune à tous les assembleurs compatibles Motorola.

Pour que l'assembleur puisse travailler correctement, il faut le fichier texte qui lui est fourni ou, si vous préférez, ce que vous avez frappé au clavier, respecte certaines règles de syntaxe bien définies. En effet, c'est grâce à ces règles que l'assembleur va savoir s'il est en présence de telle ou telle instruction ou de tel ou de tel mode d'adressage ou bien encore si le chiffre qu'on lui donne est décimal, en hexadécimal ou en binaire. Nous allons donc détailler ces dernières ci-après.

Toute ligne qui commence avec une étoile (*) en première colonne est considérée comme une ligne de commentaire. Tout ce qu'elle contient est donc ignoré par l'assembleur qui se borne à le reproduire tel quel sur le listing produit en fin d'assemblage.

Si celle-ci commence par un ou plusieurs espaces, l'assembleur s'attend à trouver ensuite un mnémonique d'instruction ou une directive d'assemblage.

Si la ligne à interpréter commence par un caractère alphanumérique, un point ou le caractère souligné (_) l'assembleur considère qu'il est présence d'une étiquette, il s'attend ensuite à trouver après celle-ci un mnémonique ou une directive d'assemblage.

Dans le cas de l'assembleur retenu ici à titre d'exemple, les étiquettes peuvent comporter jusqu'à quinze caractères mais, si vous écrivez des programmes de taille importante, nous vous recommandons de ne pas dépasser des noms d'étiquettes à huit caractères. Dans ces conditions vous disposez alors d'un nombre maximum de 2000 étiquettes différentes par programme.

Bien sûr, la souplesse permise par les étiquettes impose de respecter quelques contraintes. Il faut toujours considérer une étiquette comme unique. Si l'une d'elle définit une adresse, la même étiquette ne pourra pas être utilisée ensuite dans le même programme pour définir une adresse différente.

Le rôle des étiquettes ne se borne pas à remplacer des adresses lors de sauts. Elles peuvent en fait remplacer n'importe quelle valeur numérique utilisée dans un programme.

Autre avantage des étiquettes, cela permet d'écrire des programmes beaucoup plus lisibles ou « parlants » si vous préférez. En effet, dans le cas du 68HC11 il est préférable de lire LDAA PORTB lorsque l'on change A avec le contenu du port B plutôt que LDAA XXXX qui impose de toujours se souvenir que l'adresse XXXX est en fait celle du port B.

Précisons enfin que, comme quasiment tous les assembleurs, celui proposé supporte une étiquette particulière qui est l'astérisque (*). Sa valeur est égale à la valeur courante du PC lorsque cette étiquette est rencontrée. Ainsi par exemple :

BRA*

est une boucle sans fin car on ordonne au programme de se brancher sur l'étoile qui est justement l'adresse de l'instruction BRA.

L'assembleur proposé par Motorola fait la distinction entre majuscules et minuscules au niveau des étiquettes. Ainsi PORTB et PortB sont considérées comme des étiquettes différentes.

V.3.3 Evaluation des expressions :

La majorité des assembleurs sait réaliser un certain nombre d'opérations arithmétiques et logiques sur les étiquettes. Même si cela vous surprend à première vue, n'oubliez pas qu'une étiquette représente en fait une valeur numérique et qu'il est donc tout à fait normal de pouvoir lui faire subir des opérations. Dans le cas de l'assembleur proposé, ces dernières sont les suivantes :

- L'addition notée + ;
- La soustraction notée - ;
- La multiplication notée * ;
- La division entière notée / ;
- Le reste après division entière noté % ;
- Le ET logique noté & ;
- Le OU logique noté | ;
- Le OU EXCLUSIF logique noté ^.

Les opérations arithmétiques sont effectuées sur les étiquettes considérées comme des nombres entiers signés de 16 bits. Les opérations logiques sont effectuées bit à bit sur la taille réelle des nombres sans que celle-ci ne puisse excéder 16 bits.

V.3.4 Directives d'assemblage

Afin que l'assembleur puisse effectuer un travail aussi efficace que possible, il comprend, outre les mnémoniques des instructions, un certain nombre d'instructions particulières appelées pseudo-instructions ou encore directives d'assemblage. Nous allons voir leurs noms et fonctions.

- La directive EQU permet d'affecter une valeur permanente à une étiquette. Elle s'utilise sous la forme :

Etiquette EQU Expression

- La directive DB ou FCB permet de réserver des positions mémoire et d'y placer des valeurs définies par vos soins ; elle s'utilise sous les formes :

Etiquette DB Expression ou

Etiquette DB Expression 1, Expression 2, ..., Expression N

ou encore avec la même syntaxe mais sans être précédée d'une étiquette si celle-ci n'est pas nécessaire.

Les valeurs d'expressions utilisées sont considérées comme des octets.

- * c : qui permet l'affichage sur le listing du nombre de cycles machine utilisés par chaque instruction ;
- * cre : qui fait imprimer en fin de listing une table de références croisées des étiquettes. Cette option doit impérativement être spécifiée au moyen de la directive OPT avant l'apparition de la première étiquette du programme ;
- * | : qui fait imprimer le listing à partir de son point d'apparition ;
- * noc : (option par défaut) qui interdit l'affichage du nombre de cycles de chaque instruction ;
- * nol : (option par défaut) qui interdit l'impression du listing à partir de son point d'apparition ;
- * s : qui fait imprimer la table des étiquettes à la fin du listing.

V.4 Utilisation de l'assembleur

L'assembleur est constitué d'un fichier appelé AS11.EXE. Il marche sous environnement DOS.

la syntaxe d'appel est la suivante :

```
AS11 FICHER.XXX – OPTION1 OPTION2...
```

Où FICHER.XXX est le nom du fichier texte qui contient le programme à assembler. Ce nom peut être affecté d'un suffixe quelconque,

OPTION1, OPTION2, etc., sont les options décrites ci-après qui peuvent être utilisées seules ou combinées. Dans ce dernier cas, elles se suivent dans n'importe quel ordre séparées les unes des autres par des espaces.

L'assembleur ne produit qu'un fichier ayant le même nom que le fichier à assembler et l'extension S19. Ce fichier est au format S1 – S9.

Les messages d'erreur, sont affichés à l'écran au fur et à mesure de l'assemblage. Il est possible de les faire afficher dans un fichier pour impression ou examen ultérieur en utilisant la commande standard de 'redirection' de sortie de DOS qui est >. Ainsi :

```
AS11          demo.txt          >          erreurs.txt
```

assemble le programme demo.txt et crée le fichier erreurs.txt contenant la liste des messages d'erreurs éventuels.

V.4.1 Le format S1-S9

Ce format, très ancien puisqu'il date de l'époque du premier microprocesseur 8 bits de Motorola qui fut le 6800, connaît aujourd'hui diverses variantes puisqu'il est utilisable même avec les circuits 32 bits. Dans le cas de l'assembleur utilisé ici seuls les deux formats de base sont utilisés : S1 et S9.

Les caractères utilisés sont codés en ASCII sur 7 bits et sont donc compatibles avec tous les équipements informatiques actuels. Les chiffres sont exprimés en hexadécimal et le format utilisé est le suivant :

S NNABCDAAABCCDDEEFFGG... OOPPCK ou

- S1 est l'en-tête d'une ligne de données ;
- NN indique le nombre total d'octets qui suit ;
- ABCD est une adresse codée sur deux octets (AB et CD) ; c'est l'adresse de stockage en mémoire du premier octet de donnée qui suit ;
- AA est le premier octet de données, qui devra donc être rangé à l'adresse ABCD ;
- BB est l'octet de données suivant (qui sera en rangé en ABCD+1) ;
- Et ainsi de suite jusqu'à l'octet de donnée PP dans cet exemple ;
- CK est une somme de contrôle ou "*checksum*", codée sur un octet qui est tout simplement la somme bit à bit tronquée sur 8 bits des données qui précèdent.

La ligne précédée de S9 est quant à elle codée de la façon suivante :

S9NNABCDCK ou

- NN indique le nombre total qui suivent ;
- ABCD est une adresse optionnelle codée sur deux octets (AB et CD) ; c'est l'adresse à laquelle doit être passé le contrôle en fin de chargement du fichier
- CK est la somme de contrôle identique à celle vue ci avant.

CHAPITRE VI

INTRODUCTION

Le principe de modulation sinus-triangle suggère une implantation analogique faisant appel à un générateur de signaux triangulaire et un générateur de signaux sinusoïdaux, ces générateurs étant associés à une bascule pour détecter les instants où se produisent les intersections de la porteuse et de la référence.

Lorsqu'on utilise un système à microprocesseur pour assurer la modulation, on peut commander les semi-conducteurs de l'onduleur à partir de séquences préalablement calculées et stockées dans une mémoire table LUT (Look up Table), soit générer en temps réel les angles de commutations calculés par des formules approximatives qu'on peut implémenter. Dans ce chapitre on s'est intéressé à la technique *d'élimination sélective d'harmoniques* (SHE) en particulier à la ASHE afin d'asservir la valeur du fondamental.

VI.1 PPWM avec élimination sélective d'harmoniques (SHE) : [SUN.96]

Cette technique consiste à choisir les N angles α_i ($i=1..N$) de telle façon à éliminer N harmoniques indésirables. Une autre approche suggère l'élimination de $(N-1)$ harmoniques, la $N^{\text{ième}}$ variable restante étant laissée pour asservir le fondamental à une valeur désirée. La première méthode est dite PSHE avec élimination sélective pure, La seconde est la ASHE avec élimination sélective d'harmoniques et asservissement du fondamental.

Le but de la SHE est d'éliminer un nombre déterminé d'harmoniques indésirables. Cette technique nécessite une étude numérique avant la réalisation, ceci limite son application du fait que le calcul des angles exacts de commutations ne peut se faire en temps réel (on-line) par microcontrôleurs faute des performances de ce dernier, par contre on peut les stocker dans des mémoires (LUT), ou trouver un algorithme simple qui approxime le model de la solution pour générer les angles en temps réel avec un microcontrôleur.

La méthode de Newton-Raphson est appliquée pour la résolution du système d'équations.

• **EXPRESSION DES TENSIONS :**

En triphasé lorsque la charge est équilibrée (i.e somme des courants de chaque phase est nulle), les harmoniques multiples de trois sont inopérants.

On tirera avantage d'un choix convenable des symétries dans l'onde résultante, figure(VI.1)

Les tensions de phase sont déphasées de 120° i.e :

$$V_{bo}(\omega t) = V_{ao}(\omega t - 2\pi/3)$$

$$V_{co}(\omega t) = V_{ao}(\omega t + 2\pi/3)$$

En plus nous prendrons une symétrie de glissement

$$V(\omega t) = -V(\omega t + \pi)$$

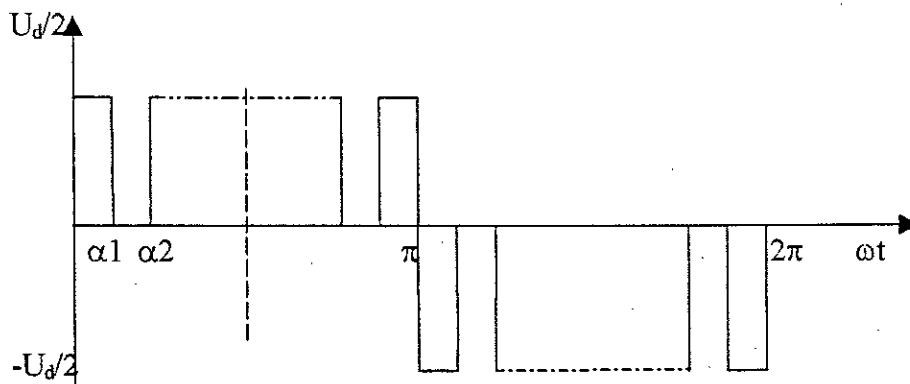


Figure VI.1 : Forme d'onde de sortie

Comme nous sommes en régime périodique nous aborderons le problème par une décomposition de la tension de sortie en série de Fourier :

$$v(\omega t) = \sum_{n=1}^{\infty} [a_n \sin(n\omega t) + b_n \cos(n\omega t)] \frac{U_d}{2}$$

avec
$$a_n = \frac{1}{\pi} \int_0^{2\pi} v(\omega t) \sin(n\omega t) d(\omega t)$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} v(\omega t) \cos(n\omega t) d(\omega t)$$

comme $\alpha_0 < \alpha_1 < \dots < \alpha_{2N+1}$ avec $\alpha_0 = 0$ et $\alpha_{2N+1} = \pi$, on obtient :

$$a_n = \frac{2}{\pi} \sum_{k=0}^{2N} (-1)^k \int_{\alpha_k}^{\alpha_{k+1}} \sin(n\omega t) d(\omega t)$$

Vu la symétrie par rapport à π (symétrique la demi-onde), le développement ne contient pas d'harmoniques pairs.

$$v(\omega t) = -v(\pi - \omega t)$$

Si en plus nous choisissons une symétrie par rapport à $\pi/2$ (fonction impaire), i.e :

les termes en cosinus (b_n) sont nuls.

Si les aires des alternances positives et négatives sont égales, la valeur moyenne (continue) est nulle.

Finalement :

$$a_n = \frac{4}{n\pi} \left[1 + 2 \sum_{k=1}^N (-1)^k \cos(n\alpha_k) \right]$$

En particulier pour le fondamental ($n = 1$)

$$a_1 = M = \frac{4}{\pi} \left[1 + 2 \sum_{k=1}^N (-1)^k \cos(\alpha_k) \right]$$

$$-\frac{4}{\pi} \leq M \leq \frac{4}{\pi}$$

M est appelé indice de modulation.

Dans la SHE pure, on égale alors à zéro les N harmoniques qui viennent juste après le fondamental, tandis que dans la SHE asservie, on élimine $(N-1)$ harmoniques indésirables et on donne une valeur au fondamental. On obtient donc un système d'équations non-linéaires en fonctions trigonométriques qu'on résout par une méthode numérique. La solution n'est pas évidente et devrait satisfaire à la condition $\alpha_1 < \alpha_2 < \dots < \alpha_N$. Quant à la convergence, elle dépend surtout du choix des conditions initiales. Les harmoniques restant de rang assez élevé sont facilement filtrés.

VI.1.1 Elimination sélective d'harmonique pure PSHE :

Dans ce cas les expressions du système sont données par :

$$\alpha_i = \frac{4}{\pi m_i} \left[1 + 2 \sum_{k=1}^N (-1)^k \cos(n_i \alpha_j) \right] = 0 \quad (\text{VI.1})$$

$$\text{donc} \quad 1 + 2 \sum_{k=1}^N (-1)^k \cos(n_i \alpha_j) = 0 \quad (\text{VI.2})$$

La solution exacte devrait satisfaire à la condition :

$$0 < \alpha_1 < \alpha_2 < \dots < \alpha_N < 90^\circ$$

La résolution du système est faite par la méthode de NEWTON-RAPHSON. [BOU.00]

La meilleure solution pour implémenter cette méthode est d'utiliser une mémoire table (LUT) parce que les trajectoires des angles suivent des fonctions de puissance, sachant que le microcontrôleur ne permet que les opérations élémentaires, donc on doit approximer les angles par des fonctions polynomiales, avec le minimum de coefficients, après le calcul des fonctions polynomiales il est apparu que ces dernière présentent une grande erreur, ce qui n'est pas intéressant.

VI.1.2 Elimination Sélective d'Harmoniques avec Asservissement de fondamental ASHE :

Dans ce cas on élimine (N-1) harmoniques indésirables et on asservis le fondamental à une valeur désirée. Le système non-linéaire devient

$$h_1 = c^{1e}$$

$$h_i = 0 \quad , \quad i = 2, \dots, N$$

Pour retrouver le système (VI.1) il suffit de mettre :

$$f_1 = h_1 - C^{1e} = 0$$

$$f_i = h_i = 0 \quad , \quad i = 2, \dots, N$$

VI.1.2.1 Cas Des Systèmes D'ordre Pair : [BOU.00]

Les angles exacts sont obtenus en résolvant le système (IV.2) pour diverses valeurs du fondamental et plusieurs ordres.

Les figures suivantes montrent la variation des angles en fonction de l'indice de modulation pour les ordres N=4,6,8,10.

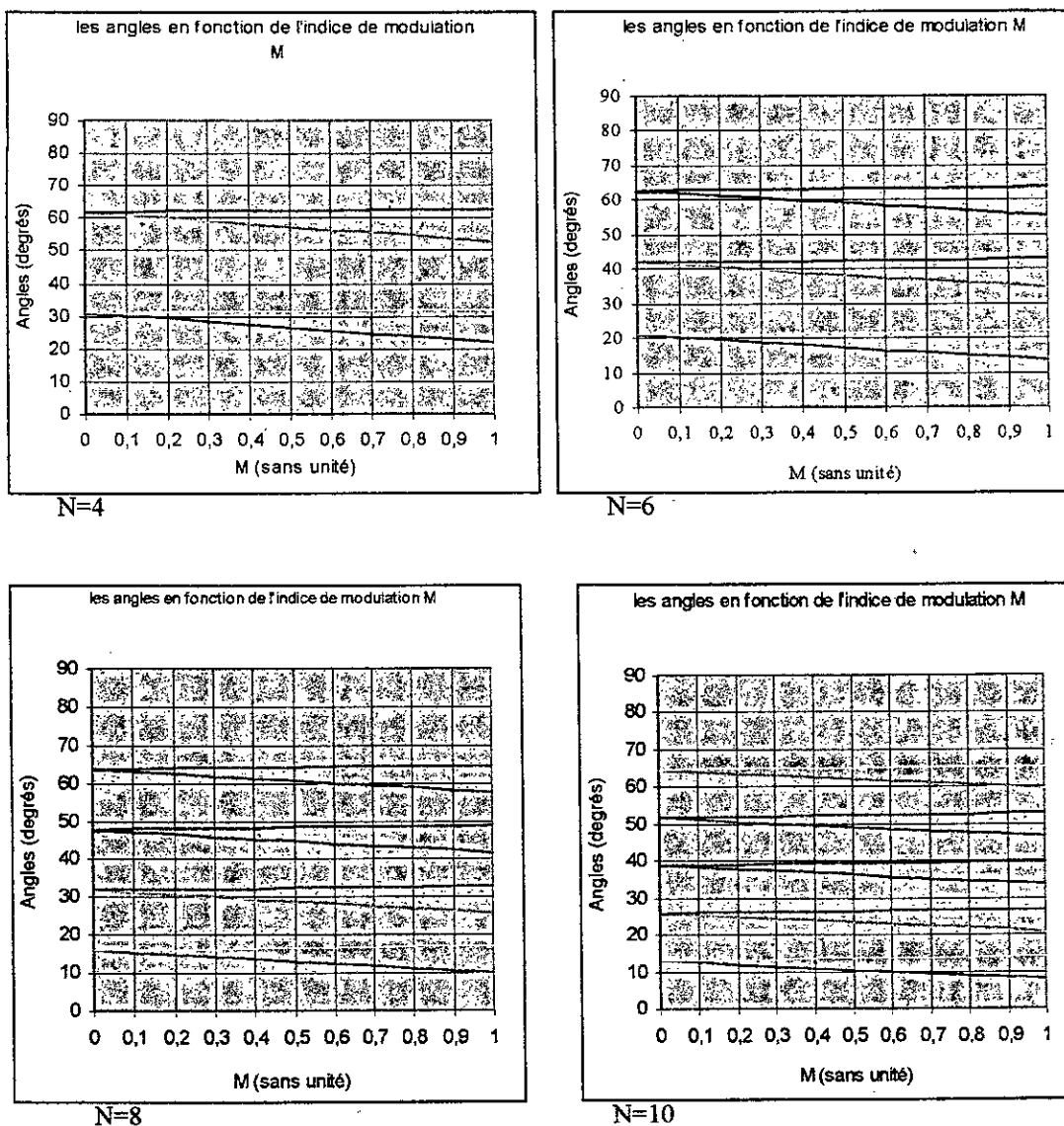


Figure VI.2 : trajectoire des angles en fonction de M pour N pair. (N=4 ; N=6 ; N=8 ; N=10)

Constatations :

- Les angles suivent des trajectoires qui peuvent très bien être approximées à des droites (pentes constantes) sauf pour des valeurs élevées du fondamental (M>0.9).
- Les droites correspondantes aux angles avec indices de même parités sont parallèles entre elles (mêmes pentes), et sont séparés de :

$$\Delta = \frac{2 \times 60}{N} \text{ degrés}$$

Donc, il suffit de connaître α_1 et α_2 pour déduire $\alpha_3, \dots, \alpha_N$.

les pentes sont données par les formules suivantes : [BOU.00]

$$P1 = 4.34784 * \log(N) - 15.0657, \quad N=2,4,6,\dots$$

$$P2 = 1, \quad N=2,4,6,\dots$$

L'étape suivante est d'approximer les angles en fonction des pentes et de l'indice de modulation, sachant que les droites correspondantes aux angles avec indices de même parités sont parallèles entre elles, et sont séparés de Δ .

Les expressions approximatives sont :

$$\alpha_1 = P1 * M + \Delta$$

$$\alpha_2 = P2 * M + \Delta$$

Pour les angles suivants, il suffit de les décaler de Δ :

$$\alpha_i = \alpha_{i-2} + \Delta \quad i=3,5,7,\dots,N.$$

$$\alpha_{i+1} = \alpha_{i-1} + \Delta$$

Afin d'implémenter cette méthode on a cherché une fonction plus facile à programmer en assembleur c'est à dire une fonction polynomiale, cette fonction est calculée par la méthode des moindres carrés.

La nouvelle pente est donnée par :

$$P1 = -0.025 * N^2 + N - 12.6$$

La figure ci-dessous montre l'erreur sur P1 pour différentes valeurs de N.

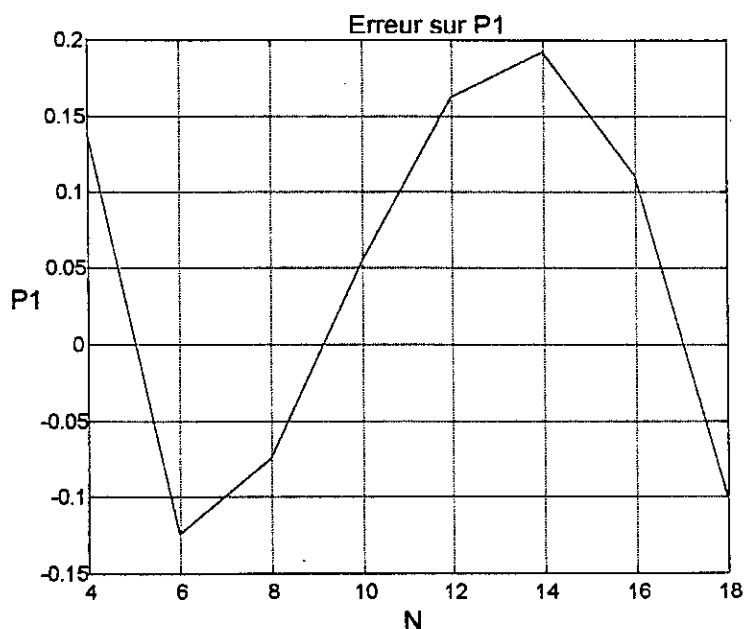


Figure VI.3 : Erreur sur P1

➤ **Organigramme du programme assembleur**

Ce programme assembleur va générer le signal PWM en calculant les différences d'angle et les cycles correspondant pour chaque fréquences, indice de modulation M et le nombre de commutation.

- Différence d'angle

$$P = -P1 = N^2 - 20 * N + 126$$

$$P2 = 1$$

$$D = 120 / N$$

$$dif1 = D + P * M$$

$$dif2 = M + P * M$$

$$dif3 = D - dif2$$

$$DERA = 30 - M$$

- Calcul des différences d'angles en cycles d'horloge

$$M = \frac{80 * M}{freq}$$

$$PM = \frac{8 * P * M}{freq}$$

$$DERA = \frac{23810}{freq} - M$$

DEB: adresse de début des différences d'angles.

freq : fréquence de travail.

DERA: dernier angle.

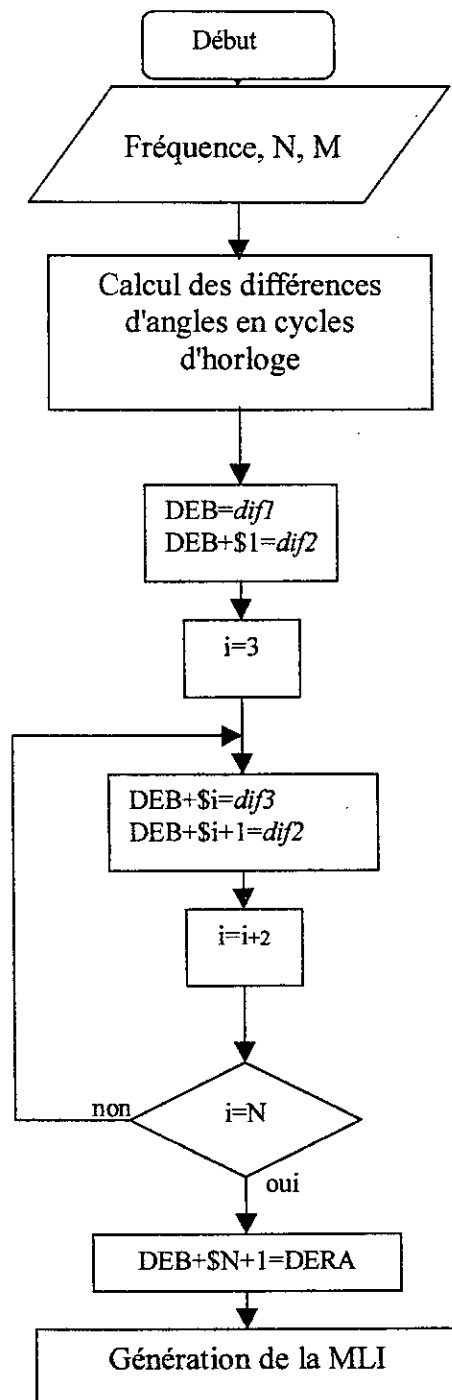
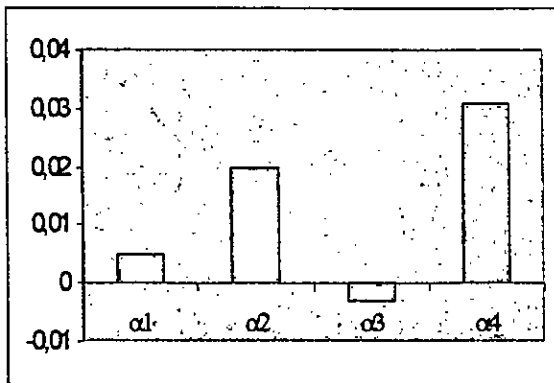
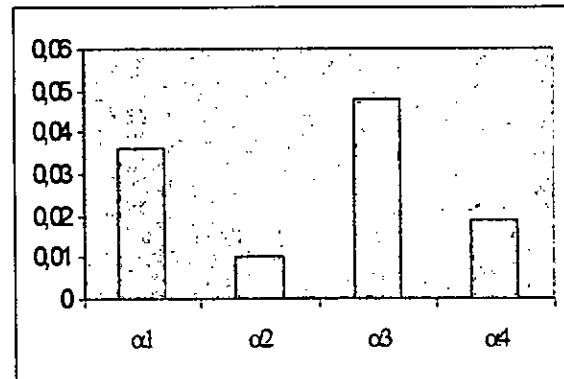


Figure VI. 4 : Organigramme de génération de la MLI cas pair

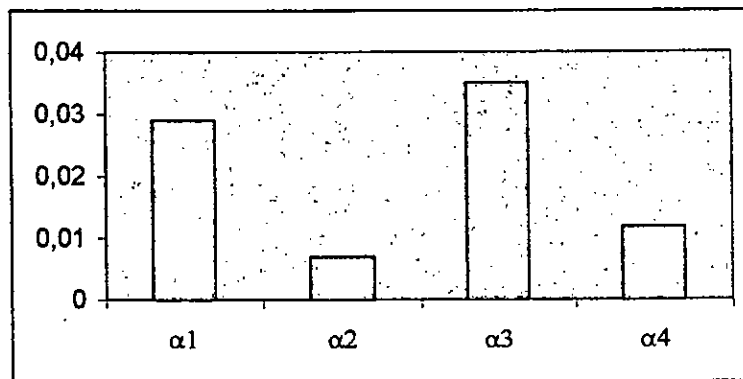
- Le temps de calcul des angles est de l'ordre des centaines de micro seconde, ce qui nous permis de dire que la génération des angles et en temps réel.
- Les figures ci-dessous montre la différence entre les angles calculé par notre programme assembleur et les angles théoriques pour différentes valeurs de fréquences et de N.



a) frequency=50hz



b) frequency=35hz



c) frequency=15hz

Figure VI.5 : Erreur sur les angles entre les valeurs théoriques et les valeurs calculées par le MC pour N=4.

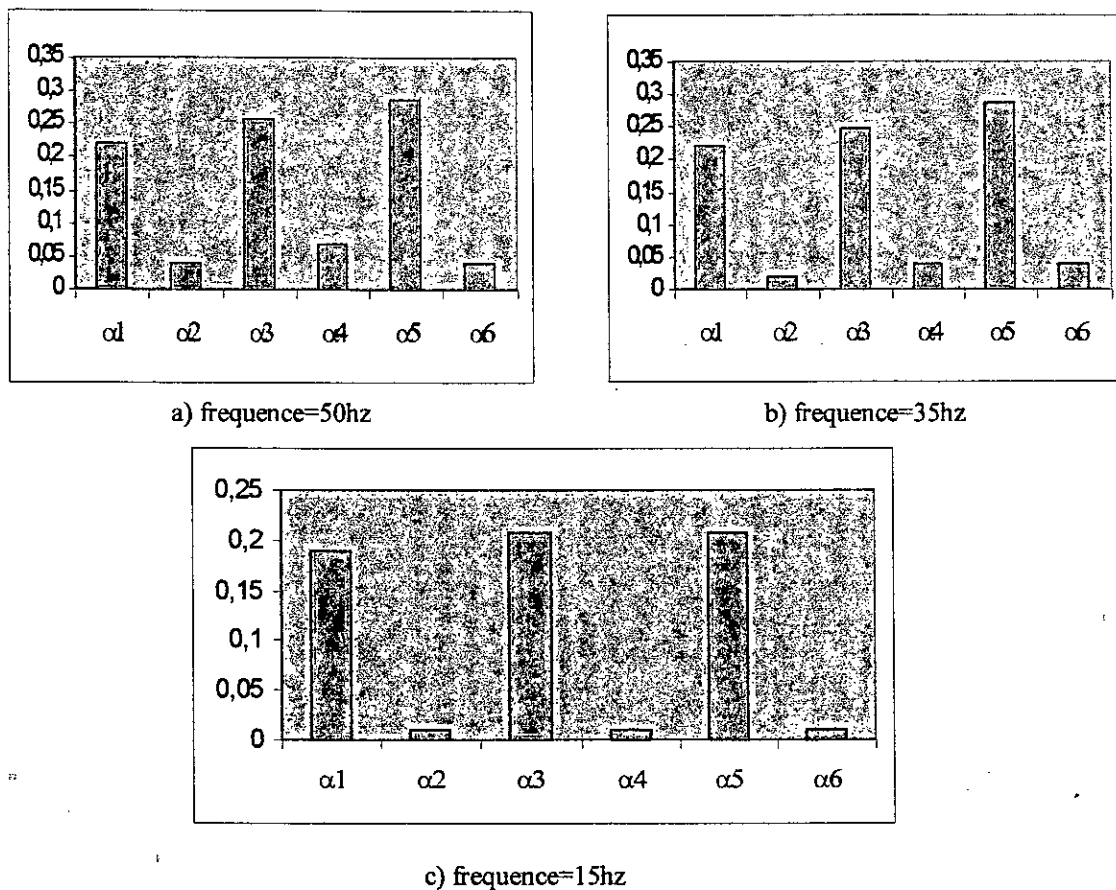


Figure VI.6 : Erreur sur les angles entre les valeurs théoriques et les valeurs calculées par le MC pour N=6.

VI.1.2.1 Cas Des Systèmes D'ordre impair : [BOU.00]

Dans ce cas, on a fait varier l'indice de modulation M de 0 à $4/\pi$, on a utilisé l'algorithme de Enjeti (ordre impair).

Chaque angle varie selon une droite, ceci nous a aidé à trouver des fonctions approximatives plus simples, la figure (VI.7) montre les trajectoires des angles en fonction de la valeur de l'indice de modulation M pour différents ordres

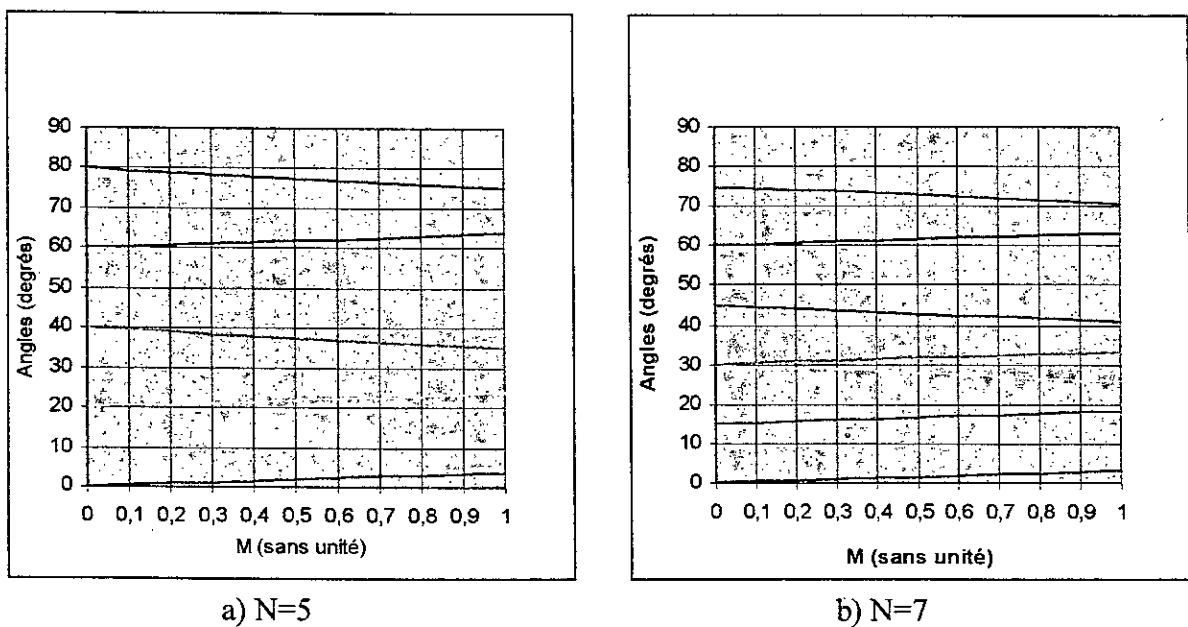


Figure VI.7 : Trajectoire des angles α , en fonction de l'indice de modulation M (ASHE) ; N impair.(a- $N=5$; b- $N=7$)

Les pentes sont données par les formules suivantes : **[BOU.00]**

$$P1=5.0391 e^{-0.07125N}$$

$$P2=-6.4384 e^{-0.05672N}$$

Pour les même raisons que pour le cas pair on a essayé de trouver des fonctions polynomiales du deuxième ordre (minimum de coefficients) qui approxime au mieux ces deux pentes avec un minimum d'erreur pour les implémenter sur le MC.

les nouvelles pentes sont données ci-dessous.

$$P1=0.025 N^2 -0.65 N +5.5$$

$$P2=-0.025 N^2 +7 N -6.7$$

La figure ci-dessous montre l'erreur sur P1 et P2 pour différentes valeurs de N.

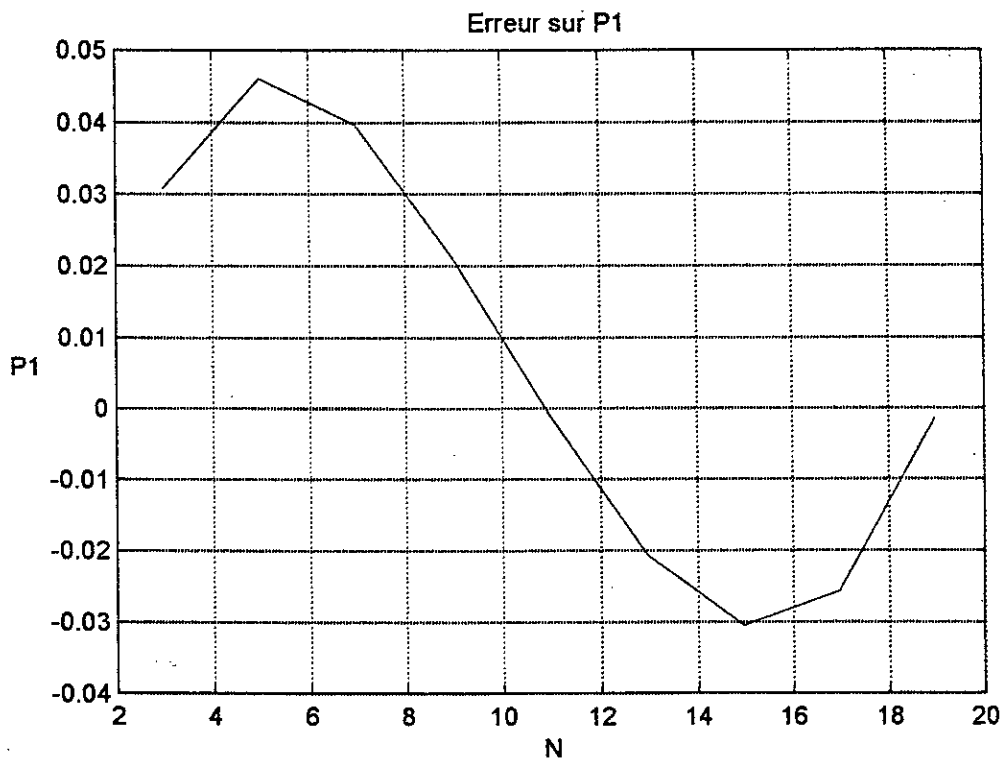


Figure VI.8 : Erreur sur P1

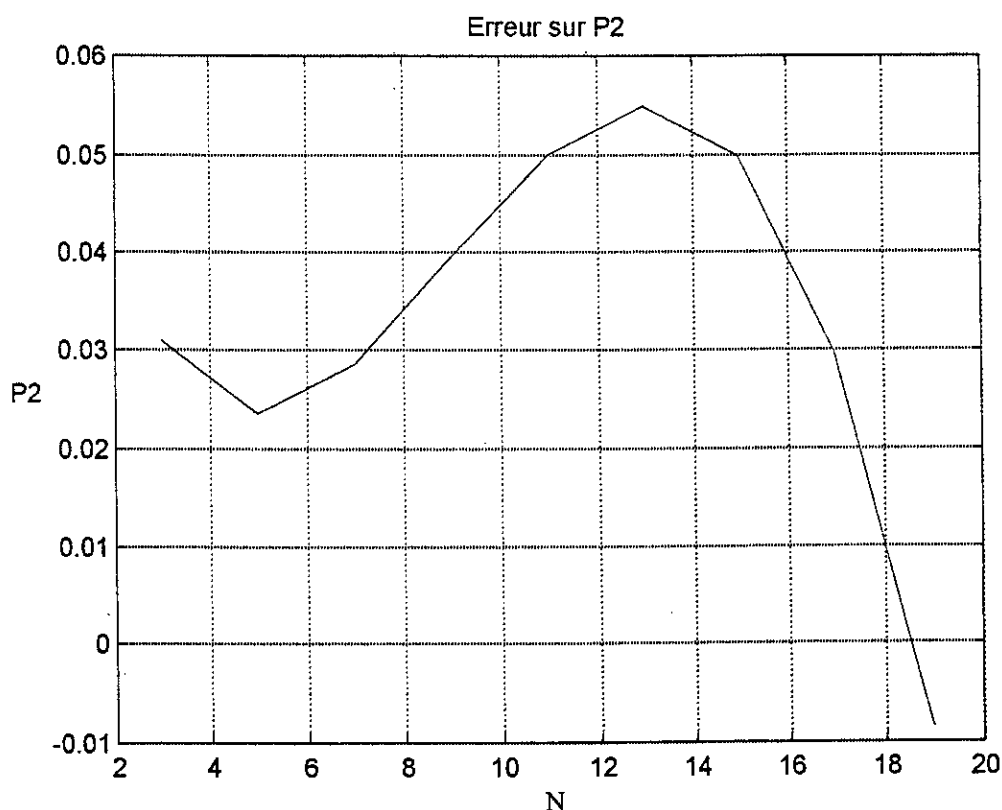


Figure VI.9 : Erreur sur P2

L'étape suivante est d'approximer les angles en fonction des pentes et de l'indice de modulation, sachant que les droites correspondantes aux angles avec indices de même parités sont parallèles entre elles, et sont séparés de :

$$\Delta = \frac{2 \times 60}{N+1} \text{ degrés}$$

Les expressions approximatives sont :

$$\alpha_1 = P1 * M$$

$$\alpha_2 = P1 * M + \Delta$$

$$\alpha_3 = P2 * M + 2 * \Delta$$

Pour les angles suivants, il suffit de les décaler de Δ :

$$\alpha_i = \alpha_{i-2} + \Delta \quad i=4,6,8,\dots,N-2$$

$$\alpha_{i+1} = \alpha_{i-1} + \Delta$$

$$\alpha_{N-1} = \alpha_{N-3} + 2 * \Delta$$

$$\alpha_N = \alpha_{N-2} + 2 * \Delta$$

➤ **Organigramme du programme assembleur**

Ce programme assembleur va générer le signal PWM en calculant les différences d'angle et les cycles correspondant pour chaque fréquences, indice de modulation M et le nombre de commutation.

- Différence d'angle

$$P1=0.25 N^2-6.5*N+55$$

$$P=-P2=0.25 N^2 -7 N+67$$

$$D=120/(N+1),$$

$$dif1=P1*M$$

$$dif2=D$$

$$dif3=D-(P1+P)*M$$

$$dif4=(P1+P)*M$$

$$dif5=D+(P1+P)*M$$

$$DERA=90-\sum dif_i$$

- Calcul des différences d'angles en cycles d'horloge

$$M = \frac{80 * M}{freq}$$

$$P1,2M = \frac{8 * P * M}{freq}$$

$$DERA = \frac{11705}{freq} - \sum dif_i$$

DEB: adresse de début des différences d'angles.

freq : fréquence de travail.

DERA: dernier angle.

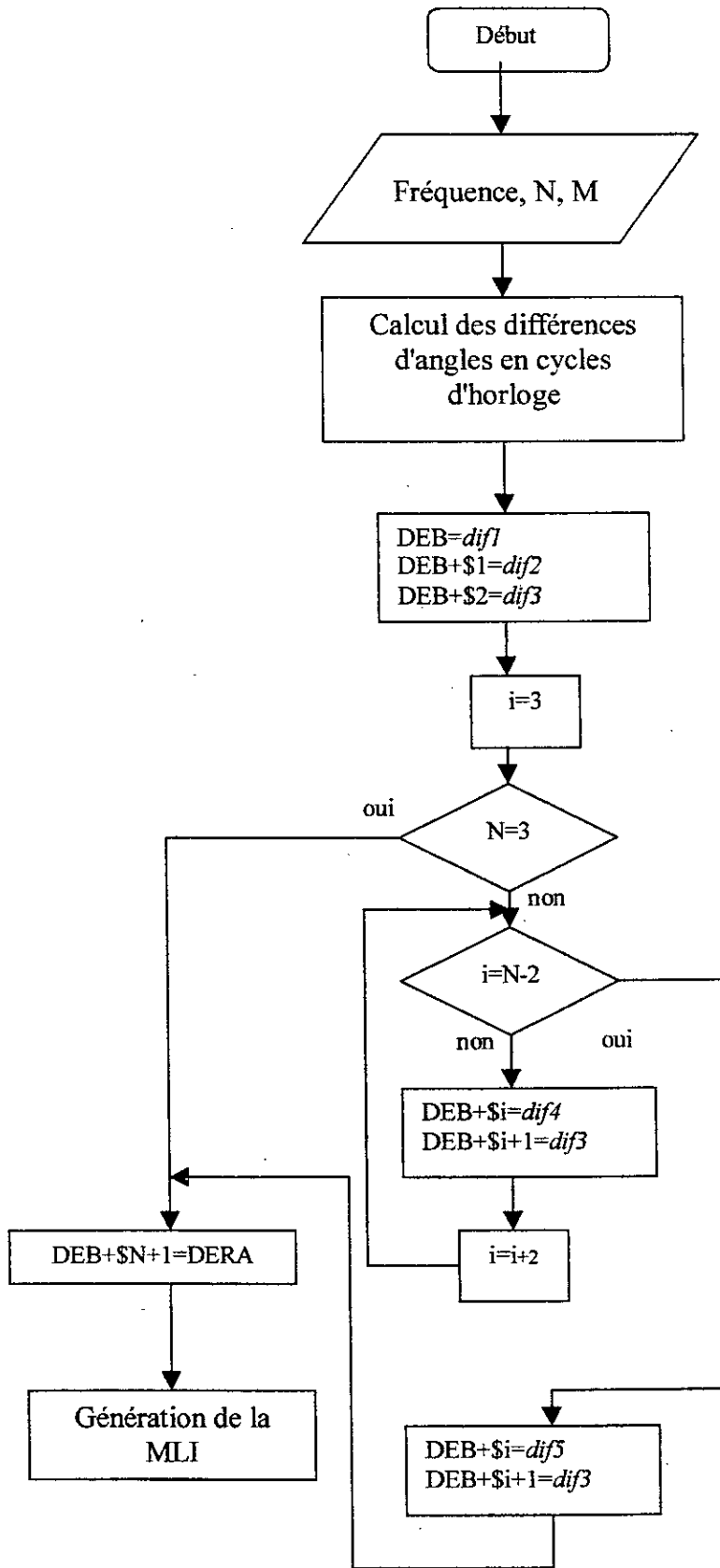


Figure VI. 10 : Organigramme de génération de la MLI cas impair

- Le temps de calcul des angles est de l'ordre des centaines de micro seconde, ce qui nous permis de dire la aussi que la génération des angles et en temps réel.
- Les figures ci-dessous montre la différence entre les angles calculé par notre programme assembleur et les angles théoriques pour différentes valeurs de fréquences et de N.

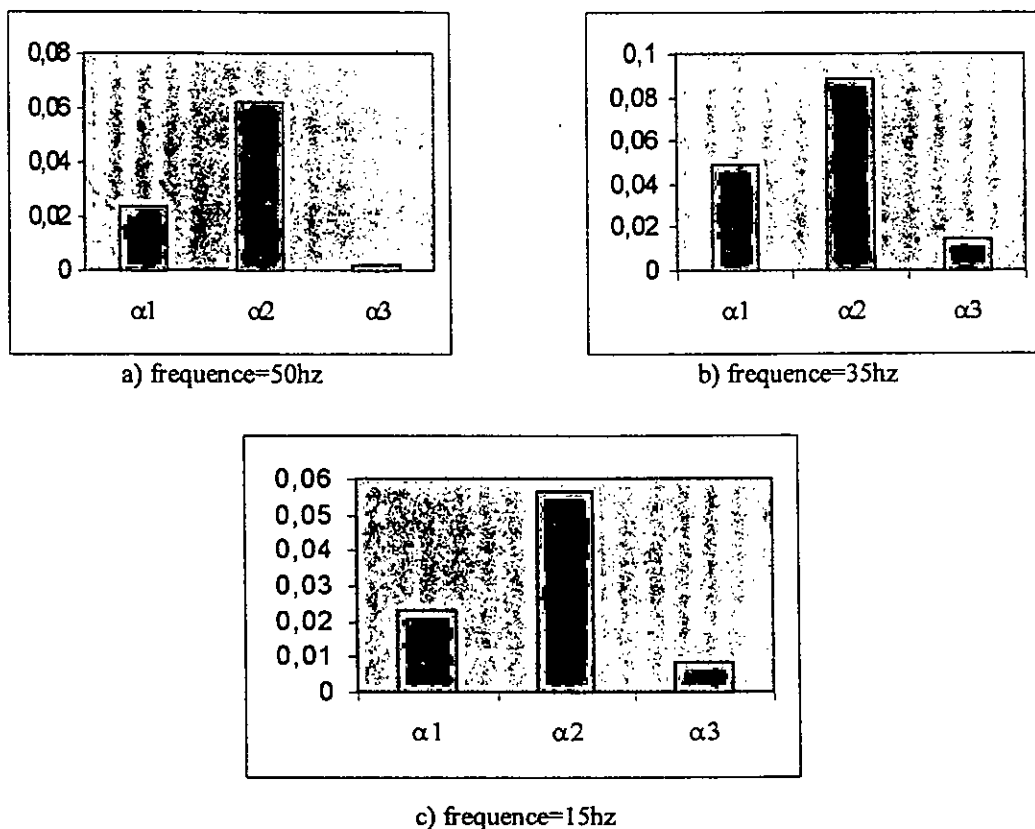


Figure VI.12 : Erreur sur les angles entre les valeurs théoriques et les valeurs calculées par le MC pour N=3.

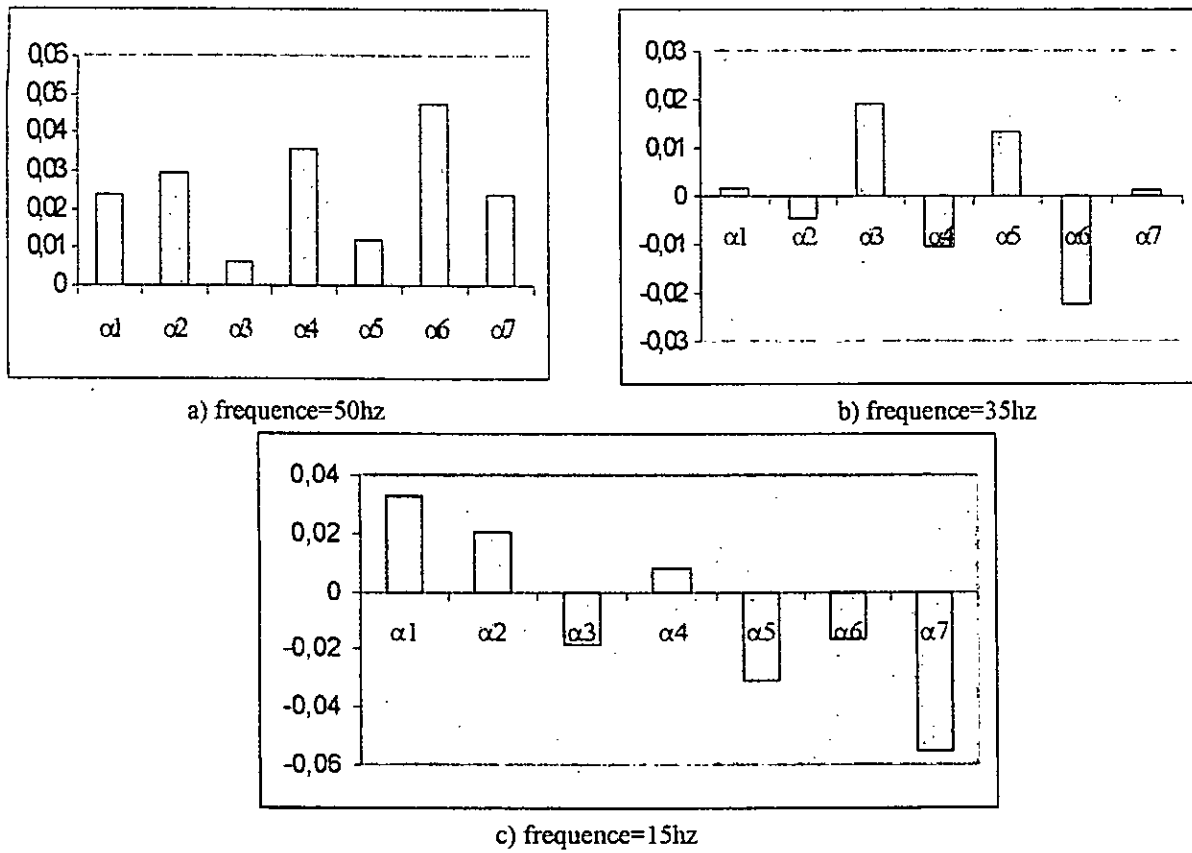


Figure VI.11 : Erreur sur les angles entre les valeurs théoriques et les valeurs calculées par le MC pour $N=7$.

Remarque :

L'ordre des harmoniques à éliminer est de la forme $6k + 1$, $k = 1, 2, \dots, N$. Chaque harmonique de tension (et de courant) d'ordre $6k-1$ et $6k+1$ cause un couple pulsé, on souhaite toujours les éliminer par paires. Pour cela N est souvent choisi impair.

I.1.3 Organigramme de génération des niveaux haut et bas :

ETATH=[PORTA6=1, PORTA5=0] Première période

ETATB=[PORTA6=0, PORTA5=0]

ETATH1=[PORTA6=0, PORTA5=1] Deuxième période

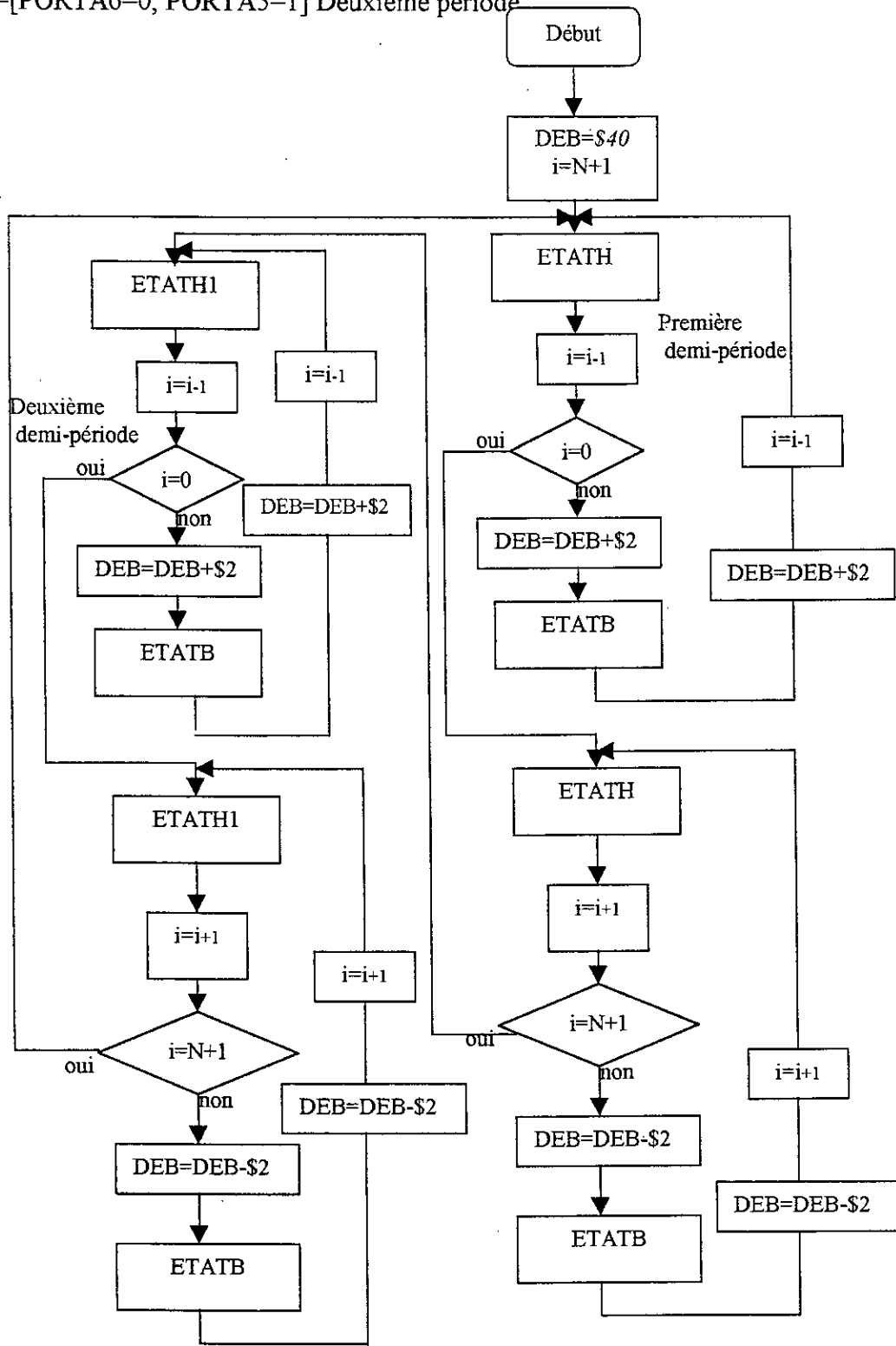


Figure VI.12 : Organigramme de génération des niveaux bas et haut de la MLI cas pair

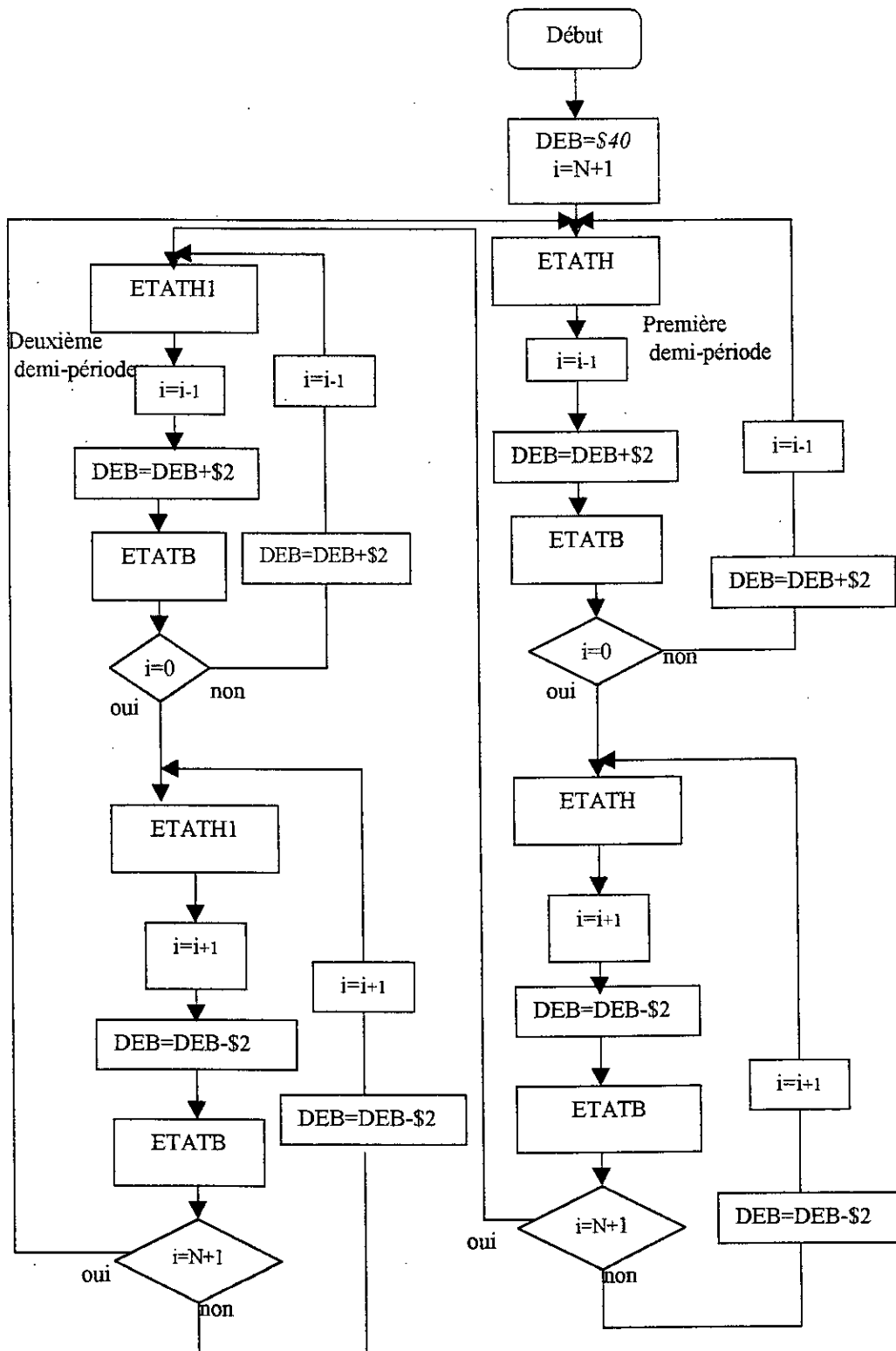


Figure VI.13 : Organigramme de génération des niveaux bas et haut de la MLI cas impair

VI.1.4 Implémentation sur microcontrôleur

Les programmes assembleur ont été testés par le simulateur **SIM68** conçu pour quelques versions du 68HC11, ensuite implémenté sur le module adapt11.

adapt11 est un outil pratique pour développer des applications à base des microcontrôleurs 68HC11, incluant une EEPROM interne ou externe selon la version.

Le tableau 1 montre la configuration du module adapt11.

RAM interne	EEPROM interne	adresse des registres	Type du microcontrôleur
256 \$0000	2k \$F800	\$10**	68HC811E2

Tableau 1 : Configuration du adapt11

Les programmes sont chargés à partir d'une interface PC série RS232, ce qui permet une programmation facile et rapide. Les programmes sont écrits en mnémonique sous format texte ensuite assemblés sous le format S-19, puis chargés à partir du PC vers le microcontrôleur par le logiciel de communication PCBUG11 dans l'EEPROM du 68HC11.

➤ **Ecriture et effacement de l'EEPROM :**

- Gardé le switch1 sur la position BOOT
- Appuyé sur le bouton RESET
- Lancé le pbug11 ensuite suivre les instructions suivantes:
 - Définir l'espace adressable de l'EEPROM

EEPROM \$F800 \$FFFF

- Enlevé la protection contre l'écriture de l'EEPROM

ms \$1035 \$10

- Effacement complet de l'EEPROM

EEPROM ERASE BULK \$F800

- Pour une programmation rapide on peut désactiver la commande Erase-Before-Write effacement avant écriture puisque l'EEPROM est entièrement effacé.

EEPROM ERASE DISABLE

- La dernière étape consiste à charger notre programme.

Loads \Mes fichiers \Mon programme.S19

Conclusion:

Les procédures d'approximation des résultats ont permis d'élaborer des algorithmes pouvant, pour un ordre et une amplitude du fondamental donnés, générer les angles α_i très proches de la solution exacte, avec une erreur très faible et un temps de calcul réduit pour la méthode ASHE pair et impair. Ce qui a permis l'implémentation de ces méthodes sur Microcontrôleur.

CONCLUSION

Conclusion Générale

L'objectif principal de ce travail est la génération d'un signal MLI avec un Microcontrôleur en temps réel, pour la commande d'un moteur triphasé.

Le problème qui se pose est de trouver un bon algorithme qu'on peut implémenter sur le Microcontrôleur 68HC11 pour générer les angles de commutation en temps réel et avec une bonne précision.

On a réussi à approcher le modèle de la solution, pour le cas pair et impair de la ASHE, par des polynômes d'interpolation du deuxième degré afin de les générer en temps réel.

Dans le cas de la ASHE impair, les résultats obtenus sont très satisfaisants puisque les angles calculés, par le Microcontrôleur, présentent une erreur qui n'atteint pas le dixième du degré pour la plupart et un temps de calcul de l'ordre des centaines des micros seconde, ce qui permet de dire que la génération des angles se fait en temps réel. En ce qui concerne le cas pair, les résultats sont moins bons sur la précision des angles pour $N > 4$ mais acceptables, puisque l'erreur ne dépasse pas $\frac{1}{2}$ degré ; le temps de calcul des angles est très minime, par rapport à celui du cas impair, pratiquement de la moitié.

Notre application a été développée par le Microcontrôleur 68HC11, qui possède un puissant jeu d'instruction, avec des modes d'adressages très souple et précis en même temps.

Comme perspective à notre travail et afin d'améliorer la précision des angles calculés tout en gardant un temps minimal, nous proposons une implémentation sur un Microcontrôleur 16 bit ou un 8bit avec une grande fréquence d'horloge.

BIBLIOGRAPHIE

[ARB.86] Kurt ARBENZ "Analyse numérique" Presses polytechniques Romandes, Lousanne 3^{ème} édition, 1986.

[BOU.83] M.BOUMEHRAT A.GOURDIN "Méthodes numériques appliquées " OPU, 1983.

[BOU.00] A.BOUDEHANE F.SEHIRI, PFE "Implémentation d'une méthode numérique pour la génération de signaux MLI " ENP 2000.

[DEF.98] Thierry DEFLANDRE Philippe MAURAS "Les harmoniques sur les réseaux électriques" Eyrolles, 1998.

[FER.94] Abdelaziz FERDJOUNI. Thèse de magister " Commande en boucle ouverte de la vitesse d'un moteur asynchrone par microprocesseur " ENP, 1994.

[LAB.95] Francis LABRIQUE Guy SEGUIER Robert BAUSIERE " Les convertisseurs de l'électronique, la conversion continu alternatif " volume 4 édition Lavoisier, 1995.

[MOK.00] Mohand MOKHTARI "Matlab 5.2 & 5.3 et Simulink 2 & 3" édition Springer, 2000.

[MAC.71] Gérard HACQUE "Mathématiques pour l'informatique, algorithmique numérique, cours " volume 1, Armand COLIN, 1971.

[SCH.86] Francis SCHEID "Analyse numérique cours et problèmes "Série Schaum, 1986.

[SEG.77] Guy SEGUIER Francis NOTELET "Electrotechnique industrielle " Bayeux, 1977.

[SEG.85] Guy SEGUIER "Electronique de puissance : les fonctions de base et leurs principales applications " 5^{ème} édition, DUNOD 1985.

[SEG.95] G.SEGUIER F.LABRIQUE R.BAUSIERE "Les convertisseurs de l'électronique de puissance, volume 4 La conversion continu-alternatif " Paris 2^{ème} édition, 1995.

[SEL.01] Yamine SELLAMI, thèse de magister "Contribution à la commande floue des systèmes. Application la commande d'un moteur asynchrone " ENP, 2001.

[SUN.96] J.SUN S.BEINEKE, et M.GROTSTOLLEN "Optimal PWM Based On Real-time, Solution of Harmonics, Elimination Equations, IEE transaction of power electronics, volume 11" no. 4, july 1996.

[TAV1.97] Christian TAVERNIER "Microcontrôleur 68HC11 et les versions UV PROM et EEPROM, Applications " DUNOD, Paris, 1997.

[TAV2.97] Christian TAVERNIER "Microcontrôleur 68HC11 et les versions UV PROM et EEPROM, Description " DUNOD, Paris, 1997.

ANNEXES

Annexe A

Voici quelques fonctions analytiques et leurs changements de variables appropriés, afin d'en ramener l'étude au cas de la droite des moindres carrés $Y=a+bX$.

Exemple:

$y = a x^b$ les changements de variables à effectuer sont $Y=\ln(y)$, $X=\ln(x)$, $a=\ln(a)$, $b=b$.

Fonctions analytiques	Formule développée
$y = a + b x^n$	$y = a + b x^n$
$y = a x^b$	$\ln(y) = \ln(a) + b \ln(x)$
$y = a e^{bx^n}$	$\ln(y) = \ln(a) + b x^n$
$y = a b^{x^n}$	$\ln(y) = \ln(a) + \ln(b) x^n$
$y = \frac{x^n}{a + b x^m}$	$\frac{x^n}{y} = a + b x^m$
$y = a x + b x^2$	$\frac{y}{x} = a + b x$
$y = a + \frac{b}{x^n}$	$y = a + b \frac{1}{x^n}$

ANNEXE B

Programme assembleur pour la génération de la MLI

```
PORTB      EQU  $1000
ETATH      EQU  $40
ETATB      EQU  $00
ETATH1     EQU  $20
DEB        EQU  $40
NB         EQU  $74
MM         EQU  $76
ADR00      EQU  $60
ADR1       EQU  $62
ADR2       EQU  $64
ADR34      EQU  $66
ADRMPM     EQU  $68
ADRPMD     EQU  $6A
DELTA      EQU  $6C
DERA       EQU  $6E
ADR78      EQU  $70
FREQT      EQU  $72
ORIGINE    EQU  $f800
```

*** INITIALISATION ***

```
ORG  ORIGINE
LDS  #$FF
```

*** CALCUL DE DELTA ***

```
LDX  #47620
LDAA #10 ;VALEUR DE L'INDICE DE MODULATION
STAA MM
LDD  #50 ;FREQUENCE DE TRAVAIL
STD  FREQT
LDAA #$01 ;NOMBRE D'ANGLE DE LA MLI
STAA NB
LDAA FREQT+$1
LDAB NB
MUL
XGDX
IDIV
XGDX
SUBD #$0003
STD  DELTA
NOP
```

*** CALCUL DE LA PENTE ***

```

LDAA  NB
CMPA  #$1
BEQ   SPEC2
LDAB  NB
MUL
STAB  ADR2
LDAA  NB
LDAB  #20
MUL
STAB  ADR1
LDAA  #126
SUBA  ADR1
ADDA  ADR2
STAA  ADR2
BRA   SAUT
SPEC2  LDAA #120
        STAA ADR2
SAUT   NOP

```

*** CALCUL DE P*M ***

```

LDAA  #$08
LDAB  MM
MUL
LDAA  ADR2
MUL
LDX   FREQT
IDIV
STX   ADR34 ,(P*M)cycles

```

```

LDAA  #80
LDAB  MM
MUL
LDX   FREQT
IDIV
STX   ADR00 ,M cycles

```

```

LDD   ADR34
ADDD  ADR00
STD   ADR78 ,(M+P*M)cycles
NOP

```

* CALCUL DE DIFFERENTES VALEUR *

* DERNIER ANGLE

```

LDD   #23810
LDX   FREQT
IDIV
XGDX
SUBB  #$03
SUBB  ADR00+$1
STD   DERA ,dernier angle en cycles

```

```
LDD DELTA
SUBD ADR78
STD ADRMPM ,calcul de(delta-(M+P*M))cycles
```

```
ADDD ADR00
STD ADRPMD ,calcul de(delta-M)cycles
```

```
LDD ADR78
SUBD #$0003
STD ADR78
NOP
```

* CALCUL DES DIFFERENCES D'ANGLE*

```
REF LDX #DEB
LDD ADRPMD
STD 00,X
LDAA NB
INX
INX
LDY ADR78
STY 00,X
INX
INX
DECA
CMPA #$00
BEQ DERNIER
LDY ADRMPM
STY 00,X
BRA REF
```

```
DERNIER NOP
LDD DERA
STD 00,X
NOP
```

**** PROGRAMME GENERATION DE LA MLI ****

```
A1 LDAA NB
LDAB #$02
MUL
ADDB #$1
STAB NB
ENCORE1 LDX #DEB
LDAA NB
LDY 00,X
LDAB #ETATH
STAB PORTB
AMEL1 DEY
BNE AMEL1
```

```

        INX
        INX
        DECA
        BEQ      A2
        LDY     00,X

        LDAB    #ETATB
        STAB    PORTB
AMEL2   DEY
        BNE     AMEL2
        INX
        INX
        DECA
        BNE     ENCORE1
*****
***
*****
A2      LDAA    NB
ENCORE2 DEX
        DEX
        LDY     00,X

        LDAB    #ETATH
        STAB    PORTB
AMEL3   DEY
        BNE     AMEL3
        DEX
        DEX
        DECA
        BEQ     A3
        LDY     00,X

        LDAB    #ETATB
        STAB    PORTB
AMEL4   DEY
        BNE     AMEL4
        DECA
        BNE     ENCORE2
*****
**
        BRA     A3
A5      BRA     A1
**
*****
A3      LDX     #DEB
        LDAA    NB
ENCORE3 LDY     00,X

        LDAB    #ETATH1
        STAB    PORTB
ASSIA1  DEY
        BNE     ASSIA1

```

```

        INX
        INX
        DECA
        BEQ    A4
        LDY    00,X

        LDAB   #ETATB
        STAB   PORTB
ASSIA2  DEY
        BNE   ASSIA2
        INX
        INX
        DECA
        BNE   ENCORE3
*****
***
*****
A4      LDAA   NB
ENCORE4 DEX
        DEX
        LDY    00,X

        LDAB   #ETATH1
        STAB   PORTB
ASSIA3  DEY
        BNE   ASSIA3
        DEX
        DEX
        DECA
        BEQ    A6
        LDY    00,X

        LDAB   #ETATB
        STAB   PORTB
ASSIA4  DEY
        BNE   ASSIA4
        DECA
        BNE   ENCORE4
*****
A6      BRA    A5
*A6     SWI

```


Programme génération de la MLI pour les nombres impairs

```

PORTA      EQU  $1000
ETATH      EQU  $40
ETATB      EQU  $00
ETATH1     EQU  $20
DEB        EQU  $40
FREQT      EQU  $72
NB         EQU  $74
MM         EQU  $76
ADR1       EQU  $78
ADR2       EQU  $7A
DELTA      EQU  $6C
DERA       EQU  $6E
PM1        EQU  $60
PM21       EQU  $62
DP21       EQU  $64
DM21       EQU  $66
P1         EQU  $68
P2         EQU  $6A
A90        EQU  $70
ORIGINE    EQU  $f800

```

*** INITIALISATION ***

```

ORG  ORIGINE
LDS  #$FF

```

*** CALCUL DE DELTA ***

```

LDX  #47620
LDAA #10 ; VALEUR DE L'INDICE DE MODULATION
STAA MM
LDD  #50 ; FREQUENCE DE TRAVAIL
STD  FREQT
LDAA #$02 ; NOMBRE D'ANGLE DE LA MLI
STAA NB
LDAA FREQT+$1
LDAB NB
MUL
XGDX
IDIV
XGDX
SUBD #$0003
STD  DELTA
NOP
NOP
NOP

```

*** CALCUL DES PENTES ***

```

LDAA  NB
LDAB  NB
MUL
LDX   #$4
IDIV
STX   ADR1

```

*PENTE2

```

NOP

```

```

LDAA NB
LDAB #$7
MUL
STD ADR2
LDD #67
SUBD ADR2
ADDD ADR1
STD P2

```

*PENTE1

```

NOP
LDAA NB
LDAB #13
MUL
LDX #$02
IDIV
STX ADR2
LDD #55
NOP
NOP

```

```

*****
*** CALCUL DE P1,P2*M ***
*****

```

```

LDAA #$08
LDAB MM
MUL
LDAA P1+$1
MUL
LDX FREQT
IDIV
STX PM1 ,(P1*M)cycles

```

```

LDAA #$08
LDAB MM
MUL
LDAA P2+$1
MUL
LDX FREQT
IDIV
STX PM21 ,(P2*M)cycles
LDD PM21
ADDD PM1
STD PM21 ,(P1+P2)*M cycles
NOP

```

```

*****
* CALCUL DE DIFFERENTES VALEURS *
*****

```

*DELTA-PM21

```

LDD DELTA
SUBD PM21
STD DM21

```

*DELTA+PM21

```

LDD DELTA
ADDD PM21
STD DP21

```

*(P1+P2)*M

```

LDD PM21
SUBD #$3
STD PM21

```

*(P1*M)

LDD PM1
SUBD #S3
STD PM1

*DERNIER ANGLE

LDD #1
LDX FREQT
FDIV
STX ADR1
LDD #S1705
LDX FREQT
IDIV
XGDX
ADDD ADR1
SUBD #3
STD A90
NOP

* CALCUL DES DIFFERENCES D'ANGLE*

LDX #DEB
LDD PM1
STD 00,X
INX
INX

LDD DELTA
STD 00,X
LDAA NB
LDAB #2
MUL
SUBD #2

REF

INX
INX
LDY DM21
STY 00,X
INX
INX
DECB
CMPB #1
BEQ DER

LDY PM21
STY 00,X
DECB
BRA REF

DER

LDAA NB
CMPA #2
BEQ DERNIER

DEX
DEX
DEX
DEX
LDD DP21
STD 00,X

DERNIER

LDX #DEB
LDAA NB

```

LDAB #2
MUL
SUBB #1
STAB ADR1
LDD A90
DIFF SUBD 00,X
SUBD #3
INX
INX
DEC ADR1
BNE DIFF
STD DERA
STD 00,X

```

```

*****
**** PROGRAMME GENERATION DE LA MLI ****
*****

```

```

LDAA NB
LDAB #302
MUL
STAB NB

A1 LDX #DEB
LDAA NB
ENCORE1 LDY 00,X

```

```

LDAB #ETATH
STAB PORTA
AMEL1 DEY
BNE AMEL1
INX
INX
DECA
BEQ A2
LDY 00,X

```

```

LDAB #ETATB
STAB PORTA
AMEL2 DEY
BNE AMEL2
INX
INX
DECA
BNE ENCORE1

```

```

*****
***
*****

```

```

A2 LDAA NB
ENCORE2 DEX
DEX
LDY 00,X

```

```

LDAB #ETATH
STAB PORTA
AMEL3 DEY

```

```
BNE AMEL3
DEX
DEX
DECA
BEQ A3
LDY 00,X
```

```
LDAB #ETATB
STAB PORTA
AMEL4 DEY
      BNE AMEL4
      DECA
      BNE ENCORE2
```

**

```
BRA A3
A5    BRA A1
```

**

```
LDX #DEB
A3   LDAA NB
ENCORE3 LDY 00,X
```

```
LDAB #ETATH1
ASSIA1 STAB PORTA
      DEY
      BNE ASSIA1
      INX
      INX
      DECA
      BEQ A4
      LDY 00,X
```

```
LDAB #ETATB
ASSIA2 STAB PORTA
      DEY
      BNE ASSIA2
      INX
      INX
      DECA
      BNE ENCORE3
```

```
LDAA NB
A4   DEX
ENCORE4 DEX
      LDY 00,X
```

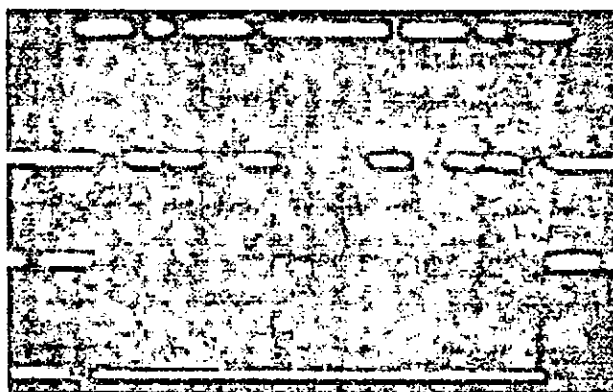
```
LDAB #ETATH1
ASSIA3 STAB PORTA
      DEY
```

```
BNE    ASSIA3
DEX
DEX
DECA
BEQ    A6
LDY    00,X
```

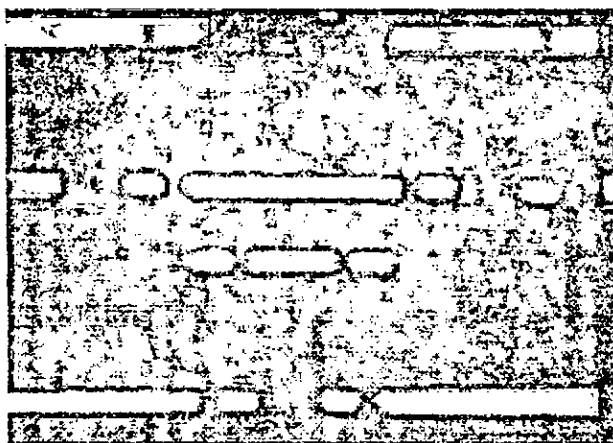
```
ASSIA4    LDAB    #ETATB
           STAB    PORTA
           DEY
           BNE    ASSIA4
           DECA
           BNE    ENCORE4
*****
A6         BRA    A5
```

ANNEXE C

Chronogrammes de sortie des signaux MLI visualisés à partir d'un oscilloscope pour le cas pair et le cas impair.



C.1) $f=35\text{HZ}$, $N=7$ et $M=0.7$



C.1) $f=15\text{HZ}$, $N=2$ et $M=0.7$