

Remerciements

En préambule à ce mémoire nous remercions ALLAH le tout- puissant de nos avoir donné la patience et le courage durant ces longues années d'études.

Nous souhaitons adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leurs aides et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Ces remerciements vont tout d'abord au corps professoral et administratif du département de l'électronique.

Nous tenons à exprimer notre reconnaissance et nos vifs remerciements à notre encadreur Dr. M. BENACHENE directrice de ce mémoire, elle était toujours à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'elle a bien voulu nous consacrer et sans elle ce mémoire n'aurait jamais vu le jour.

Nous n'oublions pas nos parents pour leurs contributions, leurs soutiens et leurs patiences.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches amis, qui nous ont toujours soutenus et encouragés au cours de la réalisation de ce mémoire.

Merci à tous.

Dédicace

*Je dédie ce travail à la source de tendresse :
mes très chers parents.*

*Tout en étant convaincue que mon succès est une
récompense pour tous leurs sacrifices, qu'ils trouvent
ici l'expression de ma plus profonde gratitude.*

À mes sœurs , À mes frères , A ma fiancée

Et à toute ma famille :

*Ouled jaafri , Tayebi , Elmajdoub , Hattab, Ben
jaafri, Jaafri , Bahidi, Kerrafi , Abdelwafi , Laalmi ,
Dadda, Dahbi, Hadj Belkacem , Merabti.....*

Et à tous mes amis et mes collègues.

À tous ceux que je connais de près ou de loin.

Ouled jaafri Hamza

*Je dédie ce travail à la source de tendresse :
mes très chers parents.*

*Tout en étant convaincue que mon succès est une
récompense pour tous leurs sacrifices, qu'ils trouvent
ici l'expression de ma plus profonde gratitude.*

À mes sœurs , À mes frères ,

Et à toute ma famille :

Hattab, Ouled jaafri ,

Et à tous mes amis et mes collègues.

À tous ceux que je connais de près ou de loin.

Hattab Kaddour

Table des matières

Remerciements.....	I
Dedicace.....	II
Table des matières	III
Table des figures.....	VI
Table des tableaux	VII
LISTE DES ABREVIATIONS.....	X
Résumé	XI
Introduction générale.....	1
I.1 Introduction.....	4
I.2 Historique.....	18
I.3 Généralités	5
I.4 Fonctionnement des moteurs pas a pas	6
I.5.1 Moteurs pas a pas rotatifs	7
I.5.1.1.Moteur pas a pas rotatif a reluctance variable	7
I.5.1.2.Moteur pas a pas rotatif a aimant permanent	9
I.5.1.3. Moteur pas a pas rotatif hybrides.....	12
I.5.2 Moteurs pas a pas linéaires.....	13
I.5.2.1 Moteur pas a pas linéaire a réluctance variable	13
I.5.2.2. Moteur pas a pas linéaire polarises	14
I.5.2.3. Caractères spécifiques aux moteurs linéaires	16
I.6 . Réalisations industrielles	16
I.7 . Performances.....	16
I.8 . Nombre de phase.....	16
I.9 . Le Mode De Séquence	17
I.9.1. Monophasé	17
I.9.2 Biphase	17
I.9.3. Demi –pas.....	18
I.10.Commande des moteurs pas a pas par des circuits intégrés spécialisés.....	18
I.11.Le Critère de choix d'un moteur pas a pas	19
I.12. Conclusion	19
Bibliographie	20

II.1 Introduction	22
II.2. Description du circuit FPGA	22
II.3. Les cinq principaux atouts de la technologie FPGA	24
II.3 .1. Performances	24
II.3. 2. Temps de mise sur le marché.....	25
II.3. 3. Coût.....	25
II.3. 4. Fiabilité.....	25
II.3. 5. Maintenance à long terme	26
II.4. Structure interne des FPGA.....	26
II.4.1. Principe de fonctionnement des FPGA.....	26
II.4.2. Ressources fonctionnelles configurables	27
II.4.2.1. Les éléments de mémorisation.....	28
II.4.2.2. Les éléments de routages.....	29
II.4.2.3. Les éléments d'entrées sorties	30
II.4.2.4. Les éléments de contrôle et d'acheminement des horloges	30
II.5. Principales applications des FPGA.....	31
II.6. Conclusion	31
Bibliographie.....	32
III.1.Introduction	34
III.2.Structure générale d'un système de commande	34
III.3.Commande des moteurs pas a pas.....	34
III.3.1.Fonctionnement a pas entiers	34
III.3.1.1.Commande en mode 1	34
III.3.1.2.Commande en mode 2	37
III.3.2.Fonctionnement a pas réduits	37
III.3.2.1.Commande en mode 3	37
III.3.2.2. Commande en mode 4.....	38
III.3.2.3.Commande en mode 5.....	39
III.4. Alimentation	40
III.4.1. Objectifs d'une bonne alimentation et classification.....	40
III.4.2. Alimentation en tension	40
III.4.2.1 Alimentation unipolaire en tension unique.....	40
III.4.2.2 Alimentation Unipolaire Bitension.....	41
III.5 Paramètres de commande des moteurs pas à pas	41

III.6 Conclusion.....	42
Bibliographie.....	43
Iv.1 Introduction.....	45
IV.2 Descriptif de la carte auxiliaire.....	45
IV.3 Alimentation de puissance de la carte et mesure.....	46
IV.4 Mise en œuvre de la commande d'un moteur pas a pas.....	47
IV.4.1 Descriptif du programme.....	47
Iv.4. 2 Mise en œuvre.....	49
IV.5 Carte FPGA de développement et de formation DE1 (Altera).....	49
IV.6. moteur pas à pas à sens de rotation unique.....	51
IV.7. Moteur pas a pas a double sens de rotation.....	52
IV.8. Conclusion.....	53
Bibliographie.....	54
Conclusion Générale.....	56
ANNEXE.....	57

Table des figures

Figure .I.1: Moteur pas à pas	4
Figure 1.2. Vue d'ensemble d'un moteur pas à pas	5
Figure.I.3 Principe de commande d'un moteur pas à pas.....	7
Figure .I.4. Structure classique d'un moteur à reluctance variable.....	8
Figure .I.5 schéma simplifié d'un moteur à aimant	9
Figure .I.6. Représentation schématique d'un moteur unipolaire.....	10
Figure .I.7. Représentation schématique d'un moteur bipolaire.....	10
figure .I.8. Séquence de commande d'un moteur bipolaire	11
Figure .I.9 Moteur pas à pas hybride	12
Figure .I.10 .Structure interne d'un moteur hybride	3
Figure .I.11 . Moteur pas à pas linéaire à réluctance variable.....	13
Figure .I.12 . Moteur pas à pas linéaire hybride biphasé	14
Figure .I.13 . Distribution du flux dans un moteur linéaire réluctant polarisé	15
Figure .I.14 .Type de fils de moteur pas à pas	17
Figure II-1.a :Architecture interne du FPGA fabriqué ALTERA (Cyclone II) EP2C20) [6].....	23
Figure II-1.b :Architecture interne du FPGA fabriqué Xilinx : XC4000 [3].	23
Figure II-1.B Architecture Interne Du FPGA Fabriqué Xilinx : XC4000 [3]	24
Figure II-2 Architecture traditionnelle d'une plateforme FPGA	26
Figure II-3 :Eléments logiques configurables (circuits Xilinx Virtex4 et Virtex6).....	27
Figure II-4.a :Eléments logiques (LAB) configurables (Logic Blocks, Cyclone II) d' Altera.	27
Figure II.4. Eléments logiques configurables	28
Figure II-5 : Éléments de mémorisation (M4K RAM) d' Altera [6].....	29
Figure II-6 :Élément de routages (Altera) [6].....	30
Figure III-1 Structure générale du commande d'un moteur pas à pas.	34
Figure III-2 Structure quadriphasée à alimentation unipolaire	35
Figure III-3 Principe d'alimentation en mode 1	35
Figure III-4 Principe de l'alimentation bipolaire en mode 1	36
Figure III-5 Principe de l'alimentation d'un bobinage biphasé en mode 1	36
Figure III-6 Principe d'alimentation en mode 2	37

Figure III-7	Principe d'alimentation en mode 3	38
Figure III-8	Principe d'alimentation en mode 4	38
Figure III-9	Pilotage par modulation d'amplitude des courants.....	39
Figure III-11	Alimentation unipolaire bitension.....	41
Figure.IV. 1	La carte auxiliaire d'un circuit L298	45
Figure.IV. 2	commander un moteur pas à pas par FPGA	46
Figure. IV. 3	schéma de programme implanté dans le FPGA.....	47
Figure.IV. 4	schéma représentés les chronogrammes	48
Figure. IV. 5	schéma représentent La machine d'état	48
Figure.IV. 6	Carte de développement et de formation DE1(Altera)	50
Figure. IV. 7	la Carte Réalisé	51

Table des tableaux

Tableau I.1.	séquence d'un moteur monophasé.....	17
Tableau I.2.	séquence d'un moteur biphasé.....	18
Tableau I.3.	séquence d'un moteur Demi -pas.....	18

LISTE DES ABREVIATIONS

FPGA	Field Programmable Gates Arrays
ASIC	Application Specific Integrated Circuit
NRE	Non Recurrent Engering
DSP	Digital signal processors
E/S	Entré/Sortie
M.R.V	Moteurs Pas A Pas A Réductance Variable
M.A.P	Moteur Pas A Pas A Aimant Permanent
CLB	Configurable Logic Blocks
PS	Programmable Switch
LUT	Look Up Table
(des PLLs , des DLLs et BDTI)	Des Eléments D'asservissement Des Horloges
VHDL	Very high speed - or scale - integrated circuits Hardware Description Language

Résumé

L'objectif de ce travail est d'élaborer la commande par circuit FPGA et VHDL d'un moteur pas à pas en temps réel. Nous avons réalisé l'interface de commande entre la carte FPGA, DE1 d'ALTERA, avec son alimentation.

Nous avons implanté les programmes de commande du moteur pas à pas à l'aide de logiciel Quartus II 9.0 sur le circuit FPGA cyclone II de la carte DE1 d'ALTERA.

Abstract

The subject of this work is to develop the FPGA and VHDL circuit control of a stepper motor in real time. We have realized the control interface between the ALTERA FPGA board, DE1 and its power supply.

We have implemented the stepper motor control programs using Quartus II 9.0 software on the ALTERA DE1 board cyclone II FPGA circuit.

المخلص

والهدف من هذا العمل هو تطوير تحكم FPGA و VHDL على محركات خطوة خطوة في زمن حقيقي ونحن نعمل من جهة على التحكم في المحركات داخل بطاقة FPGA، DE1 من شركة ALTERA، مع مصدر تغذيتها. كما نفذنا برامج السيطرة على محركات خطوة خطوة باستخدام برنامج كوارتس II9.0 على بطاقة FPGA.

Introduction générale

L'implantation des algorithmes de contrôle dans leur intégralité sur des cibles matérielles telles que les FPGA est une démarche qui nécessite une parfaite maîtrise des processus de conception et un travail spécifique d'adéquation entre l'algorithme et l'architecture de commande à intégrer. Donc, un savoir-faire méthodologique est nécessaire aux concepteurs utilisant les composants de type FPGA afin de satisfaire l'ensemble des contraintes inhérentes de l'implantation, tout en apportant une flexibilité de développement suffisante. Par ailleurs, les applications de commande sont des applications qui décloisonnent un savoir-faire dans plusieurs domaines. En effet, cela nécessite du concepteur la maîtrise d'un savoir-faire dans les domaines de la microélectronique, de l'électronique faible et grande puissance, des machines électriques et de leur commande. Cette difficulté pousse les concepteurs à préférer les implantations standards des solutions logicielles. Par conséquent, l'implantation des algorithmes de contrôle sur de nouvelles solutions matérielles tels que les FPGA doit suivre des étapes bien déterminées afin de guider le concepteur et faciliter le processus de conception [1].

Le moteur pas à pas constitue un convertisseur électromécanique destiné à transformer le signal électrique (impulsion) en déplacement (angulaire ou linéaire) mécanique. Au point de vue électrotechnique, le moteur classique ressemble à la machine synchrone, dont le stator (le plus souvent à pôles saillants) porte les enroulements de pilotage et le rotor (presque toujours à pôles saillants) est soit muni d'aimants permanents (structure dite polarisée ou active), soit constitué par une pièce ferromagnétique dentée (structure dite reluctance ou passive).

Entre le moteur et son alimentation, sont intercalés trois éléments essentiels

- une unité de calcul, qui élabore les impulsions de commande.
- un modulateur PWM, qui génère les commandes des contacteurs électroniques de commutation.
- une électronique de commutation (puissance), qui, à partir d'une alimentation, fournit l'énergie vers les enroulements appropriés du moteur.

Les FPGA, sigle anglais qui signifie « Field Programmable Gates Arrays » traduit en français par réseau de portes programmables, sont des circuits intégrés reprogrammables. Ils offrent la possibilité de réaliser des fonctions numériques plus ou moins complexes, tout comme leurs homologues figés : les ASIC. [2] [3]

Dans le premier et le deuxième chapitre de ce mémoire nous avons présenté quelques définitions et concepts des moteurs pas à pas, notamment leur fonctionnement, les technologies de leurs réalisations industrielles et enfin le critère de choix d'un moteur pas à pas. Pour bien situer notre application nous avons aussi présenté les circuits FPGA, leurs structures internes et leurs principales applications.

Il ressort que le circuit programmable de type FPGA est l'élément le plus important dans ce projet. C'est la solution choisie pour la commande des moteurs pas à pas dans notre cas.

Nous avons établi les algorithmes de commande des moteurs pas à pas en langage de description des circuits logiques VHDL.

Nous avons présenté dans le troisième chapitre la commande des moteurs pas à pas, dans ce chapitre on utilise un circuit de puissance (L298) et on utilise la carte de développement DE1 de la firme Altera contenant le circuit FPGA cyclone II EP2C20F484C7 pour commander les moteurs pas à pas. Nous avons vu les différentes commandes au plusieur. Différentes stratégies de commande des moteurs pas à pas ont été présentées : commande en pas entier et commande en demi pas et soit en mode unipolaire, ou bi-polaire, monophasé ou biphasé et dans les deux sens de rotation horaire et anti-horaire.

Selon les applications, les performances souhaitées, on rencontre différents types de moteurs à réluctance variable, qui développent un couple proportionnel au carré du courant, ne nécessitent que des alimentations unipolaires (courant généré dans un seul sens). Les autres moteurs nécessitent des courants bipolaires.

Dans le dernier chapitre, nous avons présenté les résultats de la réalisation de la commande d'un moteur pas à pas par l'association de la carte de développement DE1 et d'une carte auxiliaire intégrant les circuits de puissance. Nous avons noté que le bloc « div_1s » nous génère un signal d'horloge de période 1 seconde, en divisant par 50 millions l'horloge à 50 MHz de la carte DE1. Cette horloge d'une seconde va commander un séquenceur, permettant au moteur de fonctionner dans un sens ou dans l'autre suivant que le bouton poussoir relié à l'entrée *s* est appuyé ou non, et en mode « pas entiers une phase » ou en mode « demi-pas » suivant que le bouton poussoir associé à l'entrée « *p_demip* » est appuyé ou non.

La sortie du séquenceur envoie les signaux adéquats sur les entrées in1 à in4 du circuit L298 à chaque état du moteur pas à pas.



CHAPITRE I

moteur pas à pas

I.1 Introduction

Le moteur pas à pas est un convertisseur électromécanique qui assure la transformation d'un signal électrique impulsionnel en un déplacement mécanique (angulaire ou linéaire). Sa structure de base se présente sous la forme de deux pièces séparées mécaniquement, le Stator et le Rotor. L'interaction électromagnétique entre ces deux parties assure la rotation [4].

Il existe trois types de moteur pas à pas: les moteurs à aimants permanents et les moteurs à réluctance variable et les moteurs hybrides [5].

Les moteurs pas à pas sont des transducteurs électromécaniques qui assurent la conversion des signaux électriques digitaux ou impulsionnels en mouvement de rotation ou de translation de type incrémental.

Les moteurs pas à pas, utilisés pratiquement dans tous les composants d'un système informatique, sont très mal connus, surtout en ce qui concerne leur mode de commande.

I.2 Historique

les premiers moteurs pas -à- pas à réluctances variable ont été utilisés par la marine de guerre britannique dans les années 1920 pour déplacer les indicateurs de direction de lance -torpilles et des canons. Dans les années 1930, l'ingénieur Marius Lavet a découvert un type particulier de moteur pas à pas à aimant, connu maintenant sous le nom de moteur lavet , qui a permis le développement de ce dispositif dans le domaines de l'horlogerie grâce à sa miniaturisation et à son faible cout.

Il équipe aujourd'hui presque toutes les montres à aiguilles .Le moteur pas à pas classique est apparu dans les années 1940, mais c'est l'avènement de l'électronique numérique dans les années 1960 qui a permis son développement [6].



Figure .I.1: Moteur pas à pas

I.3 Généralités

Ils existent trois types de moteurs pas à pas : les moteurs à aimant permanent, les moteurs à réluctance variable ainsi que les moteurs hybrides. Nous verrons plus tard que les moteurs à aimants permanents se subdivisent en deux catégories.

Malgré les différences existantes entre les moteurs, le résultat recherché est l'avance d'un seul pas, c'est-à-dire la rotation de leur axe suivant un angle déterminé à chaque impulsion que l'une ou l'autre de leurs bobines recevra. Cet angle qui varie selon la constitution interne du moteur, est en général compris entre 0.9° et 90° .

Les moteurs les plus couramment rencontrés présentent des pas de :

- 0.9° soit 400 pas par tour.
- 1.8° soit 200 pas par tour.
- 3.6° soit 100 pas par tour.
- 7.5° soit 46 pas par tour.
- 15° soit 24 pas par tour.

Il est évident que les moteurs pas à pas, de par leurs technologies, présentent une très grande précision et une durée de vie quasi illimitée, l'usure mécanique étant pratiquement inexistante (absence de frottement). La figure I.2 en représente l'aspect externe. Leur domaine de prédilection sera donc ceux où la précision est de rigueur : les constituants mécaniques de l'informatique et la robotique.

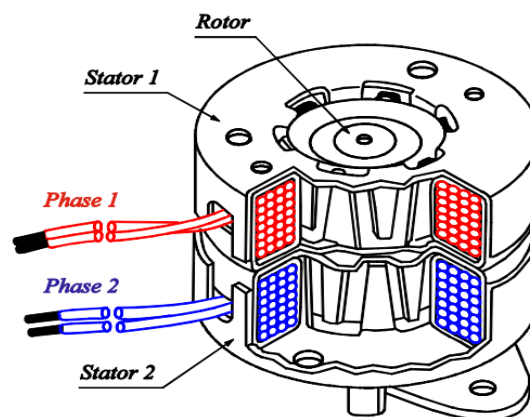


Figure 1.2. Vue d'ensemble d'un moteur pas à pas

I.4 Fonctionnement des moteurs pas à pas

Un moteur pas à pas est un appareil dont les différents enroulements sont alimentés l'un après l'autre, chaque commutation provoquant un déplacement angulaire (ou linéaire) donné, appelé "pas" et qui est la caractéristique du moteur. Le sens de rotation est évidemment défini par le sens dans lequel les enroulements ou groupes d'enroulements sont excités.

Un moteur pas à pas rotatif peut être comparé à une série d'enroulements disposés en cercle qui, lorsqu'ils sont excités successivement, réagissent en le faisant tourner d'un angle θ .

Le couple ou la réaction n'ont pas une valeur constante pendant le déplacement θ (ou x : déplacement linéaire). Cependant, la variation du couple entre le point 1 et le point 2 sera répétée du point 2 au point 3; et ainsi de suite.

Il y a deux types de moteurs pas à pas. Les premiers fonctionnent par réaction entre un champ électromagnétique et un aimant permanent. Ce type est désigné sous le nom de moteur pas à pas à aimant permanent (M.A.P). Les seconds fonctionnent par la réaction entre un champ électromagnétique et un rotor en fer doux. Ils sont désignés sous le nom de moteurs pas à pas à réluctance variable (M.R.V) et qui seront le centre de notre étude.

L'assemblage des deux principes de fonctionnements donne naissance au troisième type, qui regroupe les moteurs hybrides.

Les moteurs pas à pas ont beaucoup d'avantages sur les systèmes linéaires, ils offrent:

- Une réponse très rapide (temps de réponse jusqu'à moins de 1 milliseconde);
- Insensibilité aux variations linéaires;
- Longue durée de vie (jusqu'à 1 milliards de cycles);
- Positionnement précis;
- Insensibilité aux variations de la tension et de l'amplitude des impulsions.
- La versatilité de ces moteurs est aussi très importante. Ils peuvent être

utilisés comme:

- Moteurs à fréquence variable (élimination du variateur de vitesse);

- Moteurs à courant continu sans balais;
- Servomoteurs en circuit ouvert pour éliminer la boucle de retour;
- Moteur à entraînement numérique;
- Moteurs synchrones;
- Compteurs d'impulsions (intégrateurs).

Ils permettent en particulier un positionnement précis sans asservissement.

A chaque impulsion du signal de commande correspond au niveau du rotor un déplacement angulaire défini appelé « pas » ou « incrément mécanique ». La vitesse de rotation est fonction de la fréquence des impulsions. [7]

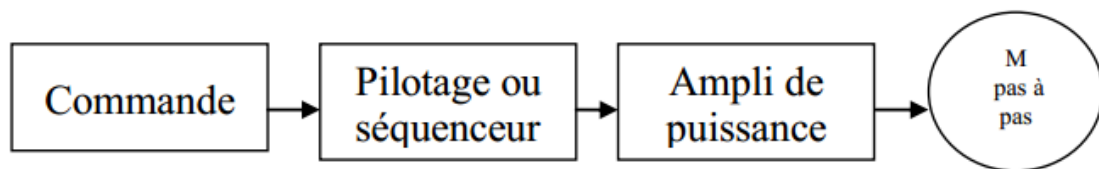


Figure.I.3 Principe de commande d'un moteur pas à pas.

I.5 Technologie des moteurs pas a pas :

Un actionneur électrique peut créer deux types de mouvements : un mouvement de rotation ou un mouvement de translation.

I.5.1 Moteurs pas a pas rotatifs

Les moteurs pas à pas peuvent être classés en fonction du phénomène physique qui est à l'origine de leur mouvement. On distingue principalement, deux catégories de moteurs pas à pas : les moteurs à reluctance variable et les moteurs polarisés.

I.5.1.1.Moteur pas a pas rotatif a reluctance variable

La rotation d'un moteur à reluctance est engendrée par la réaction, entre un champ magnétique statorique et un rotor saillant, qui conduit à une disposition alignée de la partie saillante avec le pôle créé par le champ magnétique. Dans ces conditions le flux est maximum et la réluctance du circuit magnétique est minimum.

Ce type de moteur est caractérisé par une structure dentée au niveau du rotor et du stator. Le rotor est fabriqué en acier doux non magnétique. Le nombre de bobines dans le stator

et le type de connexion déterminent le nombre de phases du moteur.

Il s'agit d'un circuit magnétique simple constitué de deux pièces ferromagnétiques, l'une fixe (stator) comportant un enroulement à N spires parcourues par un courant I et l'autre mobile (rotor) autour d'un axe fixe. Soit θ_m l'angle entre l'axe longitudinal de la pièce mobile et celui interpolaire de la pièce fixe.

A chaque impulsion de la commande, la phase suivante du stator est alimentée. On constate que les pôles les plus proches des bobines alimentées se positionnent en face de ces dernières. Suivant l'ordre d'alimentation des phases du stator, on peut choisir le sens de rotation.

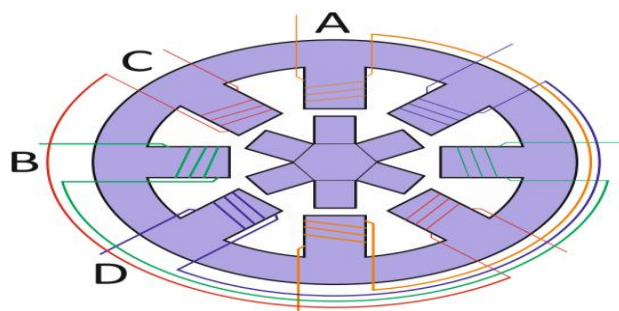


Figure .I.4. Structure classique d'un moteur à reluctance

Chaque couple de deux bobines statoriques diamétralement opposées et connectées en série, constitue une des quatre phases de la machine considérée. L'alimentation d'une phase crée un couple permettant le déplacement du rotor vers une position d'équilibre qu'il garde tant que l'alimentation est maintenue. Cet état d'équilibre correspond à une position alignée entre les dents statoriques porteuses de la phase alimentée et les dents rotoriques. Chaque nouvelle séquence d'alimentation établit un nouvel équilibre. Le moteur se déplace donc avec un pas angulaire de 15° .

Le mode de commande peut dans ce cas, de la même façon que pour les autres moteurs, être monophasé, biphasé ou demi-pas.

Pour augmenter la résolution angulaire de ces moteurs, des constructions à circuits magnétiques multiples, dites multistack, peuvent être envisagées. Ces moteurs se présentent principalement sous deux formes :

- Un empilage de plusieurs machines monophasées identiques assemblées mécaniquement sur le même arbre.
- Un stator unique, possédant plusieurs rotors magnétiquement indépendants, mais couplés mécaniquement.

Dans toutes ces machines, le rotor et les stators présentent le même nombre de dents, le mouvement étant obtenu par le décalage mécanique des rotors ou des stators, les uns par rapport aux autres.

I.5.1.2. Moteur pas à pas rotatif à aimant permanent

Les moteurs à aimant permanent sont constitués d'un stator supportant les bobinages et d'un rotor magnétique (aimant bipolaire). Cette catégorie de moteurs se subdivise en deux types : le moteur unipolaire et le moteur bipolaire.

La figure .I.5 représente le schéma simplifié d'un moteur à aimant. Le premier dessin de la figure le représente dans sa position de repos, lorsque les bobinages ne sont traversés par aucun courant. Chacun des pôles du rotor se place devant une paire des plots du stator.

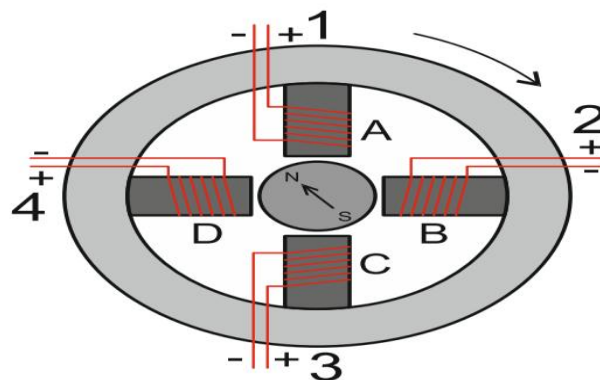


Figure .I.5 schéma simplifié d'un moteur à aimant

A. Les moteurs unipolaires

Une représentation schématisée d'un moteur unipolaire. Afin d'inverser le sens du courant, les enroulements sont réalisés au moyen de deux fils dont l'une des extrémités est reliée au pôle + ou au pôle - de l'alimentation. La commande de ce type moteur est la plus

simple de tous les moteurs pas à pas puisqu'il suffira d'alimenter les bobinages à tour de rôle pour faire tourner l'axe d'un pas.

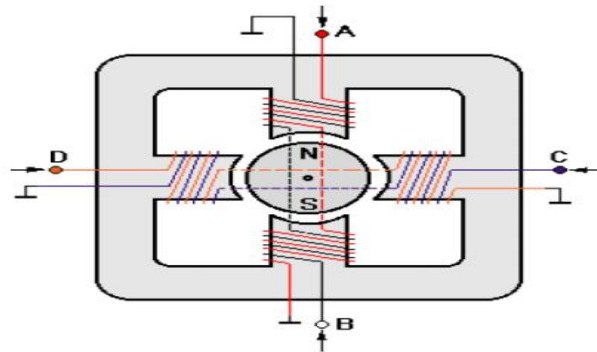


Figure .I.6.Représentation schématique d'un moteur unipolaire

Comme pour chaque type de moteur, le modèle unipolaire peut être commandé en mode monophasé, biphasé ou demi-pas.

B. Les moteurs bipolaires

Ce type de moteur nécessite une commande plus complexe que celle du moteur unipolaire, le courant devant changer de sens dans les enroulements à chaque pas effectué.

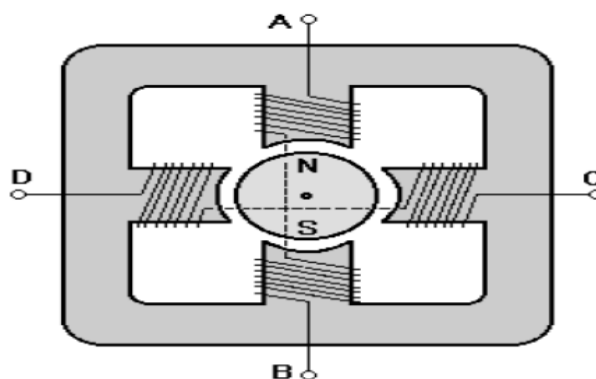
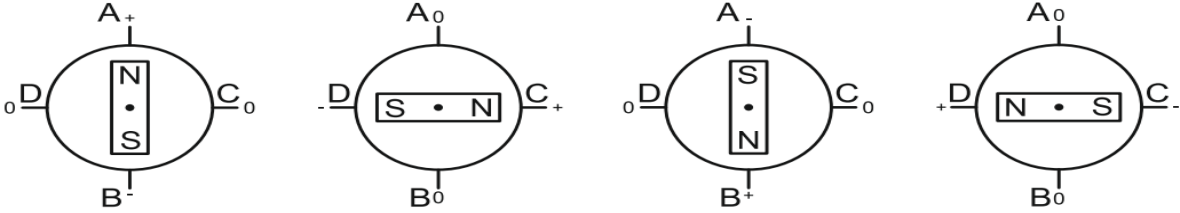


Figure .I.7.Représentation schématique d'un moteur bipolaire

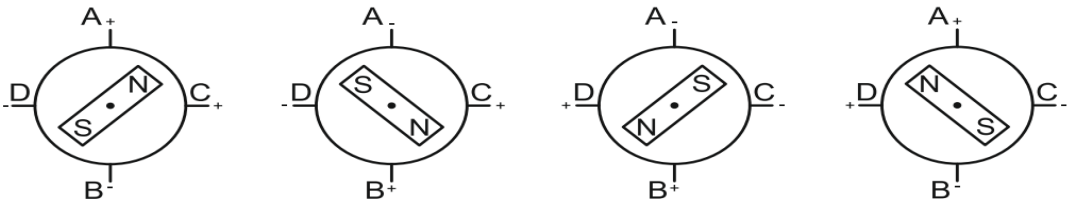
Comme pour le modèle précédent, ce moteur peut être alimenté sous trois séquences différentes, représentées par ailleurs sur la figure .I.8.

Dans le mode monophasé, le couple n'est pas très important puisqu'un seul enroulement est alimenté pour effectuer un pas.

Mode monophasé



Mode biphase



Mode demi-pas

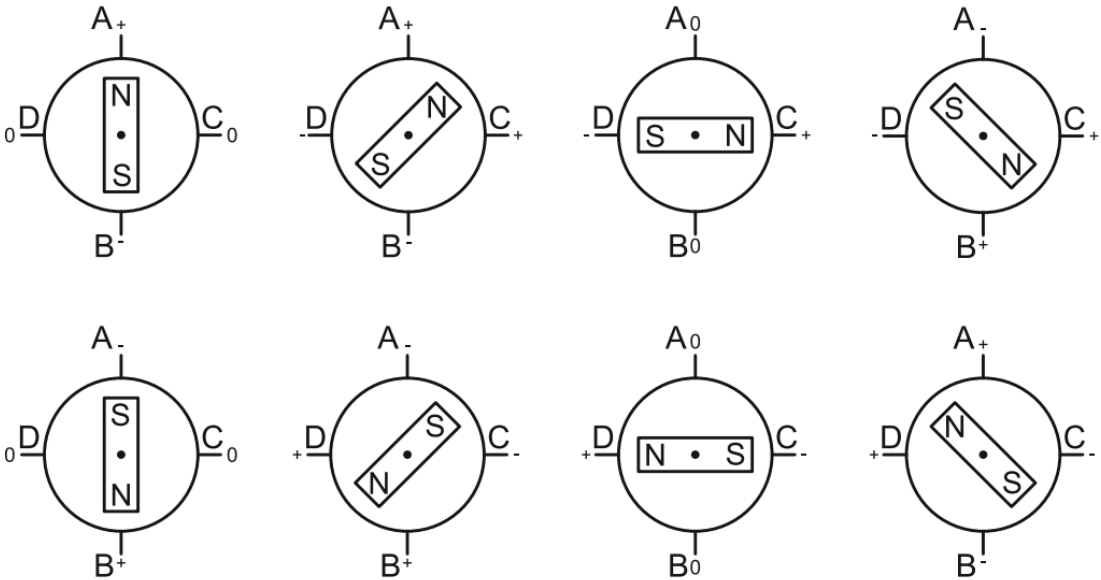


figure .I.8.Séquences de commande d'un moteur bipolaire

C'est dans le mode biphasé que le moteur développera la plus grande puissance (couple élevé) car les deux phases seront alimentées en même temps. Le mode demi-pas permet de doubler le nombre de pas qu'un moteur peut effectuer par tour, mais cette précision supplémentaire engendre un couple irrégulier. En effet, dans ce mode, la commande du moteur est un mélange de mode biphasé et monophasé. Dans ce cas, si la charge est importante, des pas risquent de « sauter », ce qui n'est pas le but recherché.

I.5.1.3. Moteur pas à pas rotatif hybrides

Les moteurs pas à pas hybrides sont généralement constitués d'un rotor denté muni d'aimants permanents. Le rotor possède généralement deux disques polaires décalés d'un angle électrique π .

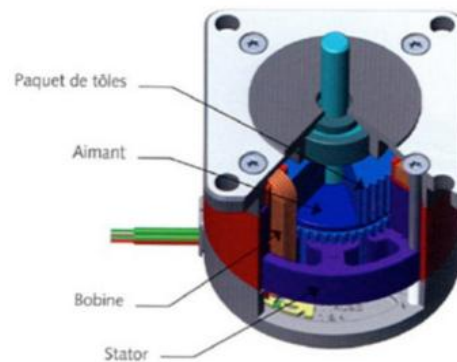
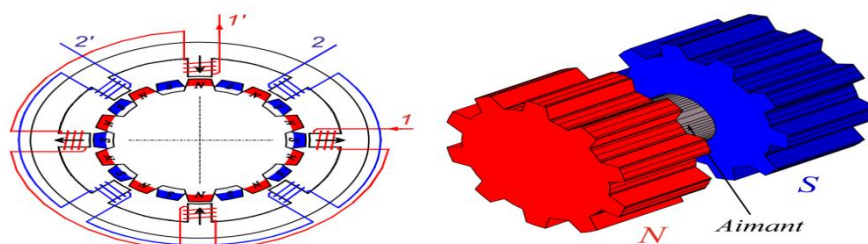


Figure .I.9 Moteur pas à pas hybride

Ce type de moteur présente à la fois les avantages du moteur à aimant permanent qui possède un couple élevé et ceux du moteur à réluctance variable qui permet d'obtenir un nombre importants de pas par tour. Toutefois l'inertie d'un tel rotor ainsi que les pertes fer sont relativement importantes et pénalisent donc cette structure.



I.5.2 Moteurs pas a pas linéaires

Tout comme pour les versions rotatives, les moteurs pas à pas linéaires se composent d'un stator portant les bobinages et d'une partie mobile se déplaçant cette fois-ci linéairement.

Cette partie mobile peut être réalisée à partir d'une pièce ferromagnétique dentée, dans ce cas la structure est dite à réluctance ou passive ; ou alors la partie mobile est munie d'aimants permanents et dans ce cas la structure est dite polarisée ou active.

I.5.2.1 Moteur pas a pas linéaire a réluctance variable

Ce type de moteurs est caractérisé par une structure dentée au niveau du stator et de la partie mobile. Le circuit magnétique est généralement assemblé à partir de tôles magnétiques de forte perméabilité. Toute fois, les machines destinées au positionnement ou à une translation lente, peuvent être réalisées avec des pièces massives. Les enroulements du circuit électrique sont généralement concentrés autour des plots du stator et donc faciles à réaliser. La figure .I.11 présente une structure linéaire à réluctance variable à deux modules stator, chaque module comporte deux plots autour des quels sont enroulées les bobines. Chaque phase de la machine est formée par la mise en série des deux bobines d'un même module. Des séparations amagnétiques sont nécessaires entre les différents modules statoriques afin d'imposer un décalage. En effet, si les plots d'un module sont alignés avec les dents de la partie mobile, les plots des autres modules stator doivent être décalés pour pouvoir créer une force de déplacement.

Chaque phase de la machine impose à la partie mobile une position d'équilibre correspondant à l'alignement de ses dents avec les plots statoriques de la phase alimentée.

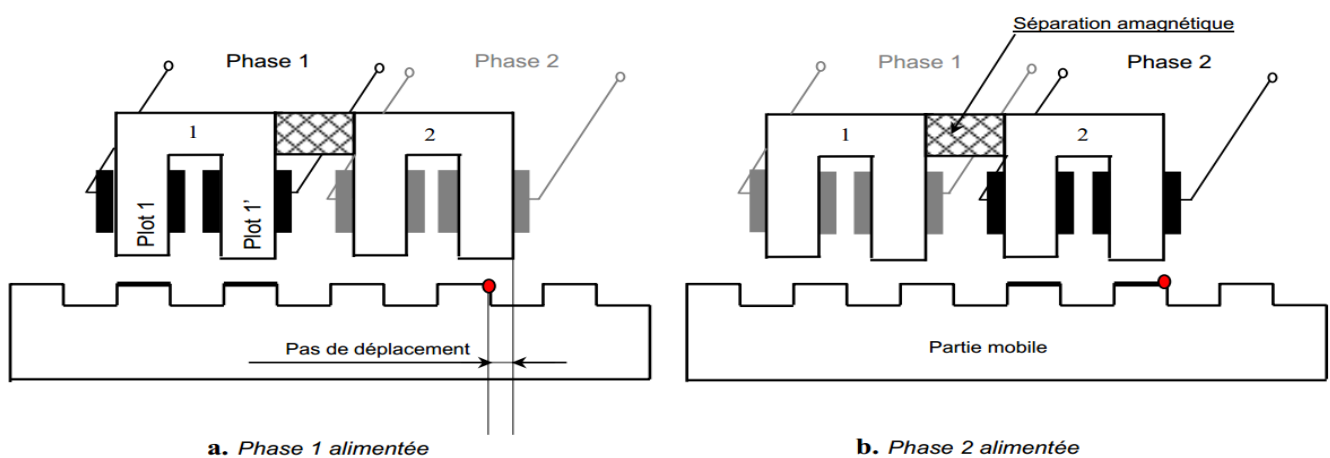


Figure .I.11 . Moteur pas à pas linéaire à réluctance variable

Sur la figure I.11.a la partie mobile est alignée avec les dents du module 1, ensuite, sur la figure I.11.b la partie mobile vient aligner ses dents avec celles du module 2.

I.5.2.2. Moteur pas a pas linéaire polarises

A . Les moteurs a aimant permanent

Ce sont des moteurs dont la partie mobile est généralement lisse et formée d'une succession d'aimants permanents montés en surface.

B . Les moteurs hybrides

Le mouvement des moteurs hybrides résulte de la superposition de la force développée par l'effet réductant des dents et de la force créée par l'aimant. La contribution des amplitudes et des périodes géométrique de ces forces permet de réaliser des caractéristiques statiques très diversifiées. En effet, l'aimant, placé dans la structure hybride, assure une certaine distribution des lignes de champ. L'alimentation des bobines produit un phénomène d'aiguillage des lignes de champs plus au moins important suivant l'intensité du courant d'alimentation. En jouant sur l'orientation des lignes de champs, il est possible de maîtriser la variation de la force résultante.

La figure I.12 représente un moteur linéaire réductant polarisé bipolaire où l'aimant et les bobines sont placés sur le même support fixe.

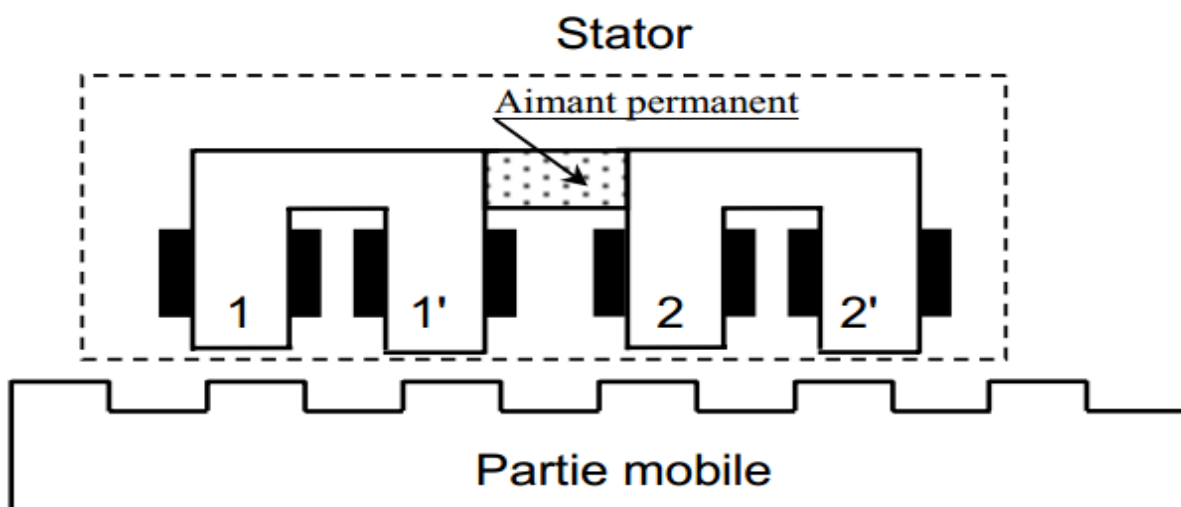


Figure I.12 . Moteur pas à pas linéaire hybride biphasé

Les deux paires de plots adjacents (1,1') et (2,2'), de la structure de la figure I.13.a sont parcourus dans le même sens par le flux de l'aimant. Les quatre bobines forment deux phases constituées chacune par deux bobines montées en série. Si le sens d'enroulement des bobines est opposé, alors le flux créé par une phase circule conformément à la distribution de la figure I.13.b. La superposition du flux créé par l'aimant à celui créé par la bobine de la phase 1 produit une nouvelle distribution des lignes de flux, donc de la force, figure I.13.c.

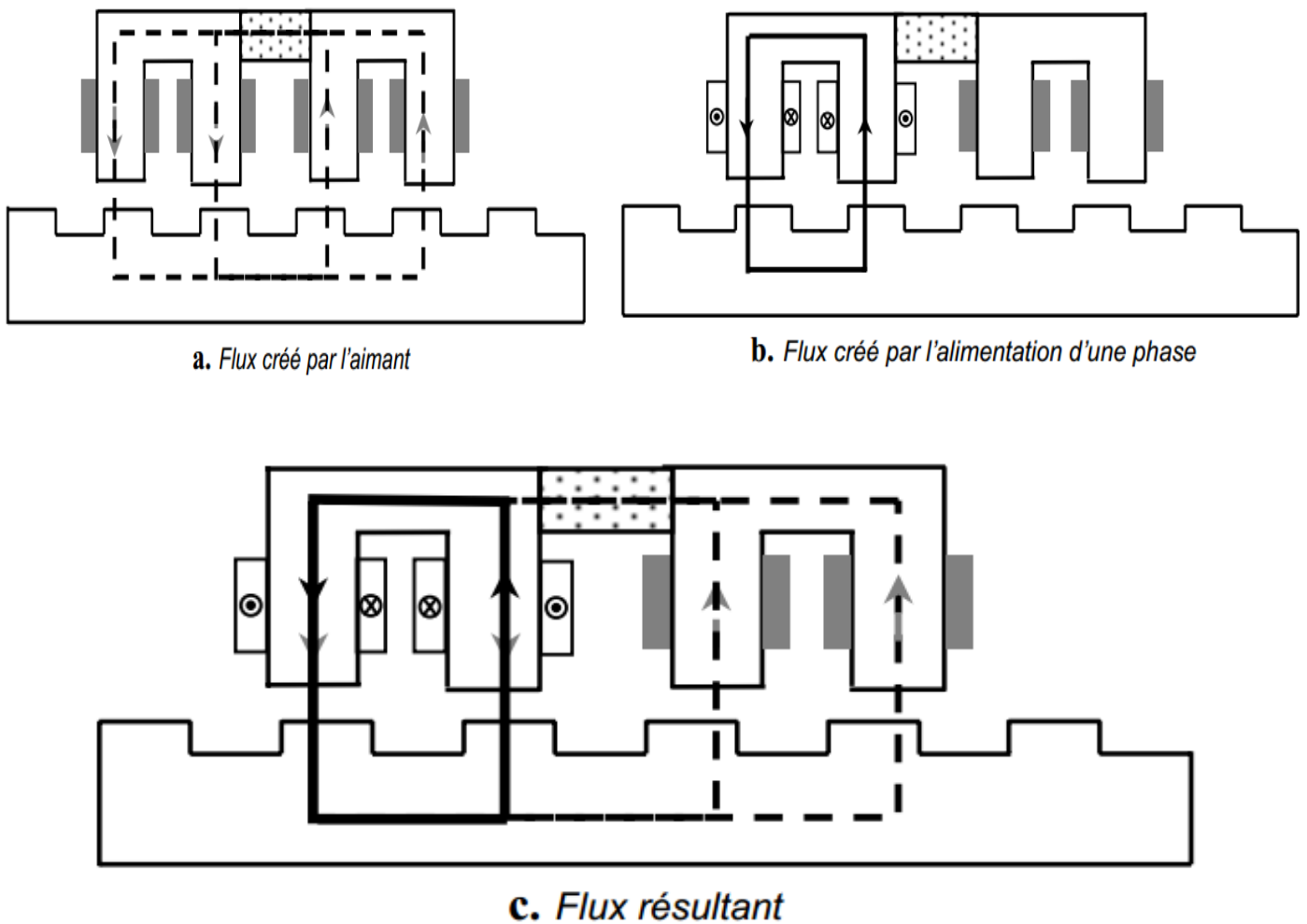


Figure .I.13 . Distribution du flux dans un moteur linéaire réductant polarisé

Pour une structure où l'aimant est fixé sur la partie mobile, les flux développés par la réluctance variable et l'aimant obéissent aux mêmes conventions que dans le cas précédent.

I.5.2.3. Caractères spécifiques aux moteurs linéaires

Un moteur linéaire génère directement une force de déplacement et un mouvement de translation contrairement aux moteurs rotatifs qui pour y arriver nécessitent l'utilisation d'organes spécifiques de transmission afin d'assurer la conversion du mouvement rotatif en linéaire.

L'attrait pour l'entraînement linéaire provient directement des avantages intrinsèques de l'entraînement direct qui sont : une grande dynamique du mouvement, une bonne précision de positionnement, une durée de vie importante, une grande fiabilité, la réduction du nombre de composants pour réaliser un axe motorisé, la flexibilité et la souplesse .

Toutefois, ces moteurs linéaires, présentent des spécificités, dues à la disposition de leurs parties mobiles par rapport à leurs parties fixes, notamment des systèmes de guidage particuliers, entraînant de nouvelles contraintes au niveau de l'entrefer magnétique. De plus, de part leur structure géométrique, les moteurs linéaires présentent des effets d'extrémités.

Ces différences technologiques par rapport aux moteurs rotatifs doivent être prises en compte au moment de la conception et de la réalisation pratique.

I.6 . Réalisations industrielles

La structure habituellement réalisée correspond à celle du schéma de principe. Le rotor est formé de tôles magnétiques. L'aimant est en ferrites ou en terre rares. Les demi-rotors dentés sont en matériau ferromagnétique feuilleté. Pour obtenir un nombre de pas par tour élevé, les plots du stator sont dentés avec le même pas que pour le rotor. Plusieurs variantes existent dans la disposition des éléments, mais toutes conduisent à des caractéristiques similaires.

I.7 . Performances

On combine un couple élevé dû à l'action de l'aimant avec une bonne résolution. On obtient ainsi couramment 200 ou 400 pas par tour et même nettement plus pour des modèles spécifiques. La machine présente un couple de détente comme tout moteur ayant des aimants et des pôles saillants. Le prix est évidemment supérieur à celui des types précédents, mais les excellentes performances en font un des moteurs pas-à-pas les plus utilisés.

I.8 . Nombre de phase

Il existe divers types de moteurs pas à pas mais la principale différence est le nombre de bobines utilisées. Configuration interne des bobines du moteur :

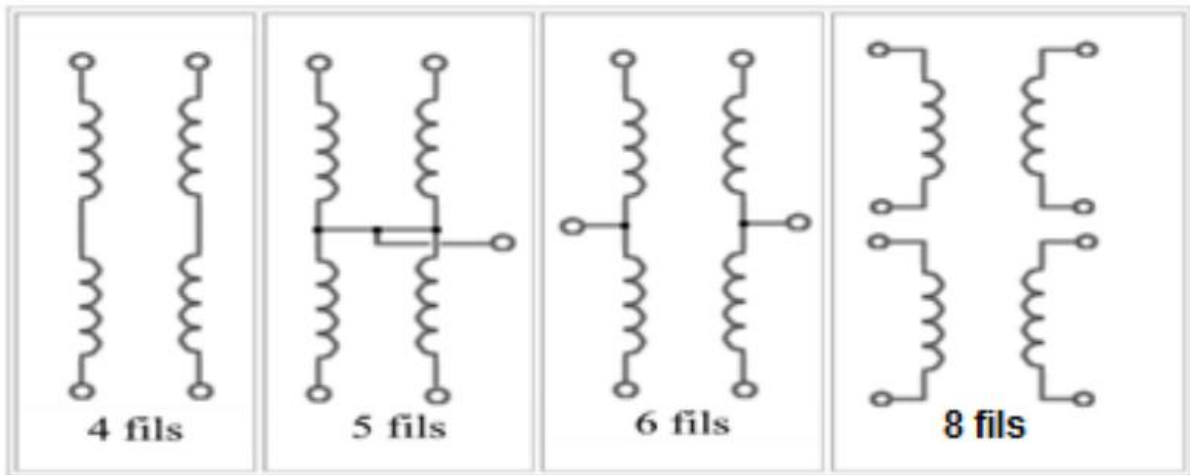


Figure .I.14 .Type de fils de moteur pas à pas

I.9 . Le Mode De Séquence

Les modes de séquence se distinguent par leur fonctionnement interne, leur séquence générée, et par la façon dont on les commande.

I.9.1. Monophasé

Cette séquence permet normalement d'alimenter une bobine à la fois, en plaçant à des positions de détente.

A	0	0	0	1
B	0	1	0	0
C	0	0	1	0
D	1	0	0	0

Tableau I.1. séquence d'un moteur monophasé

I.9.2 Biphase

Cette fonction permettra au moteur de faire un pas complet ou les positions d'équilibre sont à mi-chemin entre les positions de la séquence la plus simple.

A	1	0	0	1
B	0	1	1	0
C	1	1	0	0
D	0	0	1	1

Tableau I.2. séquence d'un moteur biphasé

I.9.3. Demi –pas

Dans ce cas, on combine les deux séquences précédentes pour placer le rotor successivement à une position de détente, à une position intermédiaire, et à nouveau à une position de détente. Ce mode permettra de multiplier par deux le nombre de pas.

L'avantage de ce mode à demi -pas qu'il ne souffre pas de problème de résonances, même s'il tourne très vite, on ne tombe jamais sur un point de résonance ou le couple s'annule.

A	1	1	0	0	0	0	0	1
B	0	0	0	1	1	1	0	0
C	0	1	1	1	0	0	0	0
D	0	0	0	0	0	1	1	1

Tableau I.3. séquence d'un moteur Demi -pas

I.10. Commande des moteurs pas à pas par des circuits intégrés spécialisés

Les deux dispositifs pour la commande du moteur pas à pas qui sont le séquenceur et l'interface de puissance avec le développement scientifique sont intégrés tout les deux dans un seul circuit qui renferme 4 étages de commande (Position, direction, amplification et vitesse). Et en reste alors qu'à délivrer les impulsions pour le nombre de pas et un but pour le sens de rotation par le moyen de deux lignes de données du port d'un microcontrôleur par exemple, pour faire tourner le moteur de plus, ces circuits intégrés protègent très bien l'unité de pilotage d'un

retour de courant. On a comme exemple de ces circuits intégrés [8]:

- Le (L297) et (1298) qui fonctionnent ensemble.
- Le (SAAI027) et (SAA1042).
- Le (SL7024M) ; le (ULN 2003) et le (ULN 2803)

I.11. Le Critère de choix d'un moteur pas à pas

Le choix se fait comme suit [8]:

- Le mode de commande, soit bipolaire ou unipolaire.
- Le nombre de pas par tour.
- La fréquence de travail
- La puissance du moteur.

I.12. Conclusion

Dans ce chapitre, nous avons donné un aperçu général sur les différents types de moteurs pas à pas, ainsi que leur mode de fonctionnement.

Voici leurs avantages :

- Rotation constante pour chaque commande (précision meilleure que 5% d'un pas).
- Existence de couple à l'arrêt.
- Contrôle de la position, de la vitesse et synchronisation de plusieurs moteurs.
- Moteur sans balai.

Et les inconvénients :

- Plus difficile à faire fonctionner qu'un moteur à courant continu.
- Vitesse et couple relativement faible.
- Couple décroissant rapidement lorsque la vitesse augmente.
- Résonance mécanique.

Bibliographie

- [1]. <http://www.techniques-ingenieur.fr/base-documentaire/energies-th4/convertisseurs-electriques-et-applications-42253210/commande-numerique-a-base-de-composants-fpga-d-une-machine-synchrone-d2902/methodologie-de-developpement-d2902niv10004.html>.
- [2] TISSERAND Arnaud : Introduction aux circuits FPGA, Présentation Séminaire MIM,2003
<http://www.irisa.fr/prive/Arnaud.Tisserand/docs/semim-at-fpga.pdf>
- [3] ALTERA Corporation : Site d'un constructeur de FPGA, Ensemble de documents techniques concernant les FPGA, 2010
<http://www.altera.com/literature/lit-index.html>
- [4] « Systèmes électromécaniques » ; Haute Ecole d'ingénierie et de Gestion Du Canton du Vaud, CD/SEM/Cours/Chap07
- [5] : A. ZENATI, S. KERROUCHE. « Commande de deux moteurs pas à pas via le port parallèle avec Delphi » Mémoire d'ingénieur d'état en électronique, Université Bejaia, Juin2009.
- [6] Pierre Mayé : « Moteurs électriques pour la robotique » ; 2^{ème} édition, octobre 2013
- [7] Hachette Technique, Claude Divoux , « Guide du technicien en électrotechnique » , éd 1999.
- [8] : H. YAKOUBI, M. SOHBI, B. OGAB. « Réalisation d'un robot mobile autonome », Mémoire d'ingénieur d'état en automatique, Université Setif, Juin 2009.



CHAPITRE II

Les circuit FPGA

II.1 Introduction

Les FPGA, sigle anglais qui signifie « Field Programmable Gates Arrays » traduit en français par réseau de portes programmables, sont des circuits intégrés reprogrammables. Ils offrent la possibilité de réaliser des fonctions numériques plus ou moins complexes, tout comme leurs homologues figés : les ASIC [1].

Les FPGA sont des circuits numériques matériels configurables dédiés à l'électronique numériques [2].

Le FPGA disposent d'une importante quantité (dépendant de la technologie utilisée) de ressources matérielles opérationnelles dont on peut configurer des fonction logique combinatoire et/ou séquentielles. Ces ressources sont, principalement, des blocs élémentaires logiques (pour réaliser des fonctions booléennes), des mémoires RAM, des opérateurs arithmétiques (qui travaillent en virgule fixe), des ressources de routage interne et des entrées/sorties. Ces ressources configurables sont reliées par un réseau dense de lignes de routage et de lignes de transport des horloges. Ces lignes de routage sont aussi configurables. En plus de ces ressources, un FPGA est composé d'une mémoire interne de configuration. Chaque point de cette mémoire correspond à la configuration d'un élément d'une des ressources opérationnelles. Cette mémoire est, dans la plupart des cas, réalisée avec une des trois technologies suivantes : ANTIFUSIBLE (la plus ancienne, configurable une seule fois), FLASH (non-volatile) ou SRAM (volatile, la plus utilisée, représente plus de 80 % du marché). Pour réaliser une application avec un FPGA il faut décrire le circuit électronique à réaliser avec un langage de description matérielle comme le VHDL (Very High Speed Integrated Circuit Hardware Description Langage). Puis il faut synthétiser cette description en circuit électronique. Cette étape et les suivantes peuvent se faire avec des logiciels gratuits fournies par le fabricant de circuit. Enfin après une étape de placement et routage qui prend en compte l'architecture du FPGA, un fichier de configuration appelé bitstream est généré. Celui-ci permet de spécifier au FPGA lors de la configuration la position des points de la mémoire de configuration[2].

II.2. Description du circuit FPGA

Un FPGA est un circuit en silicium reprogrammable. À l'aide de blocs logiques préconstruits et de ressources de routage programmables, vous pouvez configurer ce circuit afin de mettre en œuvre des fonctionnalités matérielles personnalisées, sans avoir jamais besoin d'utiliser une maquette ou un fer à souder. Il vous suffit de développer des tâches de

traitement numérique par logiciel et de les compiler sous forme de fichier de configuration ou de flux de bits contenant des informations sur la manière dont les composants doivent être reliés. En outre, les FPGA sont totalement reconfigurables et peuvent adopter instantanément une nouvelle « fonctionnalité » si vous recompilez une nouvelle configuration de circuit. Jusqu'à présent, seuls des ingénieurs particulièrement expérimentés en matière de conception de matériel numérique pouvaient utiliser la technologie FPGA.

Si les FPGA rencontrent un tel succès dans tous les secteurs, c'est parce qu'ils réunissent le meilleur des ASIC et des systèmes basés processeur. Ainsi, ils offrent un cadencement par matériel qui leur assure vitesse et fiabilité, mais sont plus rentables que les ASIC personnalisés. Les circuits reprogrammables jouissent également de la même souplesse d'exécution logicielle qu'un système basé processeur, mais ils ne sont pas limités par le nombre de cœurs de traitement disponibles. Contrairement aux processeurs, les FPGA sont vraiment parallèles par nature, de sorte que plusieurs opérations de traitement différentes ne se trouvent pas en concurrence pour l'utilisation des ressources. Chaque tâche de traitement indépendante est affectée à une section spécifique du circuit, et peut donc s'exécuter en toute autonomie sans dépendre aucunement des autres blocs logiques. En conséquence, vous pouvez accroître le volume de traitement effectué sans que les performances d'une partie de l'application n'en soient affectées pour autant [4,3.5].

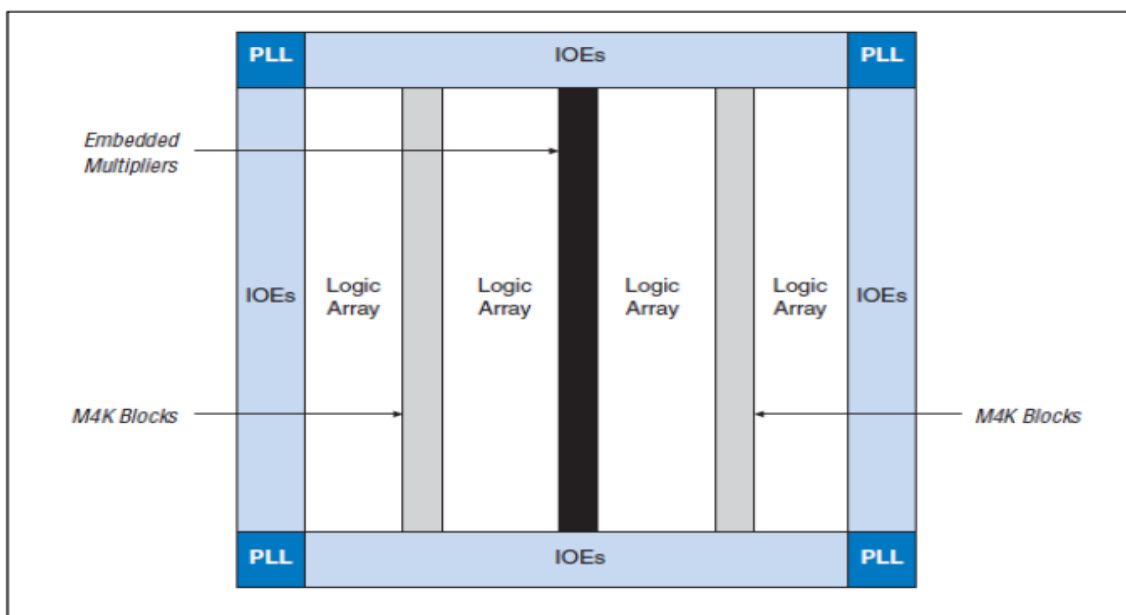


Figure II–1.a : Architecture interne du FPGA fabriqué ALTERA (Cyclone II)

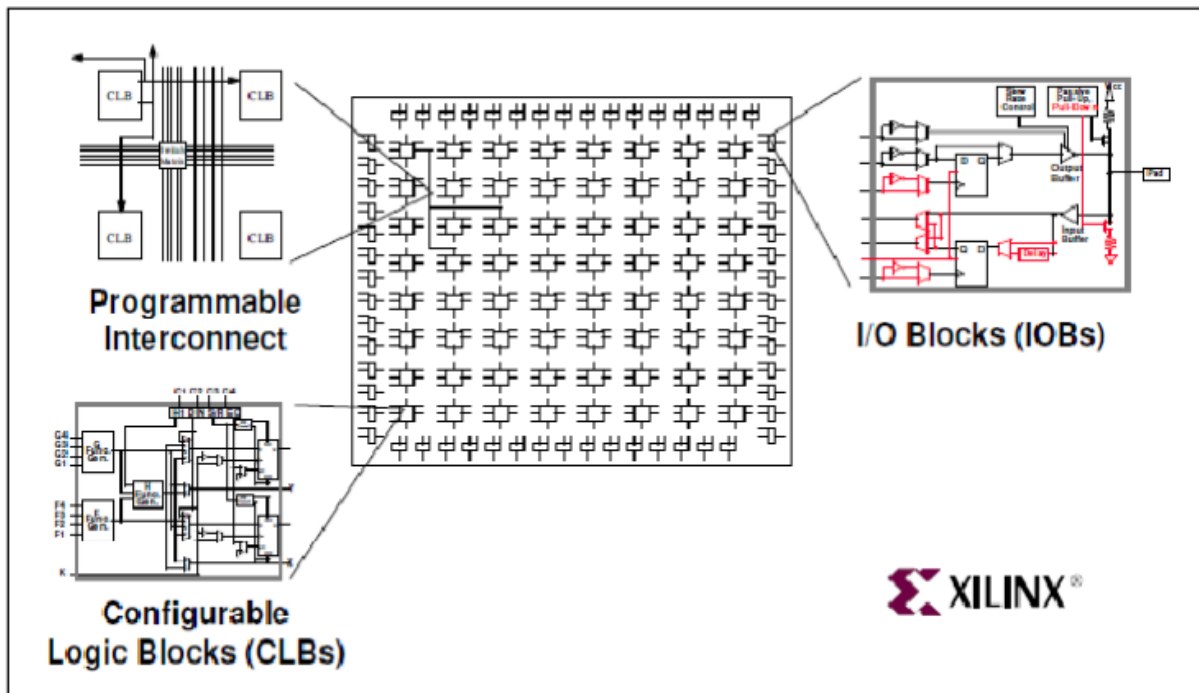


Figure II-1.B : Architecture Interne Du FPGA Fabriqué Xilinx : XC4000 [3]

II.3. Les cinq principaux atouts de la technologie FPGA

1. Performances
2. Temps de mise sur le marché
3. Coût
4. Fiabilité
5. Maintenance à long terme

II.3 .1. Performances

Comme ils tirent parti du parallélisme matériel, les FPGA offrent une puissance de calcul supérieure à celle des processeurs de signaux numériques (DSP), car ils s'affranchissent du modèle d'exécution séquentielle et exécutent plus d'opérations par cycle d'horloge. BDTI, une importante société d'analyse et de « benchmarking », a publié des études montrant que les FPGA peuvent offrir une puissance de traitement par dollar plusieurs fois supérieure à celle d'une solution DSP dans certaines applications. Contrôler les entrées et sorties (E/S) au niveau matériel permet d'obtenir des temps de réponse plus courts ainsi que des fonctionnalités spécifiques, qui répondent mieux aux besoins de l'application.

II.3. 2. Temps de mise sur le marché

Face à des préoccupations croissantes concernant les temps de mise sur le marché, la technologie FPGA représente une solution souple offrant des capacités de prototypage rapide. Ainsi, vous pouvez tester une idée ou un concept, puis le vérifier sur du matériel sans avoir à passer par le long processus de fabrication d'un ASIC personnalisé [2]. Par la suite, vous pourrez apporter les éventuelles modifications nécessaires à votre FPGA, en quelques heures au lieu de quelques semaines. Le matériel « sur étagère » actuellement commercialisé propose également différents types d'E/S déjà connectées à un circuit FPGA programmable par l'utilisateur. La multiplication des outils logiciels de haut niveau disponibles sur le marché permet de réduire le temps d'apprentissage avec les couches d'abstraction. Ces outils comprennent souvent des cœurs de propriété intellectuelle (fonctions précompilées) utiles pour le contrôle avancé et le traitement de signaux.

II.3. 3. Coût

Les coûts d'ingénierie non récurrents (NRE) des ASIC personnalisés sont bien supérieurs à ceux des solutions matérielles basées sur du FPGA. Les circuit ASIC sont largement utilisés pour produire des circuits par milliers chaque année. Cependant, la plupart des utilisateurs finaux ont besoin de matériels personnalisés pour quelques dizaines ou quelques centaines de systèmes en développement. Par nature, les circuits programmables n'impliquent ni coût de fabrication, ni longs délais d'assemblage. Les besoins de la plupart des systèmes évoluent avec le temps ; or la modification progressive d'un FPGA représente un coût négligeable comparé à la dépense considérable qu'exige la reconception d'un ASIC.

II.3. 4. Fiabilité

Tandis que les outils logiciels fournissent l'environnement de programmation, les circuits FPGA sont une véritable implémentation matérielle de l'exécution logicielle. Les systèmes basés processeur comprennent souvent plusieurs couches d'abstraction, pour aider à la planification des tâches et à la répartition des ressources entre les différents processus. La couche de driver contrôle les ressources matérielles et le système d'exploitation gère la mémoire et la bande passante du processeur. Sur chaque cœur de processeur, une seule instruction peut s'exécuter à la fois ; c'est pourquoi les systèmes basés processeur risquent toujours de voir des tâches prioritaires entrer en conflit. Les FPGA, qui n'utilisent pas de système d'exploitation, minimisent les problèmes de fiabilité car ils assurent une exécution

véritablement parallèle et un matériel déterministe dédié à chaque tâche.

II.3. 5. Maintenance à long terme

Comme nous l'avons vu, les circuits FPGA sont évolutifs et vous épargnent donc la dépense de temps et d'argent qu'implique la reconception des ASIC. Les spécifications des protocoles de communication numériques, par exemple, évoluent avec le temps. Or les interfaces basées sur ASIC peuvent poser des problèmes de maintenance et de compatibilité. Comme ils sont reconfigurables, les circuits FPGA sont capables de s'adapter aux modifications éventuellement nécessaires. À mesure qu'un produit ou qu'un système évolue, vous pouvez intégrer des améliorations fonctionnelles sans perdre de temps à reconcevoir le matériel ou à modifier l'implantation du circuit [4].

II.4. Structure interne des FPGA

II.4.1. Principe de fonctionnement des FPGA

Un FPGA, comme l'indique son nom, se compose d'une matrice de blocs logiques reprogrammables. La topologie d'interconnexion entre les différents blocs logiques fait distinguer deux structures : la topologie arborescente et la topologie maillée. La première se base sur une structure qui relie les blocs logiques d'une façon hiérarchique et récursive. La deuxième est une structure organisée sur la forme de mailles et les blocs logiques de chaque ligne sont reliés en cascade. La première topologie offre une taille plus miniature pour la plateforme, mais en revanche une simplicité critique pour l'opération de routage. C'est pour cette raison que la plupart des fabricants adoptent la topologie maillée qui est totalement le contraire. La figure II-2.

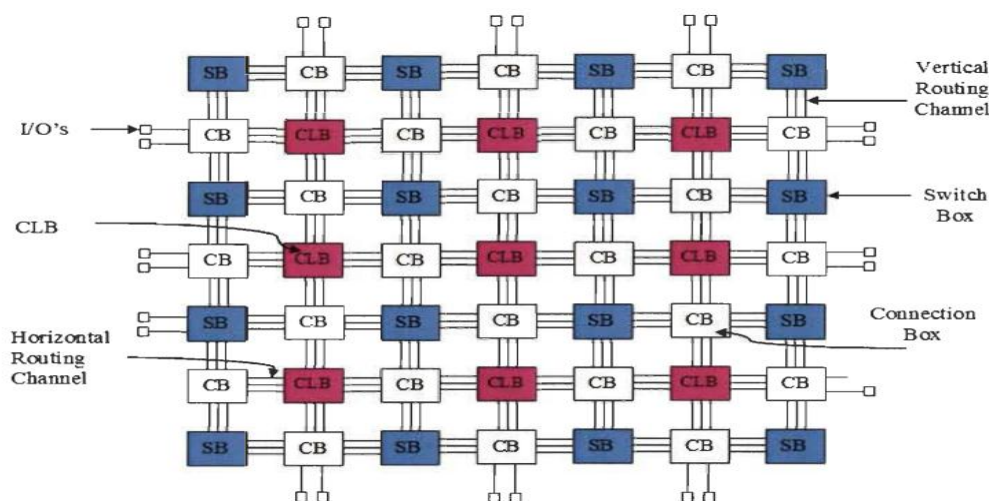


Figure II-2 Architecture traditionnelle d'une plateforme FPGA

Un FPGA peut contenir différents blocs logiques, entre les blocs d'entrée et les blocs de sortie, tout dépend des technologies utilisées et à quelle application il est désigné.

II.4.2. Ressources fonctionnelles configurables

Dans la plupart des cas l'élément logique configurable de base des FPGA se compose d'une LUT (Look Up Table) avec un nombre d'entrées allant de 4 à 8 pour les dernières générations, d'une chaîne de propagation rapide de la retenue et d'un registre de sortie afin d'assurer la synchronisation des signaux (très utile pour l'implémentation de calculs pipelinés). Ces éléments configurables peuvent être rassemblés en clusters hiérarchiques afin de favoriser une connectivité locale et rapide (cas des composants Altera). Les éléments logiques configurables sont appelés SLICE chez Xilinx. Entre deux générations successives de FPGA Xilinx, un SLICE Virtex4 ne correspond pas à un SLICE Virtex6.

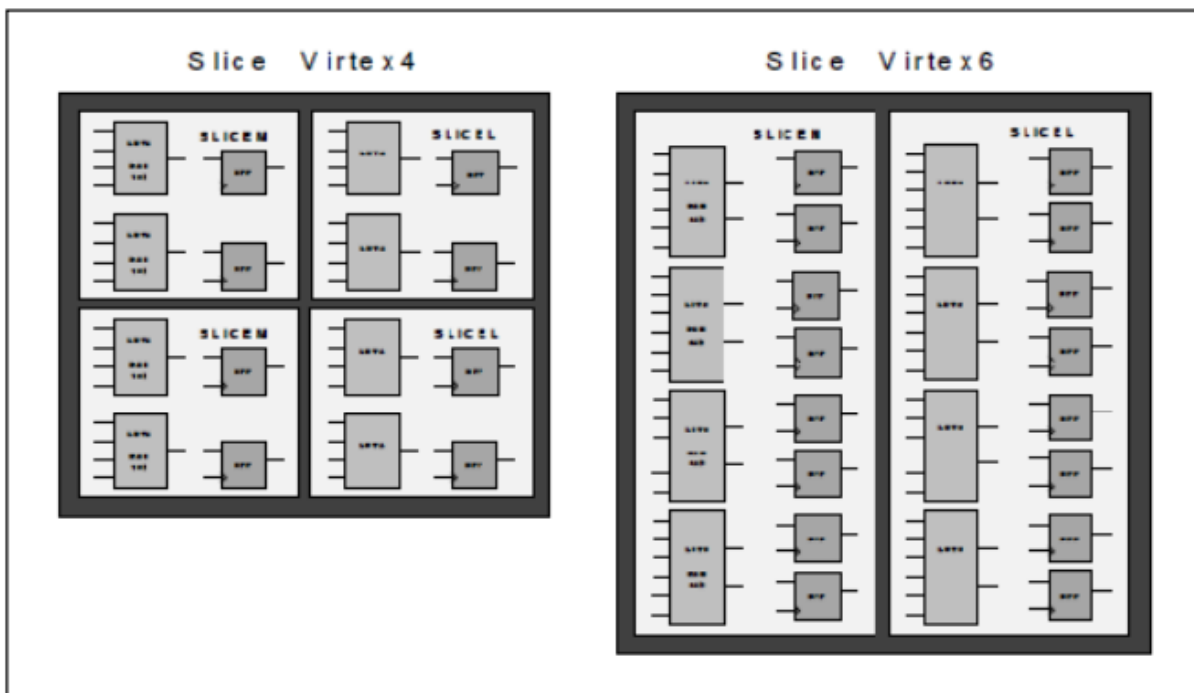


Figure II-3 :Eléments logiques configurables (circuits Xilinx Virtex4 et Virtex6).

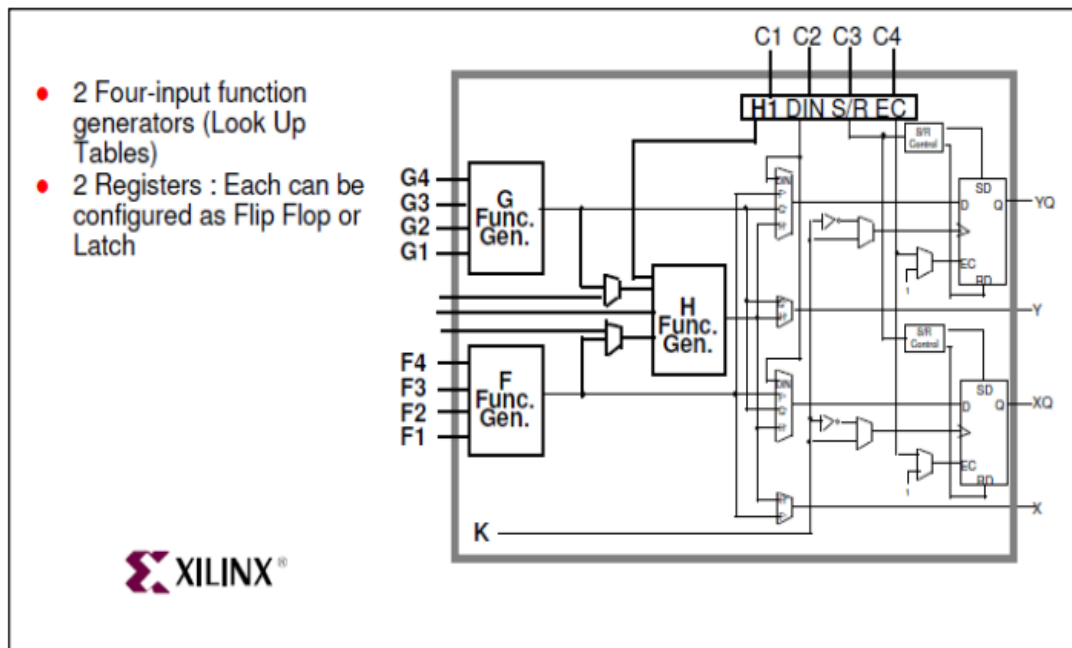


Figure II.4. Eléments logiques configurables

II.4.2.1. Les éléments de mémorisation

Les FPGAs ont changé et ne sont plus seulement utilisés pour des applications de type "glue logique", comme ce fût le cas à leur début, mais aussi pour des applications plus importantes qui demandent souvent des capacités de stockage (citons en exemple les applications du traitement d'images). La nécessité d'intégrer des blocs de mémoires directement dans l'architecture des FPGAs est vite devenue cruciale. De cette façon les temps d'accès à la mémoire sont diminués puisqu'il n'est plus nécessaire de communiquer avec des éléments extérieurs au circuit. Dans les exemples détaillés par la suite, nous verrons que ces blocs mémoires sont généralement de type RAM, et sont configurables au niveau de la largeur des mots mémorisés. Du fait de l'intégration de ces blocs, dans des architectures déjà existantes, un point crucial apparaît qui est le routage entre les parties mémoires et les ressources logiques. La caractéristique importante de ce routage est sa flexibilité. Si ce routage n'est pas assez flexible alors le circuit est difficilement routable, si il est trop flexible il y a sur consommation de surface (donc de silicium) [2,8].

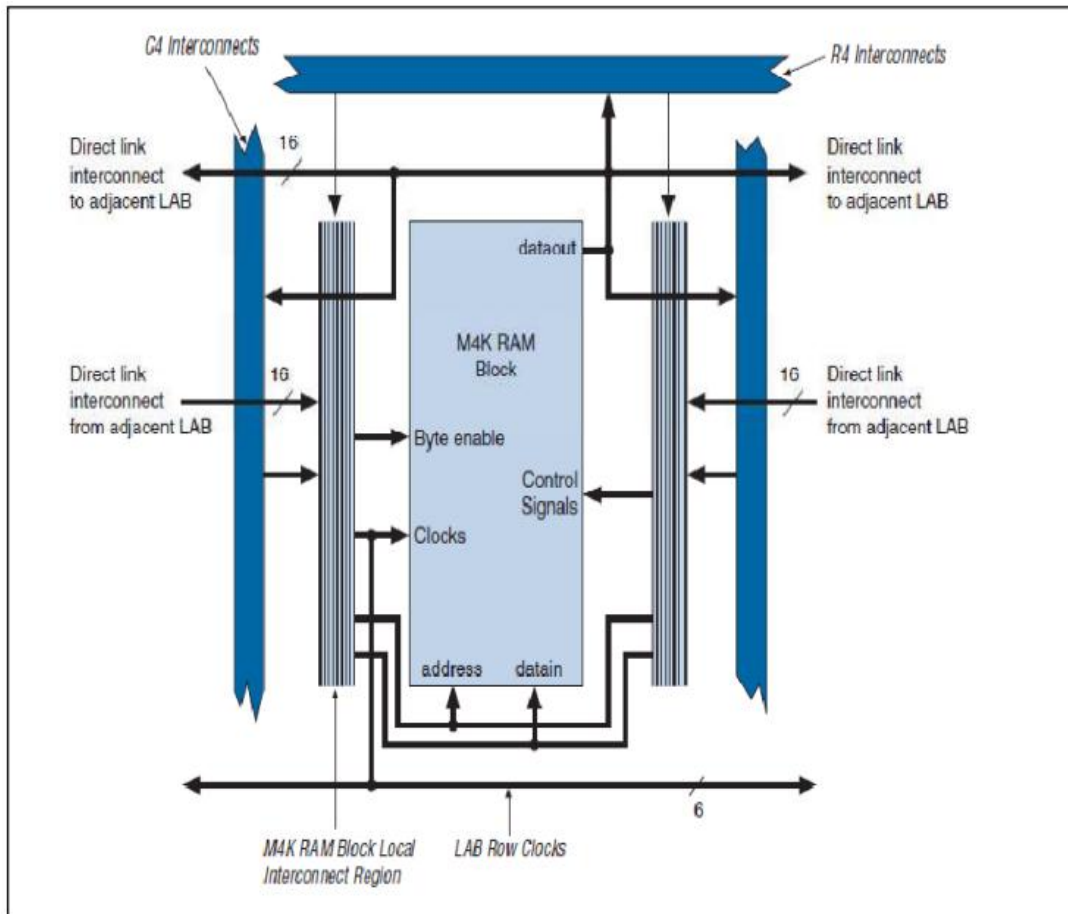


Figure II-5 : Éléments de mémorisation (M4K RAM) d'Altera [6].

II.4.2.2. Les éléments de routages

S'il y a des éléments plus importants que d'autres dans les FPGAs, ce sont les éléments de routages. En effet les ressources de routages représentent la plus grosse partie de silicium consommée sur la puce réalisant le circuit. Ces ressources sont composées de segments (de longueurs différentes) qui permettent de relier entre eux les autres éléments via des matrices de connexions. Le routage de ces ressources est un point critique du développement d'une application sur un FPGA, et les méthodes utilisées pour les ASICs ne sont plus tout à fait valables (du fait de la segmentation des ressources). Si ces éléments sont importants c'est parce qu'ils vont déterminer la vitesse et la densité logique du système. Par exemple les matrices de routage (programmable Switch) sont, physiquement, réalisées grâce à des transistors de cellules SRAMs, qui ont une résistance et une capacité, ce qui entraîne l'existence de constantes de temps [2,8,3].

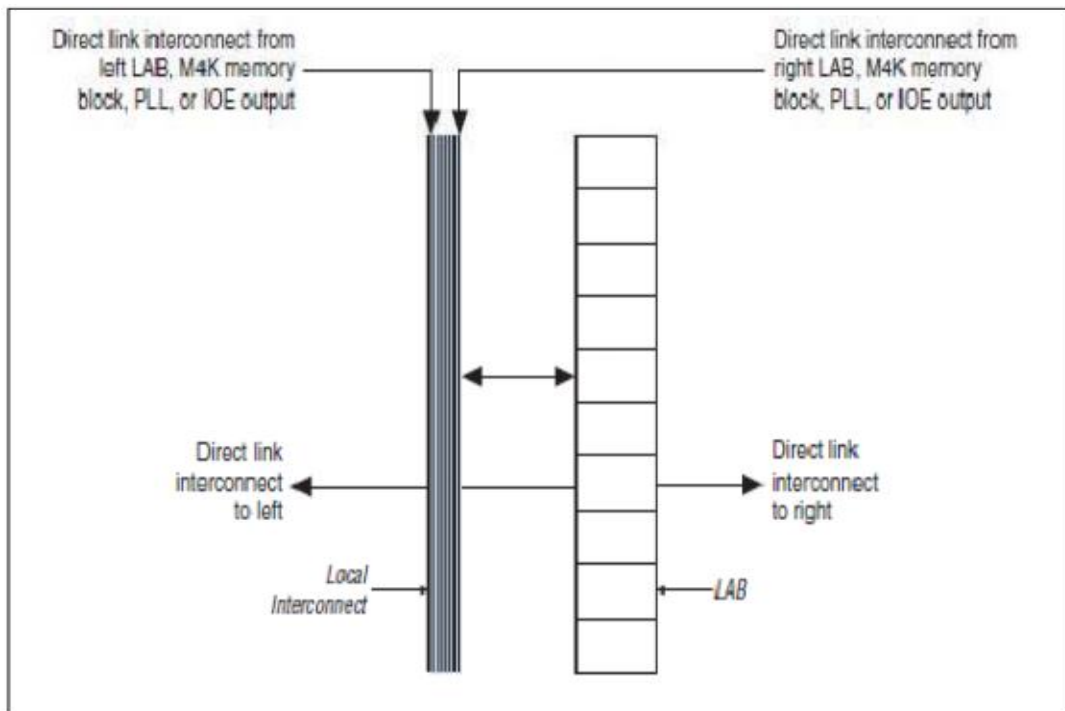


Figure II-6 :Élément de routages (Altera) [6].

II.4.2.3. Les éléments d'entrées sorties

Le composant ne vie pas seul sur une carte (ou très rarement), il appartient à un système d'ensemble pouvant contenir des parties micro programmées comme dans le cas du "Co- Design" (conception conjointe logiciel matériel). Le circuit doit donc avoir un lien avec son environnement, c'est le but des éléments d'entrées/sorties. Ceux-ci peuvent bénéficier de protections, de buffer ou d'autres éléments permettant la gestion des entrées et des sorties. En particulier il est à noter que les circuits actuels proposent différentes normes pour les niveaux d'entrées et de sorties qui par configuration peuvent être choisies afin de s'adapter à l'environnement [2,3,8].

II.4.2.4. Les éléments de contrôle et d'acheminement des horloges

Il paraît évident que dans tout système électronique relativement important, il faut disposer d'horloges et qu'elles sont souvent d'une importance capitale pour le bon fonctionnement du système (pensons aux systèmes de télécommunications par exemple). Alors, bien sur, les FPGAs sont prévus pour recevoir une ou plusieurs horloges. Des entrées peuvent être spécialement réservées à ce type de signaux, ainsi que des ressources

de routages spécialement adaptées au transport d'horloges sur de longues distances (bufferisations des lignes). Aussi, pour assurer d'avoir la même horloge dans tout le circuit (synchronisation des signaux) les circuits ont reçu des éléments d'asservissement des horloges (des PLLs ou des DLLs) qui permettent souvent de créer à partir d'une horloge d'autres horloges à des fréquences multiples de la fréquence de l'horloge incidente. Ces arbres d'horloge permettent donc de disposer sur le circuit de fonctions travaillant à des fréquences différentes.

II.5. Principales applications des FPGA

Nous allons citer les principales applications des circuits FPGA :

1. Applications Médicales
2. Application Militaire
3. Télécommunication

II.6. Conclusion

La part de marché des FPGA dans le marché globale des circuits matériels pour l'électronique numérique ne cesse d'augmenter. Les évolutions technologiques et architecturales qui ont eu successivement lieu depuis le début des années 2000 ont fait de ces circuits de réels et rentables alternatives aux classiques ASIC. Avec ces évolutions c'est tout un nouveau domaine de l'électronique numérique qui s'est ouvert. Aujourd'hui les FPGA sont utilisés dans tous les domaines, des systèmes embarqués aux systèmes de communications, ils sont au cœur d'un important champ de recherche académique et industrielle.

Bibliographie

- [1]. <http://dumas.ccsd.cnrs.fr/dumas-00574220/document>.
- [2]. <http://perso.univ-st-etienne.fr>
- [3]. <http://www.xilinx.com/>.
- [4]. <http://www.ni.com/white-paper/8043/fr/>.
- [5]. http://www.yannthoma.com/research/these/These_Thoma_chap2.pdf.
- [6]. *Cyclone II Device Handbook, Volume 1.pdf*.
- [7] *H. M. Husain Parvez, Application-Specific Mesh-based Heterogeneous FPGA Architectures: Springer New York Dordrecht Heidelberg London, 2011.*
- [8]. *WWW.ALTERA.COM*.
- [9]. <http://www.altera.com/devices/fpga/cyclone2/cy2-index.jsp>.



CHAPITRE III

Commande des moteur pas à pas par FPGA

III.1.Introduction

On utilise généralement Pour La commande des moteurs pas à pas des micro - contrôleurs associé à des circuits annexe tels que les séquenceurs (L297) et circuit de puissance (L298).

Seulement le nombre de moteur commandé est limité par le nombre de ports d'entrées/sorties des micro – contrôleurs . une solution judicieuse consiste à utiliser des circuits FPGA pour commander les moteurs pas à pas dont le nombre d'entrées/sorties dépassent largement aux des micro - contrôleurs .

III.2.Structure générale d'un système de commande

La structure générale du commande d'une moteur pas à pas indiquée dans la figure suivante :

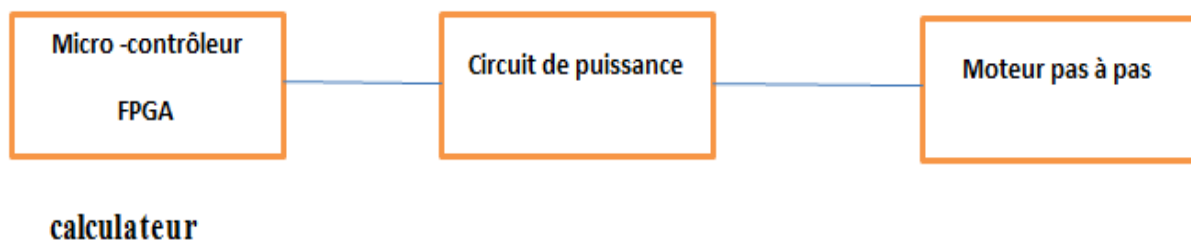


Figure III-1 :Structure générale du commande d'un moteur pas à pas.

III.3.Commande des moteurs pas a pas

Dans cette partie, différentes stratégies de commande sont présentées : deux stratégies de commande en pas entiers, dites commandes en mode 1 et 2 ainsi que trois stratégies de commandes en pas fragmentés, dites commandes en modes 3, 4 et 5.

III.3.1.Fonctionnement a pas entiers

III.3.1.1.Commande en mode 1

La commutation phase par phase de la structure quadriphasée de la figure III-2 permet au mobile d'atteindre quatre positions d'équilibre sur une période de déplacement.

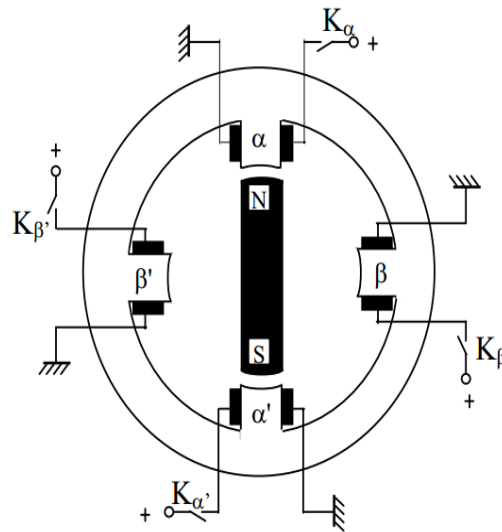
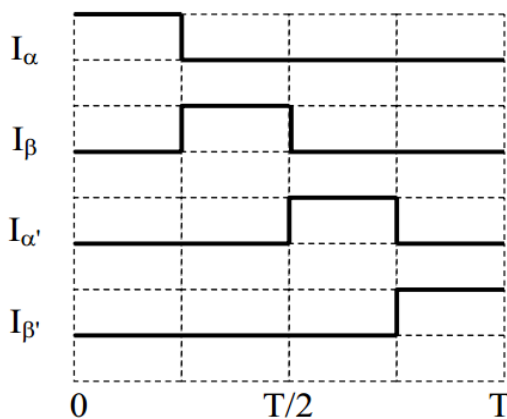
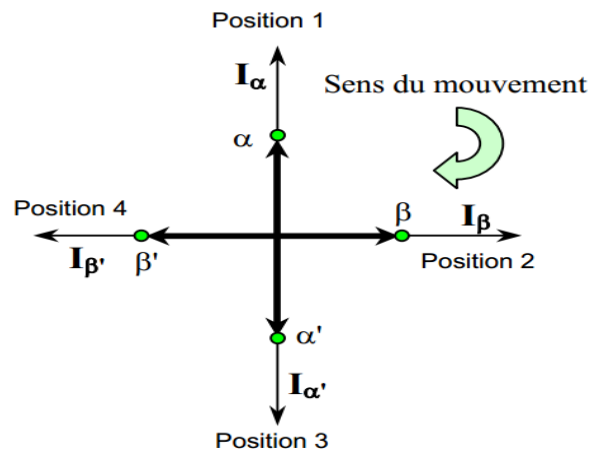


Figure III-2 Structure quadriphasée à alimentation unipolaire

Les interrupteurs K_α , K_β , $K_{\alpha'}$ et $K_{\beta'}$ de la figure III-2, sont actionnés successivement conformément au cyclogramme de la figure I.12.a ce qui conduit à quatre positions d'équilibre du mobile. La figure III-3.b symbolise le mode de commande des différentes bobines et la position relative du mobile suivant l'alimentation.



a. Cyclogramme des courants d'alimentation



b. Positions d'équilibre

Figure III-3 Principe d'alimentation en mode 1

Si les interrupteurs de la figure III-2 sont actionnés dans l'ordre inverse : K_α , K_β , $K_{\alpha'}$ et $K_{\beta'}$, le déplacement du champ et par conséquent celui de la partie mobile se fait dans le sens contraire du précédent. Ainsi, l'emplacement des positions d'équilibre est conservé mais leur occurrence est inversée.

Les connexions sérielles de α et α' , d'une part, et de β et β' d'autre part, permet d'obtenir un enroulement diphasé qu'il faut alimenter en bipolaire, figure III-4 .

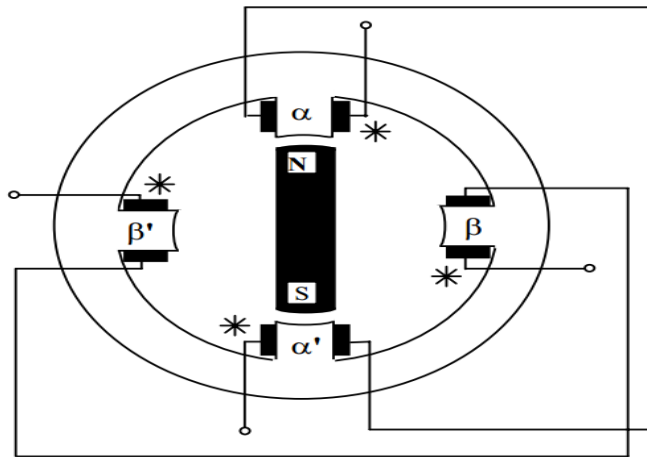
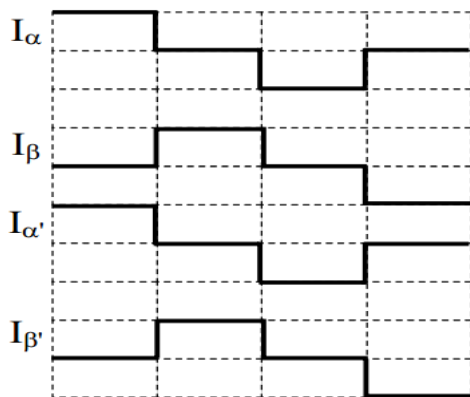
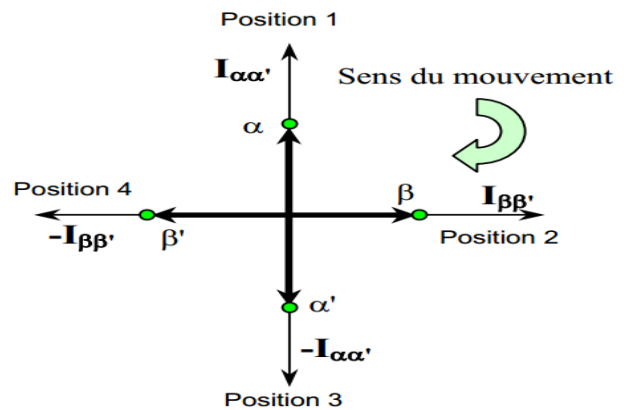


Figure III-4 Principe de l'alimentation bipolaire en mode 1

L'excitation du stator ainsi connecté, suivant le cyclogramme $(\alpha\alpha')$, $(\beta\beta')$, $(-\alpha\alpha')$, $(-\beta\beta')$ donne le même nombre de pas cyclique que l'alimentation unipolaire phase par phase, mais le couple est dans ce dernier cas plus important car deux bobines sont simultanément alimentées, figure III-5 .



a. Cyclogramme des courants d'alimentation

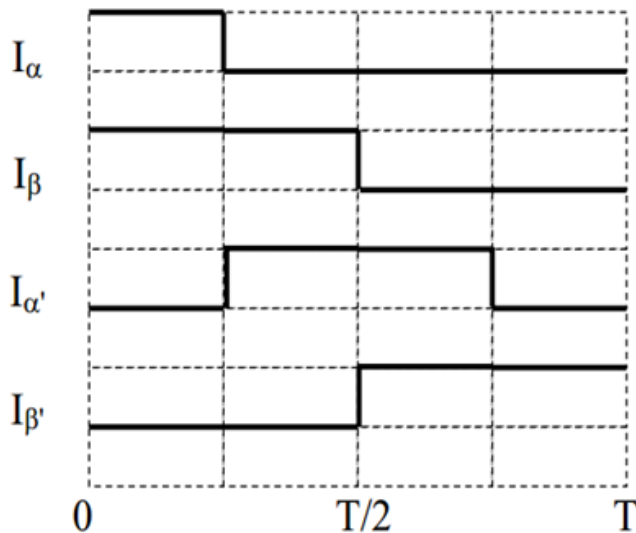


b. Positions d'équilibre

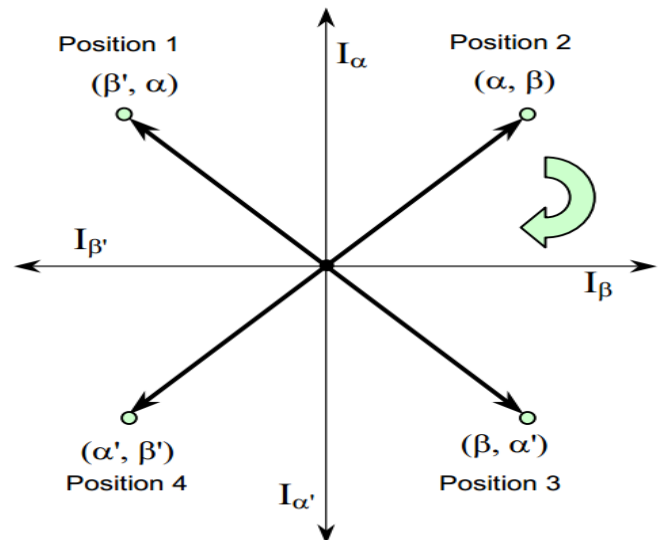
Figure III-5 Principe de l'alimentation d'un bobinage biphasé en mode 1

III.3.1.2. Commande en mode 2

Dans le cas de la commande en mode 2, les doublets de phases successives de la machine de la figure III-2 sont alimentés simultanément suivant l'ordre (α, β) , (β, α') , (α', β') , (β', α) , conformément au cyclogramme de la figure III-6.a, les positions d'équilibre correspondantes sont illustrées sur la figure III-6.b.



a. Cyclogramme des courants d'alimentation



b. Positions d'équilibre du champ

Figure III-6 Principe d'alimentation en mode 2

L'avantage de ce mode de commande est qu'il optimise l'utilisation du cuivre de la machine et donc l'exploitation de sa puissance massique, le couple développé dans ce cas est multiplié par $\sqrt{2}$ par rapport au couple développé par une commande en mode 1 unipolaire.

III.3.2. Fonctionnement à pas réduits

III.3.2.1. Commande en mode 3

Ce mode de fonctionnement est une combinaison en alternance des modes 1 et 2. Le nombre de positions d'équilibre est ainsi doublé et un déplacement en demi-pas est généré. La figure III-7.a donne l'allure des courants de phase pour la machine à aimant permanent à alimentation unipolaire de la figure III-2. Sur la figure III-7.b sont illustrées les huit positions d'équilibre correspondantes au fonctionnement en demi-pas. L'inconvénient de ce mode de fonctionnement est que le champ créé à chaque position, n'a pas toujours la même amplitude. Il est $\sqrt{2}$ fois plus important à la position 2 qu'à la position 1.

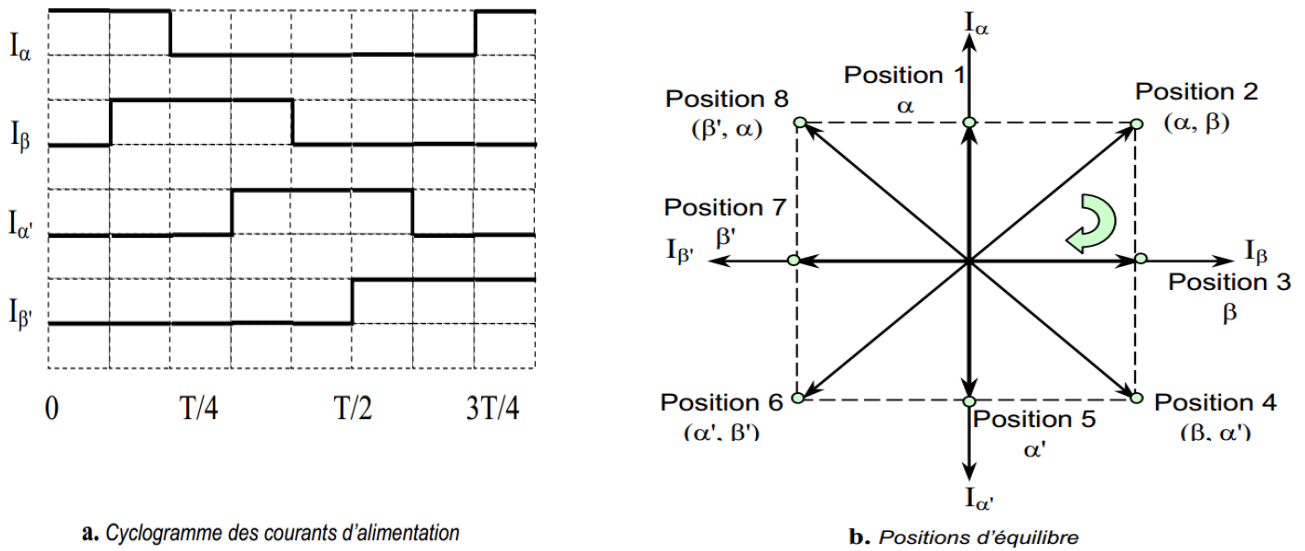


Figure III-7 Principe d'alimentation en mode 3

III.3.2.2. Commande en mode 4

La différence entre ce mode de fonctionnement et le mode 3, présenté précédemment, est que pour obtenir des vecteurs champs d'amplitudes identiques quelle que soit la position, la consigne de courant est multipliée par un facteur $\sqrt{2}$ lorsqu'une seule phase est alimentée. Ce mode permet de conserver les pertes Joule constantes au cours du fonctionnement. Les allures des courants d'alimentation sont, dans ce cas, données par le cyclogramme de la figure III-8.a et les positions d'équilibre correspondantes par la figure III-8.b.

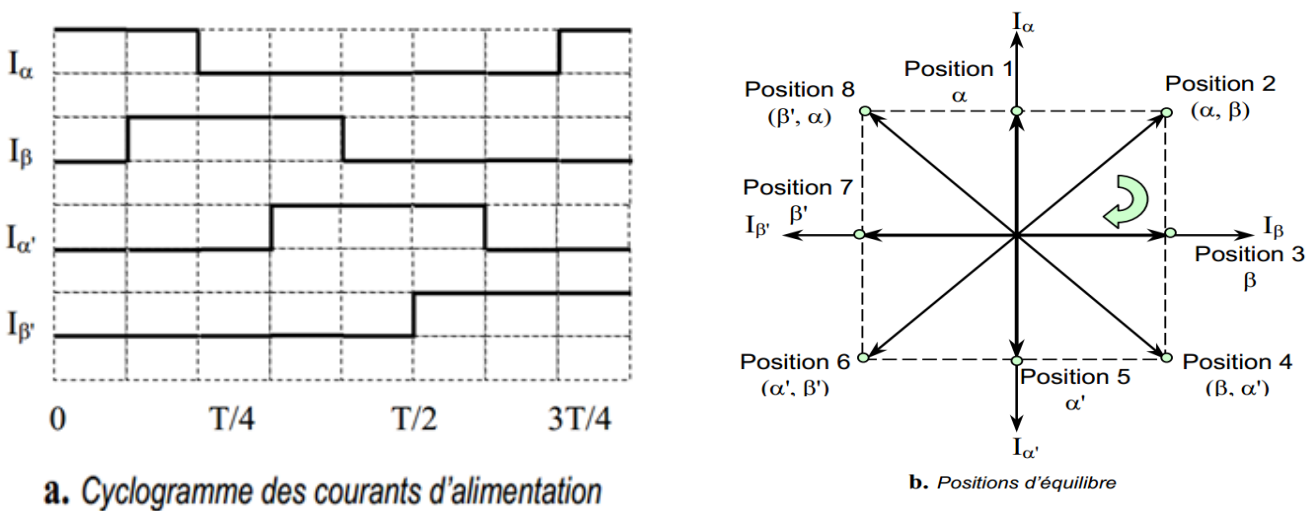


Figure III-8 Principe d'alimentation en mode 4

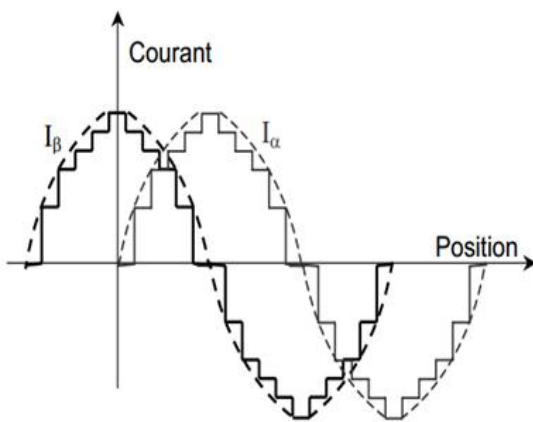
III.3.2.3. Commande en mode 5

Pour améliorer substantiellement la résolution angulaire d'un moteur pas à pas par action sur la commande, la technique de pilotage en micro pas, par variation de l'amplitude des courants, peut être appliquée .

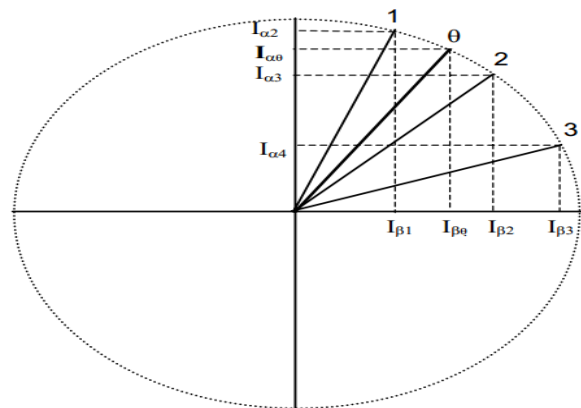
Cette technique de fragmentation de pas consiste à alimenter simultanément deux phases de lamachine par deux courants d'amplitudes différentes. Les amplitudes des courants d'alimentation varient séquentiellement en créneaux entraînant ainsi le rotor par fragments de pas .

Par ce mode d'excitation, la modulation des courants injectés dans les phases statoriques permet de positionner le rotor dans une direction quelconque.

Si les enroulements statoriques dumoteur sontalimentés par des niveaux de courant à variations sinusoïdales, figure III-9.a, alors les phases α et β doivent être alimentées simultanément par les courants $I_{\alpha\theta}$ et $I_{\beta\theta}$ afin d'obtenir parmi les n positions celle repérée par θ sur la figure III-9 .b .



a. Cyclogramme des courants d'alimentation



b. Positions d'équilibre du champ

Figure III-9 Pilotage par modulation d'amplitude des courants

III.4. Alimentation

III.4.1. Objectifs d'une bonne alimentation et classification

Un enroulement de moteur pas à pas est un circuit électrique fortement inductif siège d'une force électromotrice variable ou présentant parfois une inductance variant avec la position.

Ces éléments s'opposent à des modifications rapides du courant d'alimentation. Une bonne alimentation doit :

- établir rapidement un courant donné dans un circuit inductif,
- maintenir ce courant à sa valeur en dépit des fluctuations des paramètres du circuit, notamment ceux qui varient avec la vitesse ou la position,
- interrompre rapidement ce courant.

Selon les applications, les performances souhaitées et les moteurs utilisés, on rencontre différents types d'alimentation. Les moteurs à réluctances variables, qui développent un couple proportionnel au carré du courant, ne nécessitent que des alimentations unipolaires (courant généré dans un seul sens). Les autres moteurs nécessitent des courants bipolaires. Ceux-ci sont créés par un seul enroulement bipolaire (courant dans les deux sens) ou par deux enroulements unipolaires, fortement couplés, en opposition. Dans la suite, nous étudions ces alimentations par performance et complexité croissantes.

III.4.2. Alimentation en tension

III.4.2.1 Alimentation unipolaire en tension unique

Ce type d'alimentation, le plus ancien, est aussi le plus simple (figure III-8). Il consiste à brancher grâce à un transistor Q chaque phase sur une alimentation de tension V_E . Une diode D, placée en antiparallèle sur l'enroulement, permet l'extinction du courant quand on bloque le transistor. Si R est la résistance de la phase, le courant nominal, atteint lorsque le moteur est à l'arrêt, est donné par :

$$I = \frac{V_E}{R}$$

Dès qu'il tourne, on doit tenir compte des régimes transitoires électriques et mécaniques qui vont ralentir, voire empêcher, l'établissement des courants aux valeurs nominales. L'équation

électrique de l'enroulement est : $V_E = RI + \frac{\partial \varphi}{\partial t} = RI + L \frac{di}{dt} + u_i(t)$

avec

L : inductance de l'enroulement

R : résistance de l'enroulement (+ résistance série R_s)

U_i : tension induite de mouvement

On voit clairement que les variations de courant s'opèrent avec la constante de temps L/R ; pour obtenir un régime transitoire plus rapide et garder le même courant nominal, il suffit de mettre une résistance R_{sen} série avec l'enroulement et d'utiliser une nouvelle alimentation V'_E telle que $I = \frac{V'_E}{R+R_s}$

La nouvelle constante de temps est alors $L / (R + R_s)$. Pour une extinction plus rapide, on place en série avec la diode une résistance R_s ou une diode Zener. Cette amélioration est facile à obtenir mais se paie par des pertes thermiques dans la résistance série, qui augmentent avec le degré de l'amélioration. Aussi, ce procédé ne sera utilisé que pour les entraînements de faible puissance pour lesquels le prix de revient doit être particulièrement réduit.

L'alimentation unipolaire est parfaitement indiquée pour les moteurs à réluctance variable à condition de bien dimensionner les circuits de récupération.

III.4.2.2 Alimentation Unipolaire Bitension

Au début de la commutation les transistors Q_1 et Q_2 sont fermés simultanément. La tension V_H permet d'augmenter le gradient de courant dans la charge. Après un temps fixe t_1 ou lorsque le courant a atteint sa valeur nominale le transistor Q_2 s'ouvre et l'alimentation V_L permet de maintenir ce courant nominale au travers de D_1 . Le temps t_1 d'ouverture du transistor Q_2 peut également correspondre à une surintensité I_{MAX} favorable à la rotation du moteur. Le rapport des temps de montée est proportionnel au rapport V_H et V_L . Ce procédé est d'autant plus économique qu'il y a beaucoup de moteurs à alimenter.

III.5 Paramètres de commande des moteurs pas à pas

- **Circuit de pilotage pas à pas :**

Reçoit un signal impulsé et peut être commandé par un dispositif marche/arrêt, basculant entre deux niveaux de tension.

- **Sens :**

Déterminé par la position haute/basse du dispositif.

- **Vitesse :**

Obtenue en régulant la source d'impulsions, à l'aide d'un oscillateur .

- **Cadence de pas :**

Peut être modifiée avec le type de pilote, pour obtenir un demi-pas ou un micropas .

- **Mémoire :**

Les moteurs pas à pas, fonctionnant en multi-axes, peuvent bénéficier de systèmes de gestion avec mémoire programmable intégrée, laquelle permet un fonctionnement en mode simple ou en séquence répétée .

- **Commande système :**

Peuvent être personnalisées pour correspondre à une machine ou à un système. Les caractéristiques personnalisées en E/S autorisent l'interfaçage avec des réseaux industriels.

III.6 Conclusion

La commande des moteurs pas à pas nécessite un séquence peut être implémenté sur une unité de calcul (micro – contrôleur , circuit FPGA).

Les moteur pas à pas sont commandés soit en mode unipolaire, bi – polaire, en pas entier ou demi – pas et dans les deux sens de rotation horaire et anti – horaire.

Bibliographie

[1]. <http://biblioweb.u-cergy.fr/theses/07CERG0344.pdf>.

[2]. Simulation sur Matlab/Simulink et implémentation sur DSP/FPGA de la commande vectorielle de la machine synchrone à aimants permanents (PMSM) alimenté par un onduleur de tension à Modulation vectorielle (SVM).pdf.

[3]. Commande numérique à base de composants FPGA d'une machine synchrone_Mohamed Wissem NAOUAR L'UNIVERSITE DE CERGY PONTOISE.pdf.

[4]. WWW.ALTERA.COM.



CHAPITRE IV

*Réalisation de Commande de moteur
pas à pas par FPGA*

IV.1 Introduction

Dans ce chapitre nous allons développer la réalisation de la commande d'un moteur pas à pas par l'association de la carte de développement DE1 et d'une carte auxiliaire intégrant les circuits de puissances.

IV.2 Descriptif de la carte auxiliaire

Nous utilisons pour notre carte auxiliaire un circuit L298, au sein duquel sont implantés deux ponts en H, associé à la logique de commande, comme l'indique la structure interne de ce circuit sur la figure suivante :

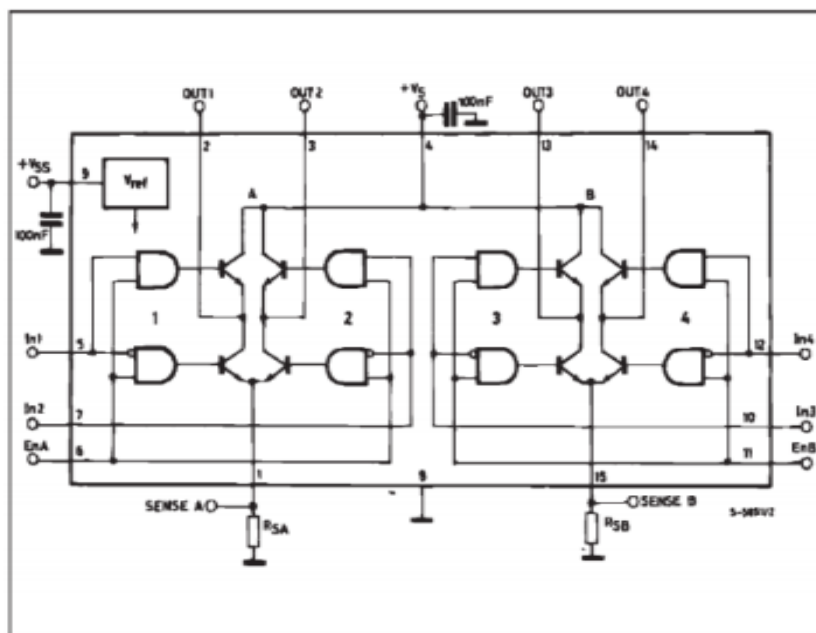


Figure.IV. 1 La carte auxiliaire d'un circuit L298

Comme on peut le noter sur la figure précédente, les diodes de puissance ne sont pas intégrées dans le L298, il faut les ajouter sur la carte, comme le montre la figure suivante. Il est alors possible de commander un moteur pas à pas bipolaire deux phases (les deux phases étant connectée respectivement en OUT1-OUT2 et OUT3-OUT4).

IV.4 Mise en œuvre de la commande d'un moteur pas a pas

IV.4.1 Descriptif du programme

Le programme implanté dans le FPGA de la carte DE1 est associé au schéma suivant:

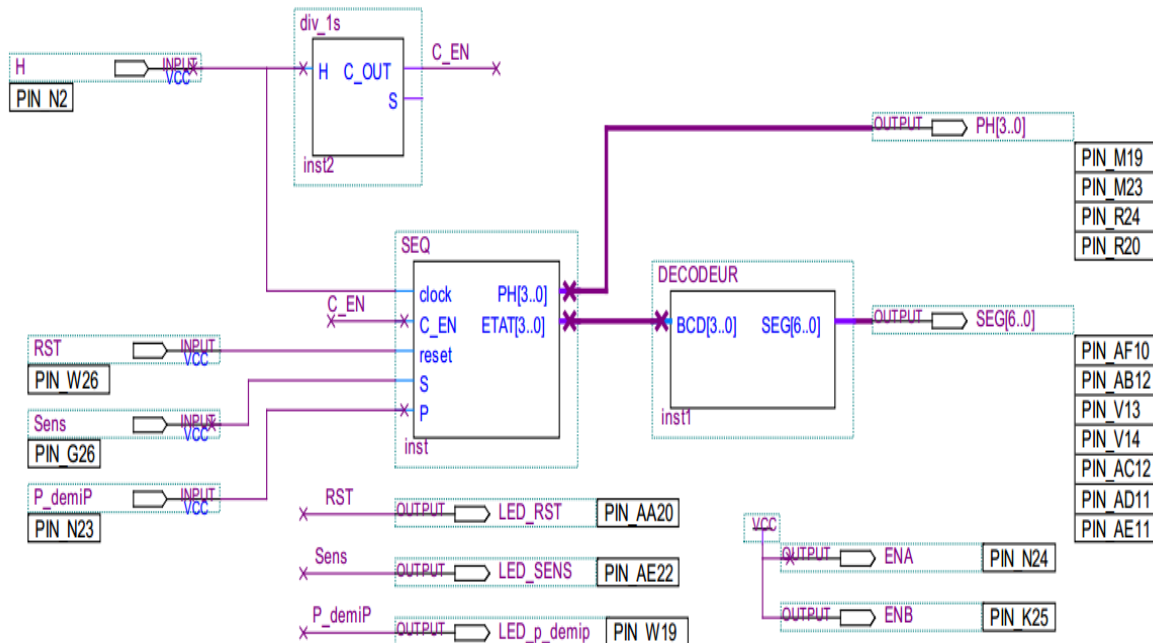


Figure. IV. 3 schéma de programme implémenté dans le FPGA

Le bloc « div_1s » nous génère un signal d'horloge de période 1 secondes, en divisant par 50 millions l'horloge à 50 MHz de la carte DE1.

Cette horloge d'une seconde va commander un séquenceur, permettant au moteur de fonctionner dans un sens ou dans l'autre suivant que le bouton poussoir relié à l'entrée « Sens » est appuyé ou non, et en mode « pas entiers une phase » ou en mode « demi-pas » suivant que le bouton poussoir associé à l'entrée « P_demiP » est appuyé ou non.

La sortie « PH[3..0] » du séquenceur envoie les signaux adéquats sur les entrées In1 à In4 du circuit L298, suivant les chronogrammes représentés ci-dessous :

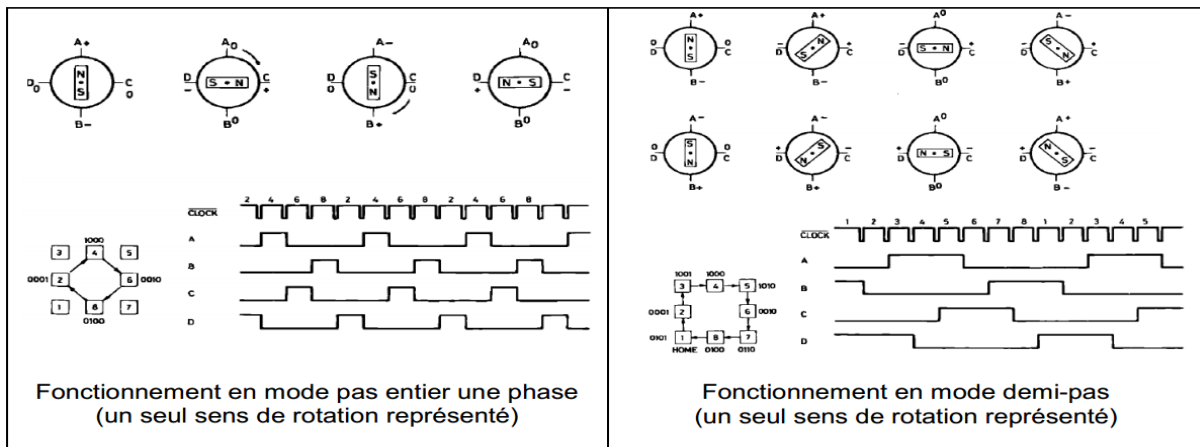


Figure.IV. 4 schéma représentés les chronogrammes

Un bouton poussoir associé à l'entrée « RST » permet de réinitialiser le séquenceur.

Des diodes électroluminescentes de la carte DE1 permettent de visualiser l'appui sur ces boutons. On notera que le séquenceur est rendu synchrone grâce à une entrée d'autorisation d'horloge « C_EN ». Contrairement au programme précédent, le rapport de division de l'horloge étant important, cette option est ici indispensable.

La machine d'état suivant décrit le fonctionnement de notre séquenceur :

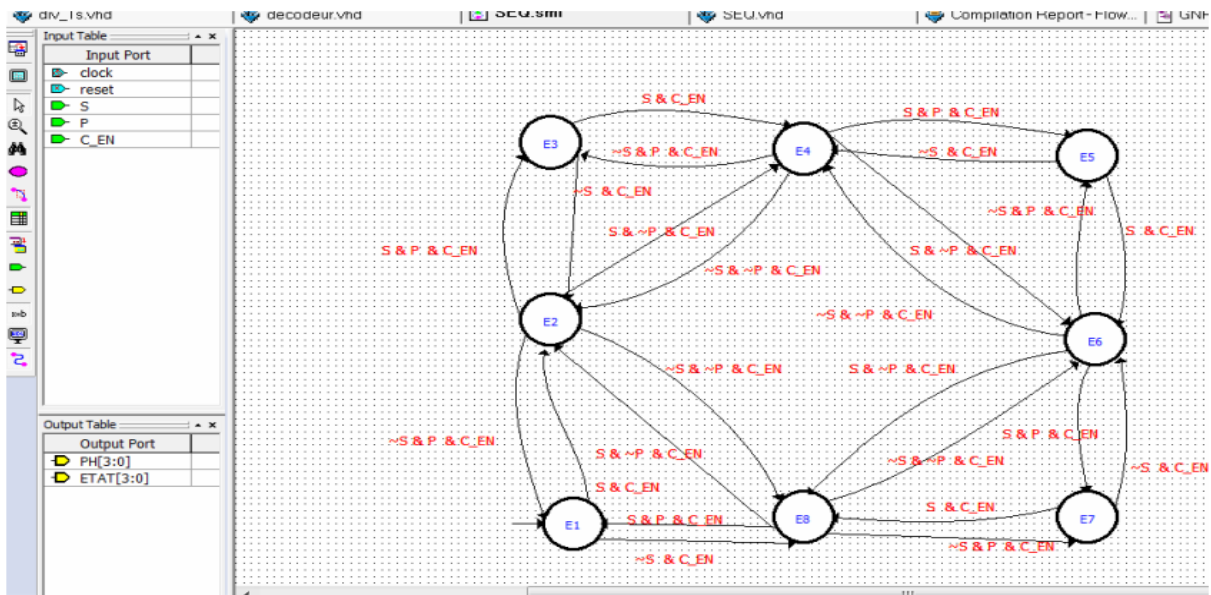


Figure. IV. 5 schéma représentent La machine d'état

Comme indiqué précédemment, le passage d'un état à l'autre se fait :

- uniquement si « C_EN » est au NL1 quel que soit les états,
- dans un sens ou dans l'autre suivant l'entrée « S »,
- en passant par tous les états, ou juste les étatspairs, suivant l'entrée « P ».

La sortie « PH[3..0] » du séquenceur commande les entrées Inx du L298, tandis que la sortie « ETAT[3..0] » permet de visualiser le numéro de l'état actif, et de l'indiquer sur un afficheur 7 segments de la carte DE1, grâce au bloc « DECODEUR », réalisant la conversion BCD / 7 segments.

Iv.4. 2 Mise en œuvre

Ouvrir un nouveau projet sous Quartus, importer le diviseur de projet précédent, le sauvegarder dans le nouveau projet afin d'éviter que sa modification n'affecte l'étude précédente, et modifier le programme pour répondre au nouveau cahier des charges.

Nous avons élaboré un programme VHDL pour le décodeur et Synthétisé le séquenceur à l'aide de l'utilitaire de Quartus et tester la solution obtenue à l'aide du simulateur de Quartus.

Nous avons Regroupé les sous ensembles en un schéma général et programmé le FPGA sans alimenté la carte auxiliaire, nous avons vérifié au niveau de l'afficheur que le fonctionnement est bien celui attendu (dans un sens dans l'autre, en mode pas entier, demi-pas).

IV.5 Carte FPGA de développement et de formation DE1 (Altera)

La carte DE1 (figure IV. 6), construite autour d'unFPGA ALTERA Cyclone II (2C20), comporte de la mémoire FLASH, SRAM et SDRAM, et de nombreux périphériques d'affichage (LEDs, LCD, VGA), sonores et de communication (USB, IrDA...).

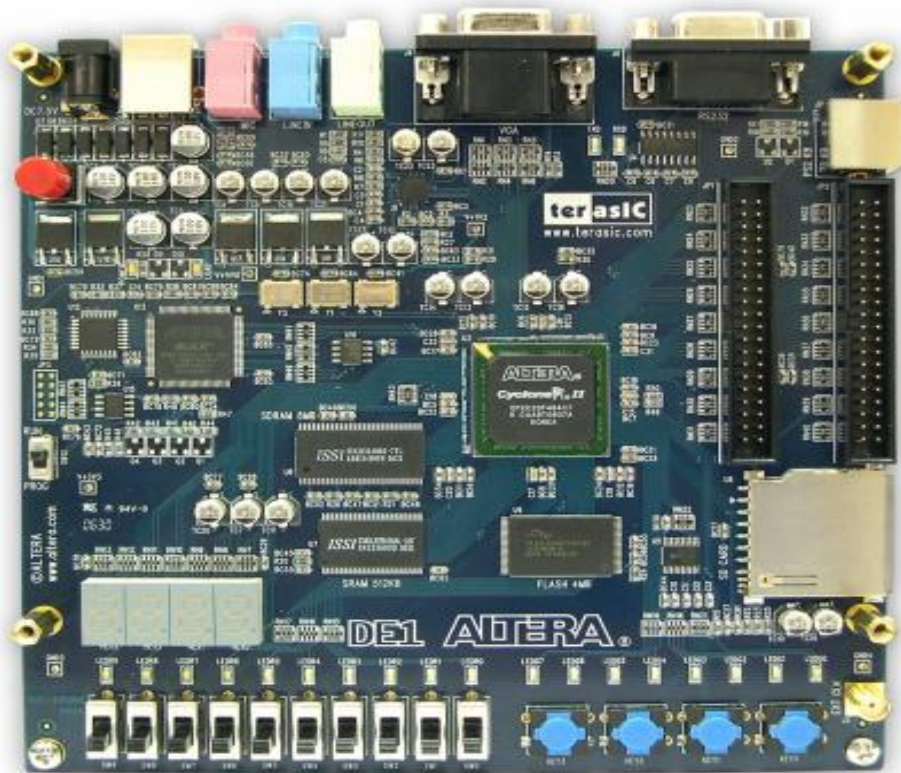


Figure.IV. 6 Carte de développement et de formation DE1(Altera)

❖ **Caractéristiques FPGA DE1 Cyclone II :**

- USB Blaster intégré pour configuration FPGA
- Cyclone II EP2C20F484C7 FPGA et EPCS4 serial configuration dispositif
- VGA DAC réseau résistance (4096 colleurs)
- Port PS/2 pour clavier ou souris
- Entrée ligne, sortie ligne, entrée micro (CODEC audio 24 bits)
- Embases d'extension (76 broches de signal) Mémoire :
 - 8-MB de SDRAM, 512-KB de SRAM, 4-MB de Flash
 - Port de carte mémoire SD fente
 - 10 commutateurs à bascule
- 4 commutateurs à bouton-poussoir antiparasités
- 4 commutateurs à bouton-poussoir antiparasités

- 10 LEDs rouges, 8 LEDs vertes
- 04 affichages à 7 segments
- Oscillateurs 27 MHz et 50 MHz, entrée d'horloge SMA externe

❖ Réalisation de commande de moteur pas à pas par FPGA

La figure .IV. 9 ci-dessous représenter la réalisation de commande de moteur pas à pas par FPGA :

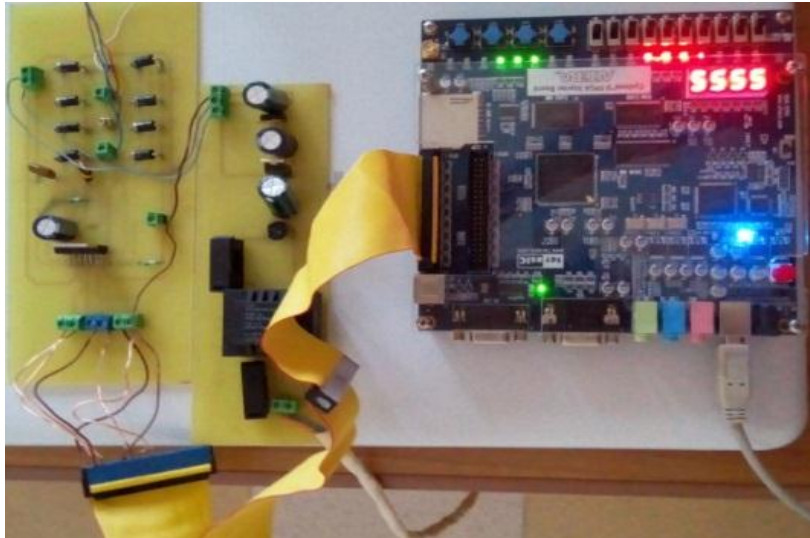
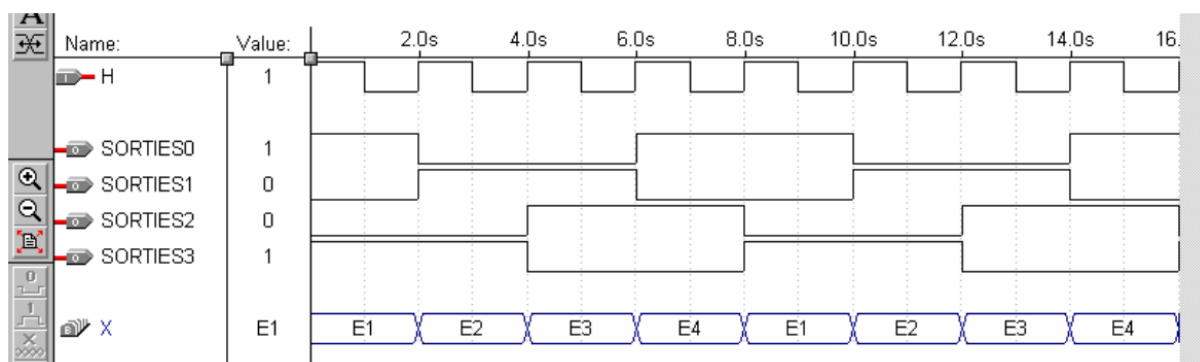


Figure. IV. 7 la Carte Réalisé

IV.6. moteur pas à pas à sens de rotation unique

le séquenceur destiné à la commande d'un moteur pas à pas, unipolaire 4 phases, à un seul sens de rotation, enfonctionnement pas entier deux phases .

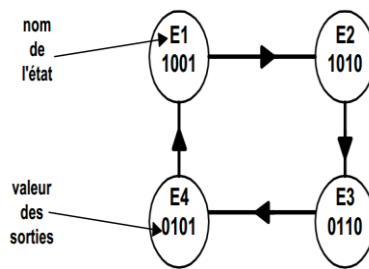
Nous avons ci-après les signaux recherchés S0, S1, S2 et S3 pour la commande des phases en fonction de l'horloge H :



On note 4 états possibles en sortie, s'enchaînant inconditionnellement dans l'ordre E1,

E2, E3, E4, E1...au rythme de l'horloge H.

A partir de ce cahier des charges, on peut en déduire le diagramme (ou diagramme à bulles) suivant :



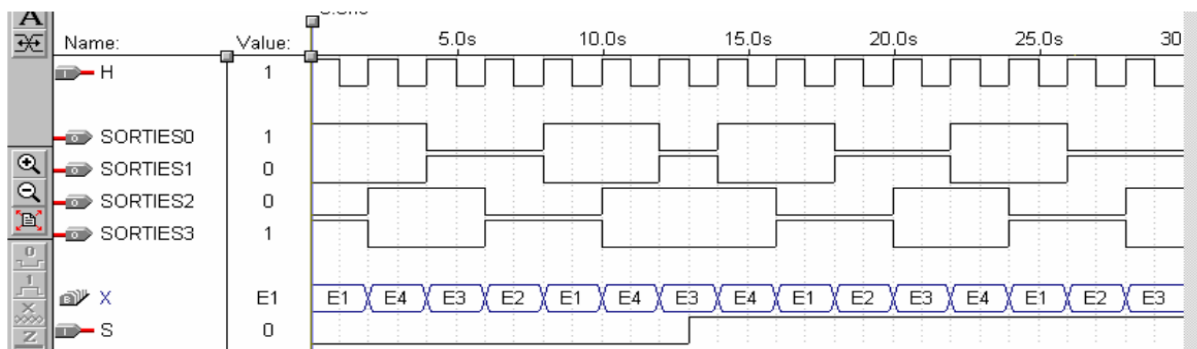
Ce diagramme représente les différents états du système, en précisant éventuellement les niveaux logiques correspondants en sortie, et les conditions de passage d'un état à l'autre. Dans notre cas très simple, à chaque coup d'horloge, on passe inconditionnellement à l'état suivant.

Le fichier VHLD associé à ce diagramme en annexe.

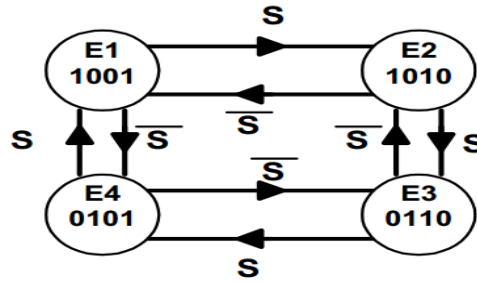
IV.7. Moteur pas a pas a double sens de rotation

Reprenons le moteur pas à pas à sens de rotation unique en introduisant cette fois une commande S permettant de choisir le sens de rotation. Nous supposons cette commande synchrone dans un premier temps.

Les chronogrammes recherchés sont alors :



Ce qui nous conduit au diagramme d'état suivant :



Remarques :

- lorsque plusieurs entrées interviennent dans un diagramme d'état, on suppose qu'une seule est susceptible de changer à un instant donné ;
- lorsque les conditions ne sont pas réalisées, l'état ne change pas (dans notre exemple, les conditions étant complémentaires, elles sont toujours réalisées) ;
- lorsque plusieurs branchements sont possibles à partir d'un état, les conditions ne doivent pas être vraies en même temps ;
- la présence de l'horloge est implicite, le passage d'un état à l'autre ne peut se faire qu'au front actif de l'horloge.

La description VHDL en annexe.

IV.8. Conclusion

Nous avons réalisé la carte d'alimentation délivrant les tensions 5v et 24v pour alimenter le circuit L298, ensuite implémenté les programmes de commande du moteur pas à pas qui contiennent les modules suivants :

- le diviseur d'horloge de 50MHz à 1Hz
- le séquenceur
- le décodeur pour afficher les états du moteur pas à pas.

Les tests en simulation et en temps ont été concluants.

Bibliographie

Electronique numérique et séquentielle par N. Richard chez Dunod

Logique programmable par L. Dutrieux et D. Demigny chez Eyrolles

Modélisation de systèmes intégrés numériques par A. Vachoux sur :

<http://lsmwww.epfl.ch/Site01OCT10/Courses/modelnum01.book.pdf>



CONCLUSION GENERALE

Conclusion Générale

Dans ce projet a été consacré à l'étude de différents types et structures de moteurs pas à pas rotatifs et linéaires ainsi que de différents modes de commande envisageables.

Cette étude a montré que pour générer un mouvement incrémental de translation, le moteur pas à pas linéaire à attaque directe représente une solution intéressante. En effet, cette solution distingue de celle utilisant un moteur pas à pas rotatif par l'absence d'organe de transmission, réduisant le nombre de composant augmentant ainsi la flexibilité du système.

Ce projet a vraiment eu lieu en deux étapes : la partie théorique avec toute la Recherche sur les composants et le fonctionnement d'un moteur pas à pas , les circuits FPGA et la partie pratique où nous avons réalisé le typon, percé la carte, soudé, effectué des tests. Il a été très intéressant de suivre le cheminement de réalisation de cette carte : en effet, dans le département STPI, nous avons pas l'occasion de concevoir une pièce.

La commande des moteurs pas à pas nécessite une séquence peut être implémenté sur une unité de calcul (micro – contrôleur , circuit FPGA).

Les moteurs pas à pas sont commandés soit en mode unipolaire, bi – polaire, en pas entier ou demi – pas et dans les deux sens de rotation horaire et anti – horaire.

La part de marché des FPGA dans le marché globale des circuits matériels pour l'électronique numérique ne cesse d'augmenter. Les évolutions technologiques et architecturales qui ont eu successivement lieu depuis le début des années 2000 ont fait de ces circuits de réels et rentables alternatives aux classiques ASIC. Avec ces évolutions c'est tout un nouveau domaine de l'électronique numérique qui s'est ouvert. Aujourd'hui les FPGA sont utilisés dans tous les domaines, des systèmes embarqués aux systèmes de communications, ils sont au cœur d'un important champ de recherche académique et industrielle.

Dans notre projet, nous avons utilisé un moteur pas à pas unipolaires de type PM 35L, à 6 fils avec une résistance interne de 120Ω entre phase et un pas de 7.5° . C'est-à-dire pour faire un tour il faut 48 pas.

ANNEXE

Annexe A:**moteur pas à pas à sens de rotation unique**

```
library ieee;
use ieee.std_logic_1164.all;

entity CDE_MOT_2 is
    port( H          : in  std_logic;
          SORTIES   : out std_logic_vector (3 downto 0));
end CDE_MOT_2;

architecture ARC_CDE OF CDE_MOT_2 is
    type TYPE_ETAT is (E1, E2, E3, E4);
    signal X:TYPE_ETAT;
begin
    process (H)
    begin
        if H'event and H = '1' then
            case X is
                when E1 => X <= E2;
                when E2 => X <= E3;
                when E3 => X <= E4;
                when others => X <= E1;
            end case;
        end if;
    end process;

    with X select
        SORTIES      <=  "1001" when E1,
                       "1010" when E2,
                       "0110" when E3,
                       "0101" when E4;
end ARC_CDE;
```

Annexe B:

moteur pas à pas à double sens de rotation

```
library ieee;
use ieee.std_logic_1164.all;

entity CDE_MOT_2S is
    port( H,S : in std_logic;
          SORTIES : out std_logic_vector (3 downto 0));
end CDE_MOT_2S;

architecture ARC_CDE OF CDE_MOT_2S is
    type TYPE_ETAT is (E1, E2, E3, E4);
    signal X:TYPE_ETAT;
begin
    process (H)
    begin
        if H'event and H = '1' then
            case X is
                when E1 =>
                    if S='1' then X <= E2;
                    else X <=E4;
                    end if;
                when E2 =>
                    if S='1' then X <= E3;
                    else X <=E1;
                    end if;
                when E3 =>
                    if S='1' then X <= E4;
                    else X <=E2;
                    end if;
                when others =>
                    if S='1' then X <= E1;
                    else X <=E3;
                    end if;
            end case;
        end if;
    end process;

    with X select
        SORTIES <= "1001" when E1,
                  "1010" when E2,
                  "0110" when E3,
                  "0101" when E4;
end ARC_CDE;
```

Annexe C:

Block div_1s

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;
--les paquetages std_logic_1164 et std_logic_unsigned de la bibliothèque ieee
--permettent respectivement l'utilisation du type std_logic et l'addition avec ce type

ENTITY div_1s IS

--H est le signal d'horloge à 50 MHz
--S est un signal de rapport cyclique 1/2, résultant de la division de l'horloge
--C_OUT est le signal de retenue qui passe au NL1 lorsque le compteur est plein
--le rapport de division dépend du modulo M
--ajuster en conséquence le nombre de bits du compteur interne X

--pour notre application on souhaite un signal à 1 Hz
--il faut donc diviser par 50 000 000, ce qui nécessite 26 bits pour X (2^26=67 108 864)

port (
        H          : IN STD_LOGIC;
        C_OUT      : OUT STD_LOGIC;
        S          : OUT STD_LOGIC);

END div_1s;

ARCHITECTURE archdiv OF div_1s IS

        SIGNAL X      : STD_LOGIC_VECTOR (25 downto 0);
        CONSTANT M: INTEGER :=50000000 ;

BEGIN
        PROCESS (H)
        BEGIN
        --compteur modulo M
                IF (H'EVENT AND H = '1') THEN
                        IF X >= M-1 THEN X <= (others=>'0');
                        --mise à 0 de tous les bits de X
                        ELSE X <= X + 1 ;
                        END IF;

                        END IF;
        END PROCESS;

        --à la moitié du comptage on change la valeur de S (rapport cyclique 1/2)
        S<='1' when X>=M/2 else '0';
        --mise à 1 de la retenue lorsque X passe à 0
        C_OUT<='1' when X=M-1 else '0';

END archdiv;

```

Annexe D:

Bloc DECODEUR

```
-- la bibliothèque ieee contient les paquetages dont la déclaration suit
library ieee;
--ce paquetage permet l'utilisation des types STD_LOGIC et STD_LOGIC_VECTOR
use ieee.std_logic_1164.all;
-- permet d'utiliser le type STD_LOGIC_VECTOR avec des entiers
use ieee.std_logic_unsigned.all;

entity DECODEUR is
    port (BCD : in STD_LOGIC_VECTOR (3 downto 0);
          SEG : out STD_LOGIC_VECTOR (6 downto 0) );
end DECODEUR;

    0
--      -----
--      | 0 |
--      |   |1
--      |   |
--      -----
--      | 6 |
--      |   |2
--      |   |
--      -----
--      3

architecture ARCH_DEC of DECODEUR is
    begin

-- attention les leds s'allument pour une sortie à 0
    SEG <= "1000000" when BCD=0 else
           "1111001" when BCD=1 else
           "0100100" when BCD=2 else
           "0110000" when BCD=3 else
           "0011001" when BCD=4 else
           "0010010" when BCD=5 else
           "0000010" when BCD=6 else
           "1111000" when BCD=7 else
           "0000000" when BCD=8 else
           "0010000" when BCD=9 else
           "1111111";

end ARCH_DEC ;
```

Annexe E:

Bloc séquenceur

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY sequenseur IS
    PORT (
        clock : IN STD_LOGIC;
        reset : IN STD_LOGIC := '0';
        S : IN STD_LOGIC := '0';
        P : IN STD_LOGIC := '0';
        C_EN : IN STD_LOGIC := '0';
        PH : OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
        ETAT : OUT STD_LOGIC_VECTOR(3 DOWNTO 0)
    );
END sequenseur;

ARCHITECTURE BEHAVIOR OF sequenseur IS
    TYPE type_fstate IS (E1,E2,E3,E4,E5,E6,E7,E8);
    SIGNAL fstate : type_fstate;
    SIGNAL reg_fstate : type_fstate;
    SIGNAL reg_PH : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
    SIGNAL reg_ETAT : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
BEGIN
    PROCESS (clock,reg_fstate)
    BEGIN
        IF (clock='1' AND clock'event) THEN
            fstate <= reg_fstate;
```

```

END IF;

END PROCESS;

PROCESS (fstate,reset,S,P,C_EN,reg_PH,reg_ETAT)

BEGIN

    IF (reset='1') THEN

        reg_fstate <= E1;

        reg_PH <= "0101";

        reg_ETAT <= "0001";

        PH <= "0101";

        ETAT <= "0001";

    ELSE

        -- reg_PH <= "0000";

        -- reg_ETAT <= "0000";

        --PH <= "0000";

        --ETAT <= "0000";

        CASE fstate IS

            WHEN E1 =>

                IF (((S = '1') AND (C_EN = '1'))) THEN

                    reg_fstate <= E2;

                ELSIF ((NOT((S = '1')) AND (C_EN = '1'))) THEN

                    reg_fstate <= E8;

                -- Inserting 'else' block to prevent latch inference

            ELSE

                reg_fstate <= E1;

            END IF;

            reg_PH <= "0101";

            reg_ETAT <= "0001";

        WHEN E2 =>

```

```
IF (((S = '1') AND (P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E3;
ELSIF (((NOT((S = '1')) AND (P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E1;
ELSIF (((S = '1') AND NOT((P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E4;
ELSIF (((NOT((S = '1')) AND NOT((P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E8;
-- Inserting 'else' block to prevent latch inference
ELSE
    reg_fstate <= E2;
END IF;
reg_PH <= "0001";
reg_ETAT <= "0010";
WHEN E3 =>
    IF (((S = '1') AND (C_EN = '1')) THEN
        reg_fstate <= E4;
    ELSIF ((NOT((S = '1')) AND (C_EN = '1')) THEN
        reg_fstate <= E2;
    -- Inserting 'else' block to prevent latch inference
    ELSE
        reg_fstate <= E3;
    END IF;
    reg_PH <= "1001";
    reg_ETAT <= "0011";
WHEN E4 =>
```

```

IF (((S = '1') AND (P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E5;
ELSIF (((NOT((S = '1')) AND (P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E3;
ELSIF (((S = '1') AND NOT((P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E6;
ELSIF (((NOT((S = '1')) AND NOT((P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E2;
-- Inserting 'else' block to prevent latch inference
ELSE
    reg_fstate <= E4;
END IF;

reg_PH <= "1000";
reg_ETAT <= "0100";

WHEN E5 =>
    IF (((S = '1') AND (C_EN = '1')) THEN
        reg_fstate <= E6;
    ELSIF ((NOT((S = '1')) AND (C_EN = '1')) THEN
        reg_fstate <= E4;
    -- Inserting 'else' block to prevent latch inference
    ELSE
        reg_fstate <= E5;
    END IF;

    reg_PH <= "1010";
    reg_ETAT <= "0101";

WHEN E6 =>

```

```
IF (((S = '1') AND (P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E7;
ELSIF (((NOT((S = '1')) AND (P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E5;
ELSIF (((S = '1') AND NOT((P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E8;
ELSIF (((NOT((S = '1')) AND NOT((P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E4;
-- Inserting 'else' block to prevent latch inference
ELSE
    reg_fstate <= E6;
END IF;
reg_PH <= "0010";

reg_ETAT <= "0110";
WHEN E7 =>
    IF (((S = '1') AND (C_EN = '1')) THEN
        reg_fstate <= E8;
    ELSIF ((NOT((S = '1')) AND (C_EN = '1')) THEN
        reg_fstate <= E6;
    -- Inserting 'else' block to prevent latch inference
    ELSE
        reg_fstate <= E7;
    END IF;
    reg_PH <= "0110";

    reg_ETAT <= "0111";
WHEN E8 =>
```

```

IF (((S = '1') AND (P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E1;
ELSIF (((NOT((S = '1')) AND (P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E7;
ELSIF (((S = '1') AND NOT((P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E2;
ELSIF (((NOT((S = '1')) AND NOT((P = '1')) AND (C_EN = '1')) THEN
    reg_fstate <= E6;
-- Inserting 'else' block to prevent latch inference
ELSE
    reg_fstate <= E8;
END IF;

reg_PH <= "0100";
reg_ETAT <= "1000";

WHEN OTHERS =>
    reg_PH <= "XXXX";
    reg_ETAT <= "XXXX";
    report "Reach undefined state";

END CASE;

PH <= reg_PH;
ETAT <= reg_ETAT;

END IF;

END PROCESS;

END BEHAVIOR;

```

Annexe F:**Les bornes du FPGA reliées au circuit L298**

Broche du L298 sur la carte auxiliaire	Couleur sur la carte auxiliaire	Broche du FPGA sur la carte DE1
5 : In1	Blanc	E14
6 : EnA	Vert	C14
7 : In2	Jaune	G15
10 : In3	Jaune	D16
11 : EnB	Vert	H12
12 : In4	Blanc	C17