

**RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE**  
**MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE**  
**SCIENTIFIQUE**

## **UNIVERSITÉ IBN-KHALDOUN DE TIARET**

**FACULTÉ DES SCIENCES APPLIQUEES**  
**DÉPARTEMENT DE GENIE ELECTRIQUE**



# **MEMOIRE DE FIN D'ETUDES**

**Pour l'obtention du diplôme de MASTER**

**Domaine : Sciences et Technologie**

**Filière : Génie Electrique**

**Spécialité : Informatique Industrielle**

## **THÈME**

**Techniques de commande  
d'un onduleur de tension  
à faible puissance**

***Préparé par : M. Hamed amar***

### **Devant le Jury :**

<b>Nom et prénoms</b>	<b>Grade</b>	<b>Université</b>	<b>Qualité</b>
NASRI DJILALI	MCA	U-Tiaret	Président
BELARBI Mustapha	MCB	U-Tiaret	Encadreur
GHELLAB Abdelkader	MCB	U-Tiaret	Examineur

**PROMOTION 2016 /2017**

## **REMERCIEMENT :**

*Nous remercions notre Dieu qui nous a donné le courage et la  
volonté pour achever ce travail.*

*Nous tenons à exprimer notre remerciement profond à Monsieur  
BELARBI Mustapha qui nous facilité le chemin de notre étude.*

*Nous remercions les membres de jury pour avoir accepté  
d'examiner ce travail.*

*A toute personne qui nous aidé de pris ou loin.*

# **Dédicaces**

*Je remercie ALLAH le tout puissant d'avoir me donné la force et le courage pour finir ce modeste travail qui est le fruit des années de quête de savoir.*

*A tous ma famille chacun par son nom*

*A mes très chers parents ...*

*A mes très chers frères et sœur*

*A mon encadreur Monsieur Mustapha BELARBI pour ses conseils et ses encouragements continus*

*Tous les enseignants qui ont contribué à ma formation durant mes études*

*Tous mes amis chacun par son nom*

*A mon collègue et frère*

*A tous mes amis qui m'ont aidé de près ou de loin pour mener à bien ce travail*

*H.Amar*

## SOMMAIRE

Liste des abréviations .....	6
Liste des figures.....	7
Introduction générale.....	9
<b>CHAPITRE I : L'onduleur de tension et ses stratégies de commande.....</b>	<b>11</b>
I.1 Introduction.....	12
I.2 Définition de l'onduleur .....	12
I.3 Classification des onduleurs .....	12
I.3.1 Onduleur autonome .....	12
I.3.2 Onduleur non autonome .....	12
I.4 Choix des composants.....	12
I.5 Différents types d'onduleur pour l'alimentation des machines asynchrones ....	14
I.5.1 Onduleur de courant.....	14
I.5.2 Onduleur de tension .....	14
I.6 Modélisation de l'onduleur de tension triphasé .....	15
I.7 Techniques de commande de l'onduleur de tension.....	16
I.7.1 Commande en pleine onde .....	16
I.7.2 Modulation de la largeur d'impulsion avec la technique « sinus-triangle » .....	18
I.8 Conclusion .....	20
<b>CHAPITRE II : Le Microcontrôleur.....</b>	<b>21</b>
II.1 Les microcontrôleurs de la firme MICROCHIP .....	23
II.2 Etude du microcontrôleur PIC16F877.....	23
II.3 Architecture interne du microcontrôleur PIC16F877.....	25
II.4 Accès à la mémoire RAM .....	26
II.4.1 Adressage direct à la mémoire RAM.....	27
II.4.2 Adressage indirect à la mémoire RAM.....	27

## Sommaire

II.5 Mémoire programme .....	27
II.6 Etude du microcontrôleur PIC18F4550.....	28
II.7 Architecture interne du microcontrôleur PIC18F4550.....	29
II.8 Organisation générale de la mémoire .....	32
II.8.1 Organisation de la mémoire programme.....	32
II.8.2 Organisation de la mémoire de données .....	34
II.8.3 Mémoire EEPROM de données.....	36
II.9 Conclusion .....	37
<b>CHAPITRE III : Implantation d'une commande PWM sur un microcontrôleur.</b>	<b>38</b>
III.1 Schéma de conception d'un l'onduleur triphasé .....	39
III.2 La commande MLI de type sinus triangle .....	41
III.3 Exemple 1 de génération d'un un signal MLI.....	41
III.4 Exemple 2 de génération d'un un signal MLI.....	48
III.5 Allocation manuelle des variables dans l'espace mémoire.....	51
III.6 Conclusion.....	57
Conclusion générale .....	58
Bibliographie.....	60

## Liste des abréviations

<b>Symbole</b>	<b>Signification</b>
<b>DC</b>	Direct Current
<b>AC</b>	Alternative Current
<b>MOSFET</b>	Metal Oxide Semiconductor Field Effect Transistor
<b>IGBT</b>	Insulated Gate Bipolar transistor
<b>V<sub>ref</sub></b>	tension de référence
<b>V<sub>p</sub></b>	tension de la porteuse
<b>F</b>	fréquence
<b>f<sub>p</sub></b>	fréquence de porteuse
<b>MLI</b>	modulation de largeur d'impulsion
<b>PWM</b>	Pulse Width Modulation
<b>PIC</b>	Programmable Interface Controller.
<b>RISC</b>	Reduced Instructions Set Computer
<b>EPROM</b>	Erasable Programmable Read-Only Memory
<b>EEPROM</b>	Electrically-Erasable Programmable Read-Only Memory ou mémoire morte effaçable électriquement et programmable
<b>RAM</b>	Random Access Memorr ou mémoire à accès direct
<b>RTC</b>	Real time clock ou horloge temps réel
<b>MCLR</b>	Master Clear
<b>W</b>	Working register
<b>FSR</b>	File Select Registre
<b>BSR</b>	Bank Select Register
<b>I2C</b>	Inter Integrated Circuit
<b>VDD</b>	Voltage Drain Drain
<b>VSS</b>	Voltage Source Source
<b>Ko</b>	kilo-octets
<b>ADC</b>	Analog to Digital Converter
<b>CAN</b>	Convertisseur Analogique/numérique
<b>LCD</b>	Liquid Crystal Display ou afficheur à cristaux liquide
<b>PCL</b>	Program Counter Low
<b>PCH</b>	Program Counter High
<b>PCU</b>	Program Counter Upper
<b>PCLATH</b>	Program Counter LATch High
<b>PCLATU</b>	Program Counter LATch Upper

## Liste des figures

### Chapitre I

Figure I- 1 : Schéma synoptique d'un onduleur .....	12
Figure I- 2 : Classification des commutateurs de puissance selon la tension et la fréquence .....	13
Figure I- 3 : Classification des commutateurs de puissance selon la densité de courant et la rapidité de commutation .....	13
Figure I- 4 : Association redresseur/onduleur de courant .....	14
Figure I- 5 : Association redresseur/onduleur de tension.....	14
Figure I- 6 : Circuit de fonctionnement de l'onduleur triphasé .....	15
Figure I- 7 : Signaux de commande en pleine onde .....	17
Figure I- 8 : Les tensions de sortie de l'onduleur à commande pleine onde.....	17
Figure I- 9 : Principe général de la technique sinus-triangle.....	18
Figure I- 10 : Principe de la MLI sinusoïdale .....	18
Figure I- 11 : La création des commandes et des tensions de sortie triphasées .....	19

### Chapitre II

Figure II- 1 : Le brochage du microcontrôleur PIC16F877 (Boitier DIL40).....	23
Figure II- 2 : Architecture interne d'un microcontrôleur PIC16F877 .....	25
Figure II- 3 : Architecture des registres internes du microcontrôleur PIC16F877.....	26
Figure II- 4 : Composition du registre d'état STATUS d'un microcontrôleur PIC16F877 .....	27
Figure II- 5 : Adressage direct et indirect vers la RAM du microcontrôleur PIC16F877.....	27
Figure II- 6 : Adressage de la mémoire programme et la pile de sauvegarde du microcontrôleur PIC16F877 .....	28
Figure II- 7 : Le brochage du microcontrôleur PIC18F4550 (Boitier DIL40).....	29
Figure II- 8 : Architecture interne d'un microcontrôleur PIC18F4550 .....	30
Figure II- 9 : Chronogramme d'exécution des instructions .....	31
Figure II- 10 : Organisation de la mémoire programme du PIC18F4550.....	32
Figure II- 11 : Organisation en pages de la mémoire de données du PIC18F4550.....	34
Figure II- 12 : Mécanisme de pagination de la mémoire de données du PIC18F4550 .....	35

### Chapitre III

Figure III- 1 : Schéma de la partie drivers de commande d'un onduleur de tension triphasé	40
Figure III- 2 : Schéma de la partie puissance d'un onduleur de tension triphasé	40
Figure III- 3 : Organigramme 1 d'une commande MLI implanté sur un PIC18F4550	42
Figure III- 4 : Signaux de commande de l'onduleur de tension commandé par l'organigr. 1	43
Figure III- 5 : Tension de sortie de l'onduleur de tension commandé par l'organigramme 1	43
Figure III- 6 : Organigramme 2 d'une commande MLI implanté sur un PIC16F877	45
Figure III- 7 : Signaux de commande résultants après implantation de l'organigramme 2	46
Figure III- 8 : Tension de sortie de l'onduleur de tension commandé par l'organigramme 2	46
Figure III- 9 : Organigramme 3 d'une commande MLI implanté sur un PIC16F877	49
Figure III- 10 : Signaux de commande résultants après implantation de l'organigramme 3	50

Figure III- 11 : Tension de sortie de l'onduleur de tension commandé par l'organigramme 3	50
Figure III- 12 : Circuit simplifié d'un générateur de fonction	51
Figure III- 13 : Génération de fonction avec un PIC16F877	52
Figure III- 14 : Remplissage de la fonction triangulaire sur la Bank1	53
Figure III- 15 : Remplissage de la fonction sinus sur la Bank2	53
Figure III- 16 : Tension de sortie générées avec l'organigramme de la Figure III- 13	54
Figure III- 17 : Agrandissement de la tension de sortie de la Figure III- 16	54
Figure III- 18 : Optimisation de l'écriture dans la mémoire EEPROM	55
Figure III- 19 : Optimisation des fonctions de génération de signaux avec un PIC16F877	56
Figure III- 20 : Tensions de sortie générées avec le code de la Figure III- 19	57

# *Introduction générale*

## *Introduction générale*

Les onduleurs de tension constituent une fonction incontournable de l'électronique de puissance, ils sont présents dans les domaines d'applications les plus variés, dont le plus connu est sans doute celui de la variation de la vitesse des machines à courant alternatif. La forte évolution de cette fonction s'est appuyée, d'une part, sur le développement de composants à semi-conducteurs entièrement commandables, puissants, robuste et rapides ; d'autre part, sur l'utilisation quasi-généralisée des techniques dites MLI ou PWM (Modulation de Largeurs d'Impulsion ou Pulse Width Modulation).

L'objectif de ce mémoire est l'étude et la simulation d'une carte de commande d'un onduleur de tension par un microcontrôleur dont le choix de ce dernier est judicieux et qui va nous bien servir dans ce projet.

Il nous a paru utile du point de vue méthodologie de commencer d'abord par la simulation d'une carte de commande pour un onduleur de tension.

Au long de ce projet nous avons utilisé le logiciel Proteus comme outil de simulation, le logiciel CCS-PICC comme outil de programmation en langage C, où chaque programme développé en PIC C sera simulé par la suite sous Proteus.

Le présent manuscrit a été rédigé comme suit :

Le premier chapitre a été réservé à l'étude des onduleurs de tension et leurs stratégies de commande.

Dans le second chapitre, nous avons plus détaillé l'architecture interne des microcontrôleurs.

Quant au dernier chapitre, nous avons détaillé les techniques d'implantation d'une commande PWM sur les microcontrôleurs.

Enfin une conclusion générale et une réflexion aux extensions qui peuvent améliorer ce travail.

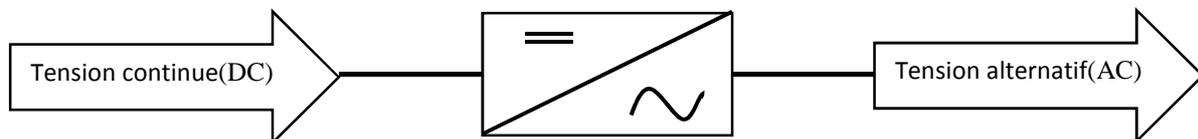
**CHAPITRE I :**  
*L'onduleur de tension*  
*et ses stratégies de commande*

## **I.1 Introduction :**

Dans le domaine de l'électronique de puissance, il existe un dispositif important, qui est l'onduleur. C'est le constituant principal de la plupart des variateurs de vitesse des machines à courant alternatif. Dans ce chapitre, nous allons présenter l'onduleur de tension, son principe de fonctionnement. Ensuite, deux stratégies de commande des onduleurs seront développées : technique de commande en plaine ondes et technique de modulation de largeur d'impulsion MLI.

## **I.2 Définition de l'onduleur :**

L'onduleur est un convertisseur statique permettant de faire un échange d'énergie entre une source de tension continue DC fournie par une batterie ou un redresseur, et une source de tension alternative AC pour alimenter des charges en courant alternatif.



**Figure I- 1 :** Schéma synoptique d'un onduleur

L'onduleur de tension permet de fournir une tension alternative d'amplitude et de fréquence réglables à partir d'une source de tension continue. Il est constitué de cellule de commutation généralement à base de transistors ou thyristors pour les grandes puissances [1].

## **I.3 Classification des onduleurs :**

Il existe plusieurs schémas de réalisation des onduleurs, chacun correspondant à un type d'application déterminé ou permettant des performances recherchées. Les onduleurs sont en général classés selon les modes de commutation de leurs interrupteurs.

### **I.3.1 Onduleur autonome :**

Cet onduleur n'est pas réversible. Il nécessite des composants commandés à la fermeture et à l'ouverture. La fréquence de fonctionnement est variable. Les instants de commutations sont imposés par des circuits externes ; d'où la charge est quelconque.

### **I.3.2 Onduleur non autonome :**

Dans ce cas, les composants utilisés peuvent être de simples thyristors commandés uniquement à la fermeture avec une commutation naturelle contrairement à l'onduleur autonome. L'application principale de ce type d'onduleur se trouve dans les variateurs de vitesse pour moteurs synchrones de très forte puissance où les thyristors sont souvent les seuls composants utilisables.

## **I.4 Choix des composants :**

Les composants de l'électronique de puissances (interrupteurs) sont déterminés par les niveaux de la puissance et la fréquence de commutation. En règle générale, plus les composants

## CHAPITRE I : L'onduleur de tension et ses stratégies de commande

sont rapides, plus la puissance commutée est faible et inversement. A titre indicatif, les transistors MOSFET sont considérés comme des composants très rapides mais de puissance relativement faible. Les transistors bipolaires sont moins rapides que les transistors MOSFET mais d'avantage plus puissants (quelques kHz à une dizaine de kW). Les transistors IGBT sont des composants de gamme standard (jusqu'à 20 kHz à une des dizaines de kW). Les thyristors GTO commutent très lentement les grandes puissances. Ces composants sus indiqués sont du type commandable à l'ouverture et à la fermeture ; ce qui n'est pas le cas pour le thyristor classique [2, 3].

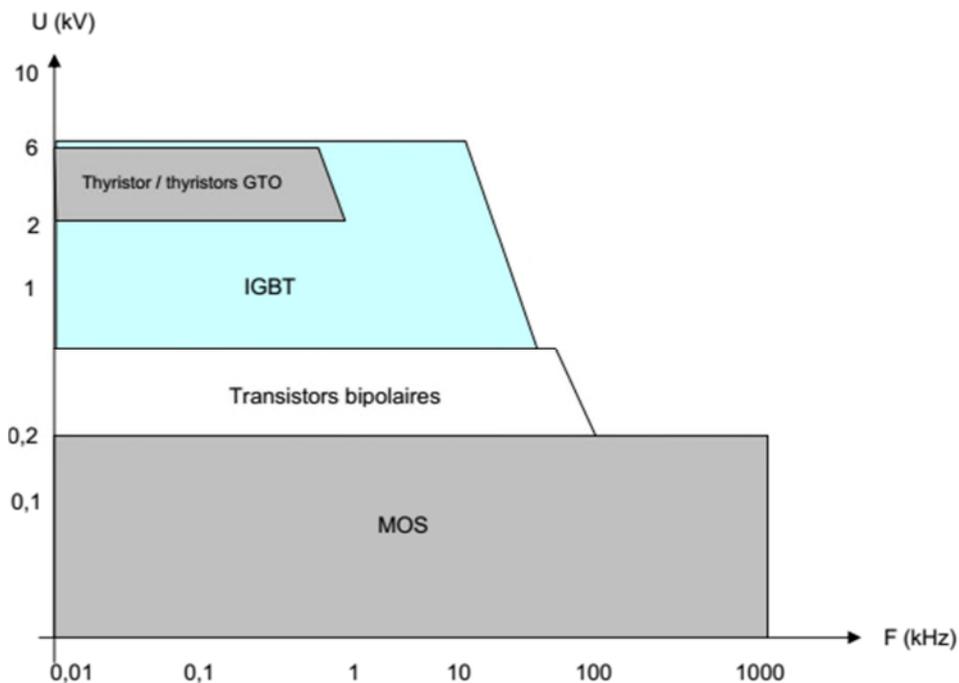


Figure I- 2 : Classification des commutateurs de puissance selon la tension et la fréquence [4]

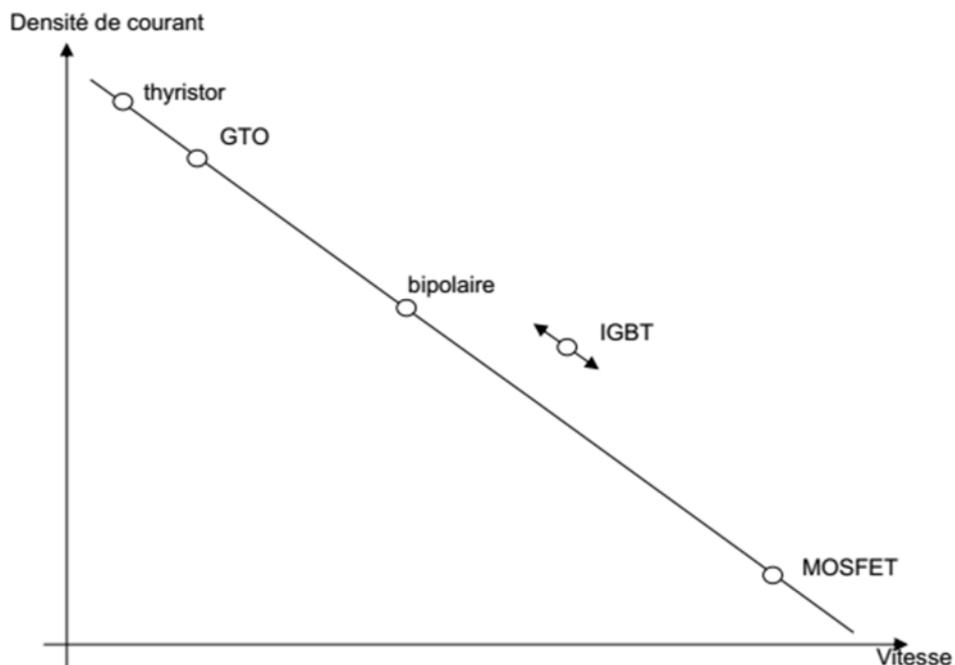


Figure I- 3 : Classification des commutateurs de puissance selon la densité de courant et la rapidité de commutation [4]

### I.5 Différents types d'onduleur pour l'alimentation des machines asynchrones :

Pour fonctionner, l'onduleur nécessite une source à tension/courant continu que l'on peut obtenir en redressant la tension du réseau triphasé par exemple. Généralement, il existe un circuit intermédiaire de liaison qui relie la sortie du redresseur à l'entrée de l'onduleur. On utilise deux types de liaisons : les liaisons à source de courant et les liaisons à source de tension. Un onduleur associé à un redresseur est alors appelé onduleur de courant ou onduleur de tension.

L'onduleur est la dernière partie du variateur de vitesse située avant le moteur. Il fournit des grandeurs électriques variables au moteur. Dans tous les cas, l'onduleur est composé de semi-conducteurs disposés par paires en trois bras. Ces semi-conducteurs commutent sur des signaux en provenance du circuit de commande [5].

#### I.5.1 Onduleur de courant :

Lorsqu'il fonctionne en source de courant, le redresseur fournit un courant constant à l'onduleur ; une inductance de lissage l'aide à maintenir le courant constant. La figure qui suit représente un convertisseur avec un onduleur de courant.

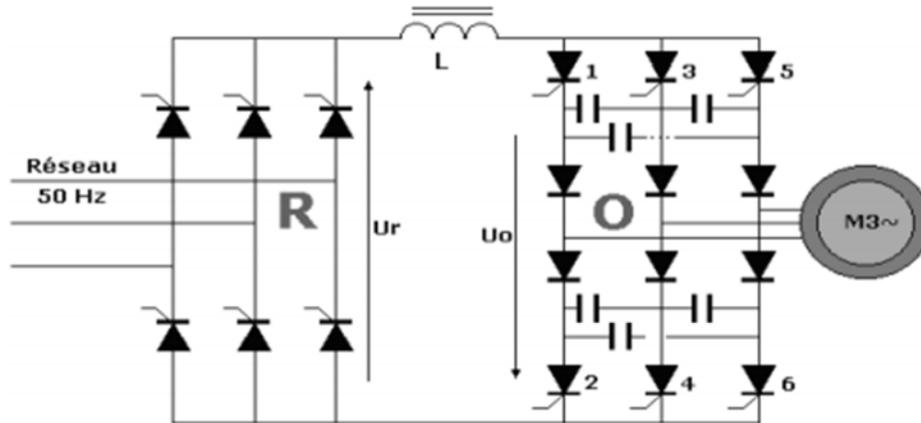


Figure I- 4 : Association redresseur/onduleur de courant

#### I.5.2 Onduleur de tension :

Lorsqu'il fonctionne en source de tension, le redresseur fournit une tension constante à l'onduleur. La présence d'un condensateur dans le circuit de liaison aide alors à maintenir une tension constante à l'entrée de l'onduleur.

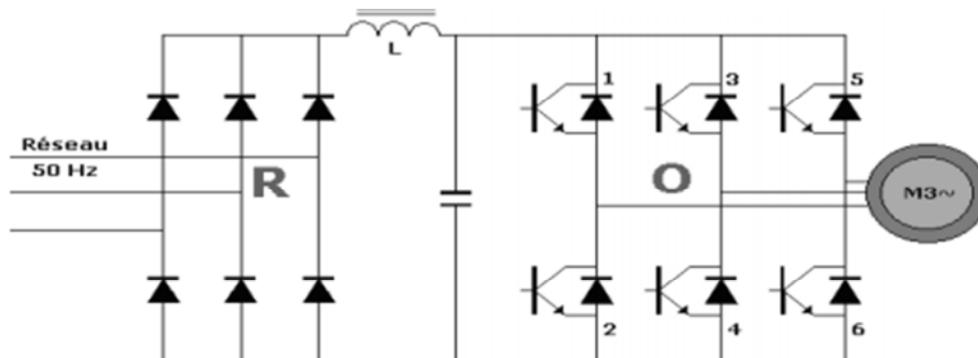


Figure I- 5 : Association redresseur/onduleur de tension

## CHAPITRE I : L'onduleur de tension et ses stratégies de commande

Pour réaliser des onduleurs destinés à alimenter une charge ordinaire d'impédance très variable à partir d'un redresseur, on choisit des onduleurs de tension. Les raisons sont multiples :

- Le fonctionnement de l'onduleur de tension est fort affecté par les imperfections de la source continue, peu par celle de la charge. Pour l'onduleur de courant, c'est l'inverse.
- Les deux onduleurs ne délivrent pas directement une tension de sortie sinusoïdale, donc il faut utiliser un filtre de sortie. Avec l'onduleur de tension, on sait exactement ce qu'on a à filtrer ; la tension à filtrer est imposée par la source continue. Avec l'onduleur de courant, la tension à filtrer dépend de la charge.

### I.6 Modélisation de l'onduleur de tension triphasé :

L'onduleur triphasé, dit deux niveaux, est illustré par son circuit de puissance de la figure qui suit. On doit distinguer d'une part les tensions de branche  $V_{AO}$ ,  $V_{BO}$ ,  $V_{CO}$  mesurées par rapport à la borne négative de la tension continue  $U$ . D'autre part, il y a les tensions de phases  $V_{AN}$ ,  $V_{BN}$  et  $V_{CN}$  mesurées par rapport à un point neutre flottant représentant une charge équilibrée montée en étoile. A partir des tensions simples, on peut tirer facilement les tensions composées  $V_{AB}$ ,  $V_{BC}$  et  $V_{CA}$ .

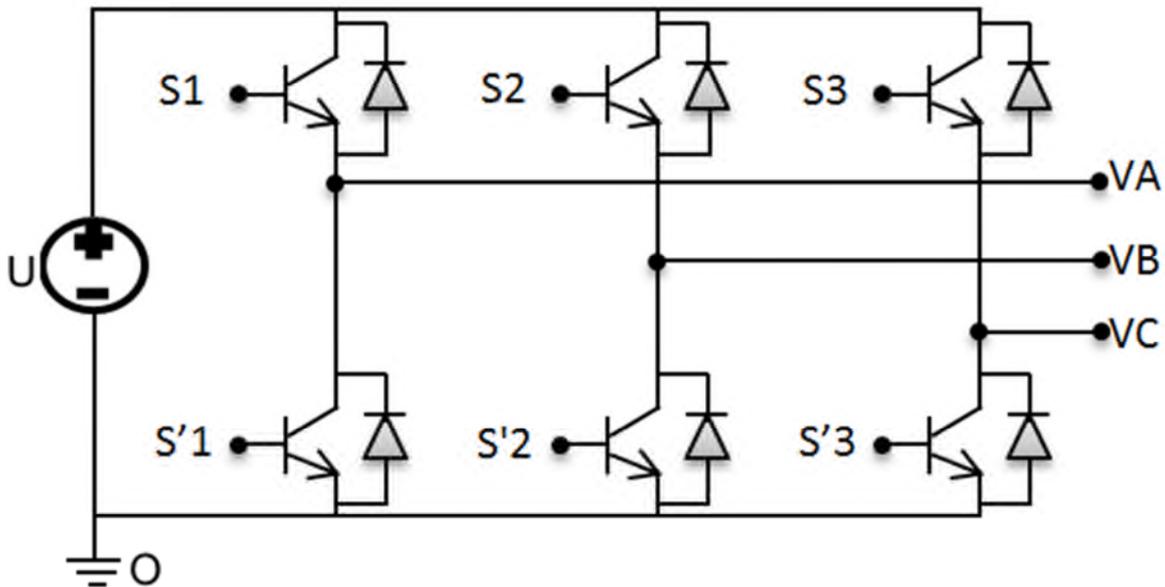


Figure I- 6 : Circuit de fonctionnement de l'onduleur triphasé

Dans le circuit de puissance de l'onduleur triphasé de la figure précédente, il est à noter que les états des interrupteurs d'un même bras sont complémentaires.

En utilisant ces états des interrupteurs, nous pouvons obtenir les tensions de branche de sortie de l'onduleur mesurées par rapport à la borne négative de la tension du côté continu comme suit :

$$\begin{cases} V_{AO} = S_1 \cdot U \\ V_{BO} = S_2 \cdot U \\ V_{CO} = S_3 \cdot U \end{cases} \quad (I.1)$$

## CHAPITRE I : L'onduleur de tension et ses stratégies de commande

Où :  $S_1$ ,  $S_2$  et  $S_3$  désignent les états des interrupteurs des phases A, B et C respectivement.

Les tensions composées sont :

$$\begin{cases} V_{AB} = V_{AO} + V_{OB} = V_{AO} - V_{BO} = (S_1 - S_2) \cdot U \\ V_{BC} = V_{BO} + V_{OC} = V_{BO} - V_{CO} = (S_2 - S_3) \cdot U \\ V_{CA} = V_{CO} + V_{OA} = V_{CO} - V_{AO} = (S_3 - S_1) \cdot U \end{cases} \quad (I.2)$$

On peut écrire l'équation (I.2) sous la forme matricielle suivante :

$$\begin{pmatrix} V_{AB} \\ V_{BC} \\ V_{CA} \end{pmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} \quad (I.3)$$

Les tensions simples sont données par l'expression suivante :

$$\begin{cases} V_{AN} = \frac{2}{3}V_{AO} - \frac{1}{3}(V_{BO} + V_{CO}) \\ V_{BN} = \frac{2}{3}V_{BO} - \frac{1}{3}(V_{AO} + V_{CO}) \\ V_{CN} = \frac{2}{3}V_{CO} - \frac{1}{3}(V_{AO} + V_{BO}) \end{cases} \quad (I.4)$$

Comme elles peuvent être exprimées sous la forme matricielle suivante :

$$\begin{pmatrix} V_{AN} \\ V_{BN} \\ V_{CN} \end{pmatrix} = \frac{U}{3} \cdot \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} \quad (I.5)$$

### I.7 Techniques de commande de l'onduleur de tension [6] :

Les onduleurs de tension peuvent être pilotés suivants plusieurs stratégies. A faible fréquence, ils sont pilotés en pleine onde, le signal de commande sera à la fréquence de la tension désiré à la sortie, et la source continu doit être réglable. A fréquence élevée, ils sont pilotés en modulation de largeur d'impulsion. Cette dernière stratégie permet de régler à la fois l'amplitude et la fréquence en gardent la source continue constante.

A fine de produire une tension de sortie proche de la sinusoïde dans les onduleurs de tension, différentes stratégies de commande ont été proposées par différents auteurs. Parmi ces différentes stratégies, nous allons présenter :

- Commande en pleine onde.
- Commande sinus triangle.

#### I.7.1 Commande en pleine onde [6] :

Pour illustrer le fonctionnement de l'onduleur, il est intéressant d'étudier le comportement en pleine onde (180°). Ce fonctionnement nous permet de mieux comprendre

## CHAPITRE I : L'onduleur de tension et ses stratégies de commande

les autres stratégies de commandes. Cette commande est appelée aussi la commande à six étapes. Dans ce type de commande chaque transistor conduit pendant ( $180^\circ$ ).

Il existe six modes de fonctionnement dans chaque cycle, et chaque mode dure ( $60^\circ$ ). Les transistors sont numérotés d'une manière qu'ils reçoivent les séquences de déplacement l'un après l'autre à un angle de ( $60^\circ$ ).

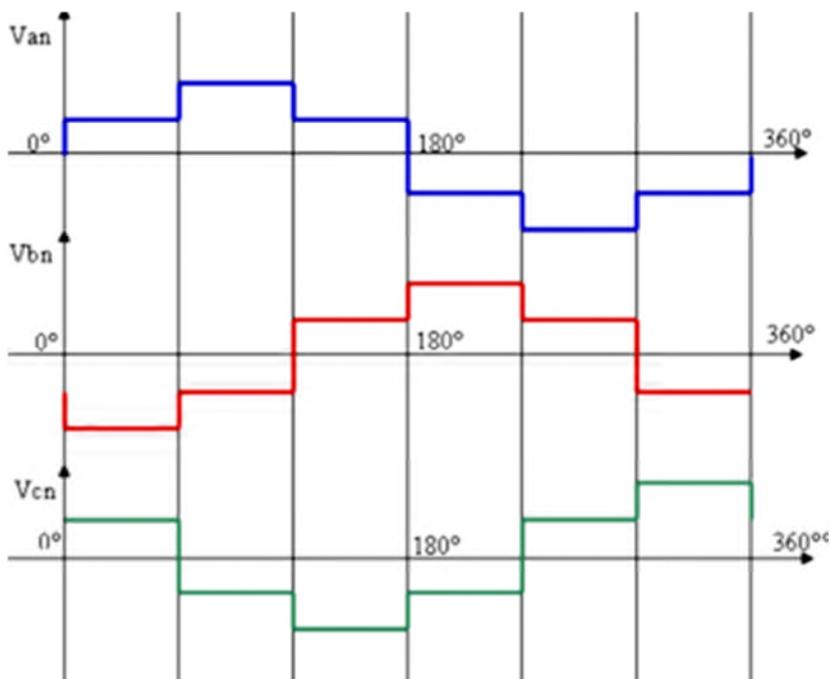
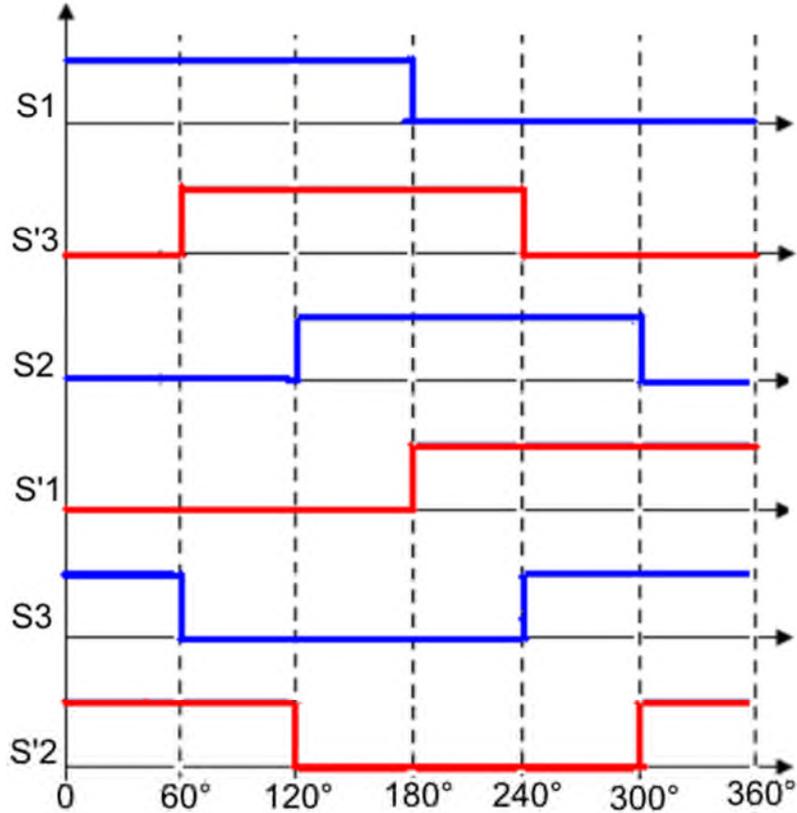


Figure I- 8 : Les tensions de sortie de l'onduleur à commande pleine onde

I.7.2 Modulation de la largeur d'impulsion avec la technique « sinus-triangle » [6] :

La modulation de largeur d'impulsion avec la technique sinus-triangle est réalisée par la comparaison d'une onde modulante basse fréquence (tension de référence) de forme sinusoïdale avec une onde porteuse haute fréquence de forme triangulaire. En triphasé, on utilise trois signaux de référence de même forme et de même fréquence mais décaler entre eux par  $120^\circ$ .

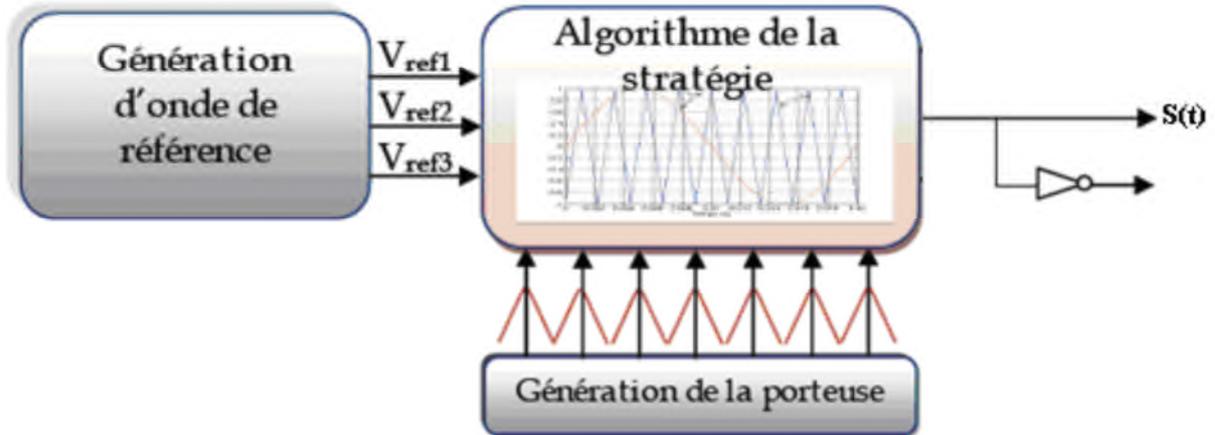


Figure I- 9 : Principe général de la technique sinus-triangle

Le principe de cette technique consiste à comparer un signal de référence (la modulante) à une porteuse comme le montre la figure I-10. Le signal de référence représente l'image de la sinusoïde qu'on cherche à produire à la sortie de l'onduleur. Ce signal est modulable en amplitude et en fréquence. Quant à la porteuse, elle définit la cadence de la commutation des interrupteurs statiques de l'onduleur, c'est un signal de haute fréquence par rapport au signal de référence. D'où, les instants de commutation sont déterminés par les points d'intersections entre la porteuse et la modulante ; alors que la fréquence de commutation des interrupteurs est fixée par la porteuse.

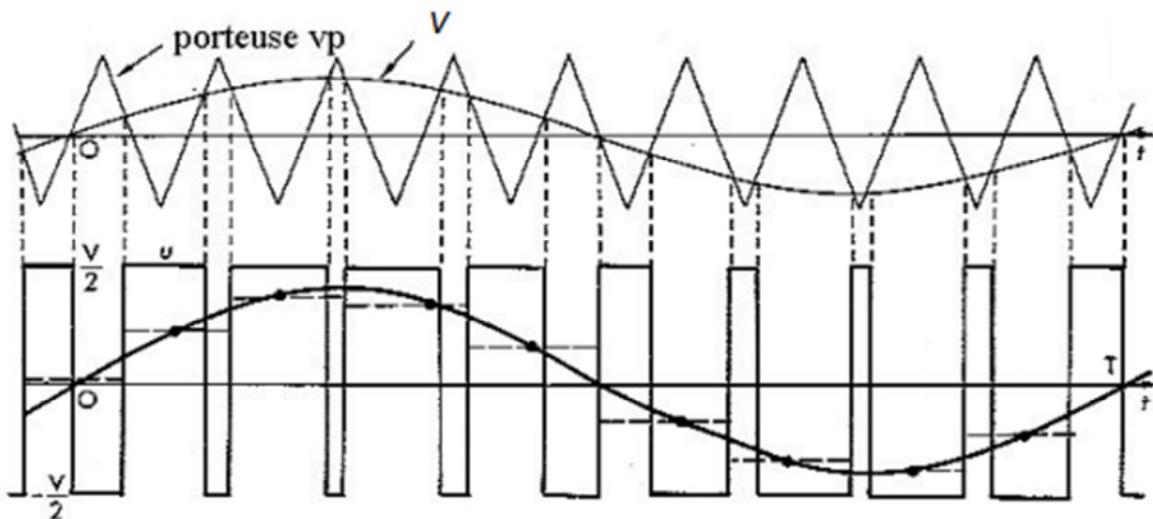


Figure I- 10 : Principe de la MLI sinusoidale [7]

Malgré qu'on ait l'avantage de variation de l'amplitude et de la fréquence de la tension de sortie de l'onduleur, cette tension n'est pas purement sinusoïdale ; elle comporte des

harmoniques. Ces derniers sont les seules responsables des parasites qui engendrent des pertes supplémentaires.

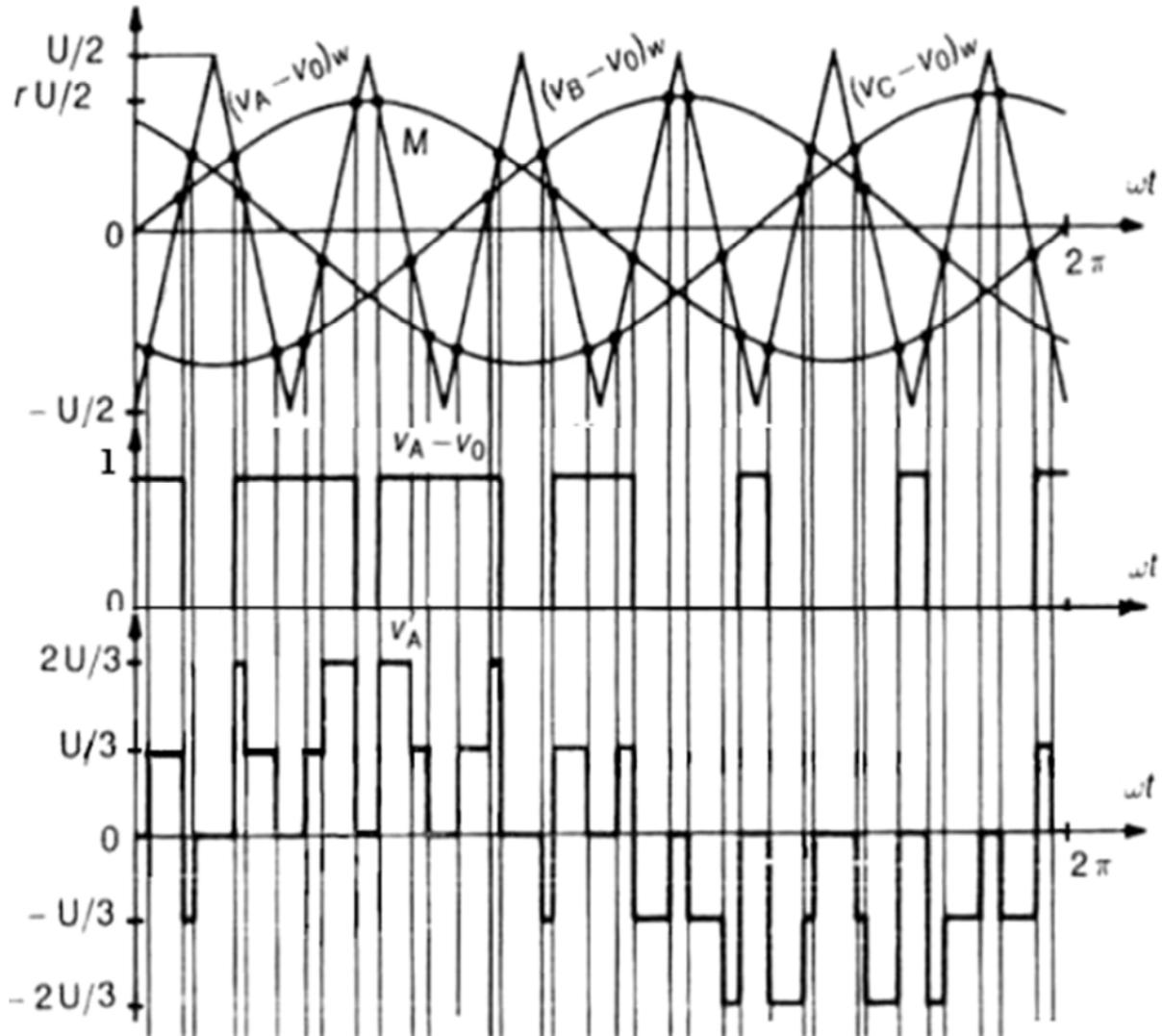


Figure I- 11 : La création des commandes et des tensions de sortie triphasées

Deux paramètres caractérisent principalement la modulation de la largeur d'impulsion ; à savoir l'indice et le taux de modulation.

**L'indice de modulation « m » :** est l'image du rapport de fréquence de la porteuse «  $f_p$  » sur la fréquence de référence «  $f$  ».

$$m = \frac{f_p}{f} = \frac{T_{ref}}{T_p} \quad (I.6)$$

**Le taux de modulation « r » :** ou le coefficient de réglage est l'image du rapport des amplitudes de la tension de référence «  $V_{ref-max}$  » sur la tension de la porteuse «  $V_{p-max}$  ».

$$r = \frac{V_{ref-max}}{V_{p-max}} \quad (I.7)$$

**I.8 Conclusion :**

Dans ce chapitre, nous avons présenté une étude sommaire sur les onduleurs. On a commencé par quelques définitions en montrant la différence entre la structure autonome et non autonome de l'onduleur. Ensuite, on a donné quelques détails sur les techniques de choix des composants en termes de rapidité et de puissance. Par la suite, nous avons montré les deux architectures des onduleurs : onduleur de tension et onduleur de courant. Comme les stratégies de commande visées basent plus sur la tension, nous avons consacré notre étude à l'étude de l'onduleur de tension. Parmi les techniques de commande existantes, nous avons présenté seulement deux techniques qu'on a jugé utile de les présenter ; à savoir la commande en pleine onde vue sa simplicité, et la commande MLI vue son usage général dans la plupart des travaux de recherches.

# CHAPITRE II :

## *Le Microcontrôleur*

Le microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires (mémoire morte pour le programme, mémoire vive pour les données), unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

Par rapport à des systèmes électroniques à base de microprocesseurs et autres composants séparés, les microcontrôleurs permettent de diminuer la taille, la consommation électrique et le coût des produits. Ils ont ainsi permis de démocratiser l'utilisation de l'informatique dans un grand nombre de produits et de procédés.

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués, comme les contrôleurs des moteurs automobiles, les télécommandes, les appareils de bureau, l'électroménager, les jouets, la téléphonie mobile, ...

Les points forts des microcontrôleurs sont nombreux et bien réels. Il suffit, pour s'en persuader, d'examiner la spectaculaire évolution de l'offre des fabricants de circuits intégrés en ce domaine depuis quelque année.

Tout d'abord, un microcontrôleur intègre dans un seul et même boîtier ce qui avant nécessitait une dizaine d'éléments séparés. Cette intégration a aussi comme conséquence immédiate de simplifier le tracé du circuit imprimé, puisqu'il n'est plus nécessaire de véhiculer des bus d'adresse et de données d'un composant à autre.

Aussi le microcontrôleur permet :

- La diminution de l'encombrement du matériel et du circuit imprimé,
- La simplification du tracé du circuit imprimé (plus besoin de tracer de bus !),
- L'augmentation de la fiabilité du système,
- La diminution du nombre de composants utilisés,
- La diminution de la consommation,

Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux :

- Moins cher que les composants qu'il remplace,
- Diminution des coûts de main d'œuvre (conception et montage),
- Environnement de programmation et de simulation évolués.

Ils permettent la gestion des cartes et se caractérisent principalement par :

- De nombreux périphériques d'entrée/sortie
- Une mémoire de programme
- Une mémoire vive (en général de type SRAM)
- Eventuellement une mémoire EEPROM destinée à la sauvegarde par programme de données à la coupure de l'alimentation,
- Un processeur 8 ou 16 bits,
- Faible consommation électrique.

## II.1 Les microcontrôleurs de la firme MICROCHIP :

Un PIC est un microcontrôleur de la firme MICROCHIP. C'est une unité de traitement d'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de faciliter l'interfaçage avec le monde extérieur sans nécessiter l'ajout de composants externes.

Les PICs sont des composants RISC (Reduced Instructions Set Computing) ou encore composant à jeu d'instructions réduit. L'avantage est que plus le nombre d'instructions est réduit, le décodage deviendra plus facile et plus rapide, et le composant fonctionne plus vite [8].

La famille des PICs est subdivisée en trois grandes familles :

- Base-line : une famille qui utilise des mots d'instructions de 12 bits.
- Mid-range : une famille qui utilise des mots d'instructions de 14 bits (dont font partie les PIC16F84, PIC16F876 et PIC16F877).
- High-end : une famille qui utilise des mots de 16 bits (les PIC18Fxxx).

Parmi ces différentes familles, les microcontrôleurs PIC16F877 et PIC18F4550 vont représenter le cerveau de notre présente étude.

## II.2 Etude du microcontrôleur PIC16F877 :

Le microcontrôleur PIC16F877 est un circuit intégré de type CMOS. Parmi ses boîtiers, on trouve le boîtier DIL40 (Dual In Line de 2x20 pattes). Chacune de ces broches lui est associée à une ou plusieurs fonctions. En effet, ces dernières peuvent jouer plusieurs rôles (entrée, sortie) tout dépend de leurs configurations qui s'effectue lors de la programmation du microcontrôleur.

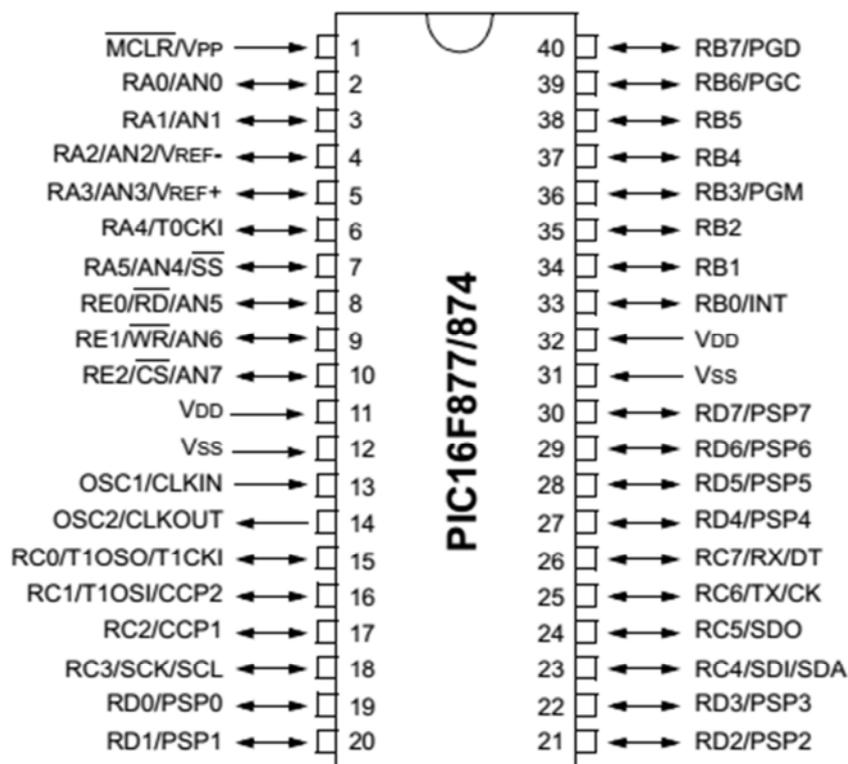


Figure II- 1 : Le brochage du microcontrôleur PIC16F877 (Boîtier DIL40) [9]

## CHAPITRE II : Le Microcontrôleur

Le boîtier précédent du microcontrôleur PIC16F877 comprend 40 broches : 33 broches d'entrées/sorties, 4 broches d'alimentation, 2 broches pour l'oscillateur et une broche (MCLR) pour la gestion de fonctionnement Run/Stop/Prog.

La présence de 4 broches d'alimentation (2 connections « VDD » et 2 connections « VSS ») s'explique par la raison de dissipation thermique. Le constructeur a donc décidé de répartir les courants. Les courants véhiculés dans le microcontrôleur sont loin d'être négligeables du fait des nombreuses entrées/sorties disponibles.

Les broches OSC1 et OSC2 ou CLKIN et CLOUT permettent de faire fonctionner l'horloge du microcontrôleur jusqu'à une fréquence de 20 MHz. Elles peuvent être branchées avec un quartz, un résonateur céramique, ou un circuit RC.

Les broches d'entrée/sortie du microcontrôleur PIC16F877 sont réparti sur 5 ports :

- Port A : 6 broches d'entrée/sortie numérotées de RA0 à RA5
- Port B : 8 broches d'entrée/sortie numérotées de RB0 à RB7
- Port C : 8 broches d'entrée/sortie numérotées de RC0 à RC7
- Port D : 8 broches d'entrée/sortie numérotées de RD0 à RD7
- Port E : 3 broches d'entrée/sortie numérotées de RE0 à RE2

Lors de la programmation, la broche MCLR (Master Clear) doit être portée à un niveau compris entre 12 V et 14 V. La programmation du microcontrôleur commence en appliquant un signal d'horloge sur la broche RB6 (broche 39) et les informations binaires transitent en série sur la broche RB7 (broche 40). Chacune des informations transitant sur la broche RB7 est validée à la retombée du signal d'horloge sur la broche RB6 [8].

La même broche MCLR pourra être exploiter pour initialiser le microcontrôleur à l'aide d'un circuit externe. Un niveau bas sur l'entrée MCLR entraîne un arrêt de fonctionnement alors qu'un niveau de 5 V remet le microcontrôleur en marche. D'une façon générale, un bouton poussoir est souvent rajouté afin qu'une réinitialisation manuelle soit possible. Lorsque le signal "RESET" est activé, tous les registres sont initialisés et le compteur programme se place à une adresse spécifique appelée "Vecteur de RESET".

En résumé, le microcontrôleur PIC16F877 se caractérise par :

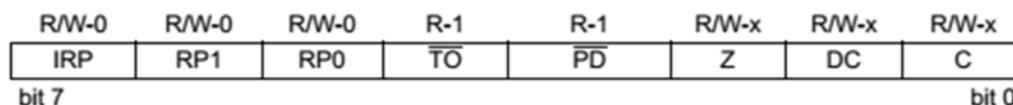
- Une tension d'alimentation entre 4.0 V et 5.5 V
- Une fréquence de fonctionnement jusqu'à 20 MHz,
- Une mémoire vive de 368 octets,
- Une mémoire EEPROM de 256 octets,
- Une mémoire morte de type FLASH de 8 Kmots (1 mot = 14 bits),
- Un chien de garde WDT pour une remise à zéro automatique,
- 33 broches d'entrées/sorties,
- Courant de sortie de chaque broche jusqu'à 20 mA
- 3 Temporisateurs : TIMER0 (8 bits), TIMER1 (16 bits avec possibilité d'utiliser une horloge externe) et TIMER2 (8 bits),
- 2 entrées CCP : Capture, Comparaison et PWM (modulation de largeur d'impulsions),
- Convertisseur analogique numérique 10 bits avec 8 entrées multiplexées.
- Une interface de communication série asynchrone et synchrone (USART/SCI),
- Une interface de communication série synchrone (SSP/SPI et I2C)





Les instructions agissent sur les indicateurs d'état comme suit : (Z pour ZERO) passe à 1 quand le résultat d'une instruction est nul, (C pour CARY) passe à 1 quand l'opération a généré une retenue, (DC pour DIGIT CARY) passe à 1 quand les 4 bits du poids faible génèrent une retenue.

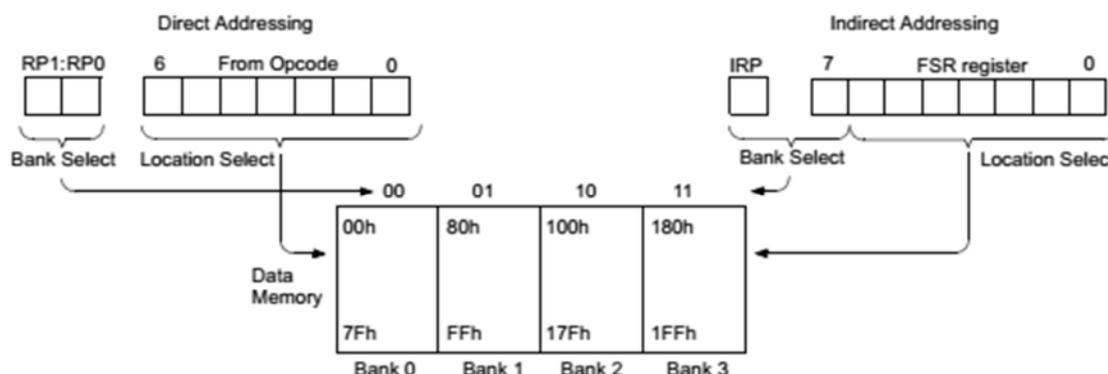
Le registre STATUS permet aussi.



**Figure II- 4** : Composition du registre d'état STATUS d'un microcontrôleur PIC16F877 [9]

### II.4.1 Adressage direct à la mémoire RAM :

Avec ce mode d'adressage, on précise dans l'instruction la valeur de l'adresse à laquelle on veut accéder. Le jeu d'instruction du microcontrôleur ne permet pas l'adressage des registres de la RAM que sur 7 bits (128 registres), l'espace mémoire est alors décomposé en 4 banques de 128 registres comme le montre la figure II.2 (Bank 0 : 00-7F, Bank 1 : 80-FF, Bank 2 : 100-17F, Bank 3 : 180-1FF). Pour accéder à 512 registres (512 octets), il faut 9 bits d'adresse ; le microcontrôleur complète les 7 bits par 2 bits situés dans le registre STATUS. Ces bits sont appelés RP0 et RP1 et doivent être positionnés correctement avant toute instruction qui accède à la RAM par l'adressage direct. [RP1, RP0] = « 00 » permet l'accès à la Bank0, « 01 » : permet l'accès à la Bank1, « 10 » : permet l'accès à la Bank2, « 11 » : permet l'accès à la Bank3.



**Figure II- 5** : Adressage direct et indirect vers la RAM du microcontrôleur PIC16F877 [9]

### II.4.2 Adressage indirect à la mémoire RAM :

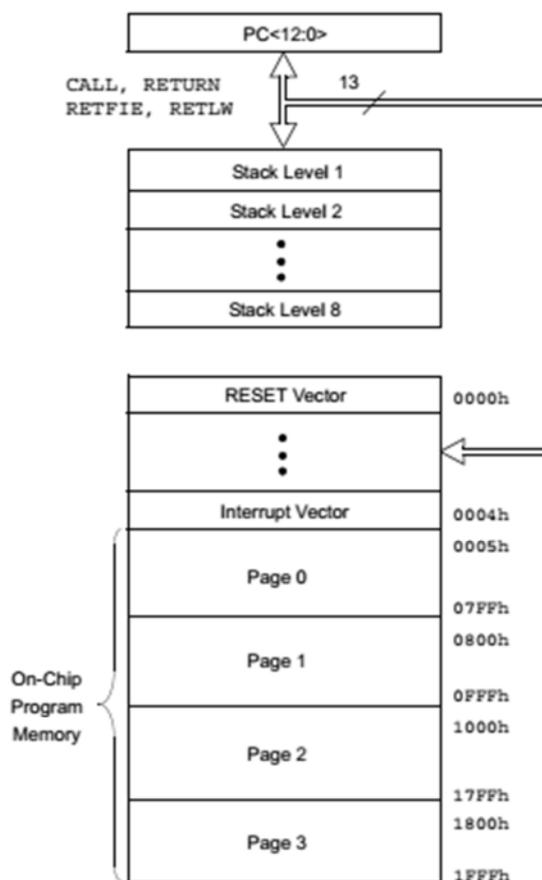
L'adressage indirect est possible en passant par un registre virtuel (INDF) dont l'adresse est contenue dans le registre FSR (File Select Register) et le bit IRP du registre STATUS. Les deux types d'adressage sont présentés par la figure suivante :

## II.5 Mémoire programme :

Le PIC 16F877 possède un compteur de programme de 13 bits ce qui permet l'adressage de 8 Kmots (instructions). Il y a deux adresses réservées :

- Vecteur de remise à zéro (RESET vector : 0000h) : adresse de début du programme
- Vecteur d'interruption (Interrupt vector : 0004h) : adresse de début du sous-programme d'interruption.

Le microcontrôleur PIC16F877 possède aussi une pile (Stack) de 8 niveaux pour la sauvegarde des adresses de retours suite aux appels du sous-programme.



**Figure II- 6 :** Adressage de la mémoire programme et la pile de sauvegarde du microcontrôleur PIC16F877 [9]

## II.6 Etude du microcontrôleur PIC18F4550 :

Le microcontrôleur PIC18F4550 est un circuit intégré de type CMOS. Dans son boîtier DIL40, il est compatible broches à broches avec le microcontrôleur PIC16F877. Chacune de ses broches est associée à une ou plusieurs fonctions de la même manière que la gamme 16F.

A la différence du microcontrôleur PIC16F877, le microcontrôleur PIC18F4550 se caractérise principalement par :

- Une fréquence de fonctionnement en mode PLL jusqu'à 48 MHz obtenue seulement avec le branchement d'un quartz externe de 20 Mhz,
- L'existence d'une horloge interne réglable sur 8 cadences de 31 kHz jusqu'à 8 Mhz,
- Une mémoire RAM de 2048 octets à la place de 368 octets,
- Une mémoire programme de type FLASH de 32 Kiloctets (16 Kx16 bits) au lieu de 14 kiloctets (8 Kx14 bits),
- 16 banques de gestion de la mémoire à la place de 4,
- 20 sources d'interruption à la place de 14,
- 4 temporisateurs (1x8 bits, 3x16 bits) à la place de 3 temporisateurs (2x8 bits, 1x16 bits),
- 13 entrées analogiques à la place de 8,
- Une interface de communication USB en plus des autres interfaces de communication,

- Un multiplicateur matériel 8x8.

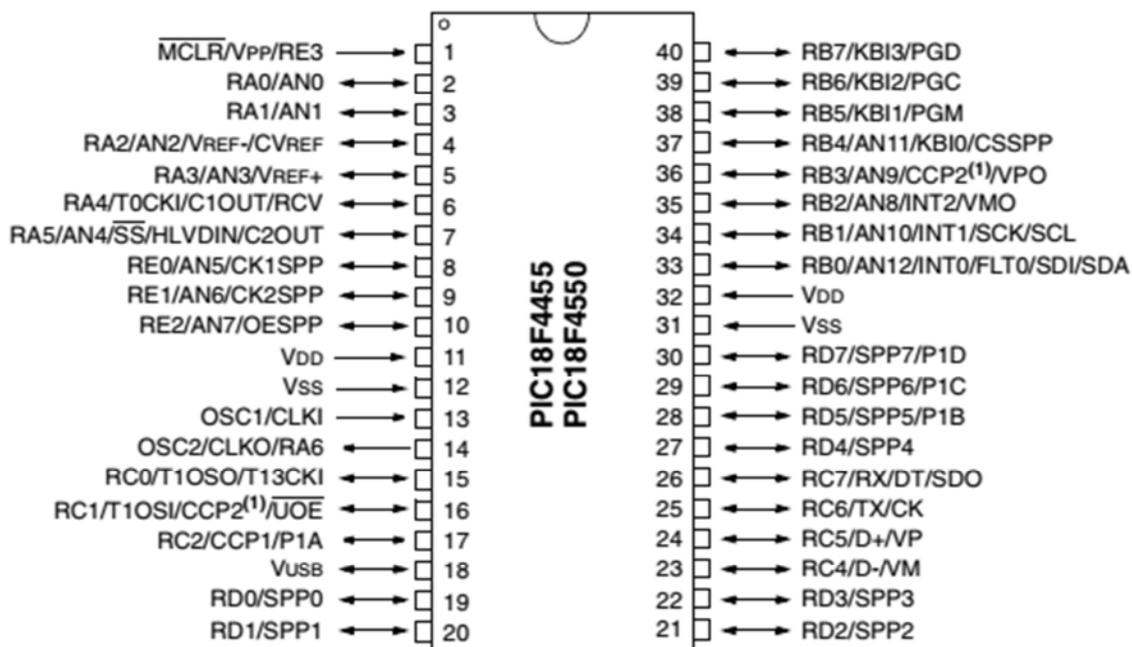


Figure II- 7 : Le brochage du microcontrôleur PIC18F4550 (Boitier DIL40) [10]

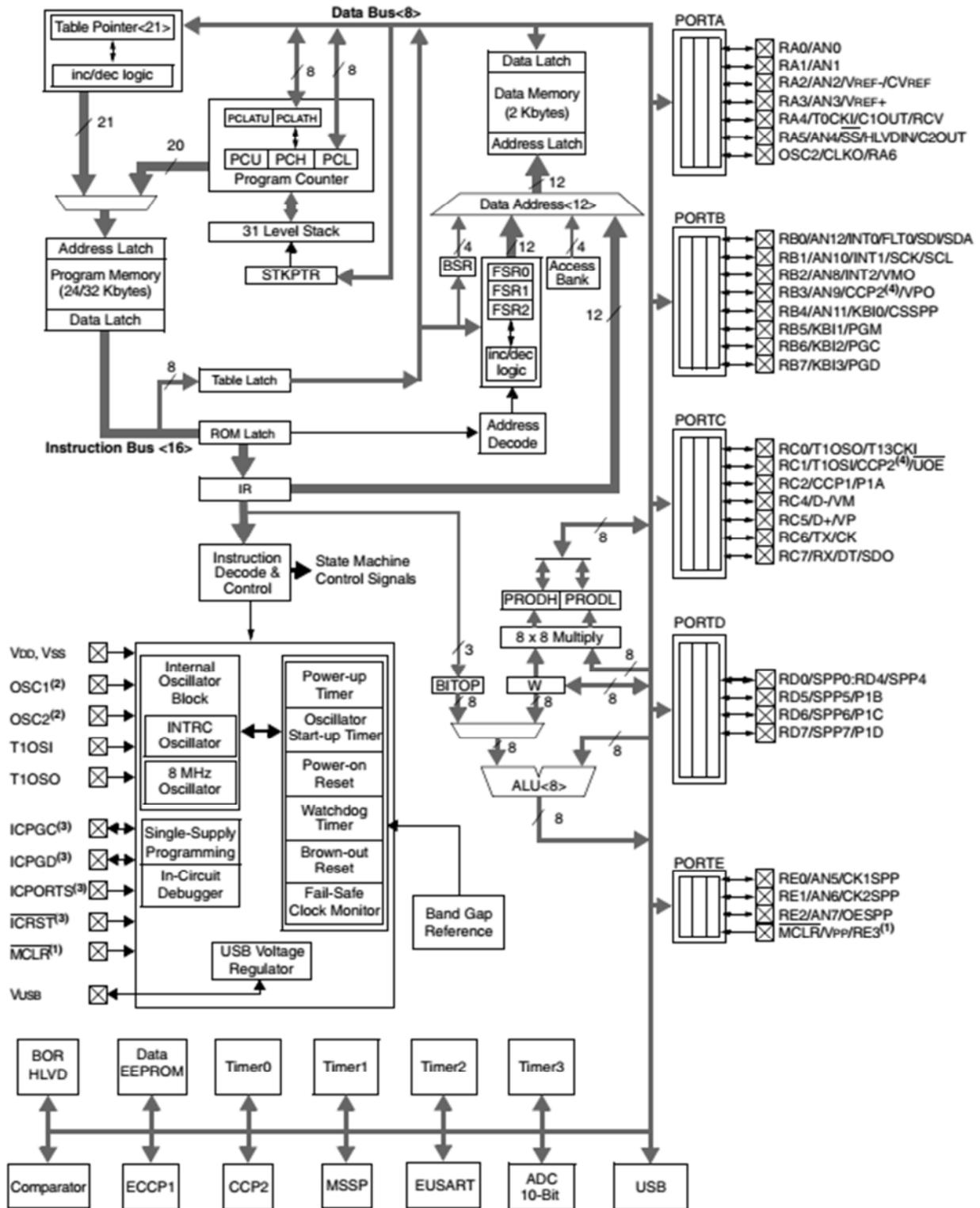
Le nombre de broches d'entrée/sortie du microcontrôleur PIC18F4550 est de 35, alors que le PIC16F877 n'a que 33. La principale différence entre les broches d'entrée/sortie de ces deux microcontrôleurs est les registres de gestion. Le microcontrôleur PIC18F4550 se compose d'un registre « LAT » mais le microcontrôleur PIC16F877 se compose uniquement du registre « PORT » et du registre « TRIS ». Le registre « LAT » est utile pour les opérations Read-Modify-Write sur la valeur que les broches d'entrée/sortie conduisent.

La broche 18 du microcontrôleur PIC18F4550 est réservée pour la fonction USB tandis que la broche 18 de microcontrôleur PIC16F877 à plusieurs fonctions. En raison de cette conception unique, le microcontrôleur PIC18F4550 n'a pas la broche d'entrée/sortie RC3.

## II.7 Architecture interne du microcontrôleur PIC18F4550 :

L'organisation interne du microcontrôleur PIC18F4550 diffère quelque peu de celle à laquelle on a l'habitude avec le microcontrôleur PIC16F877. Cela résulte bien évidemment du fait que ces circuits font partie d'une gamme évoluée.

Un schéma étant bien plus probant qu'un long discours, la figure qui suit présente l'organisation interne du microcontrôleur PIC18F4550.



- Note 1:** RE3 is multiplexed with  $\overline{\text{MCLR}}$  and is only available when the  $\overline{\text{MCLR}}$  Resets are disabled.  
**Note 2:** OSC1/CLKI and OSC2/CLKO are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to Section 2.0 "Oscillator Configurations" for additional information.  
**Note 3:** These pins are only available on 44-pin TQFP packages under certain conditions. Refer to Section 25.9 "Special ICPort Features (Designated Packages Only)" for additional information.  
**Note 4:** RB3 is the alternate pin for CCP2 multiplexing.

Figure II- 8 : Architecture interne d'un microcontrôleur PIC18F4550 [10]

Afin d'être le plus général possible, ce schéma présente quasiment toutes les ressources internes pouvant exister dans un microcontrôleur PIC18. Il est donc bien évident que sa taille peut se voir réduite pour les circuits ne disposant que d'une partie d'entre elles.

Par ailleurs, notez que la majorité des ressources qui doivent faire appel à des liaisons externes (EUSART, USB, convertisseurs analogiques/digitaux, ...) représentées en bas à gauche de la figure, partagent leurs pattes d'entrées/sorties avec celles de certains ports parallèles. Afin de simplifier le tracé de la figure, les liaisons correspondantes n'ont pas été représentées [11].

L'examen fonctionnel de ce schéma peut être conduit de diverses façons et nous partirons quant à nous de la mémoire de programme. Cette dernière est organisée en mots de 16 bits ce qui permet de coder sur un seul mot l'instruction et l'opérande ou son adresse. En effet, malgré cette taille relativement importante, certaines instructions se doivent d'être codées sur 32 bits [11].

Les données, qu'il conviendrait mieux d'appeler des instructions, issues de cette mémoire entrent ensuite dans le registre d'instructions pour être dirigées sur le décodeur d'instructions. Ce dernier agit bien évidemment sur l'unité arithmétique et logique (ALU) ainsi que sur le registre W (Working register) ou registre de travail. Ce dernier est intimement lié à l'ALU qui reçoit par ailleurs, via le bus de données spécifique à cause de l'architecture Harvard, les données issues de la mémoire de données, des entrées/sorties ou des registres des ressources internes [11].

Le compteur ordinal ou PC dispose d'une pile à 31 niveaux apte à le sauvegarder lors des appels de sous-programmes ou lors du traitement des interruptions. Même si on ne le voit pas sur la figure précédente, on remarque également qu'un registre particulier est présent près de l'ALU. Il s'agit du registre STATUS ou registre d'état, qui est commun à tous les microprocesseurs puisqu'il indique, au moyen de bits spécifiques comme a vu, le résultat de certaines instructions exécutées par l'ALU (résultat nul, débordement, ...). Notant aussi, toujours au niveau de l'ALU, la présence d'un multiplieur câblé 8 par 8 qui nous laisse donc augurer de la présence d'une instruction de multiplication, d'exécution particulièrement rapide [11].

On remarque également les registres FSR qui sont propres aux PIC et qui permettent de réaliser un adressage indexé un peu particulier, tandis que le registre BSR est utilisé dans l'adressage direct compte tenu de la pagination de la mémoire de données [11].

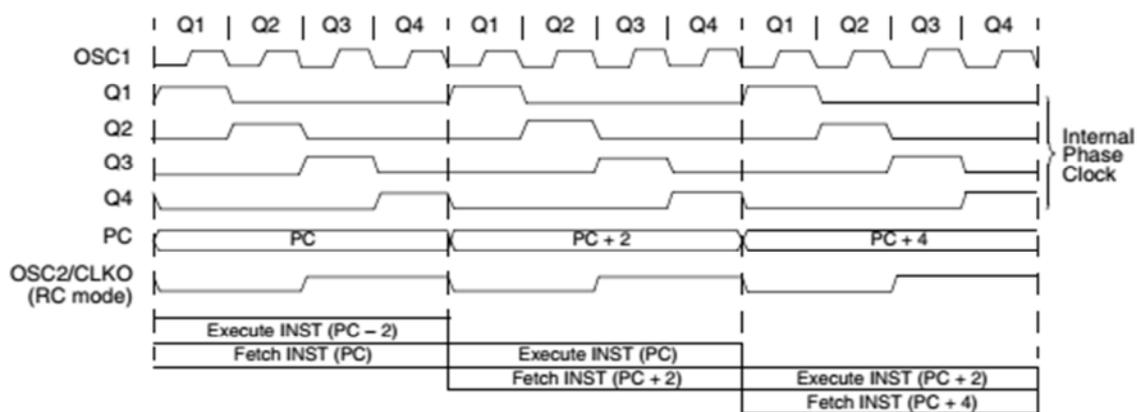


Figure II- 9 : Chronogramme d'exécution des instructions [10, 11]

Au niveau de la circuiterie d'horloge, on retrouve bien évidemment les différentes sources de reset ainsi que les divers temporisateurs. Le fonctionnement de tout cet ensemble est régi par l'horloge système. Comme le montre la figure précédente, le signal délivré par l'oscillateur, quel qu'il soit, est divisé par quatre afin de constituer l'horloge système et de générer en interne quatre cycles différents Q1 à Q4. Le PC est augmenté de deux unités à chaque cycle Q1 alors que les instructions sont recherchées en mémoire de programme et mémorisées dans le registre

d'instructions à chaque cycle Q4. Ce diagramme met clairement en évidence le fait qu'une instruction s'exécute alors que la suivante est en cours de recherche ; ce qui est le propre de la structure pipe-line des circuits à architecture RISC [11].

Cette présentation générale étant faite, on passe maintenant à la découverte de l'organisation détaillée de la mémoire de programme et de la mémoire de travail.

## II.8 Organisation générale de la mémoire [11] :

L'organisation de la mémoire de données des PIC est un peu particulière car, d'une part elle contient tout à la fois les registres des ressources internes et la mémoire de données proprement dite, et d'autre part elle fait appel à un mécanisme de pagination qui complique un peu son mode d'accès lorsque l'on n'est pas bien habitué à cette façon de faire. De nombreuses améliorations ont cependant été apportées par rapport à ce qui existe dans les PIC 10, 12 et 16.

La mémoire de programme quant à elle s'étend linéairement de l'adresse 0000h à l'adresse la plus élevée permise compte tenu de la taille de la ROM ou de l'EEPROM intégrée dans le circuit choisi. Elle n'utilise aucun mécanisme de pagination et s'avère donc d'un accès particulièrement facile.

### II.8.1 Organisation de la mémoire programme [11] :

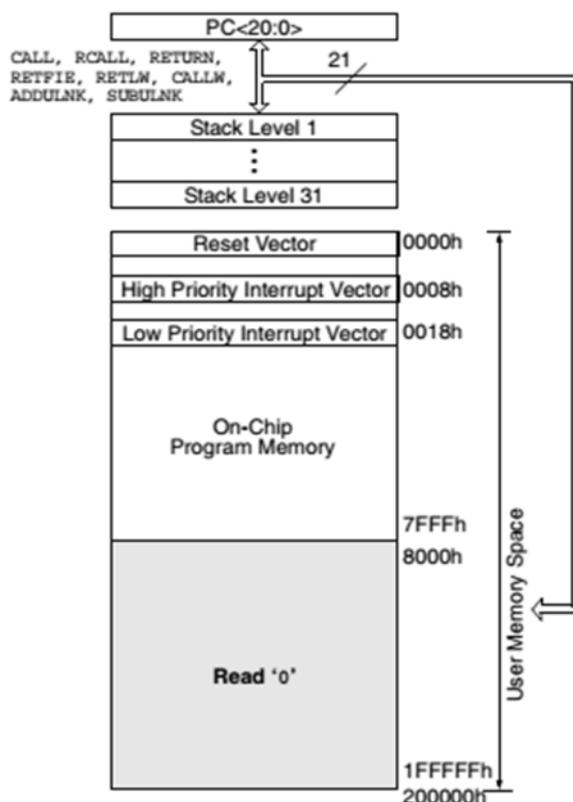


Figure II- 10 : Organisation de la mémoire programme du PIC18F4550 [10, 11]

Comme le montre la figure précédente, la mémoire de programme du microcontrôleur PIC18F4550 forme un bloc continu s'étendant de l'adresse 0000h jusqu'à la valeur maximum permise par la taille physique de la mémoire réellement implantée dans le circuit, dans notre cas 1FFFFFh (2048 octets).

Bien que les mots d'instructions des PIC soient codés sur 16 bits, cette mémoire reste organisée et gérée en octets. Le compteur ordinal ou PC, qui pointe successivement sur les diverses instructions à exécuter, évolue donc de 2 en 2 puisque chaque instruction occupe au minimum 2 octets (4 octets pour les rares instructions codées sur 32 bits). De ce fait, il est donc impossible dans un programme PIC18, de trouver une instruction à une adresse impaire. Si on programme en langage évolué tel que le CCS-PICC, tout ceci nous sera totalement transparent puisqu'il est géré automatiquement par le compilateur.

Comme sur les microcontrôleurs PIC 10, 12 et 16, le vecteur de reset se trouve à l'adresse 0000h de la mémoire de programme, tandis que les vecteurs d'interruptions démarrent à l'adresse 0008h pour celui de plus haute priorité pour s'étendre jusqu'à l'adresse 0018h pour celui de priorité la plus basse.

Avec les microcontrôleurs PIC 10, 12 et 16, le PC de ces processeurs était divisé en trois sous-registres car, comme sa taille était de 13 bits et que les mots de données manipulés par ces PIC n'étaient que des octets, il fallait bien trouver une solution pour le gérer.

Le même problème se pose ici aussi puisque le PC ne fait pas moins de 21 bits de large et que les PIC18 manipulent toujours des mots de données codés sur 8 bits (même si les instructions, elles, sont sur 16 ou 32 bits !).

Le PC des microcontrôleurs PIC18 est donc divisé ou associé à cinq sous-registres appelés :

- **PCL (Program Counter Low)** : qui correspond aux 8 bits de poids faibles du PC et qui est toujours accessible directement en lecture comme en écriture. En outre, comme les instructions sont toujours placées à des adresses paires, le bit de poids le plus faible, ou bit 0, de ce registre est toujours à 0.
- **PCH (Program Counter High)** : qui correspond aux bits 8 à 15 du PC et qui n'est jamais accessible directement, que ce soit en lecture ou en écriture.
- **PCU (Program Counter Upper)** : qui correspond aux bits 16 à 20 du PC et qui n'est jamais accessible directement, que ce soit en lecture ou en écriture.
- **PCLATH (Program Counter LATch High)** : qui est un registre intermédiaire permettant d'accéder aux bits 8 à 15 du PC contenus dans PCH, si nécessaire.
- **PCLATU (Program Counter LATch Upper)** : qui est un autre registre intermédiaire permettant d'accéder aux bits 16 à 20 du PC contenus dans PCU, si nécessaire.

Lorsque le PC évolue, suite à l'exécution d'instructions séquentielles ou d'instructions de saut et de branchement, il fonctionne comme un registre unique de 21 bits de large et on n'a donc en aucun cas à nous préoccuper de sa gestion ou bien de celle des sous-registres que nous venons d'évoquer.

Le dernier élément présent sur la figure II.10 est la pile d'une capacité de 31 mots. Elle dispose de sa propre zone mémoire indépendante tant de la mémoire de programme que de la mémoire de données. Cela se voit d'ailleurs très bien sur le synoptique de la figure II.8.

Cette pile est destinée à mémoriser les adresses de retour lors de l'exécution de sous-programmes ou de programmes d'interruptions. Elle a donc une largeur compatible avec celle du PC, à savoir 21 bits, puisque c'est le contenu de ce dernier qu'elle doit mémoriser.

II.8.2 Organisation de la mémoire de données [11] :

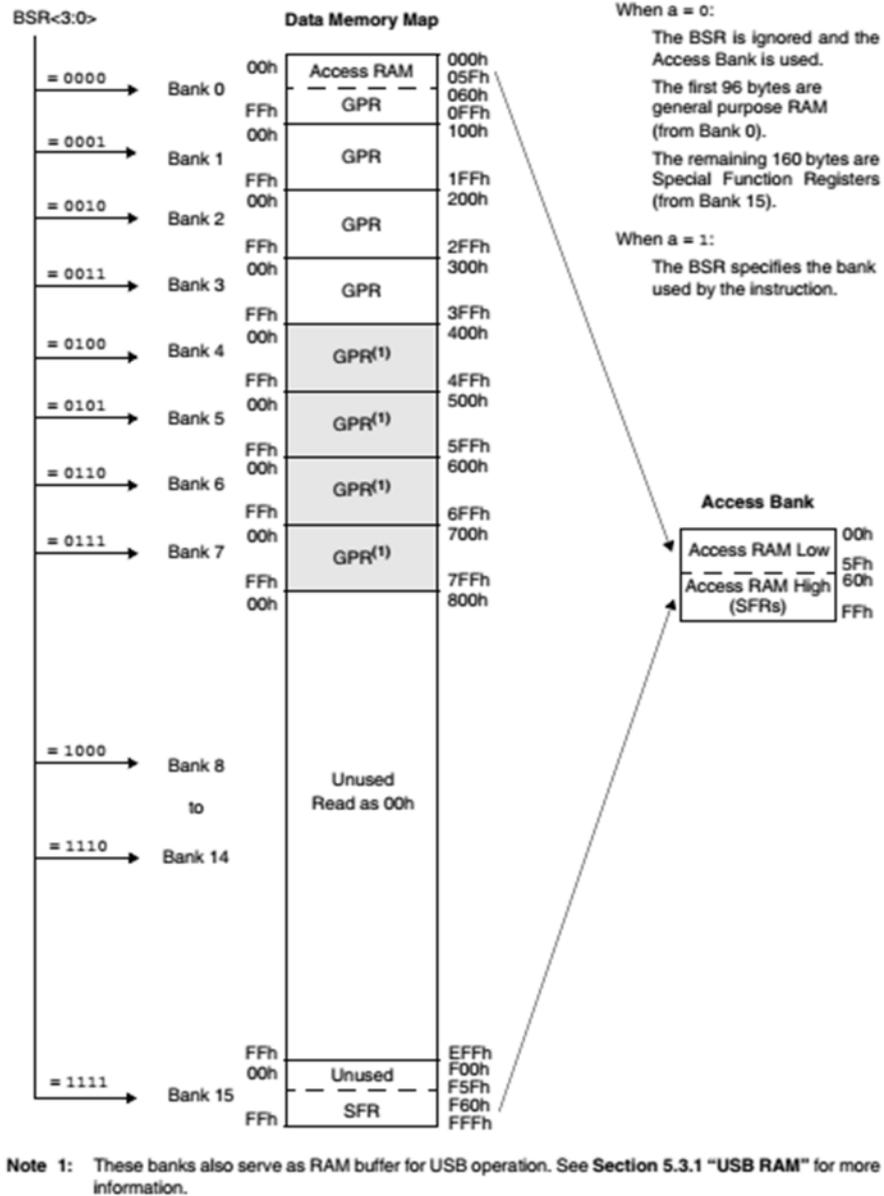
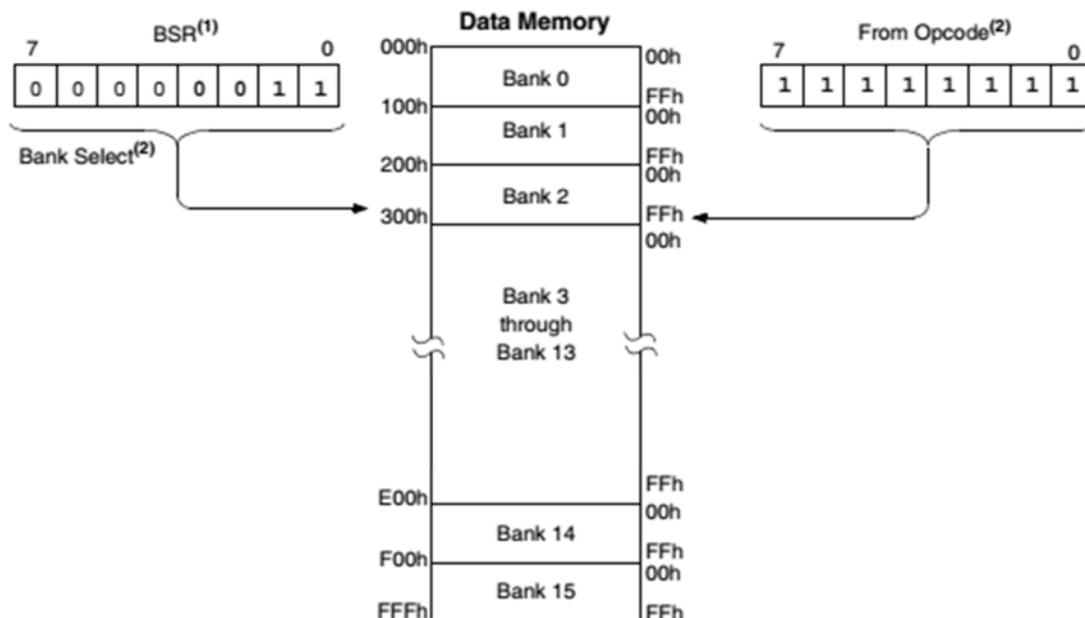


Figure II- 11 : Organisation en pages de la mémoire de données du PIC18F4550 [10, 11]

La mémoire de données des microcontrôleurs PIC18F4550 contient tout à la fois les registres de l'unité centrale, ceux des ressources internes ainsi que des registres à usage général qui constituent en fait la véritable mémoire de données. Elle utilise un mécanisme de pagination plus ou moins hérité de celui des PIC 10, 12 et 16 mais tout de même beaucoup plus souple d'emploi.

Elle est divisée artificiellement en 15 banques d'une taille fixe de 256 octets chacune comme on peut le découvrir à l'examen de la figure précédente. La mémoire vive ou mémoire de travail se trouve située dans les pages les plus basses, c'est-à-dire à partir de l'adresse 000 tandis que les registres de l'unité centrale et des ressources internes se trouvent tous regroupés dans la page la plus haute, c'est-à-dire la page 15. En fait, et pour être tout à fait précis, ces registres se trouvent même tous placés en partie haute de cette page 15 puisqu'ils s'étendent de l'adresse F60 à l'adresse FFE.

Compte tenu de la taille d'une page, qui est de 256 octets, il suffit donc de 8 bits pour adresser n'importe quel élément au sein de cette dernière. Comme le montre la figure qui suit, ces 8 bits font toujours partie intégrante du code de l'instruction.



- Note 1:** The Access RAM bit of the instruction can be used to force an override of the selected bank (BSR<3:0>) to the registers of the Access Bank.
- 2:** The MOVFF instruction embeds the entire 12-bit address in the instruction.

**Figure II- 12 :** Mécanisme de pagination de la mémoire de données du PIC18F4550 [10, 11]

Pour savoir quelle adresse complète sera réellement adressée par telle ou telle instruction, il faut donc connaître également le numéro de page mémoire utilisée, numéro qui est codé dans un registre spécifique appelé BSR (Bank Select Register) visible sur cette même figure.

La concaténation du contenu de ce registre avec l'adresse sur 8 bits contenue au sein de l'instruction donne bien une adresse unique sur 16 bits, capable donc d'adresser plus de mémoire de données.

Cette gestion de la pagination mémoire restant toutefois assez fastidieuse, Microchip a prévu divers mécanismes afin de la simplifier au maximum et de réduire le risque non négligeable d'erreur qu'elle implique.

La figure II.11 montre le principal d'entre eux avec une « page virtuelle » qui se nomme « Access Bank », ce que l'on peut traduire par la page d'accès. Cette page est toujours accessible immédiatement, quel que soit le contenu de BSR et donc quelle que soit la page normalement pointée grâce à ce registre.

Cette page d'accès présente une autre particularité, elle est divisée en deux parties : sa partie basse, de l'adresse 00 à l'adresse 5F, est une copie de la zone de RAM de mêmes adresses de la page 0 disponible pour stocker les variables de notre programme, tandis que sa partie haute, des adresses 60 à FF, n'est autre que la zone des registres internes également présente en page 15.

De ce fait, on peut, sans aucun changement de page et donc sans avoir à manipuler le registre BSR, disposer de 96 octets pour ranger nos données et variables et accéder à tous les registres internes du PIC 18 au travers de cette page d'accès.

Si ces 96 octets ne nous suffisent pas, rien ne nous interdit d'initialiser une fois pour toutes le registre BSR à 01 au début de notre programme, le faisant ainsi pointer sur la page 1. On a donc accès ainsi aux 256 octets contenus en page 1 auxquels s'ajoutent les 96 octets de la page d'accès, c'est-à-dire du début de la page 0, soit un total de 352 octets, augmenté de tous les registres internes en accès direct sans avoir à réaliser le moindre changement de page.

### II.8.3 Mémoire EEPROM de données :

Tous les PIC de la famille PIC 18 disposent de mémoire EEPROM (Electrically Erasable Programmable Read Only Memory) de données. Ce type de mémoire, qui se comporte presque comme de la mémoire vive mais avec la particularité de conserver son contenu pendant une durée indéterminée, même en l'absence d'alimentation, est extrêmement intéressant pour toutes les applications devant mémoriser des paramètres programmés par l'utilisateur d'une mise sous tension à une autre.

Cette mémoire EEPROM de données ne fait pas partie de l'espace adressable normal des PIC. Elle n'interfère donc en aucune manière, ni avec la mémoire de programme, ni avec les registres ou mémoire de données classique.

La taille de cette mémoire EEPROM de données du microcontrôleur PIC18F4550 est de 256 octets. Son adresse la plus basse part de 00 et son adresse la plus haute est FF.

Elle n'est accessible en lecture et en écriture qu'au travers de deux registres :

- - un registre EEDATA pour les données,
- - un registre EEADR pour les adresses.

Deux registres de contrôle, EECON1 et EECON2, permettent quant à eux de définir le mode de fonctionnement de cette mémoire mais, alors que le registre EECON1 est bien réel, le registre EECON2 n'est pas un registre physique et n'est utilisé d'une façon particulière que lors de l'écriture dans la mémoire.

Cette EEPROM n'utilise aucune source de haute tension externe et fonctionne dans toute la plage de tension d'alimentation autorisée pour le PIC18F4550. Sa durée de rétention des informations est annoncée par Microchip comme étant supérieure à 40 ans, soit largement plus que la durée de vie de la majorité des applications actuelles ... Elle est contrôlée par un temporisateur interne qui lui est propre, nous déchargeant ainsi de toute mesure de temps par logiciel et n'empruntant aucune ressource aux timers normaux du circuit.

Le fait de vouloir écrire à une adresse déjà occupée efface automatiquement le contenu qui s'y trouve avant d'y placer la nouvelle donnée. De ce fait, aucune commande d'effacement spécifique n'est prévue. Il n'existe pas non plus de procédure d'initialisation suite à une mise sous tension et le contenu de cette mémoire doit donc être considéré comme quelconque tant qu'il n'a pas été défini par notre programme.

## **II.9 Conclusion :**

Dans ce chapitre, nous avons fait une comparaison entre les microcontrôleurs PIC16F877 et les PIC18F4550. On a commencé par l'étude du microcontrôleur PIC16F877 tout en montrant son architecture interne. Ensuite, on a donné quelques détails sur les techniques d'accès à sa mémoire RAM, en plus de la structure de sa mémoire programme. Par la suite, nous avons étudié le microcontrôleur PIC18F4550 tout en faisant la différence avec le microcontrôleur PIC16F877.

On considère maintenant que les détails présentés nous permettent de répondre à la question qui se pose dans le chapitre suivant ; comment ce fait-t-il le même programme fonctionne sur un microcontrôleur PIC18F4550 et ne fonctionne pas sur un microcontrôleur PIC16F877 ? Cette étude sur la gestion des mémoires des microcontrôleurs PIC16F877 et PIC18F4550 est assez suffisante pour aborder le chapitre suivant.

# CHAPITRE III :

*Implantation d'une commande PWM  
sur un microcontrôleur*

### **CHAPITRE III : Implantation d'une commande PWM sur un microcontrôleur**

La conversion d'énergie du continu en alternative à base d'onduleur est utilisée dans l'entraînement électrique, les alimentations sans interruption et dans beaucoup d'autres domaines industriels où l'objectif est de produire une tension de sortie sinusoïdale contrôlable en amplitude et en fréquence.

La technique la plus employée dans la commande des onduleurs est la technique de la modulation de la largeur d'impulsions PWM (Pulse Width Modulation).

Les signaux de commande PWM des onduleurs diffèrent complètement des signaux générés par le module PWM du microcontrôleur. Ces impulsions sont d'amplitude égale (0 ou 1) mais de durées variables. Les impulsions du module PWM du microcontrôleur peuvent être appliquées sur un hacheur et non pas un onduleur.

Dans le présent chapitre, on va étudier la technique d'implantation d'une commande PWM pour onduleur sur un microcontrôleur.

#### **III.1 Schéma de conception d'un l'onduleur triphasé :**

Un onduleur triphasé est constitué principalement de trois parties :

- Génération des signaux de commande,
- Partie isolation et drivers,
- Partie puissance.

Des optocoupleurs de type 4N25 sont ultra rapides et peuvent supporter des hautes fréquences jusqu'à 4 kHz ; une fréquence suffisante pour une commande MLI.

Dans la plupart des convertisseurs de puissance, les composants de puissance sont pilotés par des Drivers. Un driver est un circuit intégré permettant le pilotage en haute fréquence 2 MOSFET ou 2 IGBT sous des tensions pouvant atteindre 600 V, en garantissant l'isolation électrique de chaque étage.

Le driver IR2113 le plus utilisé est un circuit intégré spécialisé fabriqué par « International Rectifier ». Trois drivers de ce type peuvent piloter les six MOSFETS de l'onduleur triphasé.

Les schémas qui suivent présentent le montage d'un onduleur triphasé. L'isolation entre la partie commande et la partie puissance est garantie par l'intermédiaire des optocoupleurs. Ce montage assure l'isolation électrique de l'ensemble commande-puissance. Il permet aussi l'amplification de la tension de blocage et d'amorçage des MOSFET de puissance (de 5 V vers 15 V).



### III.2 La commande MLI de type sinus triangle :

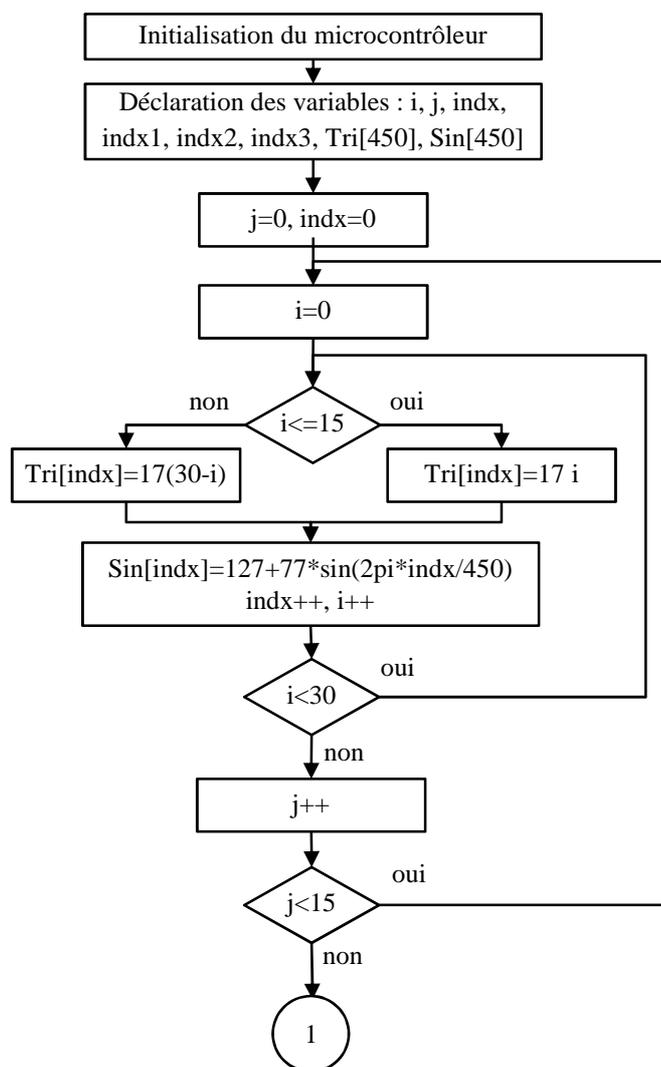
Le principe de cette méthode est de créer deux signaux, un signal sinusoïdal d'amplitude et de fréquence variable appelée « référence » et un signal triangulaire de fréquence très élevée appelé « porteuse ». Ces deux signaux sont comparés, le résultat de la comparaison sert à commander l'ouverture et la fermeture des interrupteurs du circuit de puissance. Cette méthode qui permet d'obtenir une fondamentale variable en amplitude et en fréquence est très utilisée dans les variateurs de vitesse des moteurs asynchrones.

L'étude du spectre de sortie montre que l'on obtient un fondamental dont la fréquence et l'amplitude dépendent de celle de la référence, et des harmoniques d'amplitudes importantes avec des fréquences très élevées proche de celle de la porteuse. Le filtrage est donc très facile.

Pour créer les signaux de commande MLI d'un onduleur de tension triphasé, il nous faut trois signaux sinusoïdaux décalés par 120°, plus d'un signal triangulaire de fréquence très élevé.

Le but de ce travail est de générer un signal MLI de fréquence variable avec un microcontrôleur PIC16F877 afin de commander les bras de l'onduleur.

### III.3 Exemple 1 de génération d'un un signal MLI :



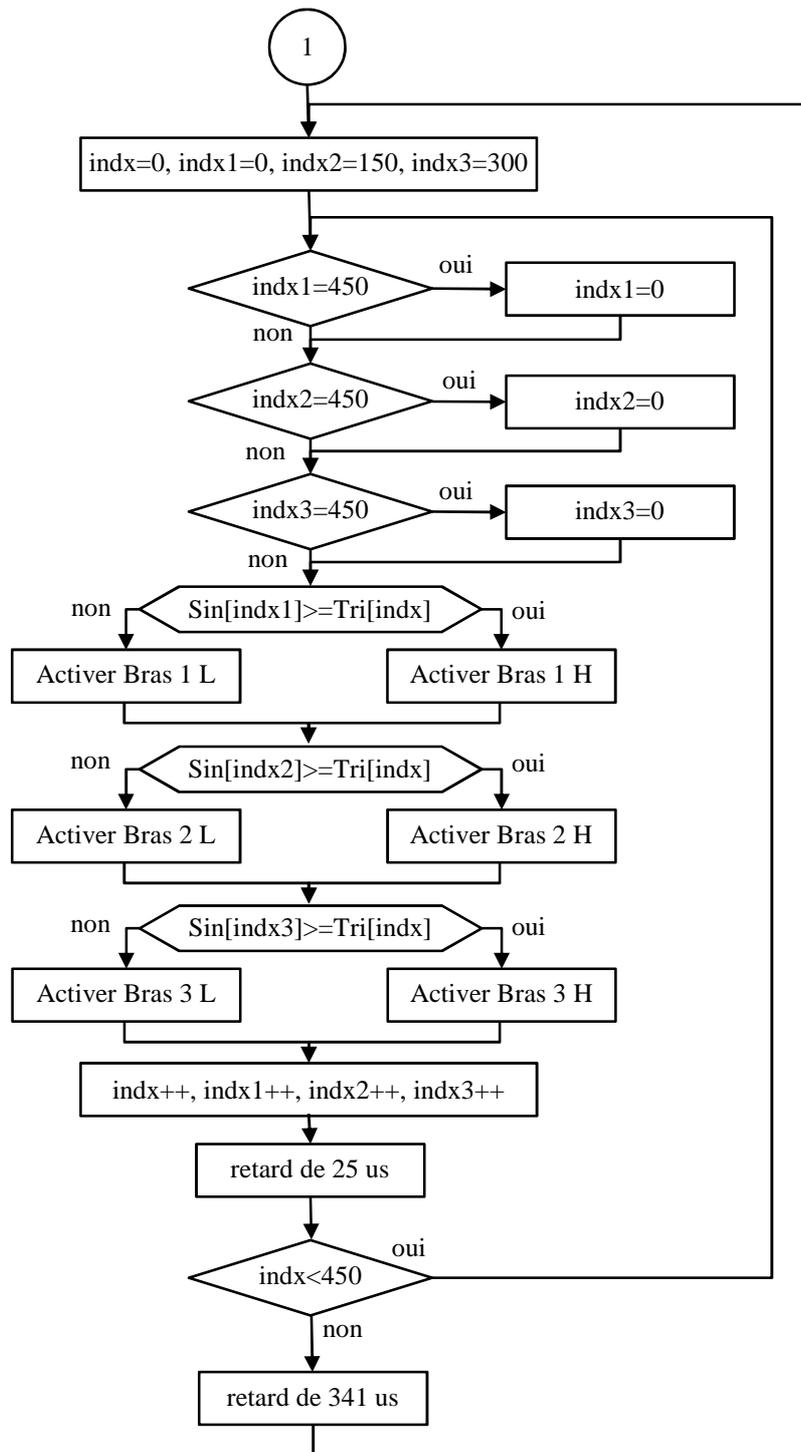


Figure III- 3 : Organigramme 1 d'une commande MLI implanté sur un PIC18F4550

L'organigramme de la figure précédente présente un exemple de génération d'une commande MLI. Comme on utilise des vecteurs de taille importante (450 dans ce cas), ce code peut être utilisé avec un microcontrôleur PIC18F4550 pour piloter un onduleur de tension. Les résultats de cette implantation sont présentés sur les deux figures qui suivent.

La Figure III- 4 représente les signaux de commande des trois bras (haut) de l'onduleur de tension. Tandis que la Figure III- 5 représente les tensions simples de la sortie de l'onduleur. Nous voyons bien la ressemblance et le déphasage  $120^\circ$  entre ces trois tensions de sortie.

### CHAPITRE III : Implantation d'une commande PWM sur un microcontrôleur

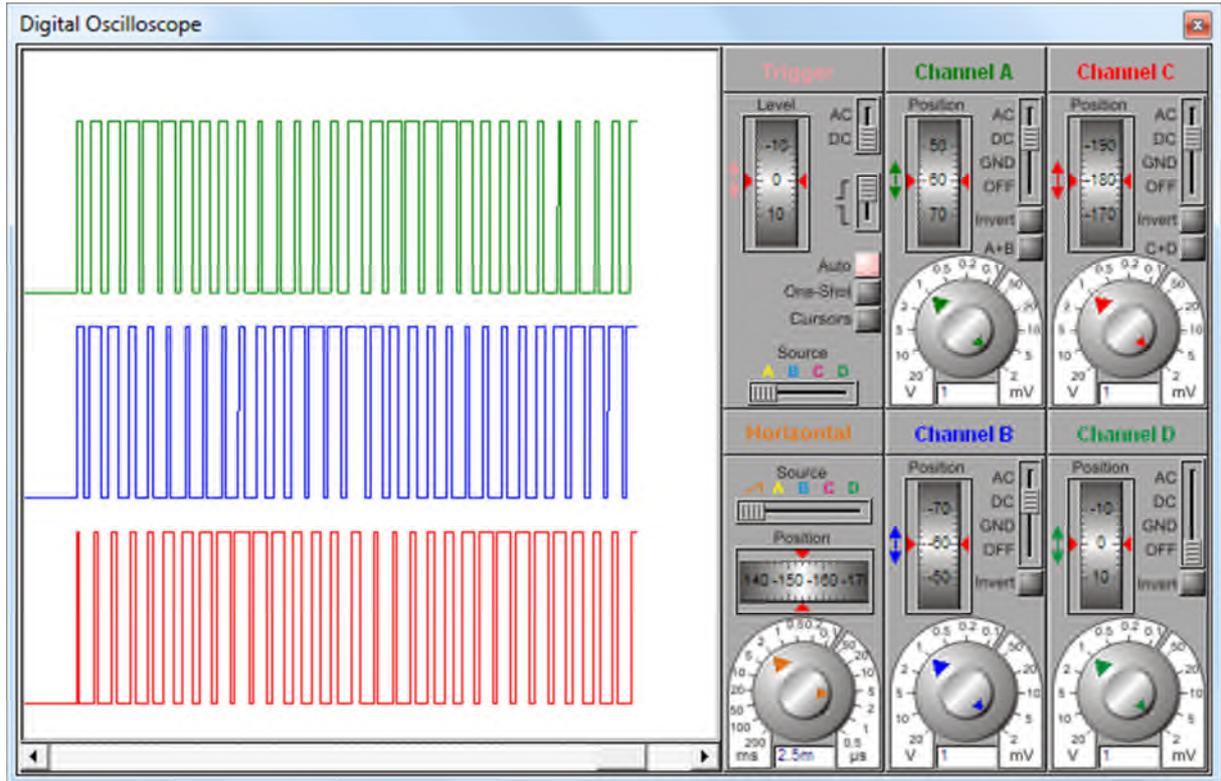


Figure III- 4 : Signaux de commande de l'onduleur de tension commandé par l'organigramme 1

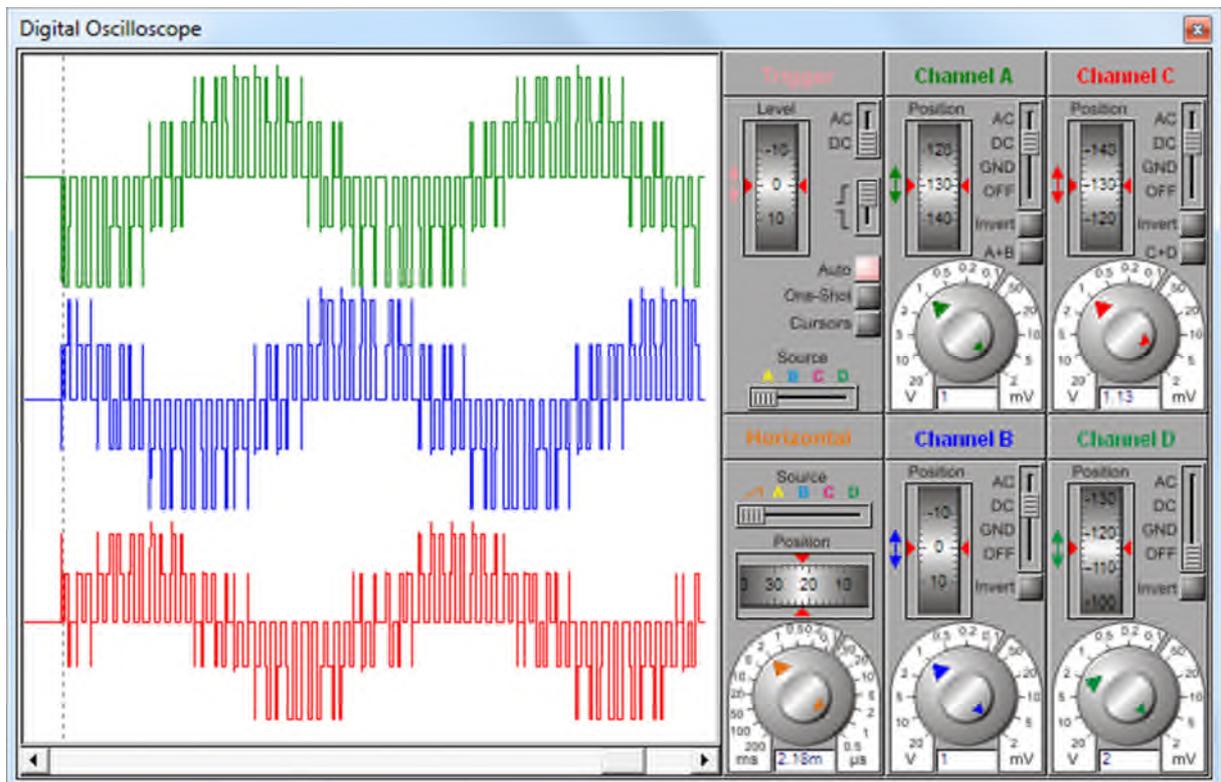


Figure III- 5 : Tension de sortie de l'onduleur de tension commandé par l'organigramme 1

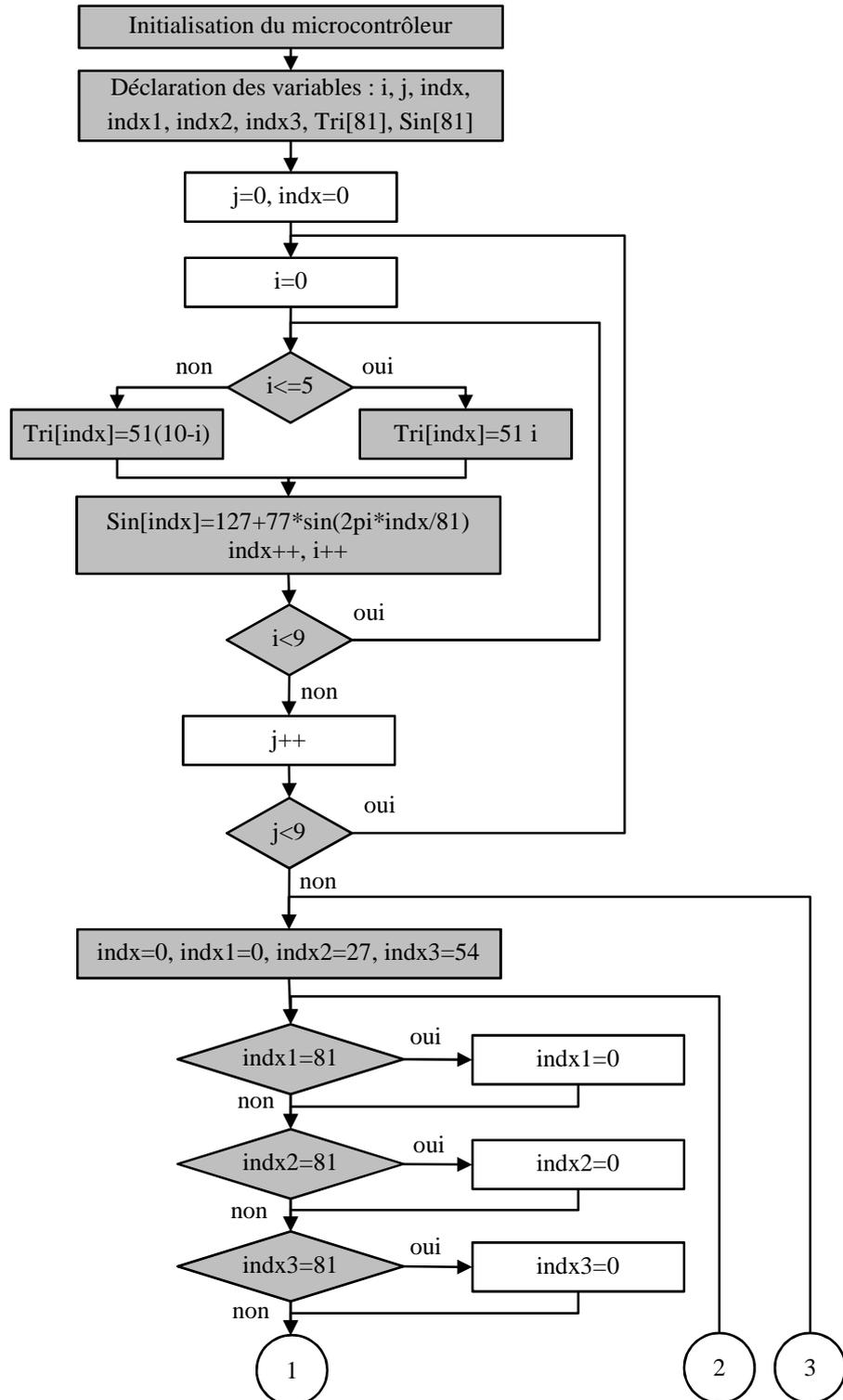
Injecter ce même code directement sur un microcontrôleur PIC16F877 n'est pas possible comme le compilateur génère une erreur d'insuffisance de mémoire. D'ailleurs, même sur le

### CHAPITRE III : Implantation d'une commande PWM sur un microcontrôleur

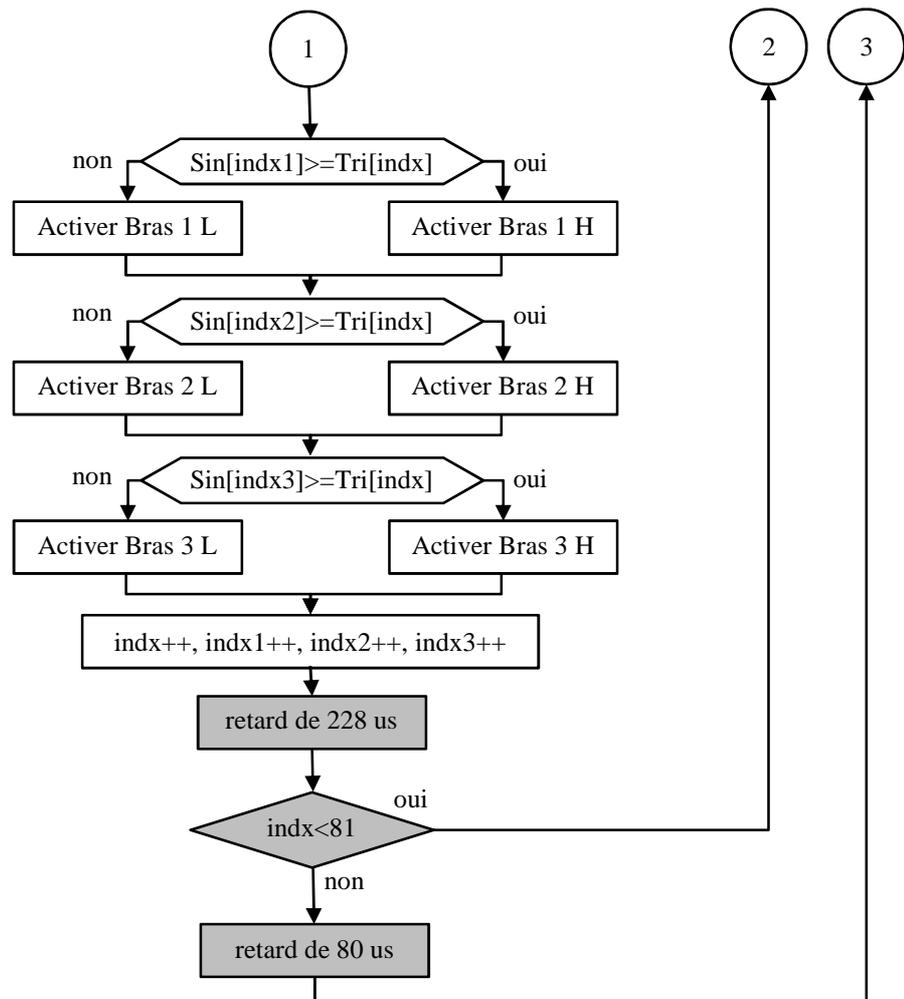
microcontrôleur PIC18F4550, ce code consomme 68% de RAM et 22% de ROM. Si on ajoute d'autres fonctions comme l'acquisition analogique, même la mémoire de ce microcontrôleur deviendra insuffisante.

Si on cherche à utiliser un microcontrôleur PIC16F877 malgré tout, on est obligé d'ajuster l'organigramme 1, afin que ce microcontrôleur trouve de l'espace pour gérer l'ensemble des variables.

L'organigramme suivant montre les changements apportés par rapport au précédent.



### CHAPITRE III : Implantation d'une commande PWM sur un microcontrôleur

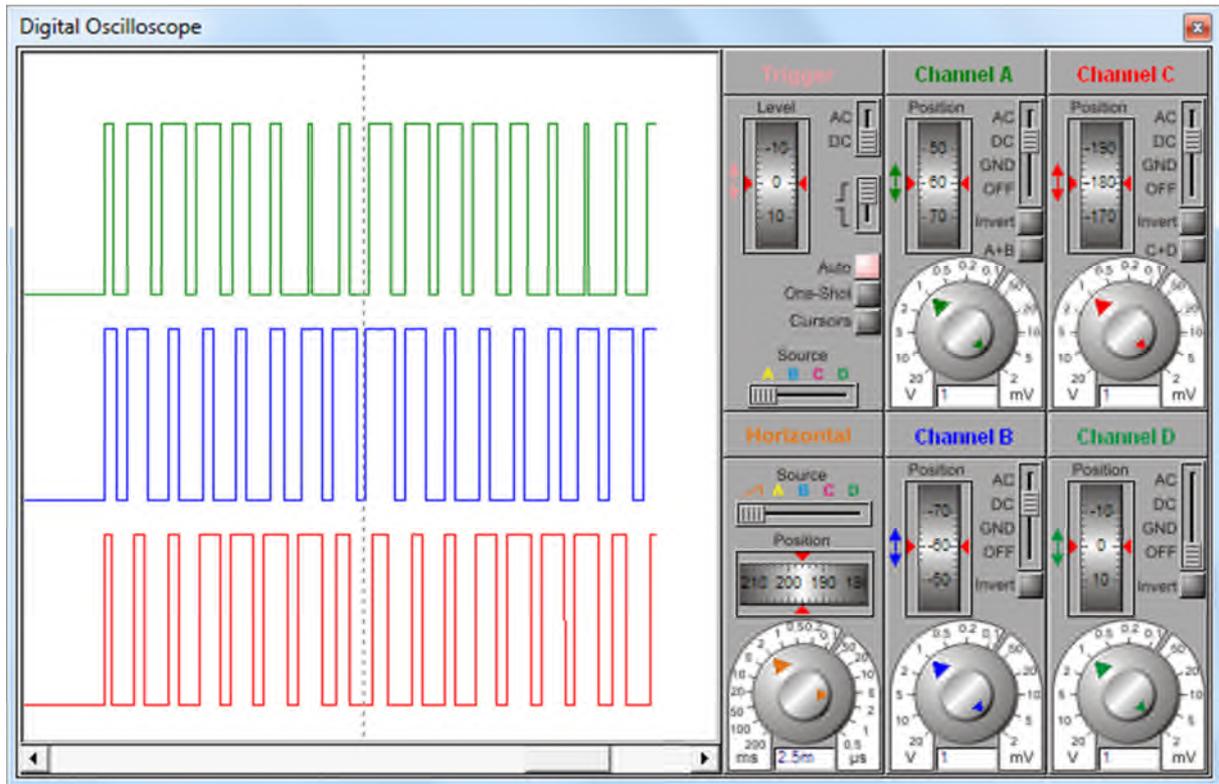


**Figure III- 6 :** Organigramme 2 d'une commande MLI implanté sur un PIC16F877

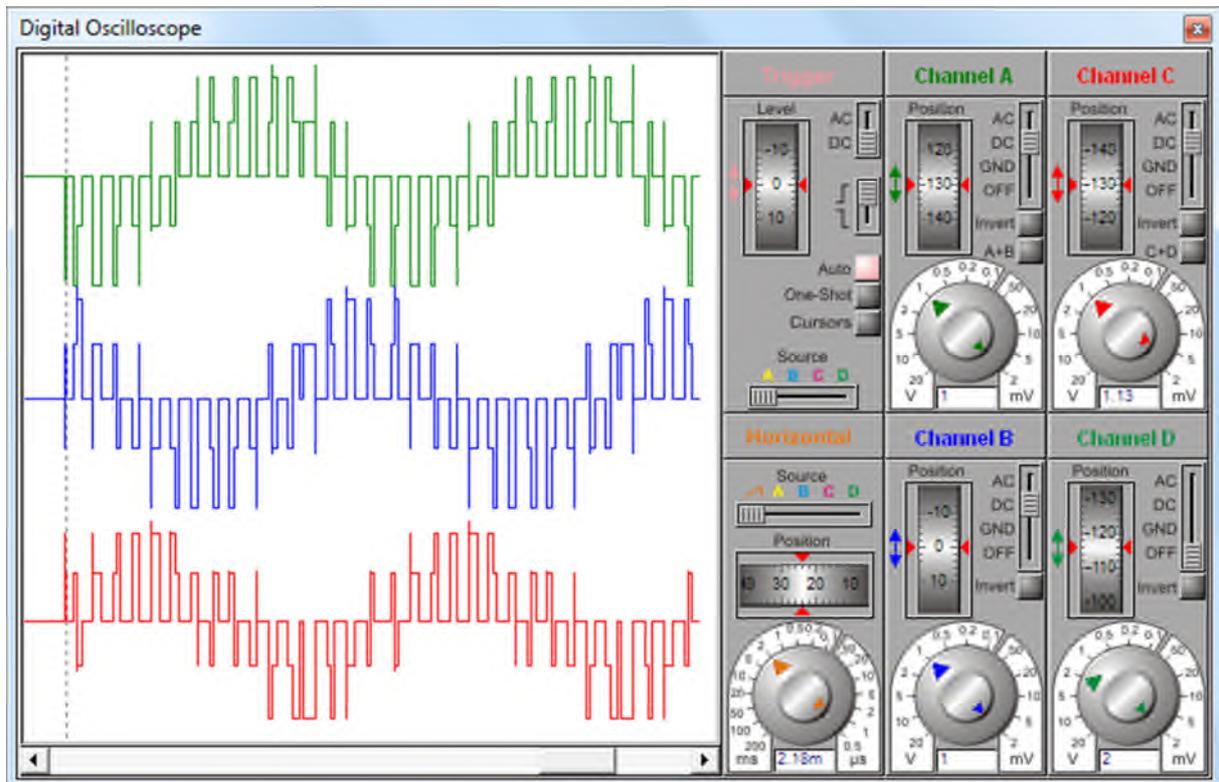
Le résultat de cet organigramme est présenté sur les Figure III- 7 et Figure III- 8. On constate visuellement qu'il s'agit pratiquement du même résultat. Mais, on sait qu'une analyse spectrale donne l'inverse, ce qu'on a généré avec le microcontrôleur PIC18F4550 est meilleur par rapport à ce qu'on a généré avec le microcontrôleur PIC16F877. Pas parce que le PIC18F4550 est meilleur par rapport au PIC16F877, mais par le nombre de points qu'on a utilisé pour générer nos signaux.

Dans l'organigramme 1, on a défini deux tableaux de stockage de données d'une taille de 450 points. Alors que dans le l'organigramme 2, la taille des tableaux était 81 points.

### CHAPITRE III : Implantation d'une commande PWM sur un microcontrôleur



**Figure III- 7** : Signaux de commande résultants après implantation de l'organigramme 2



**Figure III- 8** : Tension de sortie de l'onduleur de tension commandé par l'organigramme 2

Afin de mieux comprendre le fonctionnement de notre générateur de signaux MLI triphasés, on préfère de détailler l'organigramme de la Figure III- 3.

### CHAPITRE III : Implantation d'une commande PWM sur un microcontrôleur

Au début, il faut inclure la bibliothèque de définition du microcontrôleur PIC18F4550, et initialiser l'horloge et les fusibles de configuration. Il faut ajouter la bibliothèque « math.h » qui inclut la méthode de calcul de la fonction sinus.

Les tableaux triangle et sinus vont sauvegarder 450 valeurs de 8 bits. Ceci nous oblige de déclarer les index sur 16 bits comme ils dépassent les 255.

Après l'étape d'initialisation du microcontrôleur, on passe au calcul des tableaux de définition des deux fonctions : triangle et sinus. La fonction triangle a été définie avec 30 points significatifs. L'indice de modulation choisi est «  $m=15$  » ; c'est le nombre de répétition de la fonction triangle sur une seule période de la fonction sinus. De cette manière, nous allons définir la fonction sinus avec 450 points ( $15 \times 30 = 450$ ).

Cette fonction sinus est centrée par rapport à la valeur 127. La valeur 77 représente une profondeur de modulation «  $r$  » égale à 0.8 ( $255 \times 0.8 - 127 = 77$ ). D'où, cette fonction sinus variera entre la valeur 50 et la valeur 204.

Après remplissage des tableaux de définition des fonctions triangle et sinus avec 450 valeurs et dont les rapports de modulation sont :  $m=15$  et  $r=0.8$ , on exécute une boucle infinie qui génère les signaux de sorties après comparaison.

Comme la longueur du tableau sinus est de 450 valeurs, on a créé un décalage de  $120^\circ$  entre les index des trois signaux de sortie triphasée. On initialise l'index quand il atteint la fin du tableau de définition.

En ce moment, on fait une comparaison entre le signal triangulaire et les trois signaux sinusoïdaux pour générer les signaux de commande.

Afin d'ajuster la fréquence et retrouver le bon signal de sortie, on a corrigé les temps de calcul des différentes boucles avec un temps de retard. Cet ajustement a été fait expérimentalement pour avoir une fréquence de sortie de 50 Hz.

La fréquence maximale de sortie en utilisant un microcontrôleur PIC18F4550 avec un code de l'organigramme 1 est de 120 Hz. Alors si on utilise un microcontrôleur PIC16F877 avec le code de l'organigramme 2, on aura une fréquence maximale de 737 Hz. Ces mesures ont été obtenues expérimentalement. Ceci est relatif au nombre de point de définition des signaux triangulaire et sinus ; on a utilisé 450 points dans le premier cas, et 81 points dans le second cas.

L'indice de modulation «  $m$  » est le rapport de la fréquence de la porteuse (triangulaire) à celle de la modulante (sinus). La profondeur de modulation «  $r$  » est le rapport de l'amplitude de la modulante (sinus) à celle de la porteuse (triangle).

L'indice de modulation est plutôt un paramètre de construction pour des commandes stochastiques (non-calculées), on s'assure simplement qu'il est assez grand pour que le courant soit suffisamment peu déformé, et si la tension de sortie est à fréquence fixe, il peut être judicieux de choisir «  $m$  » pair ou impair selon la topologie ; alors que la profondeur de modulation est un paramètre de réglage, comme le rapport cyclique d'un hacheur.

La tension efficace en sortie est proportionnelle à la tension de bus DC et à la profondeur de modulation ; si elle est inférieure à 1 et que l'indice de modulation est suffisant.

### CHAPITRE III : Implantation d'une commande PWM sur un microcontrôleur

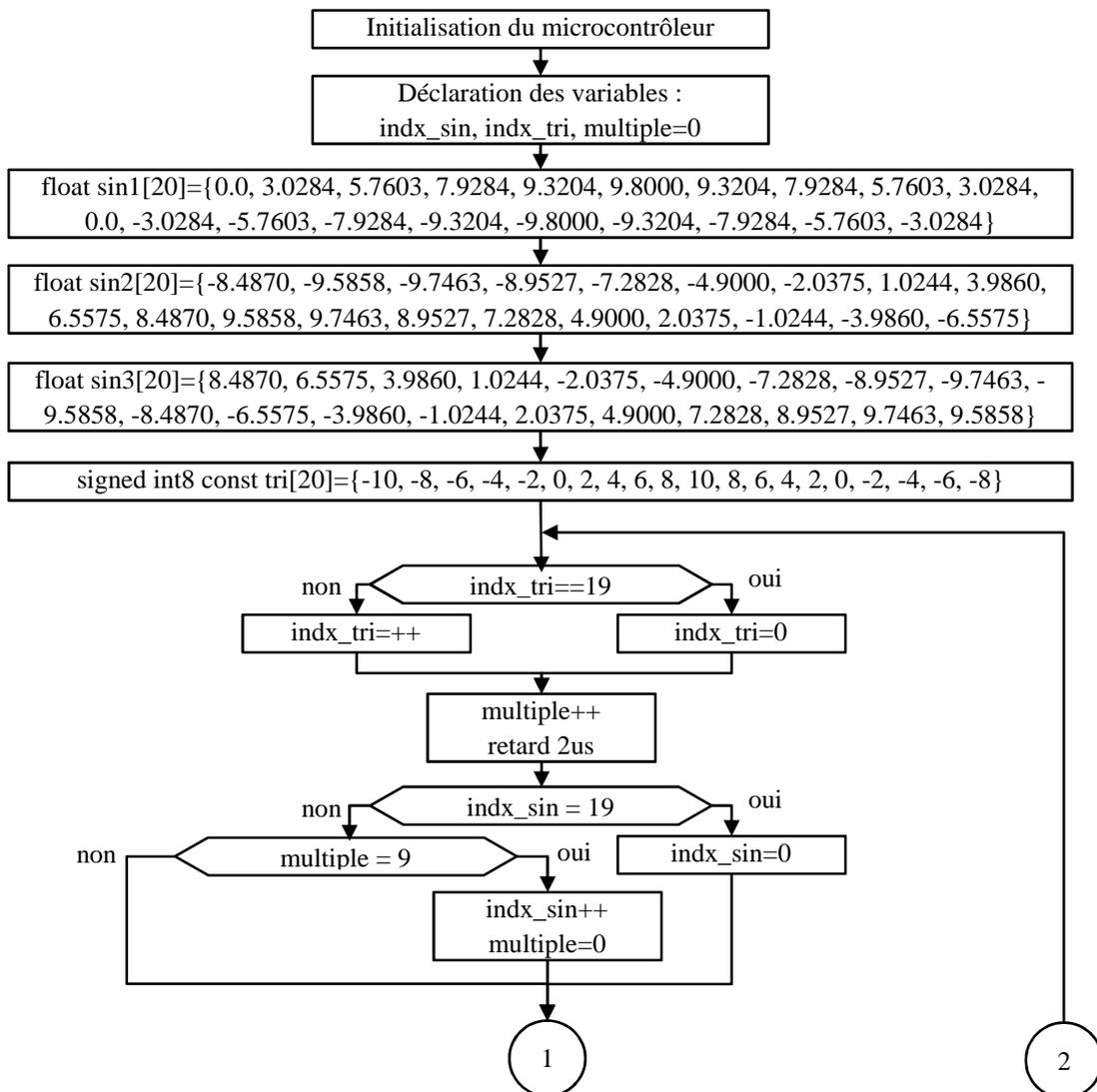
Avec une commande simple (modulante sinusoïdale), il n'est pas judicieux d'amener la profondeur de modulation au-delà de 1, car des harmoniques basse fréquence apparaissent, et la déformation des courants est accrue.

Avec le code de l'organigramme 1, l'ajustement de la fréquence de sortie est fastidieux, et n'est pas possible de le faire en temps réel. De même la variation de la valeur efficace de sortie n'est possible que lorsqu'on programme le microcontrôleur une nouvelle fois.

Pour remédier à ça, il est préférable de concevoir un code dont la valeur « m » et « r » sont réglable de l'extérieur avec des potentiomètre par exemple ; sans le recours à la programmation ; ceci rend le système flexible.

#### III.4 Exemple 2 de génération d'un un signal MLI :

On peut générer des signaux MLI pour piloter un onduleur de tension d'une autre manière. Dans le présent exemple, on a préféré de stocker en mémoire les valeurs précalculées des signaux : triangulaire et sinusoïdales. L'organigramme de calcul est présenté sur la figure suivante :



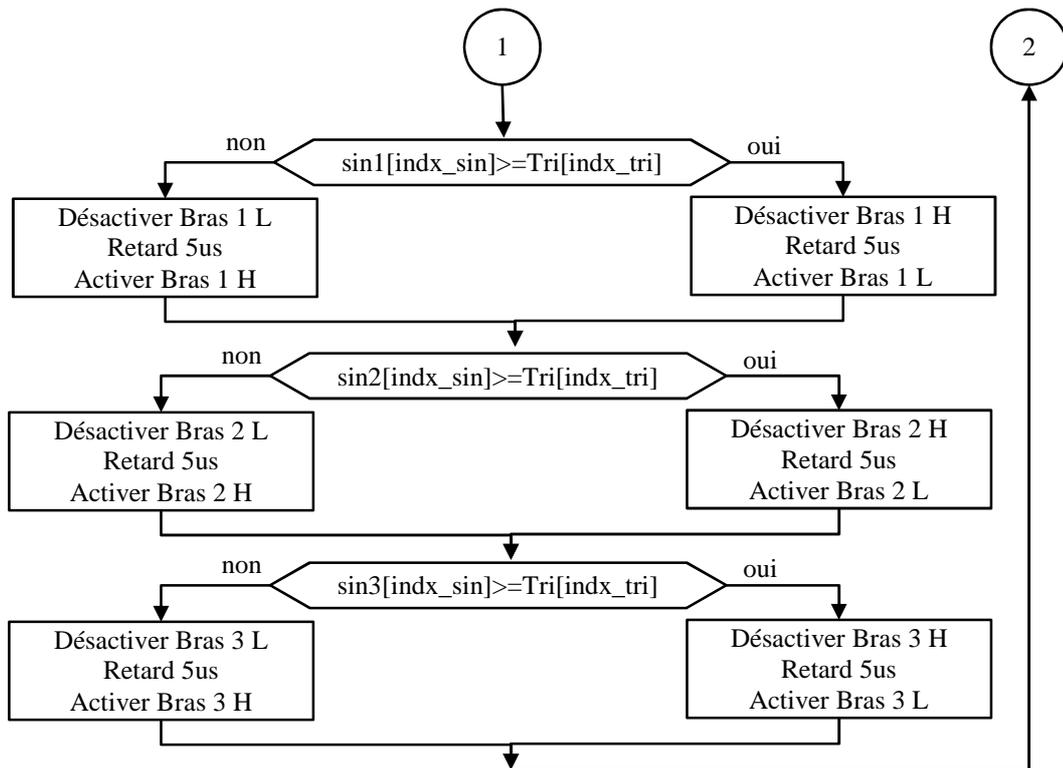


Figure III- 9 : Organigramme 3 d'une commande MLI implanté sur un PIC16F877

Par rapport aux autres organigrammes, celui de la figure précédente possède quelques changements. Les valeurs de la fonction triangulaire et des trois fonctions sinusoïdales ont été calculées avant puis intégrées dans le microcontrôleur. Ceci nous permet de gagner le temps de calcul effectué avec la fonction « sin ».

En plus, on a utilisé deux variables de positionnement sur les tableaux : triangle et sinus. La variable « multiple » est utilisée comme indice de modulation « m » qui représente le rapport de la fréquence de la porteuse sur la fréquence de la modulatrice. Un temps de retard a été introduit pour éviter un court-circuit lors du changement de la commutation sur le même bras. Malgré que la commutation soit complémentaire, le temps d'ouverture et de fermeture des composants de puissance n'est pas le même. C'est pour cette raison qu'il faut ajouter un temps de retard de commutation.

Les tableaux de définition des fonctions sinus définies sur 20 valeurs de type float (32 bits). Comme a pris 20 valeurs, le décalage de l'index des trois signaux par 120° n'est pas possible (20/3 ne donne pas une valeur entière). Ce choix a été fait à volonté pour saturer la RAM par les trois tableaux.

Le tableau triangle a été définie par 20 valeurs de type 8 bits signé au lieu du type float seulement pour qu'on puisse voir leur emplacement en mémoire par la suite.

L'indice de modulation « m » a été choisi égale à 9 ; quand l'indice de positionnement dans le tableau triangle avance par 9 pas, l'indice de positionnement dans le tableau sinus avance par 1 pas.

Notant qu'on a ajouté la directive « const » quand on a créé le tableau triangle pour le déplacer en mémoire programme afin de libérer l'espace dans la mémoire RAM. Si on enlève

### CHAPITRE III : Implantation d'une commande PWM sur un microcontrôleur

cette directive, la compilation du programme deviendra impossible pour problème d'insuffisance d'espace de stockage des variables.

L'implantation de cet algorithme sur un microcontrôleur PIC16F877, nous a donné les résultats suivant :

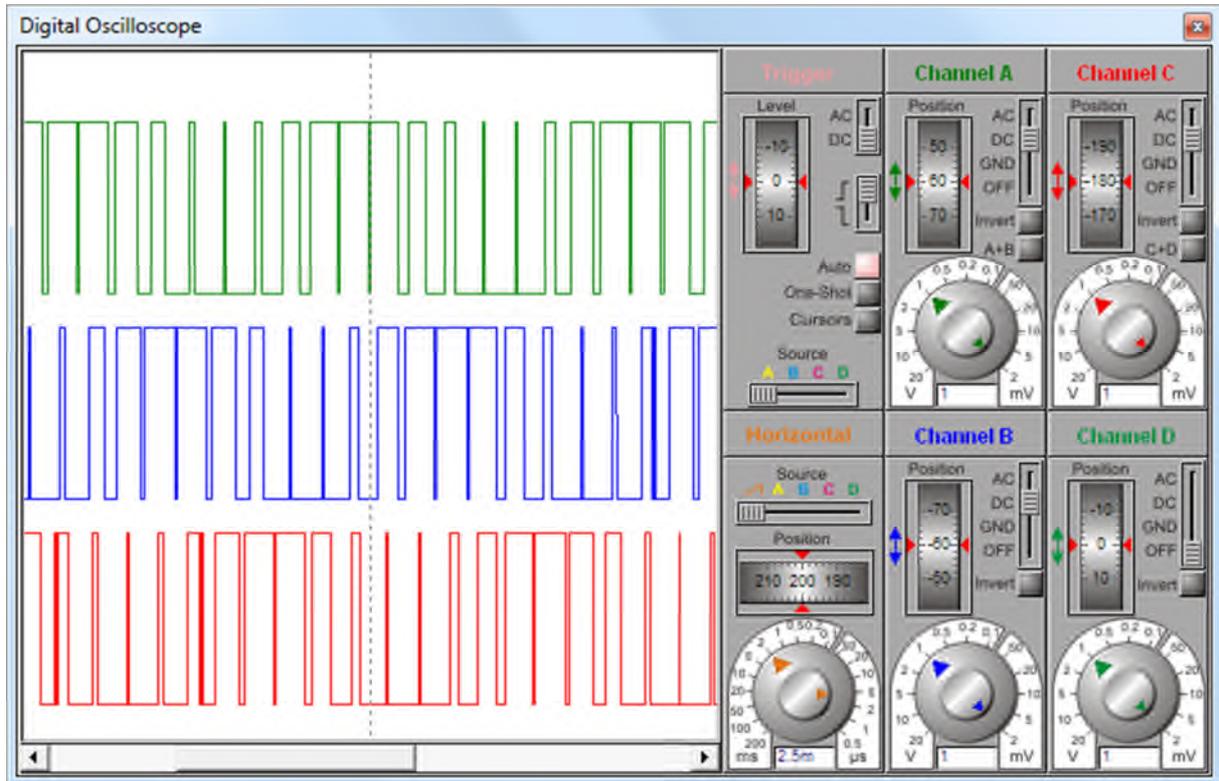


Figure III- 10 : Signaux de commande résultants après implantation de l'organigramme 3

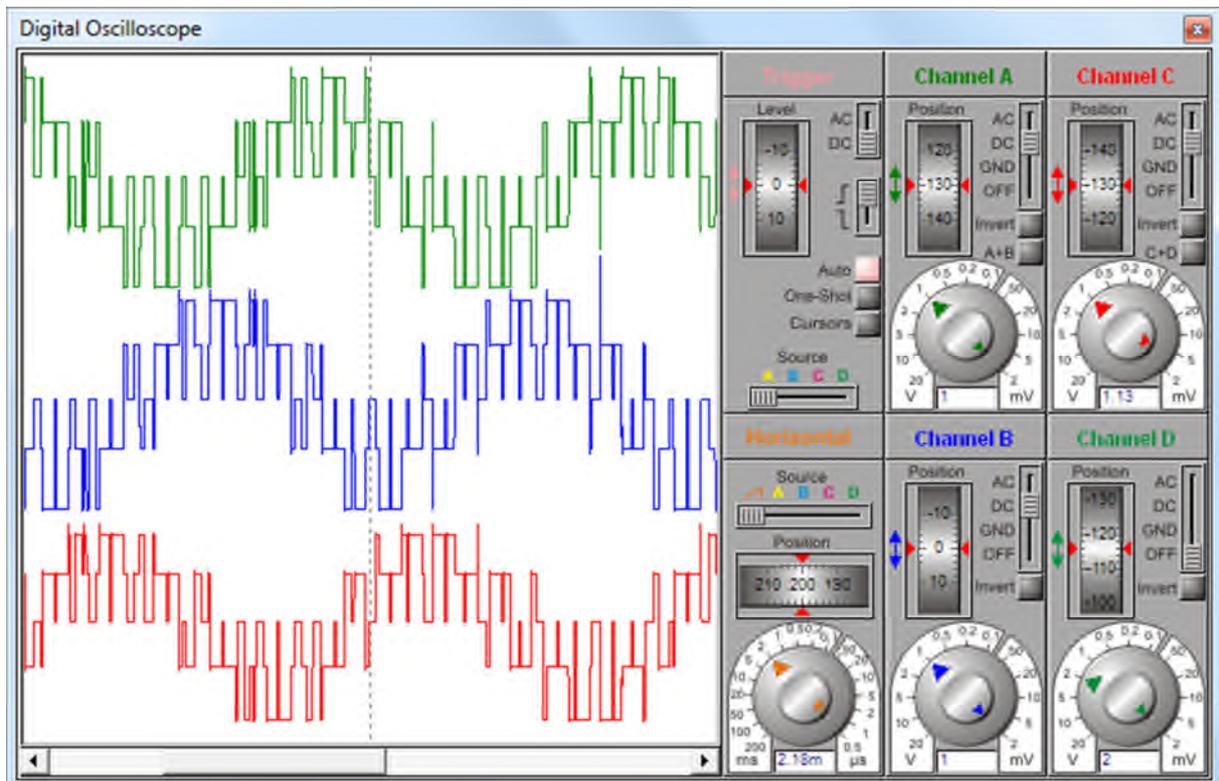


Figure III- 11 : Tension de sortie de l'onduleur de tension commandé par l'organigramme 3

### III.5 Allocation manuelle des variables dans l'espace mémoire :

Pour mieux comprendre les techniques d'implantation des différentes stratégies de commande sur les microcontrôleurs, on a préféré de réaliser un circuit générateur de fonction qui nous permet de tester les signaux générés réellement à l'intérieur d'un microcontrôleur. Ce circuit n'est que l'association des ports « B » et « D » du microcontrôleur avec des convertisseurs numérique analogique DAC. Les signaux générés en interne vont être présenté graphiquement après amplification de la sortie du circuit DAC. Ce montage est représenté par le schéma de la figure suivante :

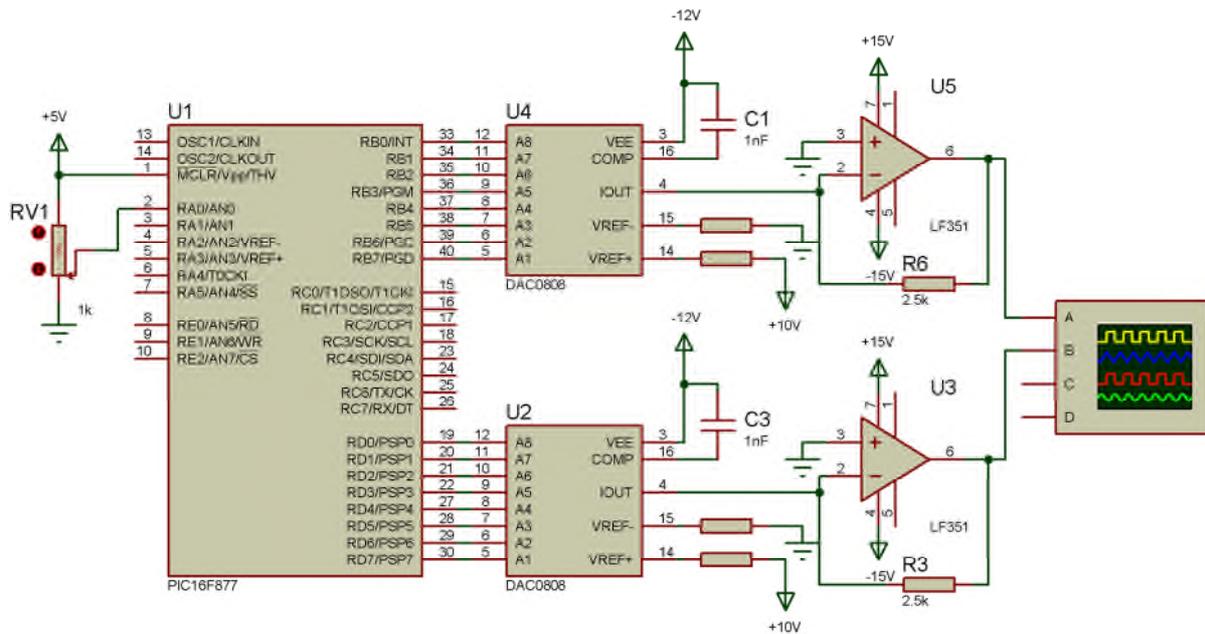
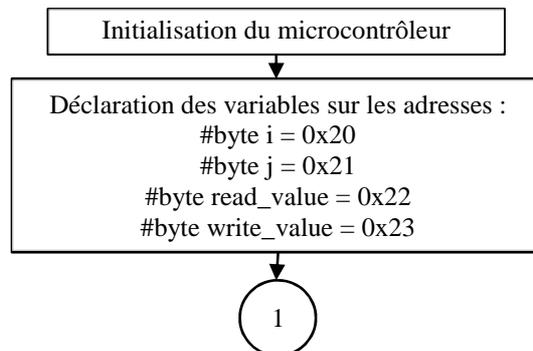


Figure III- 12 : Circuit simplifié d'un générateur de fonction

Comme on l'a constaté sur la figure II.3, la Bank0 dispose de 96 registres à usage générale d'une taille de 8 bits (adresse 20h jusqu'à 7Fh) ; ces derniers nous permettent le stockage de nos données. La Bank1 dispose de 80 registres libres (adresse A0h jusqu'à EFh). Tandis que la Bank2 dispose de 96 registres libres (adresse 110h jusqu'à 16Fh). De même pour la Bank3, elle dispose de de 96 registres libres (adresse 190h jusqu'à 1EFh).

On cherche, dans l'exemple qui suit, à calculer les valeurs des tableaux triangle et sinus et de les positionner dans des emplacements bien définis dans la mémoire. En plus, on va calculer le temps mis pour l'exécution de ces différentes fonctions.



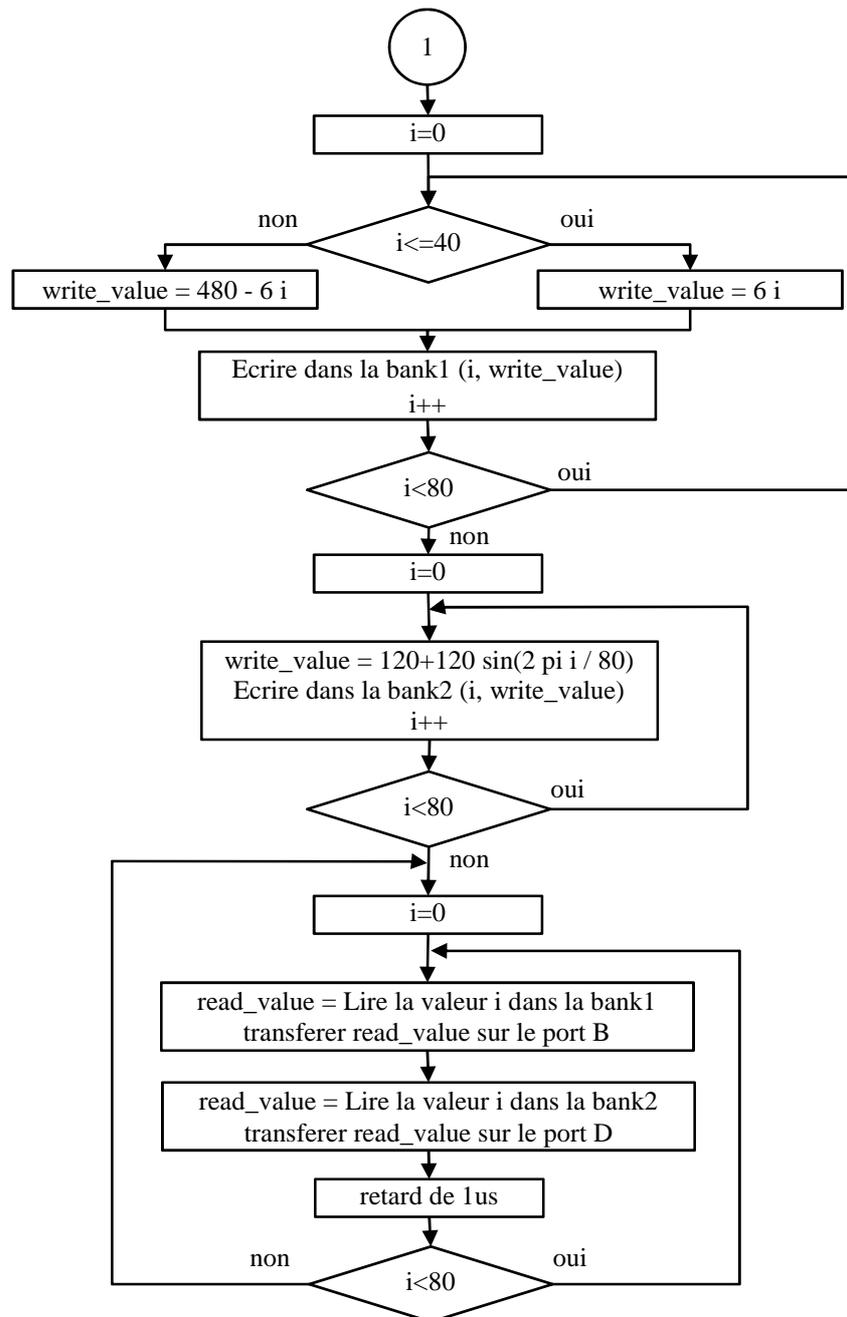


Figure III- 13 : Génération de fonction avec un PIC16F877

Dans l'organigramme précédent, nous avons alloué des espaces 8 bits pour nos variables de travail  $i$ ,  $j$ ,  $read\_value$  et  $write\_value$  aux adresse 20h, 21h, 22h et 23h respectivement de la Bank 0.

Comme la Bank1 ne dispose que de 80 registres libres, on a essayé de définir notre fonction triangulaire sur 80 points ; 40 points pour la montée de la dent de scie et les autres 40 points pour la descente. Puisqu'on travaille en 8 bits, les valeurs de cette fonction peuvent varier de 0 jusqu'à 255. Quand on divise les 255 sur les 40 points, on trouve 6.375 qui n'est pas entier. Alors, on a préféré de prendre un pas de 6 et la fonction variera entre 0 et  $6 \times 40 = 240$ .

Avec une horloge de 20 MHz, le temps total d'exécution de ces calculs et l'écriture sur la RAM est de 1.0972 ms.

Address	0	255	0	25	187	63	0	255
0080	0	255	0	25	187	63	0	255
0088	0	7	0	1	0	0	0	0
0090	0	0	255	0	0	0	0	0
0098	2	0	0	0	0	0	230	7
00A0	0	6	12	18	24	30	36	42
00A8	48	54	60	66	72	78	84	90
00B0	96	102	108	114	120	126	132	138
00B8	144	150	156	162	168	174	180	186
00C0	192	198	204	210	216	222	228	234
00C8	240	234	228	222	216	210	204	198
00D0	192	186	180	174	168	162	156	150
00D8	144	138	132	126	120	114	108	102
00E0	96	90	84	78	72	66	60	54
00E8	48	42	36	30	24	18	12	6
00F0	0	0	0	0	0	0	0	0
00F8	0	0	0	0	0	0	0	0

Figure III- 14 : Remplissage de la fonction triangulaire sur la Bank1

La Bank2 dispose de 96 registres libres, mais pour faciliter le programme par la suite, on préfère de définir pour le moment la fonction sinus sur 80 points comme on a fait pour la fonction triangulaire. Vu que cette dernière varie entre 0 et 240, on a préféré que la fonction sinus varie de la même manière. C'est pour cette raison qu'on l'a centré par rapport à 120.

Avec la même horloge de 20 MHz, le temps total, d'exécution des calculs de la fonction sinus et l'écriture sur la RAM, est de 217.13 ms. L'augmentation de ce temps de calcul est dû à la fonction « sin » qui prend un temps de 2.3846 ms pour chaque valeur.

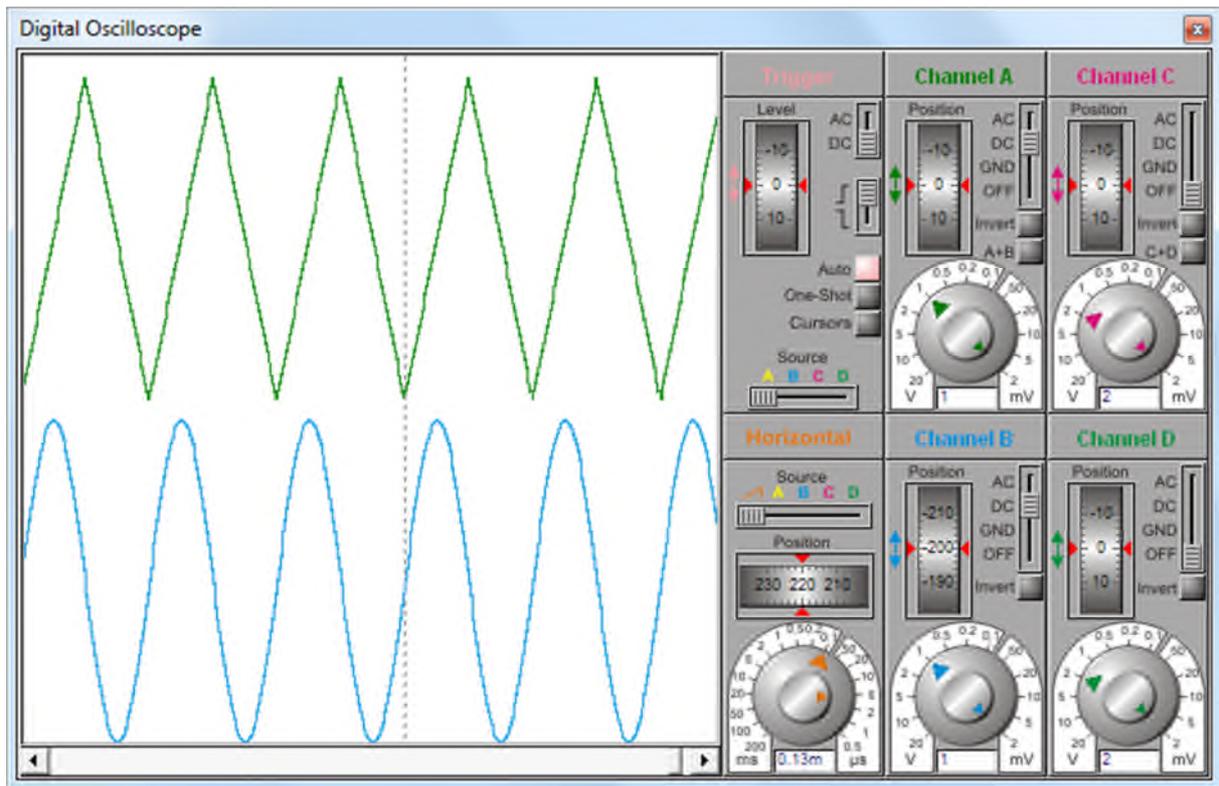
Address	0	140	0	25	113	0	202	0
0100	0	140	0	25	113	0	202	0
0108	0	0	0	0	136	81	102	161
0110	120	128	138	148	157	165	174	182
0118	190	197	204	211	217	222	226	230
0120	234	236	238	239	240	239	236	236
0128	234	230	226	222	217	211	204	197
0130	190	182	174	165	157	148	138	129
0138	119	110	101	91	82	74	65	57
0140	48	43	35	28	22	17	13	9
0148	5	3	1	0	0	0	1	3
0150	5	9	13	17	22	28	35	42
0158	48	57	65	74	82	91	101	110
0160	0	0	0	0	0	0	0	0
0168	0	0	0	0	0	0	0	0
0170	0	0	0	0	0	0	0	0
0178	0	0	0	0	0	0	0	0

Figure III- 15 : Remplissage de la fonction sinus sur la Bank2

Le calcul de la fonction « sin(x) » avec CCS-PICC se fait par la décomposition de Taylor de la fonction « cos(x-pi/2) ».

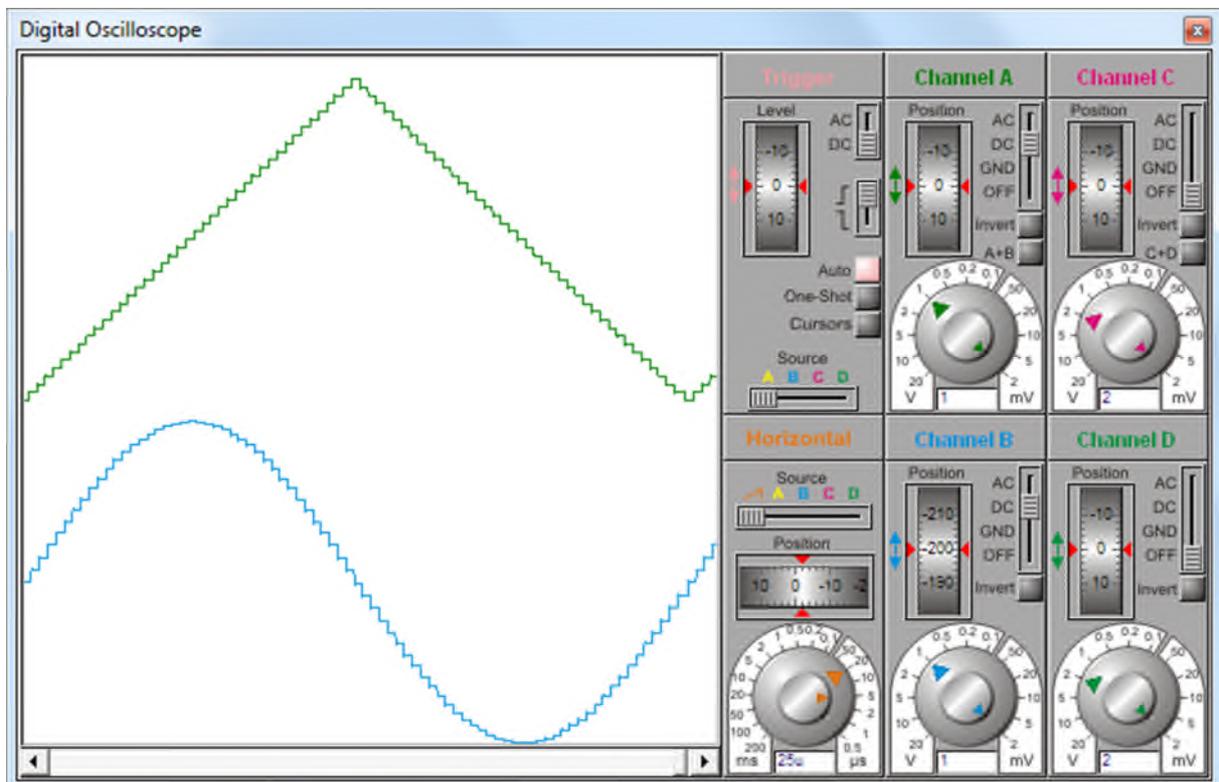
$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + \frac{(-1)^n \cdot x^{2n}}{(2 \cdot n)!} \quad (III.1)$$

Par la suite, la fréquence maximale qu'on peut atteindre avec ce circuit pour générer les deux fonctions en même temps est de 2.49 kHz. Sachant qu'on a défini l'évolution de ces fonctions avec 80 points. Si on veut augmenter la fréquence de sortie, on doit minimiser le nombre de points de définition.



**Figure III- 16 :** Tension de sortie générées avec l'organigramme de la Figure III- 13

Malgré qu'on voie des courbes lisses sur la figure précédente, il ne faut pas oublier qu'elles sont créées par des valeurs numériques. Si on agrandit les courbes, nous allons voir le phénomène de pixellisation.



**Figure III- 17 :** Agrandissement de la tension de sortie de la Figure III- 16

### CHAPITRE III : Implantation d'une commande PWM sur un microcontrôleur

Pour lisser les courbes plus, il faut utiliser plus de points de définition. Si on utilise plus de points, il nous faut plus d'espace de stockage sur notre microcontrôleur PIC16F877 ; or, l'espace de stockage de ce dernier est limité. On ne peut stocker nos valeurs que dans la RAM, dans la mémoire programme en utilisant la directive « const » si ces valeurs ne changent pas, ou dans la mémoire EEPROM. Autrement, il faut revoir la méthode de programmation.

Si nous voulons stocker nos valeurs sur la mémoire EEPROM comme elle dispose de 256 octets, il faut savoir que le temps de lecture d'une seule valeur est de 2  $\mu$ s, alors que le temps d'écriture d'une seule valeur dans cette mémoire est de 10 ms.

En plus, la mémoire EEPROM support un maximum de 1 000 000 d'opération d'effacement/écriture. A cet effet, il faut bien optimiser les séquences d'écriture sur cette mémoire en faisant une simple lecture et comparaison afin d'éviter l'écriture des mêmes valeurs.

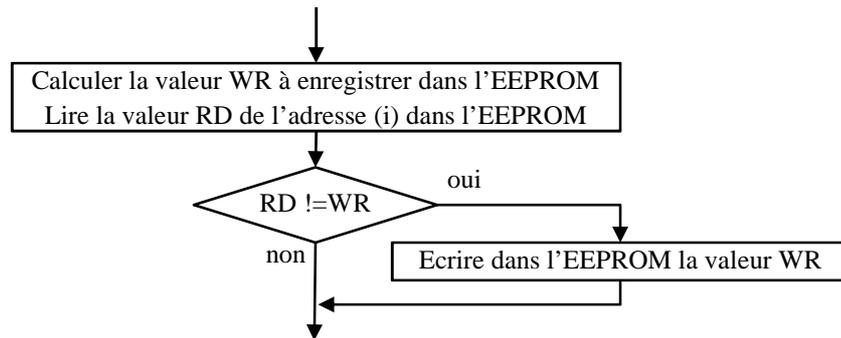
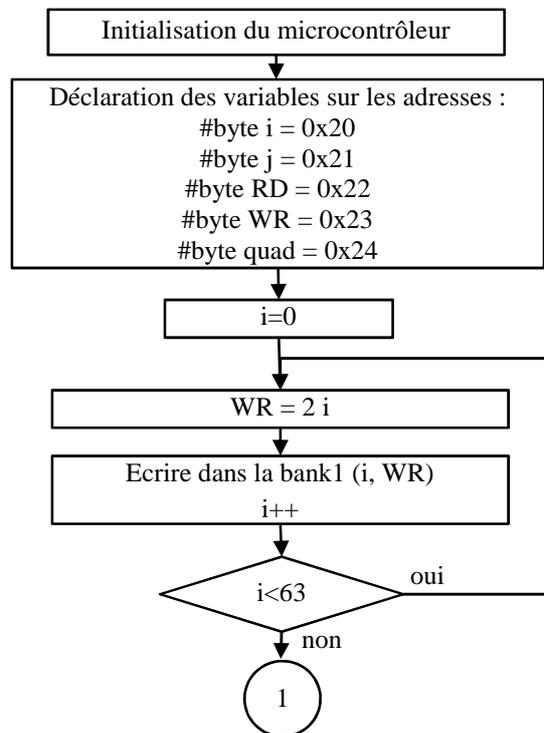


Figure III- 18 : Optimisation de l'écriture dans la mémoire EEPROM

Dans l'organigramme qui suit, on a amélioré la définition des fonctions. On a stocké seulement les points de définition du quart des signaux, et on a déduit les trois quarts avec symétrie.



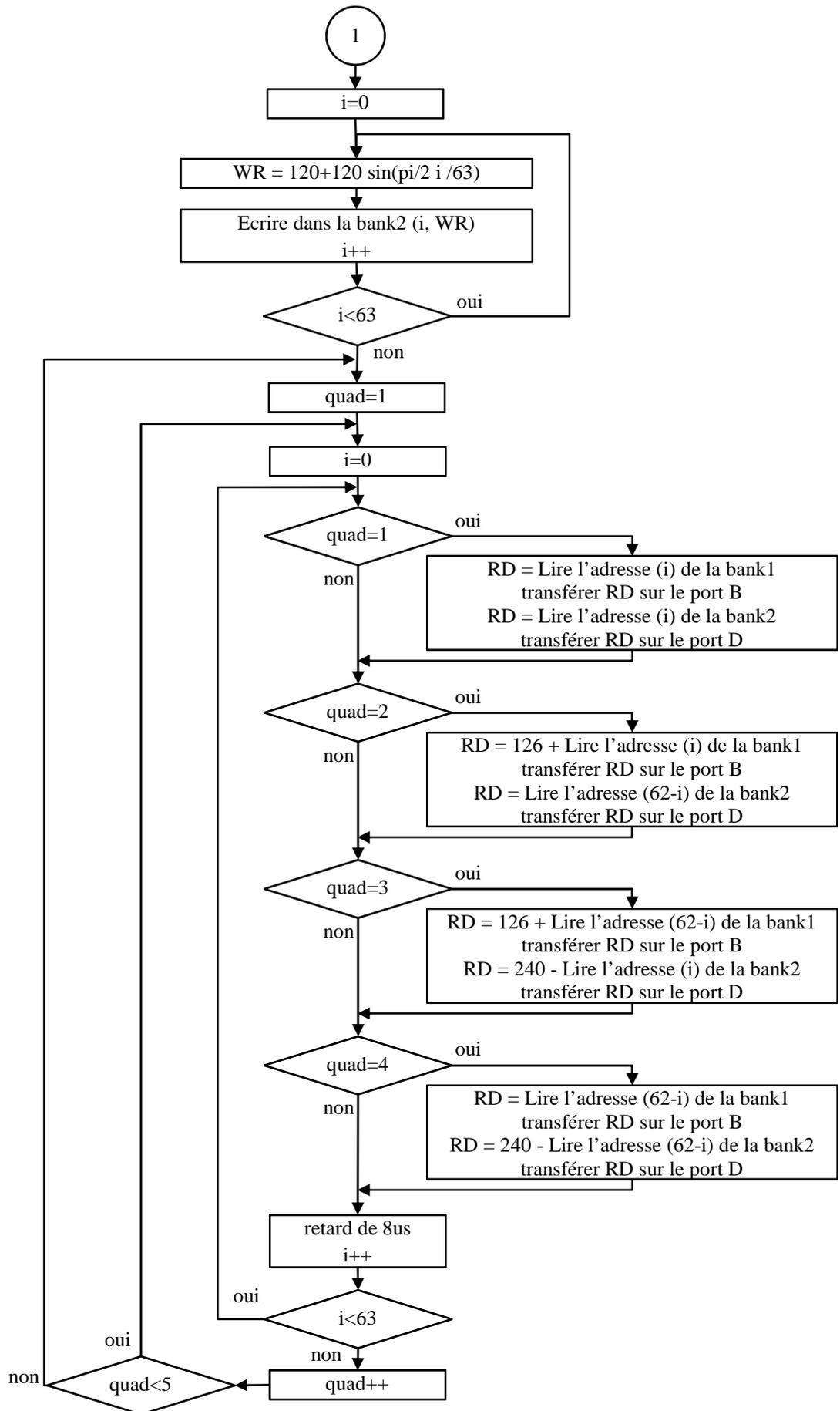


Figure III- 19 : Optimisation des fonctions de génération de signaux avec un PIC16F877

### CHAPITRE III : Implantation d'une commande PWM sur un microcontrôleur

La figure qui suit montre les résultats d'optimisation des fonctions de génération des signaux. Malgré qu'on ait utilisé 63 points de définition, les courbes de la figure Figure III- 20 sont très lisse par rapport à ceux de la figure Figure III- 17 généré avec 80 points.

Cette amélioration a été apporté lorsqu'on a bien su comment définir nos fonctions. On a utilisé 63 points de définition du 1/4 du signal ; c'est-à-dire en réalité on a défini notre fonction avec  $63 \times 4 = 252$  points.

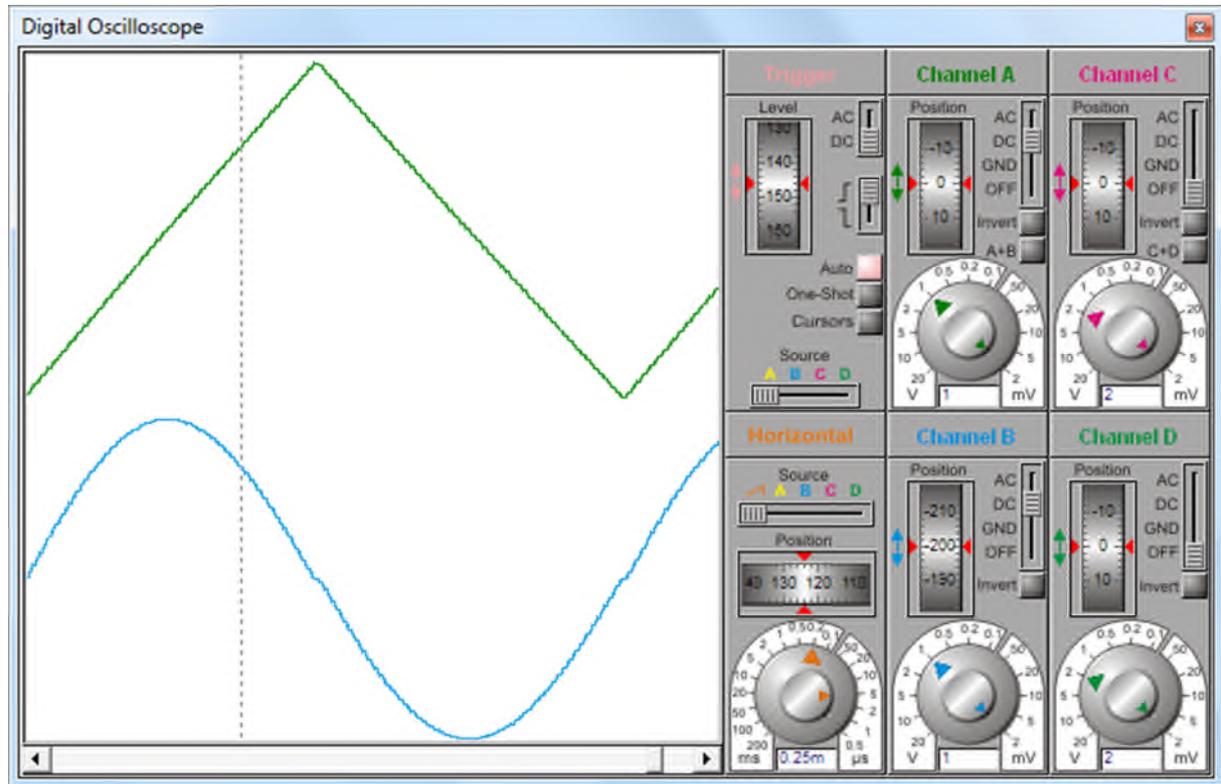


Figure III- 20 : Tensions de sortie générées avec le code de la Figure III- 19

#### III.6 Conclusion :

Nous avons vu dans ce chapitre les techniques qu'il faut les connaître pour programmer un microcontrôleur. Si ce dernier n'exécute pas une tâche demandée bien comme il faut, il faut savoir qu'une limite existe ou bien matériel ou de programmation.

Nous avons bien vu que pour avoir un bon résultat ce n'est pas seulement la capacité mémoire du microcontrôleur ; les techniques de programmation jouent un rôle très important. Le programmeur a un rôle très important lors de la réalisation de ces tâches, nous avons confirmé qu'avec un nombre réduit de point de définition plus de l'optimisation du code, nous pouvons générer un très bon signal qu'avec un nombre élevé de points de définition sans optimisation.

Si on veut par exemple varier la fréquence de sortie en utilisant un potentiomètre, il faut tenir en compte le temps mis pour la conversion analogique numérique. De même, si on veut ajouter un afficheur LCD ou une communication RS232, il faut prendre en considération le temps nécessaire pour l'exécution de ces différentes tâches.

En plus, il faut bien choisir l'emplacement des tâches, car même si on place une tâches d'une milliseconde à l'intérieur d'une boucle de 1 000 fois, elle perdra une seconde.

## *Conclusion générale*

## *Conclusion générale*

Le travail présenté dans ce mémoire concerne les techniques de commande d'un onduleur de tension à faible puissance. La génération des signaux MLI par un microcontrôleur est intéressante car elle permet de varier la fréquence ; d'où la variation de vitesse de rotation du moteur électrique.

Le premier chapitre de ce mémoire a été consacré à l'étude de la commande d'un onduleur de tension à faible puissance ; à savoir la commande en pleine onde vue sa simplicité, et la commande MLI vue son usage général dans la plupart des travaux de recherches.

Le second chapitre donne une comparaison détaillée sur tous les aspects des microcontrôleurs PIC16F877 et les PIC18F4550 ; leurs architecture interne, les techniques d'accès à leur mémoire RAM, la structure de la mémoire programme, ... nous avons étudié le microcontrôleur PIC18F4550 tout en faisant la différence avec le microcontrôleur PIC16F877.

Finalement, le dernier chapitre aborde l'implantation de la technique de commande MLI sur les microcontrôleurs. Nous avons présenté des exemples avec leur code source en langage PIC-C. En analysant les résultats obtenus par simulation, nous avons proposé des solutions aux problèmes rencontrés lors de la programmation.

Ce travail a mis en évidence l'importance de la réalisation pratique, il est vrai que les systèmes de simulation nous aident, mais ne remplace pas l'expérience. Donc, nous ne devrions pas faire confiance trop aux résultats de simulations qui ne tiennent l'architecture matériel réelle, parce que la connaissance est basée sur l'expérience. Nous pouvons développer par exemple un algorithme sophistiqué mais qui ne trouve pas un noyau pour son implantation ; d'où sa validation expérimentale n'aura pas lieu.

## Bibliographie

- [1] O. YAHIAOUI et T. LABZA, Etude et simulation d'association onduleur de tension-moteur asynchrone et réglage par mode glissant, Université de M'sila: Mémoire de fin d'études Ingénieur d'Etat en génie électrotechnique, 2006/2007.
- [2] Y. KHALED, Estimation en ligne de l'état et des paramètres du moteur asynchrone triphasé, Université de Biskra: Mémoire de Magister, 2005.
- [3] K. YAKOUB, Réduction des effets de la tension homopolaire dans les associations onduleurs multi niveaux moteur à induction, Université de Batna: Mémoire de Magistère, 2005.
- [4] C. Chevassu, «Composants de l'électronique de puissance,» [http://mach.elec.free.fr/divers/polycop\\_composants\\_elec\\_puiss.pdf](http://mach.elec.free.fr/divers/polycop_composants_elec_puiss.pdf), 2005.
- [5] Y. DIAF et D. BEN-ANTER, Modelisation et commande d'un onduleur triphasé piloté par MLI à structure multiniveaux, Université Ouargla: Mémoire de fin d'études Master en génie électrique, 2011/2012.
- [6] F. BERREZZEK, Etude des différentes techniques de commande des onduleurs à MLI associés à une machine asynchrone, Université de Annaba, 2006.
- [7] E. Paris-Londres, Moteurs asynchrones, 1994.
- [8] [http://www.academia.edu/5163800/Chapitre3.\\_microcontrolleur](http://www.academia.edu/5163800/Chapitre3._microcontrolleur).
- [9] Microchip, PIC16F877 Data Sheet 28/40-Pin 8-Bit CMOS FLASH Microcontrollers, DS30292D, 2013.
- [10] Microchip, PIC18F2455/2550/4455/4550 Data Sheet, DS39632C, 2006.
- [11] C. Tavernier, Microcontrôleurs PIC 18 : Description et mise en œuvre (2ed), Dunod, Paris: ISBN 978-2-10-057775-0, 2012.

## ملخص:

إن موجات التوتر هي الأكثر استعمالا في المجال الصناعي، نظرا لمميزاتها المتعددة كالسعر، الاستعمال و الصيانة.

في هذه المذكرة قمنا بدراسة بعض تقنيات التحكم في موج التوتر في ثلاث محاور المحور الاول دراسة موج التوتر تقنيتي التحكم (تقنية MLI و تقنية pleine onde) دراسة نظرية أظهرنا فيها تعريف و تصنيف موج التوتر مستقل وغير مستقل وانواعه وكيفية عمل موج التوتر الثلاثي الأطوار. في المحور الثاني قمنا بدراسة عامة حول المتحكمات الدقيقة وعائلتها وخصائصها، اخترنا في دراستنا نوعين هما PIC16f877A و PIC18f4550 وقمنا بالمقارنة بينهما. في المحور الأخير درسنا تقنية تحكم MLI باستعمال المتحكم الدقيق وشرحنا عدة أمثلة بينا فيها كيفية برمجته بلغة C وتحليل النتائج عن طريق المحاكات في ISIS وفي الأخير استنتجنا الحلول الممكنة.

## Résumé :

Les onduleurs de tension sont les plus utilisés dans les industries à cause de leurs faible prix, très approuvés du côté de la réalisation et facile du côté de la commande électrique.

Dans ce mémoire, on fait l'étude des techniques de commande d'un onduleur de tension à faible puissance. Ce travail est divisé sur trois chapitres :

Le premier chapitre traite la théorie d'un onduleur de tension et ses stratégies de commande : techniques plaines ondes et modulation de largeur d'impulsion MLI. Nous avons fait la classification des onduleurs ainsi que leur fonctionnement.

Le deuxième chapitre concerne l'étude des familles des microcontrôleurs, leurs caractéristiques et fonctionnement. On s'est intéressé en particulier aux PIC16F877A et PIC18F4550, tout en démontrant la déférence entre les deux.

Le dernier chapitre illustre l'implantation des microcontrôleurs en citant des exemples de programmation en langage CCS-PICC. En analysant les résultats obtenus par simulation ISIS, des solutions aux problèmes rencontrés lors de la programmation ont été présentées.