



République Algérienne Démocratique et Populaire Ministère de
l'Enseignement Supérieure
et de la Recherche Scientifique



Université Echahid Hamma Lakhdar d'El-Oued

**FACULTE DE TECHNOLOGIE DEPARTEMENT
DE GENIE MECANIQUE**

Mémoire de Fin d'Étude

Domaine : Sciences et Technologie

Niveau : 2^{ème} Année Master

Spécialité : Electromécanique

Thème

**Commande d'un moteur Pas à Pas en Utilisant
le Microprocesseur 8086 sur PROTEUS**

Présenté par :

 **TOURQUI M^{ed} EL FETEH LARBI**

 **DAHA IDRISSE**

 **YAHYAI MOHAMMED**

Encadré par :

 **Mr. GUIA HOSSAM**

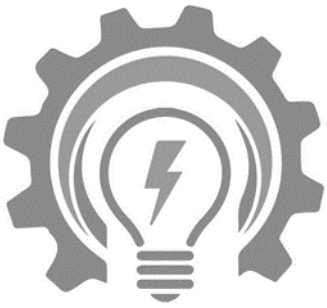
Jury par :

Dr.LAOUAMER MOSBAH(président)

Dr.REMHA SOUHAIB (examineur)

2022-2023

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
قُلْ هُوَ اللَّهُ أَحَدٌ
لِلَّهِ الْحُكْمُ وَإِلَيْهِ أُنزِلَتْ السُّرُورُ



REMERCIEMENT

Nous tenons tout d'abord, à remercier Dieu tout puissant, de nous avoir donné la santé et la volonté
pour accomplir ce travail.

Ainsi, je tiens également à exprimer mes vifs remerciements à mon encadreur Mr. HOUSSEM GUIA
pour avoir d'abord proposée ce sujet, pour le suivi continu tout le long de la réalisation de ce mémoire
et qui n'a pas cessée de me donner ses conseils.

L'ensemble des enseignants de l'institut d'électrotechnique pour leurs efforts pendant les années
d'étude.

En fin Nous adressons notre remerciement les plus profonds et les distinguées à tous ceux qui nous
aidons de proche ou de loin pour accomplir ce travail.

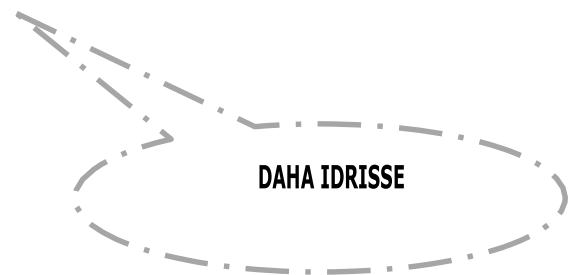
DEDICACES

Je dédie ce travail à:

Mes très chers parents pour leurs sacrifices et leurs
encouragements durant toutes mes études. Mes sœurs,
Mes frères.

Mon encadreur pour ses orientations, son encouragement et son aide.

Mes amies, et Mes collègues.



DEDICACES

A tout la famille

Pour tout le confort et le bonheur qu'elle m'a procuré, pour le respect
et

l'encouragement qu'elle m'a fait sentir, et pour avoir fait de moi ce que
je suis.

A mes adorables frères et sœurs,

Dont je suis fière pour leur soutien morale et leur amour.

A tous mes chers amis et ceux qui m'aiment,

Pour leurs encouragements et leurs soutiens morales.

A mon encadreur

Qui m'ont soutenu à réaliser ce travail, dans lequel je souhaite qu'il
trouvent le résultat de leurs encouragement.



YAHYAI MOHAMMED

DEDICACES

J'ai l'honneur de dédier ce travail à mes chers parents, pour leur aide
et leurs soutient continuel.

.Ma mère, la lumière des yeux et source de ma volonté

.Mon père qui a sacrifié tout pour je puisse étudier

A mes frères et ma sœur

.A tout ma famille

.A mes copains et copines

A tout personne qui ma connait et m'aimé surtout ma promotion



TOURQUI Med EL FETEH LARBI

Table des matières Table des matières

Remerciement	I
Dédicace	II
Table des matières.....	IV
Liste des figures.....	VII
Liste des tableau.....	VIII
Introduction Générale	01
Chapitre I : Généralité sur les moteurs pas à pas	
1. Introduction.....	02
2. Principe de fonctionnement.....	02
3. Différents type.....	03
3.1. Nombre de Phase	03
: 3.1.1. Moteurs bipolaires	03
3.1.2. Moteurs unipolaires	
3.2. La construction.....	04
: 3.2.1. Moteur à aimants permanents	04
: 3.2.2. Moteurs à réluctance variable	05
: 3.2.3. Moteur hybride	07
3.3. Comparaison des performances pour les trois types de moteurs pas à pas	09
4. Avantage et inconvénients.....	09
5. Les applications des moteurs pas à pas.....	10
6. Command des moteurs pas à pas	11
6.1. La commande en boucle ouverte	12
6.1. La commande en boucle fermé	13
7. Conclusion	14
Chapitre II : Généralité sur le microprocesseur 8086	
1 Introduction	15
2. Structure interne et externe	15
2.1. Structure interne	15
2.1.A. L'unité d'interface de bus.....	15
2.1.B. L'unité d'exécution.....	15
2.1.C. Les Registres	16
2.2. Structure externe.....	16
3. Programmation assembleur	16
3.1. Structure d'un programme assembleur	23
3.1.1. Les directives	23
3.1.2. Les instructions	23
3.1.3. Les modes d'adressage	23
4. Applications des microprocesseurs	25
5. Conclusion	27
	29
Chapitre III: Simulation Sur PROTEUS	

1. Introduction.....	30
2. Description de PROTEUS	31
3. Description des différents composants utilise.....	31
3.1. Le microprocesseur	32
3.2. Périphérique I/O 0588A	32
3.3. 47HC373 Latch	37
3.4. Un convertisseur analogique numérique ADC0804.....	38
3.5. Moteur pas à pas	40
3.6. L293D	41
4. Description du schéma de commande.....	43
4.1. Variation de la vitesse.....	43
4.2. L'Affichage.....	45
5. Description du programme	45
6. Résultats de simulation	49
7. Conclusion	50
Conclusion Générale	51
BIBLIOGRAPHIE.....	55



Liste des figures

Nombre	Titre	Page
1	Schéma de principe d'un moteur pas-à-pas	02
2	le principe de fonctionnement du moteur pas à pas	03
3	moteur pas à pas bipolaire	04
4	moteur pas à pas unipolaire	04
5	Moteur pas-à-pas à aimant permanentant	05
6	moteur à réluctance variable	06
7	Schéma de principe de fonctionnement du moteur pas-à-pas à reluctance variable	07
8	Moteur pas à pas hybride	08
9	Principe d'un moteur pas-à-pas	11
10	Commande en boucle ouverte	12
11	Commande par auto-asservissement	13
12	Organisation interne du 8086	16
13	Brochage et Schéma fonctionnel du 8086	19
14	branchement d'un générateur d'horloge	20
15	Chronogramme de séparation de bus A/D	20
16	Chronogramme de sens de transfert de données sur le bus de données	21
17	Adressage des entrées sorties	26
18	Adressage relatif	26
19	Microprocesseur 8086	32
20	Le 8255A	33
21	Jetons d'abord un coup d'œil au diagramme des broches de l'Intel 8255A –	35
22	Discutons maintenant de la description fonctionnelle des broches dans 8255A.	35
23	Latch/74HC373	37
24	DIAGRAMME DE CONNEXION DIP	38
25	DIAGRAMMES LOGIQUE	39
26	Brochage	40
27	Schéma d'application typique	40
28	Moteur pas à pas unipolaire et bipolaire	41
29	Pilote L293D	42
30	Le L293D	42
31	SCHÉMA	43
32	Broche De Connexions	44



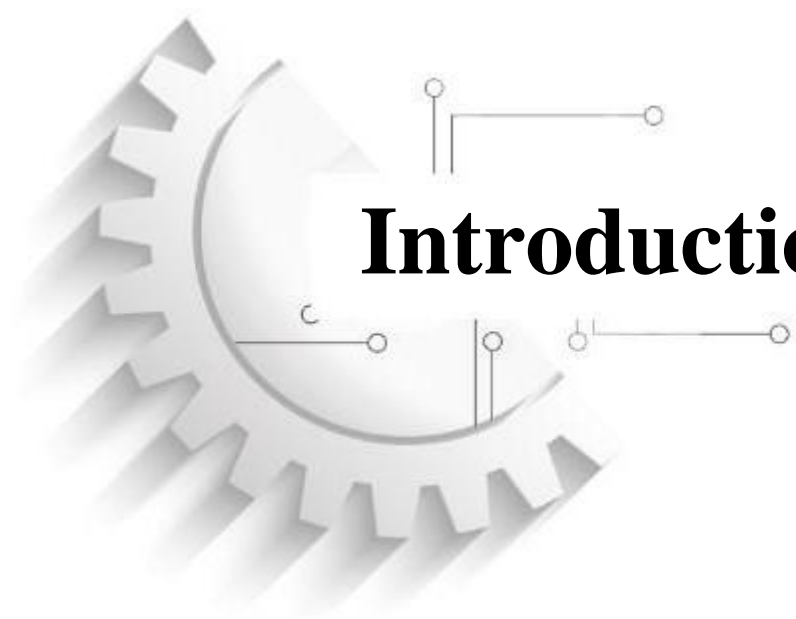
Liste des figures

33	schéma de commande d'un moteur pas à pas	45
34	Changer le sens de rotation	49
35	Changer la vitesse de rotation	49



Liste des Tableaux

Nombre	Titre	Page
01	comparaison des performances pour les trois types de moteurs pas à pas	09
02	signal de lecture de l'octet de poids fort du bus de données	22
03	Leur résultat dépend des conditions suivantes	36
04	le tableau montrant leurs différents signaux avec leur résultat.	37
05	TABLE DE VÉRITÉ	38
06	Informations de commande	39



Introduction Générale

Introduction Générale

Le moteur pas-à-pas constitue un convertisseur électromécanique destiné à transformer le signal électrique en déplacement (angulaire ou linéaire) mécanique. Ces moteurs se sont développés dû à l'apparition des microprocesseurs. Le principe d'un moteur pas à pas est connu depuis longtemps, mais son développement a commencé dans les années 1960 grâce au développement de l'électronique numérique. Lors de la modification de la vitesse de déplacement et du sens de rotation, il devient impératif d'utiliser des moteurs pas à pas. Il est utilisé partout où la précision de positionnement est requise. [1]




Il existe de nombreux systèmes de contrôle dédiés aux moteurs pas à pas. Aujourd'hui, la commande la plus courante est celle via un micro-ordinateur, grâce à l'interface qui permet d'établir une communication entre l'unité centrale et les périphériques. Les chapitres 1 et 2 présentent les moteurs pas à pas et les microprocesseurs d'un point de vue théorique, tandis que le chapitre 3 est consacré à la simulation du système de commande des moteurs pas à pas. [3]

Objectif du mémoire

L'objectif de notre étude est de simuler un système de commande d'un moteur pas à pas. Pour objectif d'allumer éteindre un moteur pas à pas (unipolaire ou bipolaire), de changer son sens de rotation et de varier la vitesse.

Organisation du mémoire

Le travail présent dans ce mémoire sera reparti en quatre chapitres :

-  **Le premier chapitre** nous avons étudié les moteurs pas-à-pas, leurs électroniques de puissance et leurs commandes. Et nous avons présenté les trois grandes familles du moteur pas à pas, et les caractéristiques de chacun.
-  **Dans le deuxième chapitre** on est parlé sur le microprocesseur 8086 ses structures interne et externe, leur programmation en langage assembleur et ses différents domaines d'application.
-  **Le troisième chapitre** est consacré à la mise en œuvre du contrôle du moteur pas à pas. Nous terminerons par une conclusion générale sur le travail effectuée.

A stylized, light gray gear is positioned on the left side of the page. Overlaid on the gear is a white rectangular area containing a simplified circuit diagram. The diagram includes a small circle at the top, a horizontal line, and several other circles and lines representing electrical connections.

Chapitre I :

Généralité sur les moteurs pas à pas

1. Introduction

Dans ce chapitre, nous présenterons qu'un moteur pas à pas est un convertisseur électromécanique qui assure la conversion d'un signal électrique pulsé en un déplacement mécanique (angulaire ou linéaire). Sa structure de base est constituée de deux composants séparés mécaniquement, un stator et un rotor. Interaction électromagnétique entre ces deux pièces assure la rotation. [2]

Ces moteurs ne se sont développés qu'à partir de 1960, dû à l'apparition des microprocesseurs, aux progrès de l'électronique de puissance et des matériaux. Ils permettent d'obtenir une grande précision grâce à leur conception mécanique et électrique, d'où, on les utilise surtout dans les applications demandant un positionnement précis, en boucle ouverte (sans asservissement). [1]

2. Principe de fonctionnement des moteurs pas à pas

Le fonctionnement d'un moteur pas à pas nécessite la présence des éléments suivants:

- Une unité de commande (microprocesseur par exemple) qui fournit des impulsions dont la fréquence est proportionnelle à la vitesse de rotation du moteur, elle imposera également le sens de rotation.
- Un séquenceur qui aiguillera les impulsions sur les différentes bobines du moteur.
- Une alimentation de puissance. [3]

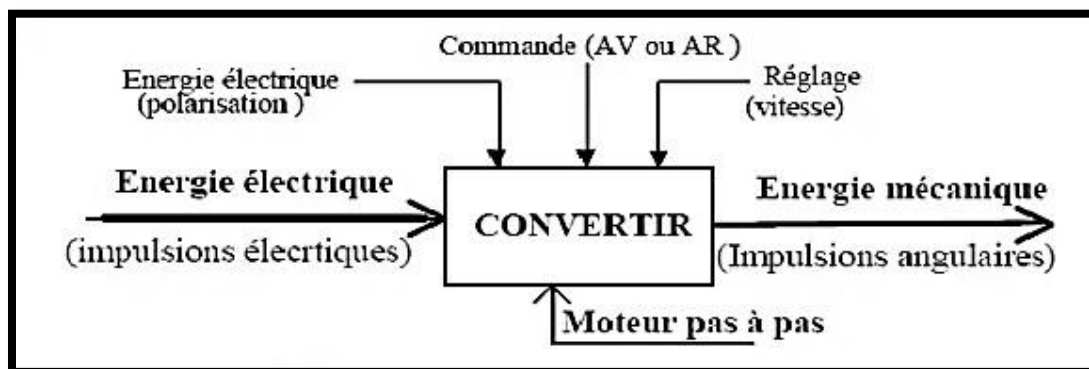


Figure 01 : Schéma de principe d'un moteur pas-à-pas

Le moteur pas-à-pas constitue un convertisseur électromécanique destiné à transformer le signal électrique (impulsion ou) 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 réluctance ou passive). [8]

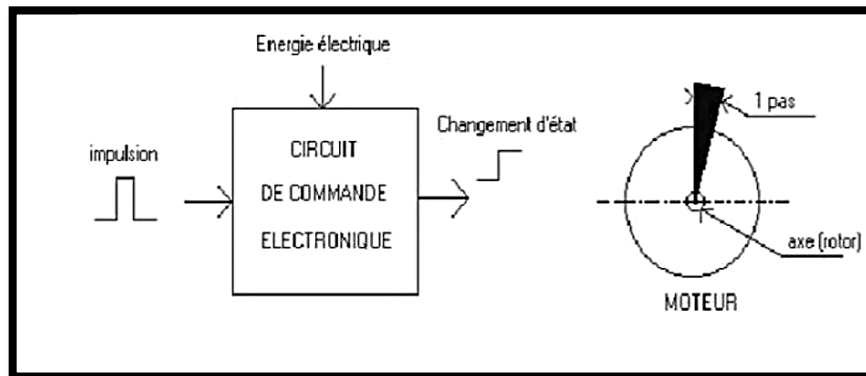


Figure 02 : le principe de fonctionnement du moteur pas à pas

Ces moteurs comme tous les autres moteurs comportent un rotor et un stator. Plus le nombre d'aimants et de bobines augmente, plus le nombre de pas sera plus grand, et plus la précision augmente. [9]. Un moteur pas à pas transforme une impulsion électrique en une énergie mécanique permettant le déplacement angulaire du rotor, appelé « pas ». [10]

3. Les différents types des moteurs pas à pas

Tous les moteurs pas-à-pas comprennent un stator portant des bobines dans lesquelles ; Le courant est commuté par l'électronique de commande. Par contre, les différences apparaissent au niveau du rotor. [12]

Les types des moteurs pas-à-pas peut être divisés selon deux critères de base :

- Nombre de Phase
- Selon la construction

3.1. Nombre de Phase

Selon l'alimentation des bobines, on distingue deux types :

1. Moteurs bipolaires :

Les phases d'un moteur bipolaire sont alimentées une fois dans un sens et d'une autre fois dans le sens inverse. Ils créent une fois un pôle nord et l'autre fois un pôle sud d'où le nom de bipolaire. [12]

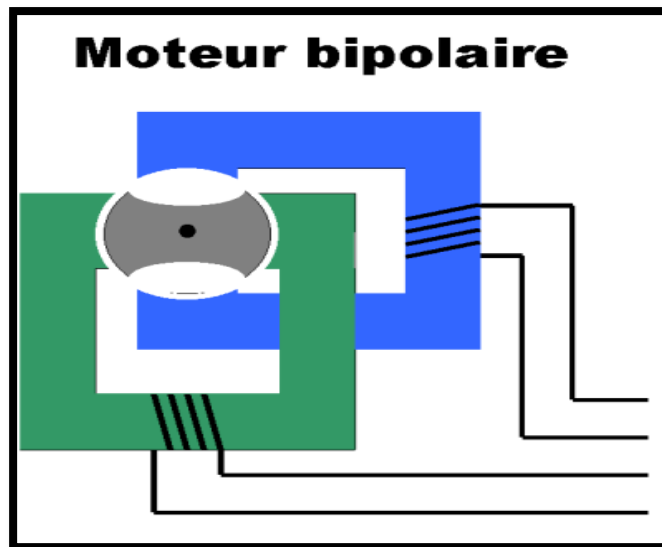


Figure 03 : moteur pas à pas bipolaire

2. Moteurs unipolaires :

Les phases sont alimentées dans un seul sens. Généralement les moteurs bipolaires possèdent 2 phases et les moteurs unipolaires possèdent 4 phases. [12]

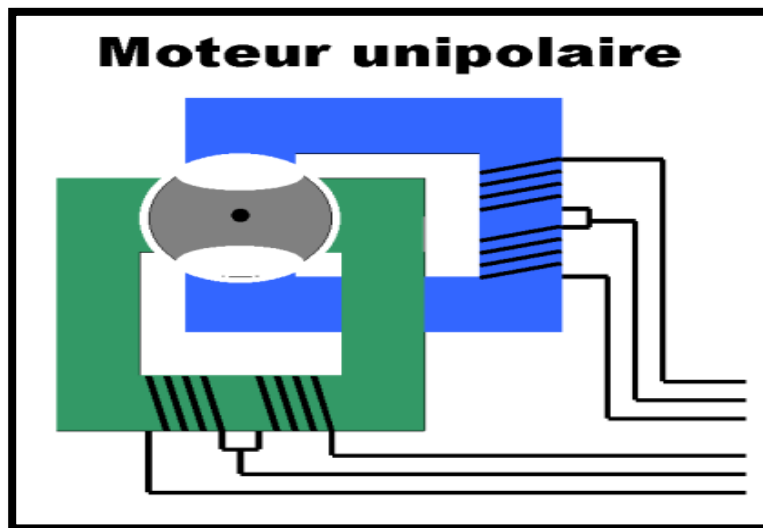


Figure 04 : moteur pas à pas unipolaire

3.2. Selon la construction

Selon la conception on distingue trois principaux types de moteurs pas à pas :

a. Moteur à aimants permanents :

a.1. Description

Les moteurs à aimants permanents sont constitués par le rotor qui possède des pôles NORD et SUD. À cause des aimants permanents, le rotor reste freiné à sa dernière position lorsque le bloc d'alimentation cesse de fournir des impulsions. Une façon simple de voir le système, est

de placer une boussole entre deux aimants. Suivant la bobine qui est alimentée et le sens du courant, l'aimant va s'aligner avec le champ. [4]

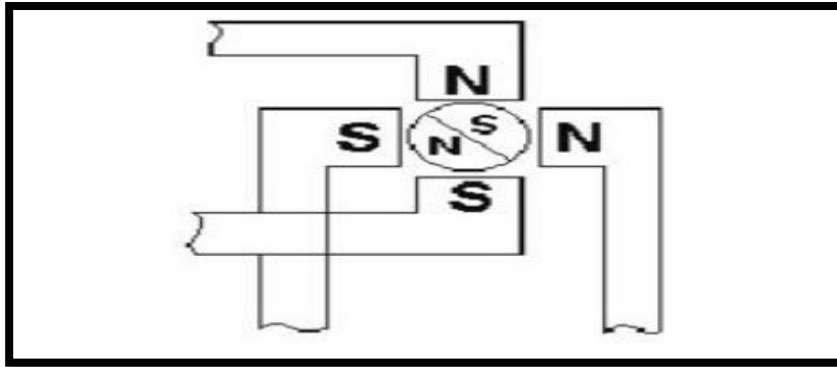


Figure 05 : Moteur pas-à-pas à aimant permanentant

Contrairement au cas précédent, ici le moteur (Permanent Magnet motor PM). Possède 2 aimants au rotor. Dans ce cas, sans courant dans les bobinages statoriques, il existe bel et bien un couple réactif. En observant ce couple il est facile de déterminer le nombre de par tour du moteur. [14]

En principe ce type de moteur, grâce à la construction particulière du rotor, n'a pas de variations de perméance propre et mutuelle entre bobinages statoriques. Dans sa version simple, le moteur pas à pas à aimants permanents peut être excité par un stator à structure saillante ou à structure lisse. Le rotor possède un seul aimant.

a.2 Principe de fonctionnement :

Le principe de ce type de moteur est différent à celui de moteur unipolaire. Le courant doit changer de sens dans les enroulements à chaque pas effectué. Ce moteur, comme le modelé précédent peut-être alimenté sous 3 trois séquences différentes. [14]

Le rotor porte des aimant permanents et considérons le stator comporte quatre bobines et le rotor est un aimant bipolaire [12]

b. Moteurs à réluctance variable :

b.1. Description

Le moteur à réluctance variable possède un rotor en acier doux non magnétique. Ce rotor est Constitué d'un nombre de pôles supérieurs à celui du stator. A chaque impulsion de la commande, la phase suivante du stator est alimentée. On constate que les pôles du rotor 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. [4]

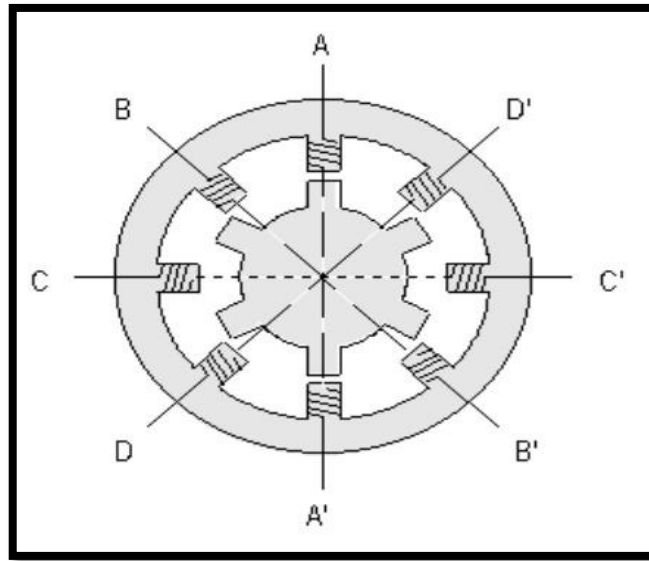


Figure 06 : moteur à réluctance variable

Le moteur pas-à-pas réluctant (Variable reluctance motor VR) comporte n_b bobines distinctes au stator et un rotor ferromagnétique sans aimant ($n_a=0$). Sans courant dans les bobinages statoriques, ce type de moteur ne présente pas de couple réluctant. Les circuits magnétiques du rotor et du stator sont assemblés à partir de tôles magnétiques de haute perméabilité (fer-silicium ou même fer-cobalt). Certaines machines, destinées au positionnement ou à une rotation très lente, peuvent avoir un rotor ou/et un stator en fer massif. Il existe un grand nombre d'astuces de construction pour augmenter le nombre de pas par tour, comme par exemple les structures bifilaires qui ne seront pas étudiées dans cette recherche. Pour minimiser la force magnétomotrice et augmenter la puissance massique, l'entrefer des moteurs pas à pas à réluctance variable doit être aussi petit que possible. Plusieurs géométries du rotor et du stator ont été ainsi imaginées par les constructeurs pour remplir des cahiers des charges particuliers. [12]

On distingue trois géométries de base, à savoir :

- Les moteurs pas à pas à réluctance variable à plots statoriques non dentés
- Les moteurs pas à pas à réluctance variable à plots statoriques dentés
- Les moteurs pas à pas à réluctance variable à plots statoriques non dentés décalés

b.2 Principe de fonctionnement

Le positionnement se fait suivant le principe de la réluctance minimale. L'axe des deux dents du rotor s'aligne avec l'axe des deux bobines alimentées pour minimiser la réluctance du trajet offert aux lignes de champ. A chaque commutation, ce sont les dents les plus proches de plots alimentés qui s'alignent en face d'eux. On peut avoir les 3 modes de fonctionnement : mode monophasé, mode biphasé et mode demi-pas. [14]

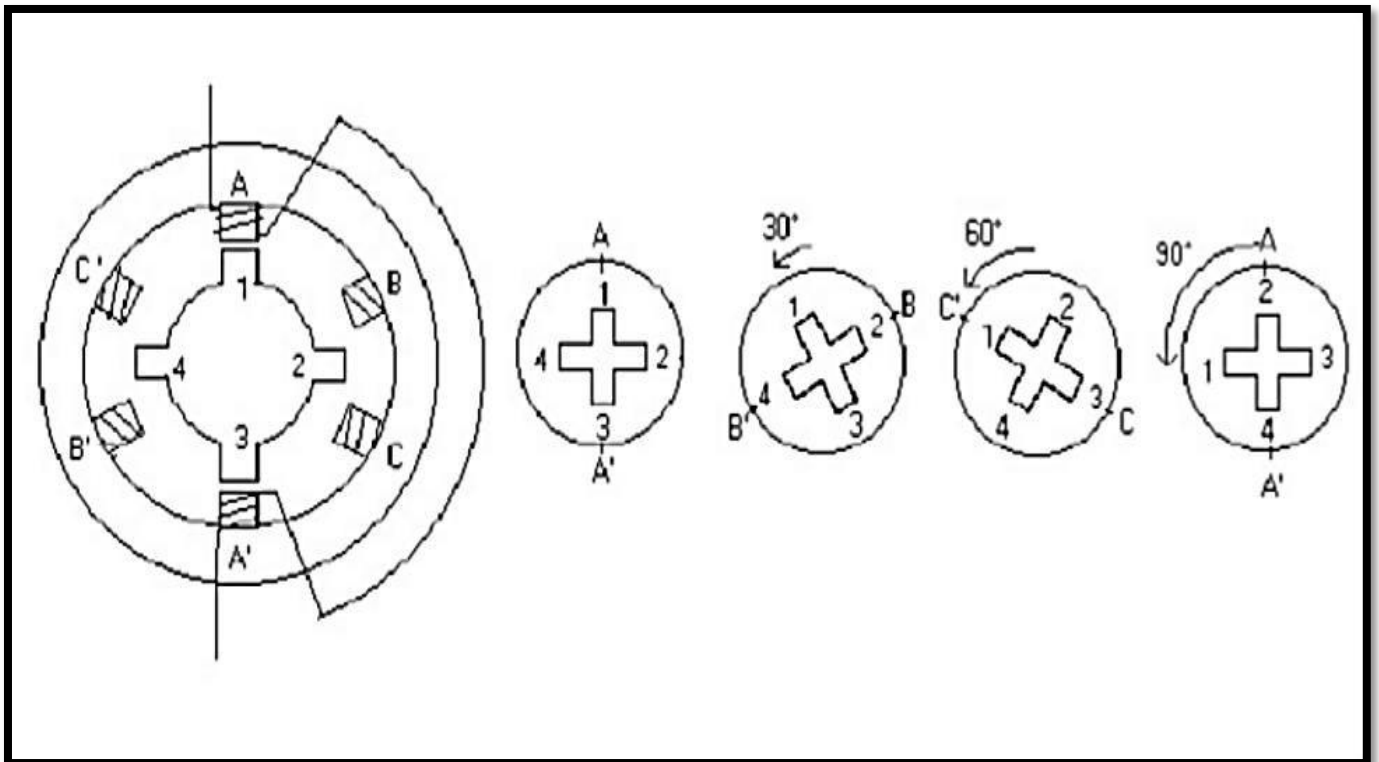


Figure 07 : Schéma de principe de fonctionnement du moteur pas-à-pas à réluctance variable

c Moteur hybride :

c.1. Description

Les moteurs hybrides sont à haute résolution. Le nombre de pas par tour est très important. Ce Moteur utilise les deux principes précédents, ce qui permet d'exploiter les caractéristiques intéressantes de chacun des deux. Le rotor comporte en périphérie des dentures en matériau perméable, polarisées par aimant. [4]

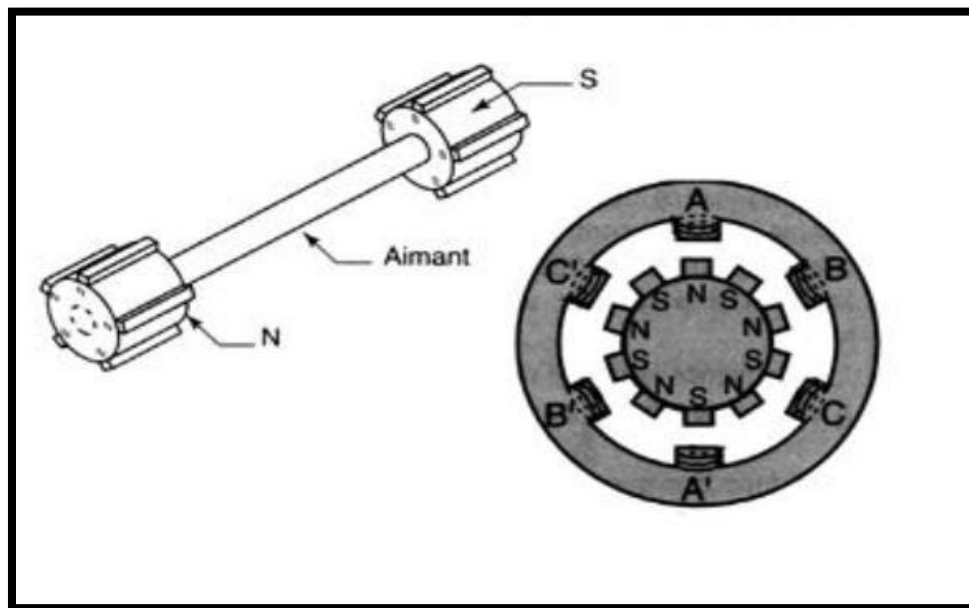


Figure 08 : Moteur pas à pas hybride

En combinant les structures des deux moteurs précédents, c'est-à-dire en plaçant les aimants du moteur à aimants permanents dans un circuit ferromagnétique, on crée un nouveau type de moteur appelé moteur réluctant polarisé ou moteur hybride (Hybrid motor : HB). Dans ce cas, il existe un couple réluctant provoqué par la variation de perméances propres associées à chaque aimant et à chaque bobine. [12]

p : nombre de pôles magnétiques vu par le stator

N_{dr} Nombre de dents sur une couronne rotorique

m : nombre de phases (doit être pair)

L'augmentation du nombre de plots statoriques alimentés simultanément permet d'augmenter le nombre de dents du rotor, et donc de diminuer le pas angulaire du rotor. Le même résultat s'obtient par la subdivision des plots en plusieurs dents. Les moteurs pas à pas hybrides comptent parmi les moteurs pas à pas les plus fabriqués. Ils existent aussi bien en structure aux circuits simples (single stack), qu'en structure multiple (multi stack). Dans presque tous les cas, les plots dentés du stator présentent le même pas que les dents aimantées du rotor L . [12]

c.2. Principe de fonctionnement

Le nombre de dents au rotor est différent de celui du stator. Quand on alimente une paire de bobines, le rotor place les dents Nord et Sud de telle façon que le flux traversant le rotor soit maximale. [14]

3.3. Comparaison des performances pour les trois types de moteurs pas à pas

Le tableau ci-dessous présente les avantages de l'un par rapport à l'autre [3]

Type de moteur pas à pas	Moteur à réluctance variable	Moteur à aimants permanents	Moteur hybride
Résolution (nb de pas par tour)	Bonne	Moyenne	Elevée
Couple moteur	Faible	Elevée	Elevée
Sens de rotation	Dépend : - de l'ordre d'alimentation des phases	Dépend : - de l'ordre d'alimentation des phases - du sens du courant dans les bobines	Dépend : - de l'ordre d'alimentation des phases - du sens du courant dans les bobines
Fréquence de travail	Grande	Faible	
Puissance	Quelques Watts	Quelques dizaines de Watts	Quelques KWatts
Inconvénients	Pas de mémoire de position		

Tableau 01: comparaison des performances pour les trois types de moteurs pas à pas

4. Les avantages et les inconvénients des moteurs pas à pas :

L'avantage le plus important du moteur pas à pas qu'il est très simplifié puisqu'il existe des circuits intégrés qui transforment directement un train d'impulsion en commande des phases, en tenant compte du sens désiré. Un autre avantage réside dans le fait que le moteur pas à pas ne nécessite pas trop d'entretien, et que son usure est faible. De plus, il est possible de bloquer l'arbre sous tension sans que ceci ne nuise au moteur. [5]

Comme inconvénients, il faut noter que la rotation se fait par coups et à une base vitesse, qu'il oscille lorsqu'il passe d'un pas à l'autre et que si le couple de charge est supérieur au couple moteur. [5]

Avantages :

- L'angle de rotation du moteur est proportionnel au nombre d'impulsion d'entrée. On peut déterminer l'angle exact de rotation.
- Existence d'un couple à l'arrêt appelé « couple de maintien » (si les enroulements sont alimentés).
- Fort couple à basse vitesse.
- Durée de vie quasi illimitée puisque l'usure mécanique est pratiquement inexistante (absence de frottement), sans nécessiter d'entretien. Par conséquent, la durée de vie dépend de celle du roulement. [6]
- Commande en boucle ouverte, qui signifie qu'aucune rétroaction (boucle de retour) n'est nécessaire.
- Positionnement précis.
- Un éventail de vitesse peut être réalisé étant donné que la vitesse est proportionnelle à la fréquence d'impulsion. Il est possible de le faire tourner très lentement en contrôlant la vitesse.
- Faible prix.

Inconvénients :

- Vitesse de rotation réduite. Le moteur peut en effet ne pas suivre toutes les impulsions de commande.
- Risque de perte de pas (et donc de dysfonctionnement du contrôle en position) en cas de perturbation anormale notamment de la charge ou de fréquence excessive des impulsions.
- Des résonances peuvent se produire si la commande n'est pas bien contrôlée.
- Vibrations souvent génératrices de bruit acoustique. [6]

5. Les applications des moteurs pas à pas

Le moteur pas à pas a été conçu à partir de deux démarches logiques très différentes [11]

- Les moteurs pas à pas sont très utilisés dans toutes les applications mécaniques où l'on doit contrôler simplement la position ou la vitesse d'un système en boucle ouverte typiquement dans les systèmes de positionnement et d'indexation .
- Ils sont par exemple utilisés en robotique (positionnement des axes, vitesse variable du robot), les imprimantes classiques et imprimantes 3D, les scanners, les photocopieurs et les platines vinyle de DJ.
- En informatique (déplacement d'une tête de lecture d'une lecture, déplacement de la tête d'impression, avance du papier pour les imprimantes).
- En astronomie (positionnement du télescope pour suivre un astre) et dans la réception TV par satellites (positionnement d'une parabole).
- Ils sont présents aussi dans les pompes à perfusion, pousse-seringues, système de positionnement sur machine industrielle et machine-outil. [3]

6. Commande d'un moteur pas-à-pas :

On constate que le système est beaucoup plus simple.

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]

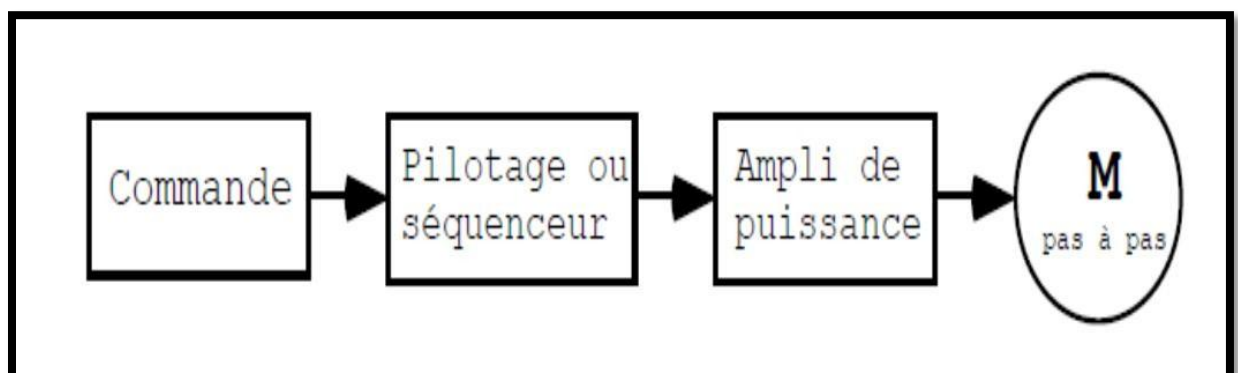


Figure 09 : Principe d'un moteur pas-à-pas

Les moteurs pas à pas tirent leur nom du fait qu'ils sont capables d'assurer un déplacement mécanique constitué d'incrémentaires unitaires appelés pas, donc dans ce type de moteur l'application d'une ou plusieurs impulsions de commande se traduit par une quantité égale de

déplacement. De même, l'application d'une impulsion de commande à fréquence fixe fait tourner le moteur à une vitesse de rotation quasi constante. La nature synchrone des moteurs pas à pas permet de convertir des informations numériques en mouvement incrémentiel en contrôlant la position ou la vitesse sans avoir besoin de capteurs de position. [13]

La commande assure la succession des ordres de commutations de phase à un rythme compatible, avec les fonctions à réaliser compte tenu du caractère synchrone du moteur pas à pas, cette commande peut s'effectuer en boucle ouverte, cependant pour accroître la fiabilité ou les performances. Cette commande peut s'effectuer en boucle fermée (la commande en régime auto asservissement). [13]

6.1. La commande en boucle ouverte

Dans le premier type de commande, les impulsions sont appliquées au moteur selon un ordre préétabli. Une alimentation de puissance contrôle et alimente séquentiellement les phases. [14]

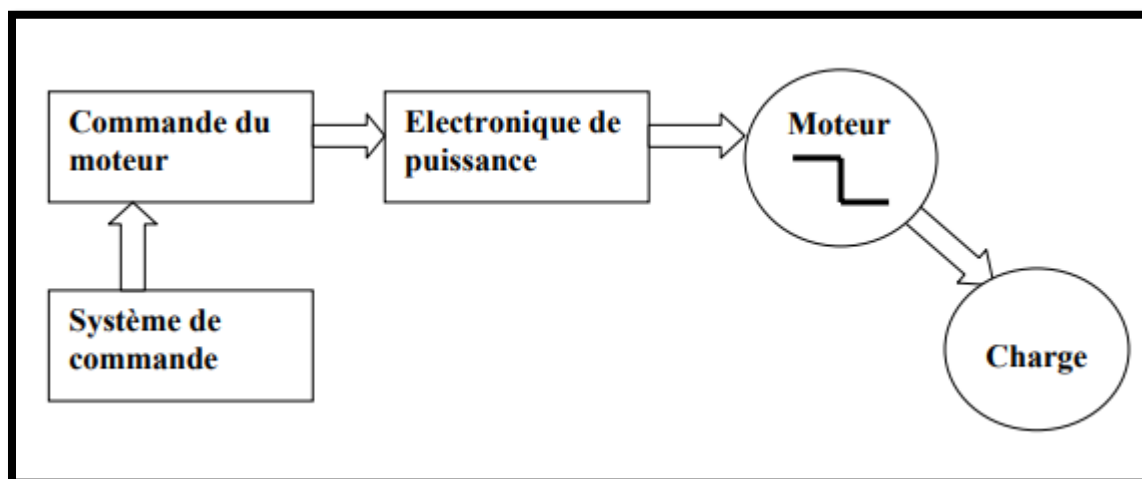


Figure 10 : Commande en boucle ouverte

Cette commande permet d'exploiter le caractère synchrone du moteur pas à pas pour contrôler un déplacement incrémental. L'avantage de cette méthode est son prix modeste. [14]

Les désavantages principaux sont:

- ❖ La possibilité de perte de synchronisme en cas de phénomène d'instabilité.
- ❖ La sensibilité aux variations de charge qui imposent des marges de sécurité importantes.

Les outils d'analyse sont les courbes qui représentent le déplacement et les plans de phase où l'on représente la vitesse en fonction du déplacement. [14]

6.2. La commande en boucle fermée

D'autres solutions utilisent des capteurs supplémentaires qui fournissent des informations sur la vitesse ou la position. Bien qu'un asservissement efficace puisse ainsi être obtenu, le prix associé à cet élément mécanique supplémentaire peut augmenter significativement. Aussi, utiliser un capteur de position pour un moteur dont la principale caractéristique est de permettre un mouvement incrémental en boucle ouverte peut sembler contradictoire.

Étant donné que tout capteur supplémentaire doit être évité, cette approche utilise le moteur lui-même comme capteur pour accéder aux informations de vitesse ou de position. Les méthodes de mesure brutes permettent d'extraire ces informations à partir de grandeurs directement mesurables aux bornes du moteur (tension, courant). La réalisation de moteur auto-asservi deviendra possible. [14]

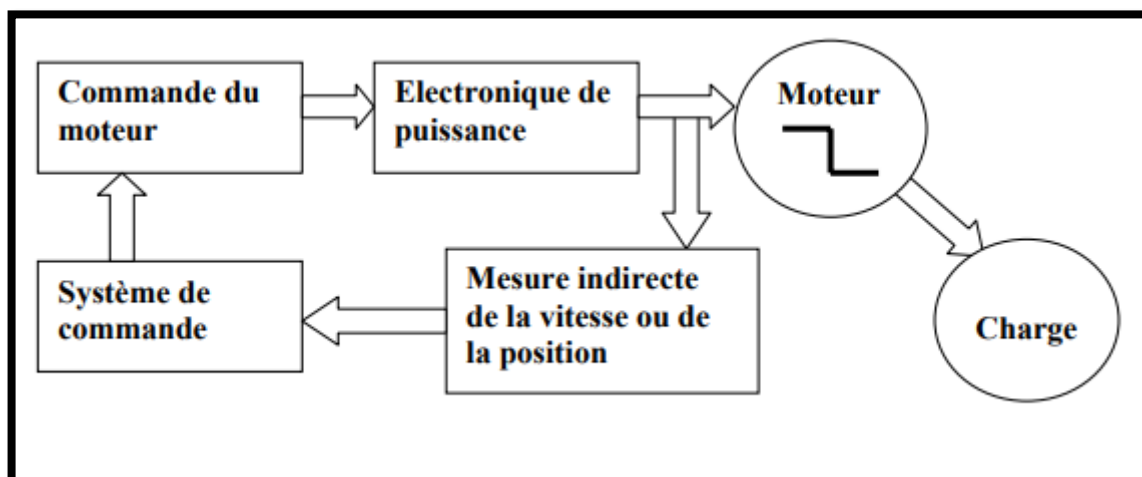


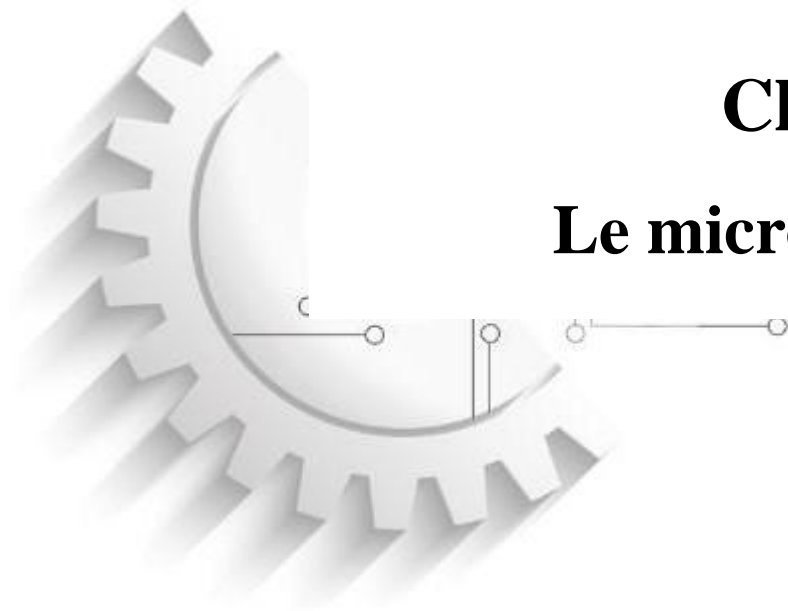
Figure 11 : Commande par auto-asservissement

7. Conclusion

Dans ce chapitre, nous avons étudié les moteurs pas-à-pas, leurs électroniques de puissance et leurs commandes. Et nous avons présenté les trois grandes familles du moteur pas à pas, et les caractéristiques de chacun. Ces moteurs sont simples à réaliser et peuvent présenter une faible inertie, ce qui nous permet d'avoir une accélération importante. Les moteurs à réluctance variable sont beaucoup moins utilisés actuellement, car ils nécessitent un usinage très précis ; par conséquent leur prix est élevé pour un nombre de pas par tour important. Ces moteurs se sont développés dû à l'apparition des microprocesseurs. Le microprocesseur fera l'objet du prochain chapitre.

Chapitre II :

Le microprocesseur 8086



1. Introduction

Le microprocesseur, noté aussi C.P.U. (Central Processing Unit) est un circuit intégré complexe appartenant à la famille des VLSI (Very Large Scale Intégration) capable d'effectuer séquentiellement et automatiquement des suites d'opérations élémentaires.

Le processeur 8086 d'Intel est à la base des processeurs Pentium actuels. Les processeurs successifs (de PC) se sont en effet construits petit à petit en ajoutant à chaque processeurs des instructions et des fonctionnalités supplémentaires, mais en conservant à chaque fois les spécificités du processeur précédent. C'est cette façon d'adapter les processeurs à chaque étape qui permet qu'un ancien programme écrit pour un 8086 fonctionne toujours sur un nouvel ordinateur équipé d'un Pentium IV.

2. Structure interne et externe

2.1. Structure interne

Dans le microprocesseur 8086 ; il existe deux unités internes distinctes :

- L'unité d'interface de bus **BIU**
- L'unité d'exécution **UE**

A. BIU : elle a le rôle de récupérer et stocker les informations à traiter, et d'établir les transmissions avec les bus du système. Pour cela l'UIB divise la mémoire en quatre segments de 64 Ko, chaque segment étant spécifié par l'un des registres de segment CS, DS, SS et ES.

B. UE : récupère les instructions de la file d'attente et utilise BIU pour récupérer ses opérandes de la mémoire et y stocker leurs résultats. Par conséquent, tous les problèmes d'adressage sont traités par l'BIU. Les deux unités fonctionnent simultanément, d'où une accélération du processus d'exécution d'un programme (fonctionnement selon le principe du pipeline). [15]

La Structure interne du le microprocesseur 8086 peut être représenté par le diagramme suivant

:

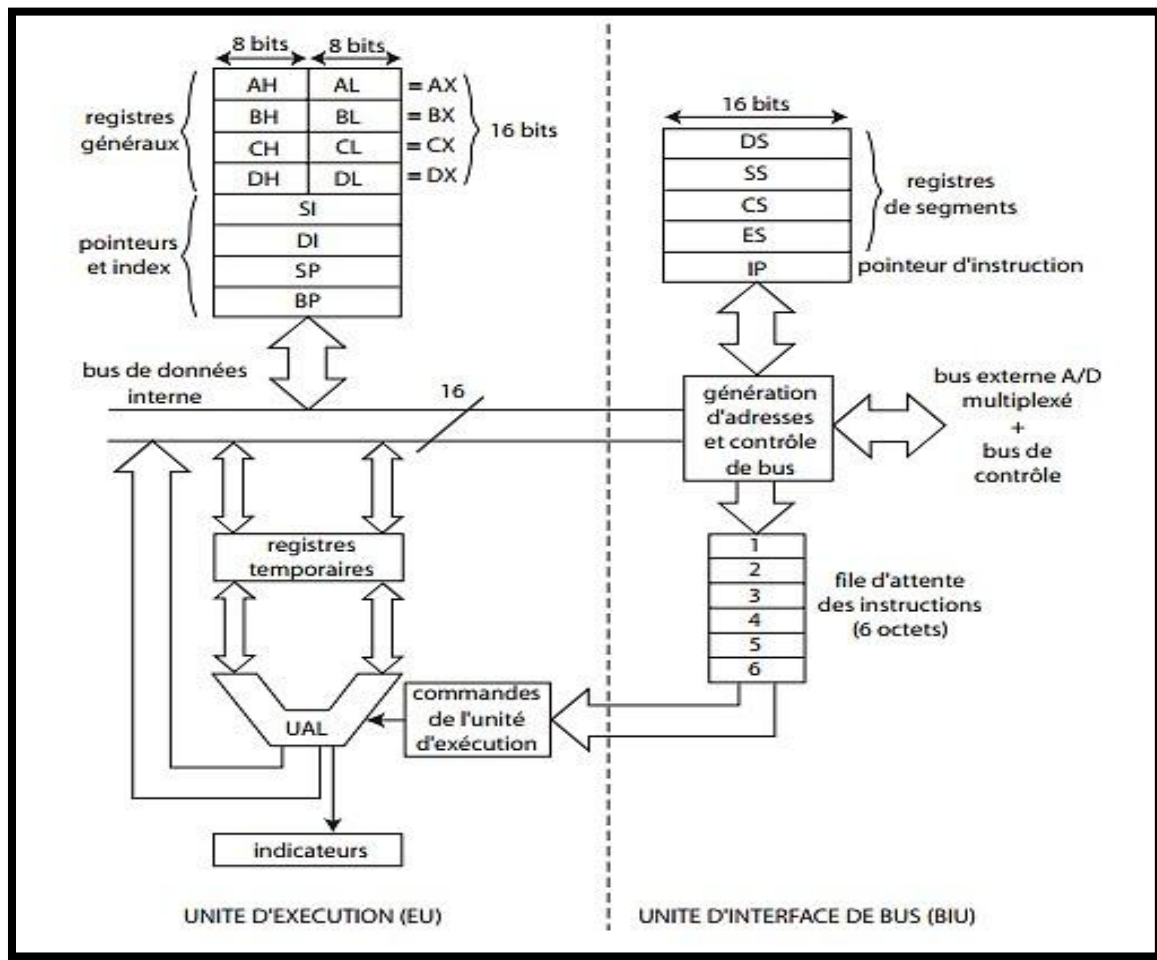


Figure 12 : Organisation interne du 8086

C. Les Registres du microprocesseur 8086

Le microprocesseur 8086 contient 14 registres répartis en 4 groupes :

Registres généraux : 4 registres sur 16 bits.

AX = (AH,AL) ;

BX = (BH,BL) ;

CX = (CH,CL) ;

DX = (DH,DL)

Ils peuvent être également considérés comme 8 registres sur bits. Ils servent à contenir temporairement des données. Ce sont des registres généraux mais ils peuvent être utilisés pour des opérations particulières. [16]

Registres de pointeurs et d'index : 4 registres sur 16 bits.

Pointeurs :

SP : Stack Pointer, pointeur de pile (la pile est une zone de sauvegarde de données en cours d'exécution d'un programme) ;

BP : Base Pointer, pointeur de base, utilisé pour adresser des données sur la pile.

Index :

SI : Source Index ;

DI : Destination Index.

Ils sont utilisés pour les transferts de chaînes d'octets entre deux zones de mémoire.

Les pointeurs et les index contiennent des adresses de cases mémoire. [15]

Registres de segments : 4 registres sur 16 bits.

CS : spécifie le segment contenant le code du programme, alors recherchez les instructions dans ce segment.

DS : spécifie le segment contenant l'opérande.

SS : spécifie le segment associé à la pile dont le sommet est pointé par le registre SP. La pile est utilisée pour les adresses de retour et les paramètres de sous-programme et pour sauvegarder et restaurer le contexte pendant les interruptions.

ES : signifie segment de données supplémentaire.

Ces 4 parties peuvent se chevaucher et même se confondre. Les opérandes et instructions sont référencés en se déplaçant dans l'un de ces 4 segments. L'adresse réelle en mémoire est obtenue en ajoutant le contenu du registre de segments multiplié par 16 à ce déplacement. Cette astuce permet d'accéder à 1M octets (2²⁰) de mémoire. La mémoire est accessible octet par octet ou sous forme de mots de 16 bits. Cependant, l'accès à un mot de 16 bits se fait toujours en 2 transferts sur les 8088 et 80188, et également sur les 8086 et 80186 lorsque l'adresse du mot est impaire. Par conséquent, sur ces deux derniers microprocesseurs, il est utile de faire attention à "l'alignement" des adresses paires pour les opérandes 16 bits si l'on ne veut pas voir une dégradation significative des performances. [15]

Pointeur d'instruction et indicateurs (flags) : 2 registres sur 16 bits. Pointeur d'instruction : **IP**, contient l'adresse de la prochaine instruction à exécuter.

Flags:

				O	D	I	T	S	Z		A		P		C
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CF : indicateur de retenue (carry) ;

PF : indicateur de parité ;

AF : indicateur de retenue auxiliaire ;

ZF : indicateur de zéro ;

SF : indicateur de signe ;

TF : indicateur d'exécution pas à pas (trap) ;

IF : indicateur d'autorisation d'interruption ;

DF : indicateur de d'incrément/décément ;

OF : indicateur de dépassement (over flow) [16]

2.2. Structure externe

Le microprocesseur Intel 8086 est un microprocesseur 16 bits, apparu en 1978. C'est le premier microprocesseur de la famille Intel 80x86. Il se présente sous la forme d'un boîtier DIP (Dual In-line Package) à 40 broches : [17]

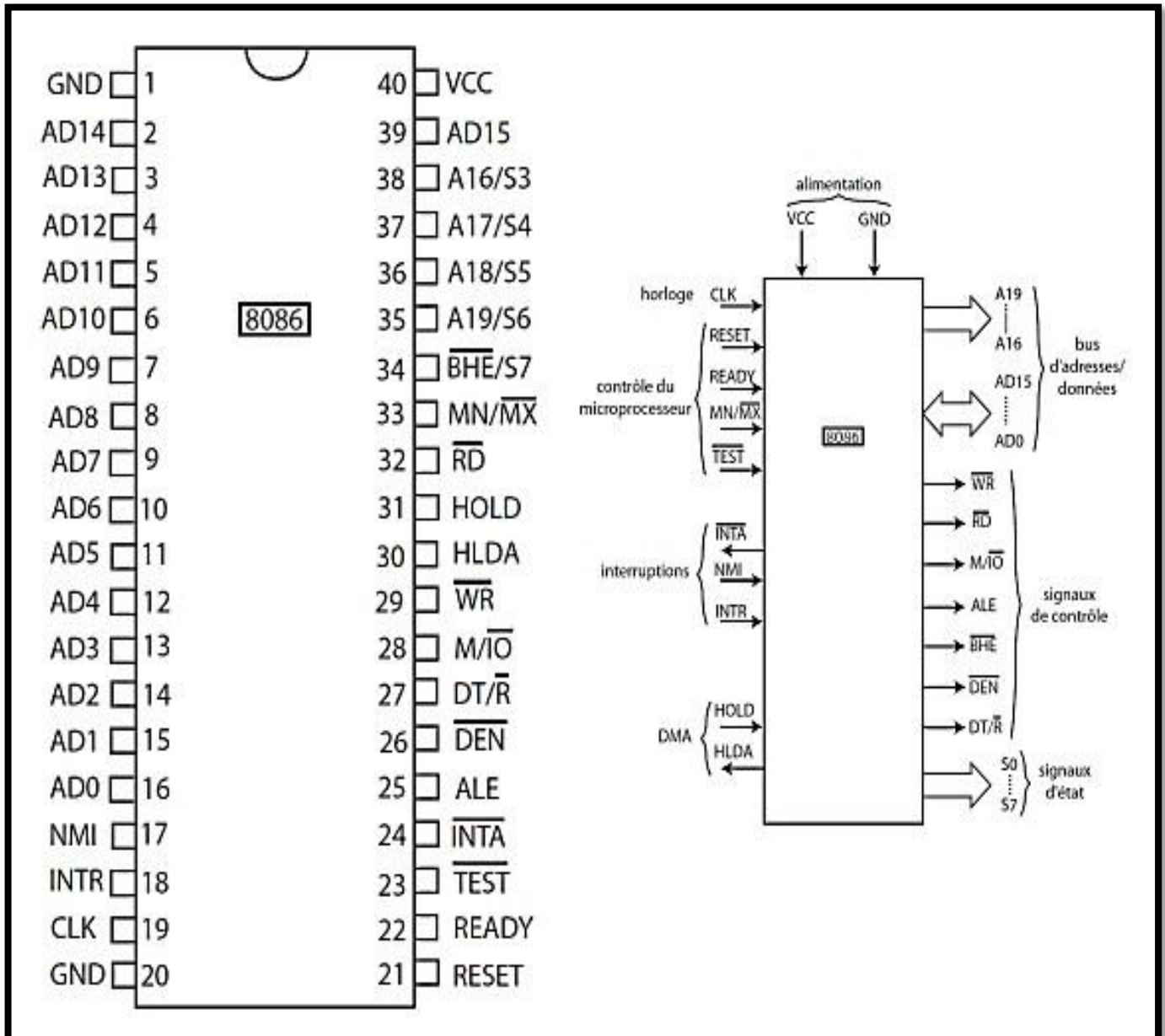


Figure 13 : Brochage et Schéma fonctionnel du 8086

Brochage et Fonctions des Pins du 8086 :

CLK : entrée du signal d'horloge qui cadence le fonctionnement du microprocesseur. Ce signal provient d'un générateur d'horloge : le 8284.

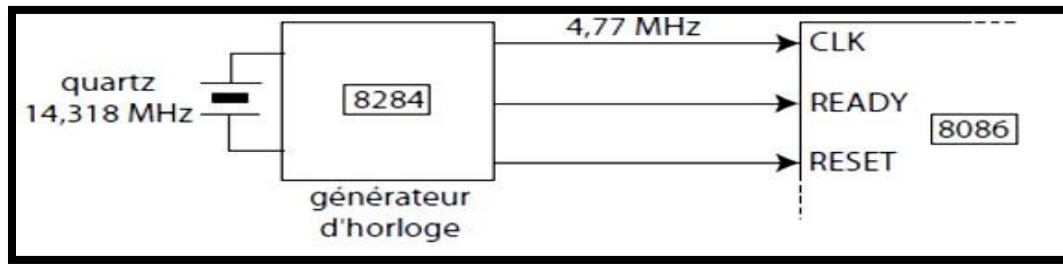


Figure 14 : branchement d'un générateur d'horloge

AD0 à AD15 : Ces lignes représentent 16 bits de bus d'adresse multiplexés avec 16 bits de lignes de données.

Pendant T1, elles représentent des lignes d'adresse A15-A0.

Pendant T2, T3, T4, elles représentent des lignes de données D0-D15.

A16 $\bar{}$ à A19 $\bar{}$: Ces lignes adresse sont multiplexées avec les lignes d'état. Pendant T1, elles représentent des lignes d'adresse A19-A16.

Pendant T2, T3, T4, elles représentent des signaux d'état S6-S3.

ALE (Address Latch Enable): Ce signal est une impulsion active pendant T1, elle indique que l'information qui circule dans bus A/D est une adresse.

Elle est fournie par le Cpu pour verrouiller les lignes d'adresse au cours des cycles T2, T3, T4.

$\bar{}$ Read, signal de lecture d'une donnée.

$\bar{}$ Write, signal d'écriture d'une donnée. [17]

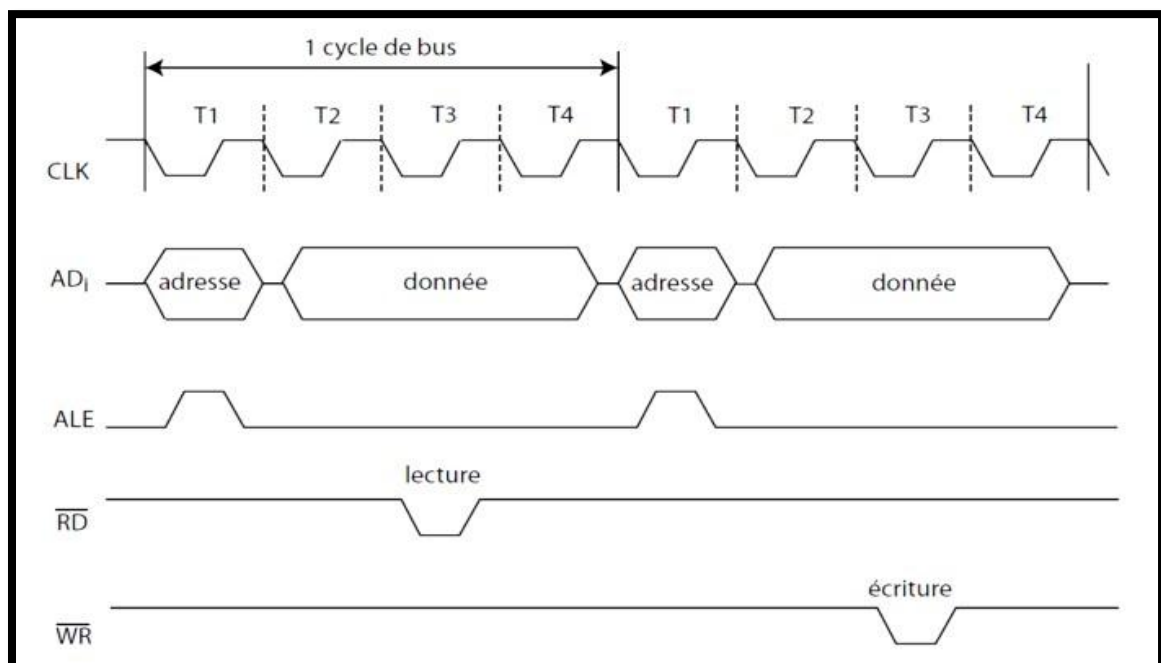


Figure 15 : Chronogramme de séparation de bus A/D

DEN : Data Enable, indique que des données sont en train de circuler sur le bus A/D (équivalente d'ALE pour les données).

DT/ \overline{R} Data Transmit/Receive, indique le sens de transfert des données :

- DT/ \overline{R} = 1 : données émises par le microprocesseur (écriture) ;
- DT/ \overline{R} = 0 : données reçues par le microprocesseur (lecture).

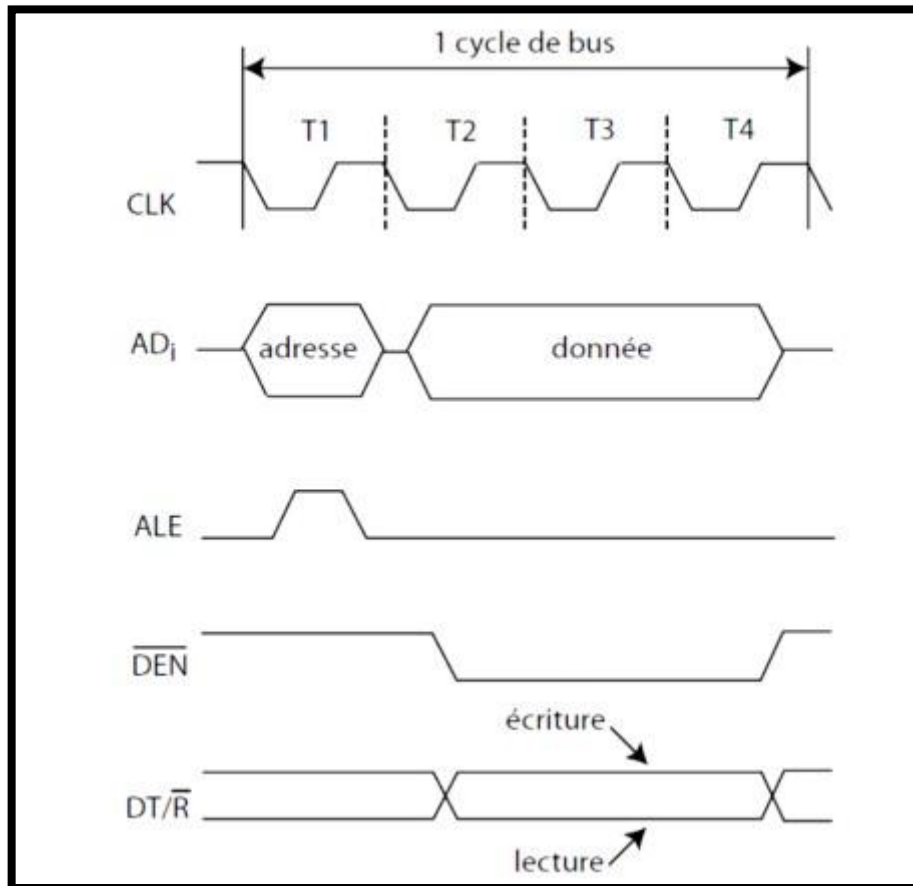


Figure 16 : Chronogramme de sens de transfert de données sur le bus de données

M/ \overline{I} Memory/Input-Output, indique si le CPU adresse la mémoire ($M/\overline{I} = 1$) ou les entrées/sorties ($M/\overline{I} = 0$).

S0 à S7 : signaux d'état indiquant le type d'opération en cours sur le bus.

READY : Entrée de synchronisation avec la mémoire.

TEST : Entrée de mise en attente du microprocesseur d'un évènement extérieur.

RESET : Entrée de remise à zéro du microprocesseur.

MN/ \overline{M} indique dans quel mode le processeur doit fonctionner

HIGH : minimum mode.

LOW : maximum mode. [17]

NMI et INTR : Entrées de demande d'interruption.

INTR : interruption normale,

NMI (Non Maskable Interrupt) : interruption prioritaire.

INTA : Interrupt Acknowledge, indique que le microprocesseur accepte l'interruption.

HOLD et HLDA : Signaux de demande d'accord d'accès direct à la mémoire (DMA).

BHE : Bus High Enable, signal de lecture de l'octet de poids fort du bus de données. Il est utilisé avec A0 pour sélectionner le mot entier, octet fort, octet faible ou aucun. [17]

Tableau 02 : signal de lecture de l'octet de poids fort du bus de données

\overline{BHE}	A0	Sélection des données
0	0	les deux octets (mot complet) (D15-D0)
0	1	octet fort (adresse impaire) (D15-D8)
1	0	octet faible (adresse paire) (D7-D0)
1	1	aucun octet

Connexions de base

GND: se connecter à 0V.

VCC: se connecter à 5V.

MN / $\overline{\square}$ se connecter à 5V (mode minimum).

NMI et INTR: se connecter à 0V (pas de support pour les interruptions).

CLK: se connecter à la sortie CLK du générateur d'horloge.

HOLD: se connecter à 0V (pas d'accès direct à la mémoire).

TEST : se connecter à 0V (pas d'attente pour les coprocesseurs).

READY: se connecter à 5V (pas de cycles d'attente pour les périphériques lents).

RESET: se connecter à zéro de la sortie du générateur d'horloge. [16]

3. La programmation en assembleur du microprocesseur 8086

Chaque microprocesseur reconnaît un jeu d'instructions connu sous le nom de jeu d'instructions fixe du fabricant. Pour les microprocesseurs classiques, le nombre d'instructions reconnaissables varie entre 75 et 150 (microprocesseur CISC : Complex Instruction Set Computer). Il existe également des microprocesseurs à très petit nombre d'instructions (microprocesseur RISC : Reduced Instruction Set Computer) : 10 à 30 instructions, qui réduisent le temps d'exécution du programme. Une instruction est définie par son opcode, qui est une valeur binaire difficile à manipuler pour les humains. Par conséquent, nous utilisons une notation symbolique pour représenter les instructions : les mnémoniques. Un programme composé de mnémoniques est appelé un programme en assembleur. [19]

3.1. Structure d'un programme assembleur

Structure de l'assembleur Comme tout programme, un programme écrit en assembleur (programme source) est constitué de définitions, de données et d'instructions, le tout écrit sur une seule ligne de texte. [16]

3.1.1. Les directives :

Les directives sont des informations fournies par le programmeur au compilateur. Il ne se traduit pas en instructions en langage machine. Les directives sont des instructions qui instruisent l'assembleur. Par exemple, les instructions sont utilisées pour créer de l'espace mémoire pour des variables, définir des constantes, etc. [16]

Pour déclarer une directive, utilisez la syntaxe suivante :

{Nom} directive {opérande} {commentaire}.

- Les champs d'opérandes dépendent de l'instruction
- Le champ Notes à la même fonctionnalité (voir paragraphe suivant).
- Le champ nom indique le nom de la variable : c'est un champ optionnel (selon la directive).

3.1.2. Les instructions:

Un programme en assembleur a une forme bien particulière. Chaque ligne d'instruction est composée de quatre champs. De gauche à droite, on a :

- le champ étiquette, qui peut être vide

- le champ mnémonique (le nom de l'instruction)
- le champ opérande (les arguments de l'instruction), qui peut être vide
- le champ commentaire, qui peut être vide.

Les instructions peuvent être classées en groupes :

A. Instructions de transfert

Ils sont utilisés pour déplacer les données de la source vers la cible :

- Registre vers mémoire;
- Registre vers registre;
- Mémoire vers registre. [19]

B. Les instructions arithmétiques

Les instructions arithmétiques de base sont l'addition, la soustraction, la multiplication et la division, y compris les variations. Plusieurs modes d'adressage sont possibles. [15]

- Addition: ADD operande1,operande 2
- Soustraction: SUB operande1,operande 2
- Multiplication : MUL operande, ou operande est un registre ou une case memoire.
- Division : DIV operande, ou operande est un registre ou une case memoire.

C. Les instructions logiques

Ce sont des instructions qui manipulent les données au niveau du bit. Les opérations logiques de base sont :

- ET;
- OU;
- OU exclusif;
- Complément à 1;
- Complément à 2 ;
- Décalages et rotations [19].

D. Les instructions de branchement

Les instructions de branchement (ou de saut) permettent de modifier l'ordre d'exécution des instructions du programme selon certaines conditions. Il existe 3 types de sauts :

- Saut inconditionnel;

- Sauts conditionnels;
- Appel de sous-programmes. [15]

3.1.3 Les modes d'adressage du 8086 :

Le "mode d'adressage" fait référence à la manière dont les opérandes sont structurés en instructions. Le microprocesseur 8086 dispose de 7 modes d'adressage :

- Adressage registre** : ce mode utilise les registres internes de μp , dans ce mode il faut respecter la taille de différents registres. [18]
- Adressage immédiat** : C'est un mode qui utilise des registres internes ou des unités de stockage pour manipuler les valeurs limites. [18]
- Adressage direct** : Comme son nom l'indique, c'est la réalisation d'opérations entre les registres internes μp et la mémoire. [18]
- Adressage indirect par registre** : c'est le mode d'accès entre la mémoire et les registres internes de μp par un intermédiaire, ces sont les registres pointeurs : (SI, DI, BX). [18]
- Adressage basé** : tout simplement, il utilise le registre de base comme pointeur entre la mémoire et les registres internes de μp ; les registres de base sont (BX, BP). [18]
- Adressage indexé** : de la même manière que base, mais il utilise le registre d'index comme pointeur ; le registre d'index est (SI, DI). [18]
- Adressage basé indexé** : Comme son nom l'indique, il s'agit d'une combinaison de modes d'adressage. [18]
- Adressage vectorisé** : Ce mode d'adressage n'est utilisé que pour gérer les interruptions.
- Adressage des entrées sorties (I/O)** : dans ce mode, le bus d'adresse devient de taille 16 bits → l'espace adressable par I/O est 64 KO. Il utilise comme instructions d'accès (IN, OUT). [18]

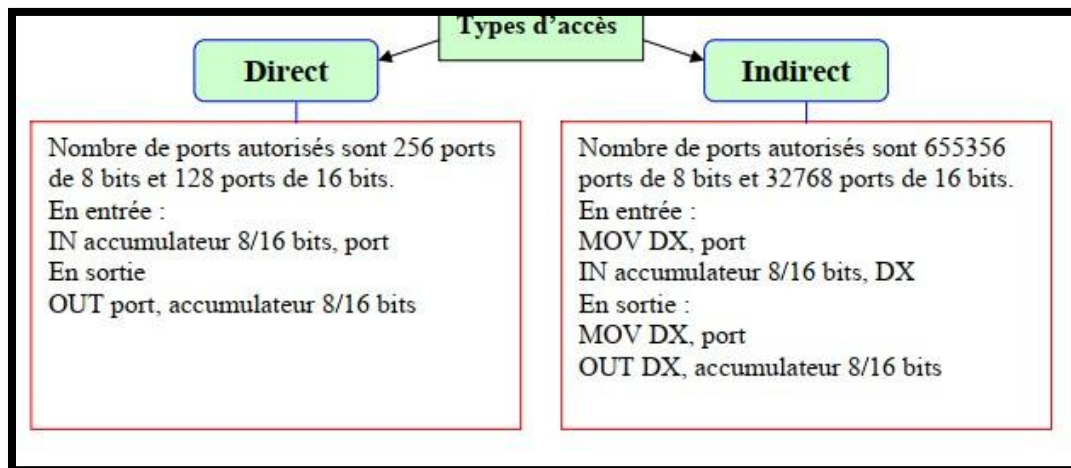


Figure 17 : Adressage des entrées sorties

j) **Adressage relatif** : ce type d'adressage est utilisé par les instructions de saut.

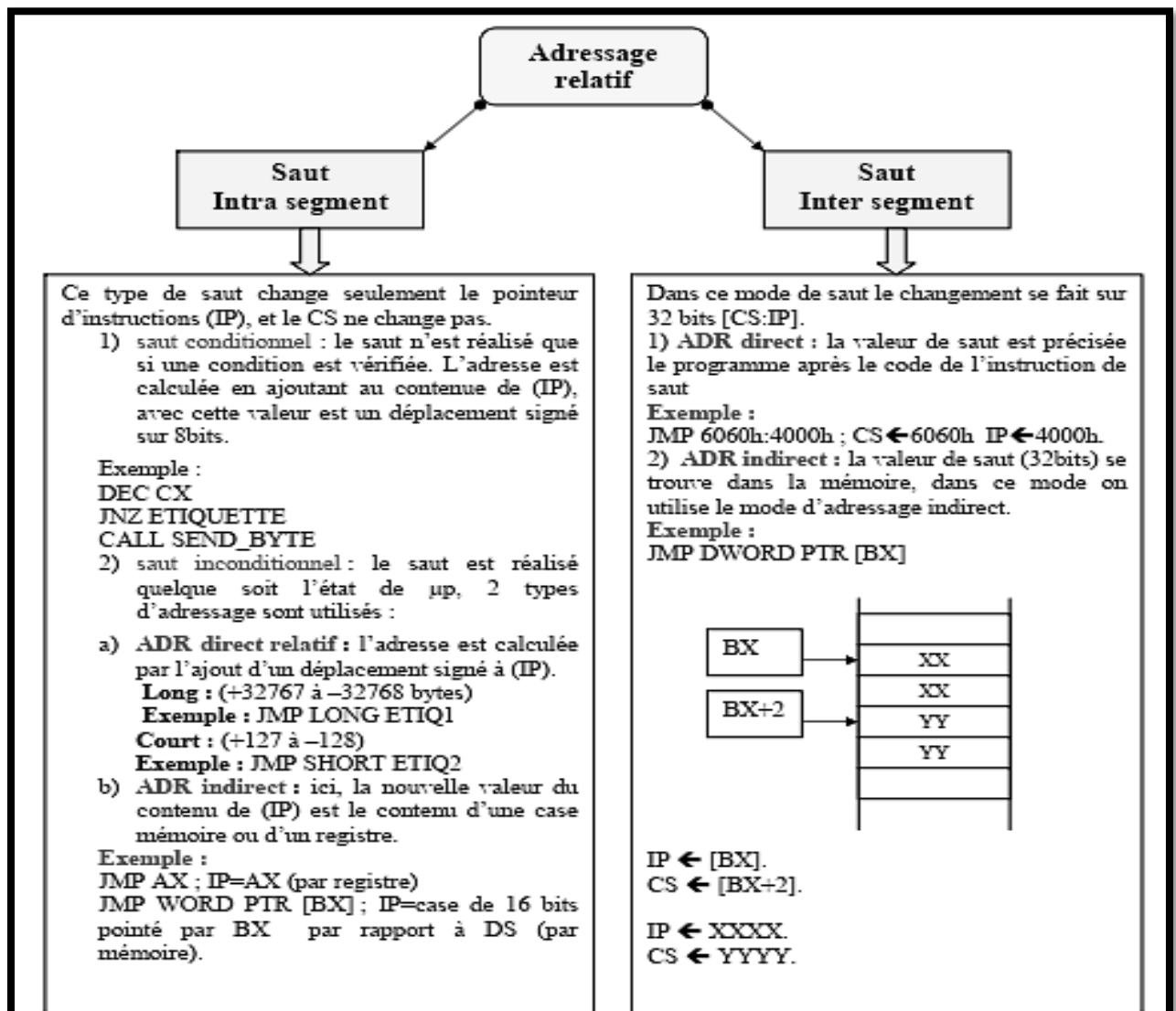


Figure 18 : Adressage relatif

4. Applications des microprocesseurs

Un microprocesseur facilite la vie quotidienne et à de vastes applications dans tous les domaines [20]. Ils ont de nombreuses applications telles que :

- ✓ Les systèmes basés sur microprocesseurs sont adaptés aux applications dédiées dans les industries
- ✓ Comme le contrôle des processus, le contrôle des machines et des équipements, l'instrumentation et ainsi de suite
- ✓ Utilisant des multiplexeurs et des multiplexeurs, il peut effectuer la mesure des tâches et le contrôle automatique
- ✓ Les systèmes à microprocesseur remplacent la logique électromécanique
- ✓ La mesure, l'affichage et le contrôle du courant,
- ✓ Tension, KW, KWH, KVA, facteur de puissance, angle de phase, fréquence, température, contrainte, contrainte,
- ✓ Pression, force, déviation, vibration, communication
- ✓ Contrôle de la circulation, contrôle des ascenseurs, contrôle des outils industriels, robotique, et ainsi de suite
- ✓ Le thermostat programmable permet le contrôle de la température à domicile
- ✓ Cafetières haut de gamme, machines à laver et horloges radio contiennent la technologie des microprocesseurs
- ✓ Certains autres articles de maison qui contiennent des microprocesseurs sont : micro-ondes, grille-pain, téléviseurs, magnétoscopes,
- ✓ Lecteurs DVD, fours, cuisinières, lave-linge, systèmes stéréo, ordinateurs domestiques, réveils,
- ✓ Appareils de jeu portatifs, thermostats, systèmes de jeux vidéo, machines à pain, lave-vaisselle,
- ✓ Systèmes d'éclairage domestique et même certains réfrigérateurs avec contrôle de température numérique [20]

En médecine et instrumentation

De nombreux dispositifs médicaux, comme une pompe à insuline, sont généralement contrôlés par un microprocesseur. Les microprocesseurs remplissent diverses fonctions et comme le traitement des données des bio-capteurs, le stockage des mesures et l'analyse des résultats ; il est également très utile dans le domaine de l'instrumentation. Générateurs de fonction,

compteurs de fréquence, synthétiseurs de fréquence, analyses de spectre et de nombreux autres instruments sont disponibles, lorsque des microprocesseurs sont utilisés comme contrôleur.

Dans les communications, l'industrie du téléphone est la plus importante

On cette industrie, les microprocesseurs sont utilisés dans les postes téléphoniques numériques, les centraux téléphoniques et les modems. L'utilisation du microprocesseur dans la communication par satellite, la télévision, a rendu possible la téléconférence, le système de réservation ferroviaire et de réservation des compagnies aériennes utilise également la technologie des microprocesseurs. [20]

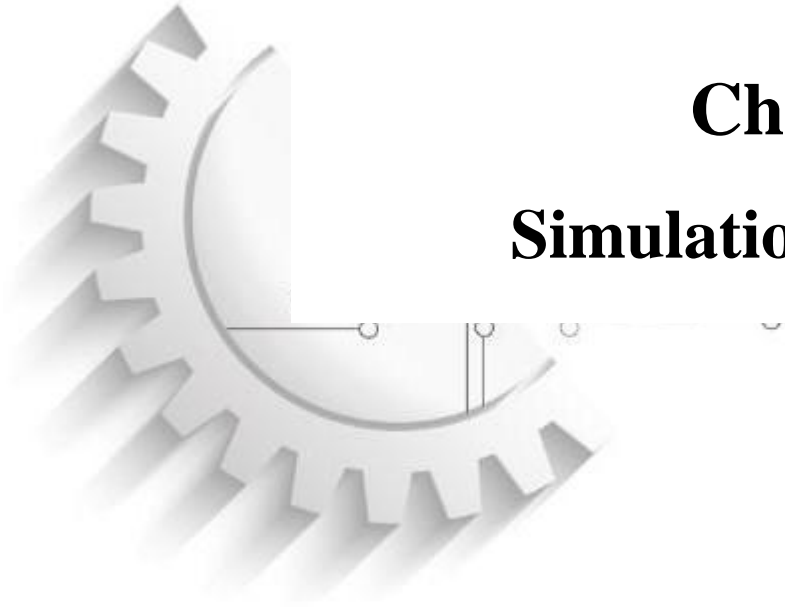
5. Conclusion

Dans ce deuxième chapitre on est parlé sur le microprocesseur 8086 ses structures interne et externe, leur programmation en langage assembleur et ses différents domaines d'application.

En conclusion on peut dire que le microprocesseur 8086 se compose essentiellement de deux unités : la BIU qui fournit l'interface physique entre le microprocesseur et le monde extérieur et l'EU qui comporte essentiellement l'UAL de 16 bits qui manipule les registres généraux de 16 bits aussi. Il est à la base des processeurs Pentium actuels. Les processeurs successifs (de PC) se sont en effet construits petit à petit en ajoutant à chaque processeur des instructions et des fonctionnalités supplémentaires, mais en conservant à chaque fois les spécificités du processeur précédent.

Chapitre III :

Simulation Sur PROTEUS



1. Introduction

Dans ce chapitre nous allons simuler un système de commande d'un moteur pas à pas en utilisant un microprocesseur 8086. Pour objectif d'allumer éteindre un moteur pas à pas (unipolaire ou bipolaire), de changer son sens de rotation et de varier la vitesse. La simulation est réalisée à l'aide du logiciel ISIS PROTEUS.

2. Description de PROTEUS

PROTEUS est une compilation de programmes de conception électronique et de simulation, développée par Labcenter Electronics qui comprend les deux programmes principaux : ISIS et ARES.

ISIS (Intelligent Schematic Input System), permet de concevoir le schéma électrique du circuit que nous souhaitons réaliser avec des composants très variés, en plus de simuler les résultats des composants et des variables ajoutées par l'utilisateur, ceux-ci peuvent aller de la valeur d'une résistance à la programmation d'un microcontrôleur.

ARES (logiciel avancé de routage et d'édition) est l'outil de routage, de localisation et d'édition des composants. Il est utilisé pour la fabrication des cartes de circuits imprimés, permettant d'éditer généralement les couches de surface (Top Copper) et de soudure (Bottom Copper), ainsi que différentes caractéristiques des pistes et les "pads", il nous permet également d'obtenir les fichiers gerber de notre conception.

Le lancement de PROTEUS donne un environnement classique de type Windows, constitué d'une fenêtre principale, et d'un ensemble de barres d'outils. Outre le menu classique permettant la gestion des fichiers, de l'affichage, et des options des projets, la fenêtre principale comprend une Zone de travail destinée au développement des circuits à simuler et à tester.

Une Bibliothèque d'objets affiche la liste des objets (circuits électriques, électroniques,) utilisés dans l'application en cours.

Les différentes Touches magnétoscope constituées des raccourcis permettant le lancement de la simulation, ainsi que la mise en pause, l'exécution pas à pas, et l'arrêt de la simulation.

3. Description des différents composants utilisés

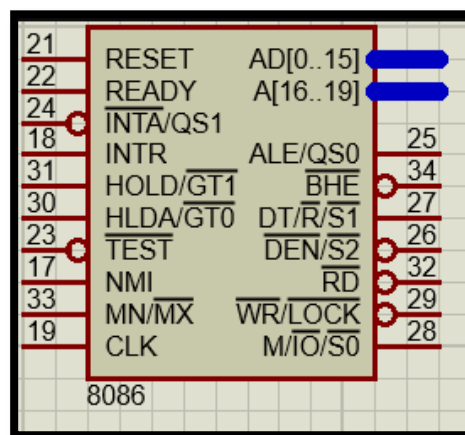
Dans notre simulation nous avons utilisé les composants suivants :

- Un microprocesseur *0808*
- *47HC373* Latch
- Périphérique I/O *0588A*
- Moteur pas à pas
- Un convertisseur analogique numérique *ADC0804*
- *L293D*
- Des Résistances

- LED
- Switch
- Un potentiomètre
- Des Batteries

3. 1. Microprocesseur 8086

Le microprocesseur est chargé de la partie traitement des informations et envoyer des ordres vers les autres composants. Pour notre application nous avons utilisé le microprocesseur 8086 d'Intel, qui a 20 bits pour le bus d'adresse et 16 bits pour le bus de données. Il est utilisé ici pour contrôler tous les appareils, y compris le moteur pas à pas, la



LED, l'affichage. [21]

Figure 19 : Microprocesseur 8086

3. 2. Interface périphérique programmable

Le 8255A est un périphérique d'E/S programmable à usage général conçu pour transférer les données des E/S afin d'interrompre les E/S dans certaines conditions selon les besoins. Il peut être utilisé avec presque tous les microprocesseurs.

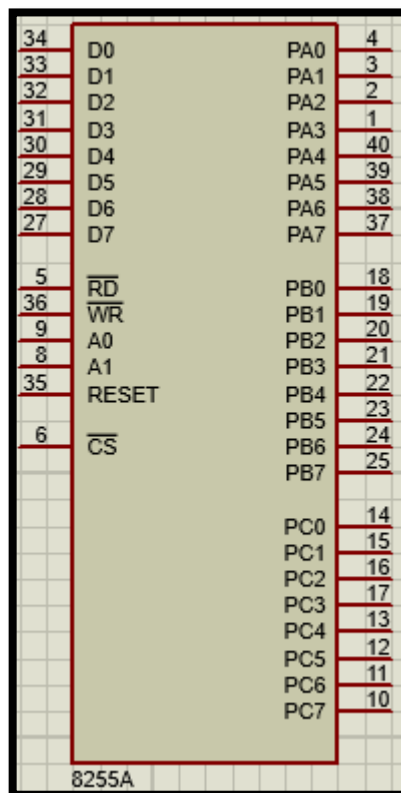


Figure 20 : Le 8255A

Il se compose de trois ports d'E/S bidirectionnels 8 bits (lignes 24I/O) qui peuvent être configurés selon les besoins. [21]

3.2.1. Ports de 8255A

8255A possède trois ports, à savoir PORT A, PORT B et PORT C. Le port A contient un loquet/tampon de sortie 8 bits et un tampon d'entrée 8 bits. Le port B est similaire au port A. Le port C peut être divisé en deux parties, à savoir le PORT C inférieur (PC0-PC3) et le PORT C supérieur (PC7-PC4) par le mot de contrôle.

Ces trois ports sont divisés en deux groupes, c'est-à-dire que le groupe A comprend le PORT A et le PORT C supérieur. Le groupe B comprend le PORT B et le PORT C inférieur. Ces deux groupes peuvent être programmés dans trois modes différents, c'est-à-dire que le premier mode est nommé mode 0, le deuxième mode est nommé mode 1 et le troisième mode est nommé mode 2. [21]

3.2.2. Modes de fonctionnement

La norme 8255A dispose de trois modes de fonctionnement différents –

- ⇒ **Mode 0** – Dans ce mode, les ports A et B sont utilisés comme deux ports 8 bits et le port C comme deux ports 4 bits. Chaque port peut être programmé en mode d'entrée ou en mode de sortie où les sorties sont verrouillées et les entrées ne sont pas verrouillées. Les ports n'ont pas de capacité d'interruption.
- ⇒ **Mode 1** – Dans ce mode, les ports A et B sont utilisés comme ports d'E/S 8 bits. Ils peuvent être configurés comme ports d'entrée ou de sortie. Chaque port utilise trois lignes du port C comme signaux de négociation de liaison. Les entrées et les sorties sont verrouillées.
- ⇒ **Mode 2** – Dans ce mode, le port A peut être configuré comme port bidirectionnel et le port B en mode 0 ou en mode 1. Le port A utilise cinq signaux du port C comme signaux d'établissement de liaison pour le transfert de données. Les trois signaux restants du port C peuvent être utilisés soit comme simple E/S, soit comme poignée de main pour le port B.

3.2.3. Caractéristiques de 8255A

Les principales caractéristiques de 8255A sont les suivantes :

- Il se compose de 3 ports d'E/S 8 bits, c'est-à-dire PA, PB et PC.
- Le bus d'adresse/de données doit être démultiplexé de l'extérieur.
- Il est compatible TTL.
- Il a amélioré la capacité de conduite DC.

3.2.4. Architecture

La figure suivante montre l'architecture de 8255A –

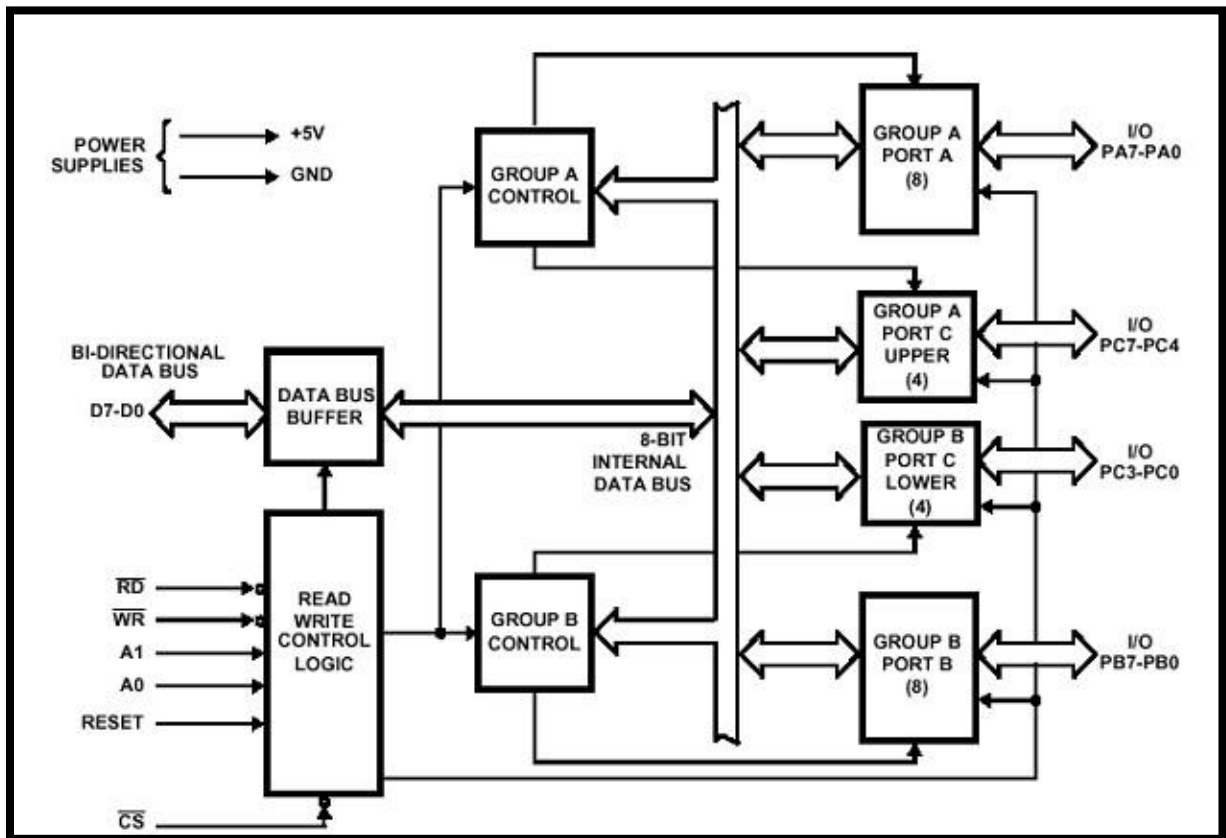


Figure 21 : Jetons d’abord un coup d’œil au diagramme des broches de l’Intel 8255A –

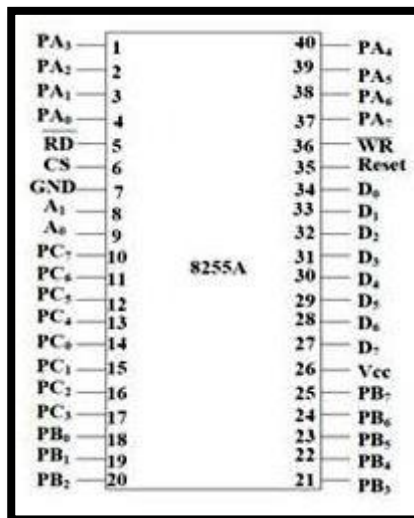


Figure 22 : Discutons maintenant de la description fonctionnelle des broches dans 8255A.

Tampon de bus de données

Il s’agit d’un tampon 8 bits à trois états, qui est utilisé pour interfacer le microprocesseur avec le bus de données système. Les données sont transmises ou reçues par le tampon selon les

instructions du processeur. Les mots de contrôle et les informations d'état sont également transférés à l'aide de ce bus. [21]

Logique de contrôle de lecture/écriture

Ce bloc est responsable du contrôle du transfert interne / externe des données / contrôle / mot de statut. Il accepte l'entrée de l'adresse CPU et des bus de contrôle, et émet à son tour une commande aux deux groupes de contrôle.

CS : Il signifie Chip Select. Un LOW sur cette entrée sélectionne la puce et permet la communication entre le 8255A et le processeur. Il est connecté à l'adresse décodée, et A0 et A1 sont connectés aux lignes d'adresse du microprocesseur.

Tableau 03 : Leur résultat dépend des conditions suivantes

CS	A ₁	A ₀	Result
0	0	0	PORT A
0	0	1	PORT B
0	1	0	PORT C
0	1	1	Control Register
1	X	X	No Selection

WR : Il signifie écrire. Ce signal de commande active l'opération d'écriture. Lorsque ce signal est faible, le microprocesseur écrit dans un port d'E/S ou un registre de contrôle sélectionné.

RÉINITIALISATION : Il s'agit d'un signal élevé actif. Il efface le registre de contrôle et définit tous les ports en mode d'entrée.

RD : Il signifie Read. Ce signal de commande active l'opération de lecture. Lorsque le signal est faible, le microprocesseur lit les données à partir du port d'E/S sélectionné du 8255. A0 et A1 Ces signaux d'entrée fonctionnent avec RD, WR et l'un des signaux de commande. [21]

Tableau 04 : le tableau montrant leurs différents signaux avec leur résultat.

A ₁	A ₀	RD	WR	CS	Result
0	0	0	1	0	Input Operation PORT A → Data Bus
0	1	0	1	0	PORT B → Data Bus
1	0	0	1	0	PORT C → Data Bus
0	0	1	0	0	Output Operation Data Bus → PORT A
0	1	1	0	0	Data Bus → PORT A
1	0	1	0	0	Data Bus → PORT B
1	1	1	0	0	Data Bus → PORT D

3. 3. Latch/74HC373

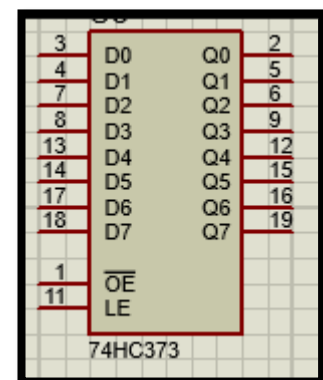
Verrou est utilisé comme mémoire pour obtenir l'adresse du microprocesseur pour lire ou écrire sur les périphériques d'E/S.

OE : Activer la sortie lorsqu'elle est faible.

D0-D7 : entrée de données

Q0-Q7 : sortie de données

LE: verrou activer l'entrée.

**Figure 23 : Latch/74HC373**

Le SN54/74LS373 se compose de huit loquets avec des sorties à 3 états pour bus applications système organisées. Les tongs semblent transparentes pour les données (les données changent de manière asynchrone) lorsque Latch Enable (LE) a la valeur HIGH. Lorsque LE est LOW, les données qui respectent les temps de configuration sont verrouillées. Les données apparaissent dans le bus lorsque l'option Output Enable (OE) est LOW. Lorsque l'OE est HIGH, la sortie du bus est en l'Etat d'impédance élevée. [21]

Le SN54/74LS374 est un bascule Octal de type D haute vitesse et basse consommation doté d'entrées de type D séparées pour chaque bascule et de sorties à 3 états pour les applications

orientées bus. Une horloge tamponnée (CP) et une activation de sortie (OE) sont courantes à toutes les tongs. Le SN54/74LS374 est fabriqué à l'aide de Power Schottky et est compatible avec toutes les familles TTL de Motorola.

- Huit loquets dans un seul emballage.
- Sorties à 3 états pour l'interfaçage des bus.
- Hystérésis sur l'activation du loquet.
- Entrées de type D déclenchées par les bords.
- Horloge positive mise en mémoire tampon.
- Hystérésis sur l'entrée de l'horloge pour améliorer la marge de bruit.
- Les diodes de serrage d'entrée limitent les effets de terminaison à grande vitesse.

LS373

D _n	LE	OE	O _n
H	H	L	H
L	H	L	L
X	L	L	Q ₀
X	X	H	Z*

H = HIGH Voltage Level
 L = LOW Voltage Level
 X = Immaterial
 Z = High Impedance

* Note: Contents of flip-flops unaffected by the state of the Output Enable input (OE).

Tableau 05 : TABLE DE VÉRITÉ

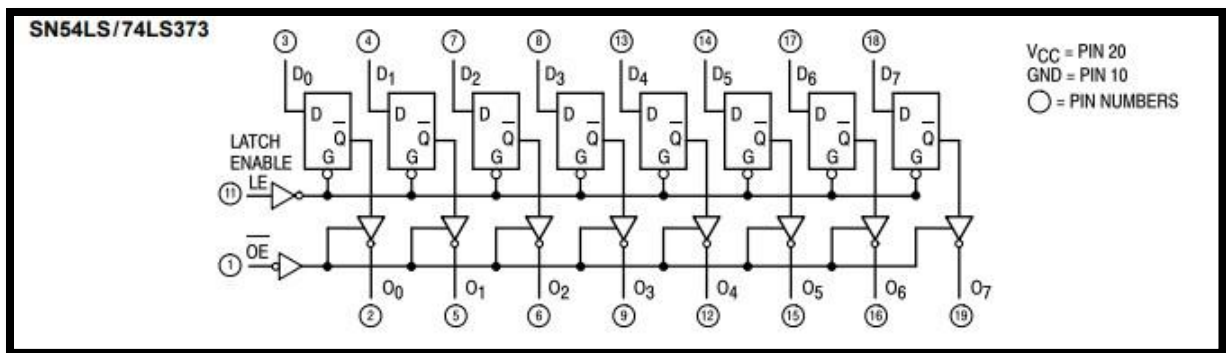


Figure 24 : DIAGRAMMES LOGIQUE

3. 4. ADC

3.4.1. Description

La famille ADC0802 sont des convertisseurs A/N CMOS 8 bits à approximation successive qui utilisent une échelle potentiométrique modifiée. Ces convertisseurs apparaissent au processeur

comme des emplacements de mémoire ou des ports d'E/S, et aucune logique d'interfaçage n'est donc requise. [21]

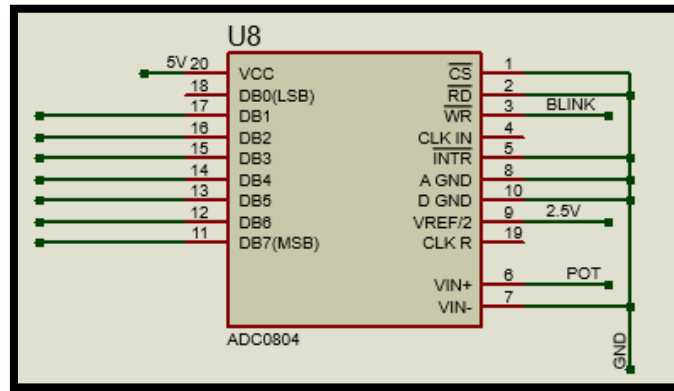


Figure 25 : Verrou activer ADC

3.4.2. Fonctionnalités

- Compatible avec les bus 80C48 et 80C80/85 - Aucune interfaçage logique requise
- Temps de conversion < 100µs
- Interface facile avec la plupart des microprocesseurs
- Fonctionnera en mode « autonome »
- Entrées de tension analogique différentielle
- Fonctionne avec des références de tension de bande interdite
- Entrées et sorties compatibles TTL
- Générateur d'horloge sur puce
- Plage d'entrée de tension analogique de 0V à 5V (alimentation simple + 5V)
- Aucun réglage zéro requis

Tableau 06 : Informations de commande

PART NUMBER	ERROR	EXTERNAL CONDITIONS	TEMP. RANGE (°C)	PACKAGE	PKG. NO
ADC0802LCN	±1/2 LSB	V _{REF/2} = 2.500V _{DC} (No Adjustments)	0 to 70	20 Ld PDIP	E20.3
ADC0802LCD	±3/4 LSB		-40 to 85	20 Ld CERDIP	F20.3
ADC0802LD	±1 LSB		-55 to 125	20 Ld CERDIP	F20.3
ADC0803LCN	±1/2 LSB	V _{REF/2} Adjusted for Correct Full Scale Reading	0 to 70	20 Ld PDIP	E20.3
ADC0803LCD	±3/4 LSB		-40 to 85	20 Ld CERDIP	F20.3
ADC0803LCWM	±1 LSB		-40 to 85	20 Ld SOIC	M20.3
ADC0803LD	±1 LSB		-55 to 125	20 Ld CERDIP	F20.3
ADC0804LCN	±1 LSB	V _{REF/2} = 2.500V _{DC} (No Adjustments)	0 to 70	20 Ld PDIP	E20.3
ADC0804LCD	±1 LSB		-40 to 85	20 Ld CERDIP	F20.3
ADC0804LCWM	±1 LSB		-40 to 85	20 Ld SOIC	M20.3

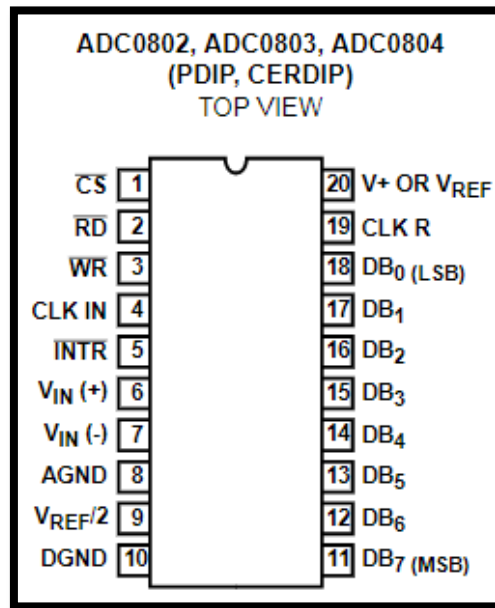


Figure 26 : Brochage

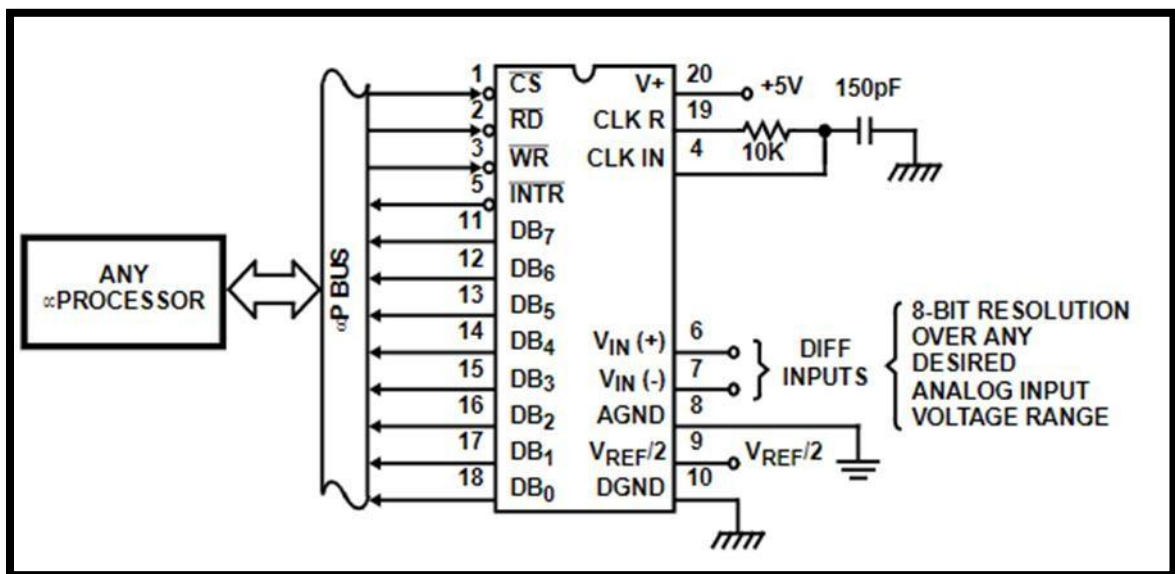


Figure 27 : Schéma typique d'application

Le Convertisseur analogique-numérique est utilisé pour lire la tension analogique du potentiomètre et la convertir en numérique, puis envoyer cette valeur au périphérique d'E/S.

CS : est mise à la terre pour faire fonctionner l'ADC (actif bas).

WR : définissez et réinitialisez ce bit pour convertir les données de l'analogique au numérique.

RD : après la conversion de l'analogique au numérique, nous avons réglé cette broche sur bas pour envoyer les données des registres internes aux broches de sortie (DB0-DB7).

DB0-DB7 : nous obtenons les données numériques qui sont équivalentes à la tension sur potentiomètre et cette valeur sera modifiée pour obtenir différentes valeurs de vitesse du moteur (en faisant varier le délai entre chaque étape). [21]

3. 5. Moteur pas à pas

- Moteur à courant continu qui se déplace par étapes discrètes donnant l'illusion de rotation
- Il divise la rotation complète en nombre de pas, Chaque moteur pas à pas aura un angle de pas fixe
- Ce moteur est entraîné par L293D Motor Driver
- Moteur pas à pas sont essentiellement deux types : unipolaire et bipolaire [21]

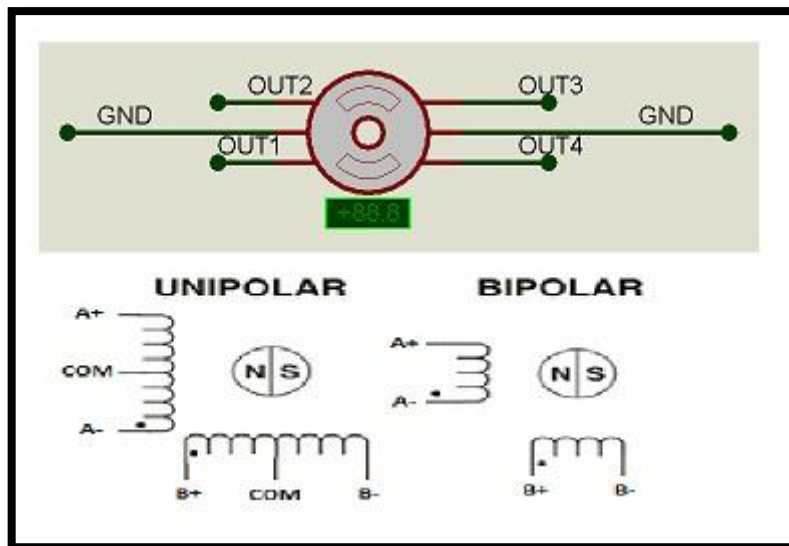


Figure 28 : Moteur pas à pas unipolaire et bipolaire

Unipolaire : A 5 ou 6 fils, cela se produit en attachant chaque 2 bobines d'une extrémité, puis il y a 2 fils communs comme indiqué, unipolaire peut être utilisé comme bipolaire comme nous l'avons fait dans le projet.

Bipolaire : n'a que 4 fils sortant de deux bobines, il n'y a pas de fils communs, il ne peut pas être utilisé comme unipolaire. [21]

3. 6. Pilote L293D

Le pilote L293D est un pilote de moteur à 16 broches utilisé comme interface entre le microprocesseur et le moteur pas à pas, il est utilisé pour entraîner le moteur pas à pas car le microcontrôleur ne peut pas fournir le courant requis par le moteur. Le L293D est conçu pour fournir jusqu'à 600 mA à des tensions de 4,5 V à 36 V Broches L293D :

- EN1,EN2 : deux broches d'activation sont connectées à 12 V.

- VSS et VS : deux broches d'alimentation sont connectées à 12 V.
- GND : est relié au sol.
- IN1 .. IN4 : Quatre broches d'entrée sont connectées au port A dans 8255.
- OUT1 .. OUT4 : Quatre broches de sortie sont connectées au moteur pas à pas.

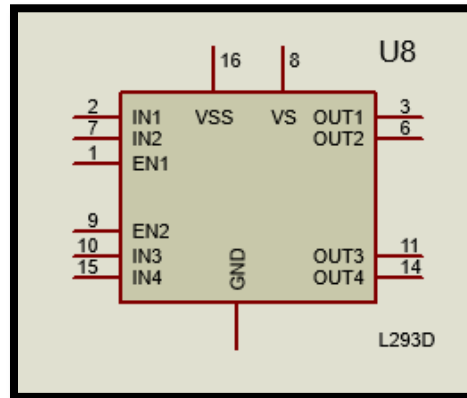


Figure 29 : Pilote L293D

Le dispositif est un circuit d'attaque monolithique intégré à haute tension et à courant élevé à quatre canaux conçus pour accepter les niveaux logiques DTL ou TTL standard et les charges inductives d'entraînement (telles que les solénoïdes de relais, les moteurs DC et pas à pas) et les transistors de puissance de découpage. [21]

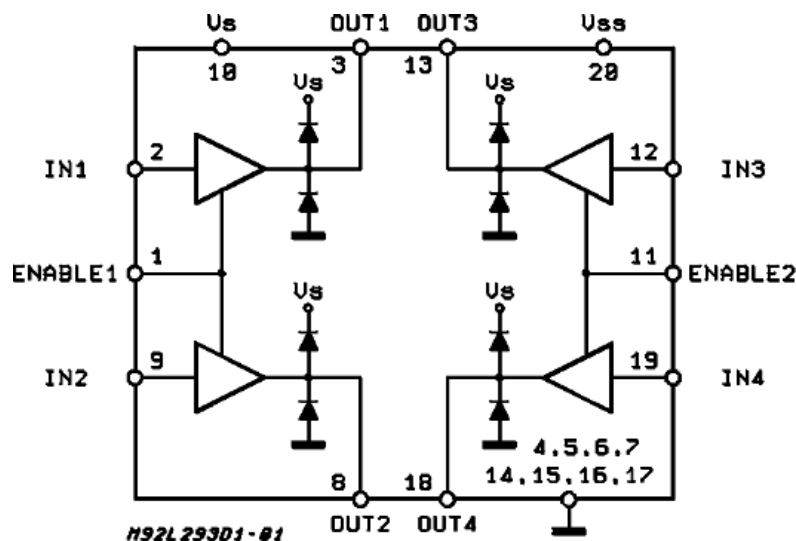


Figure 30 : SCHÉMA

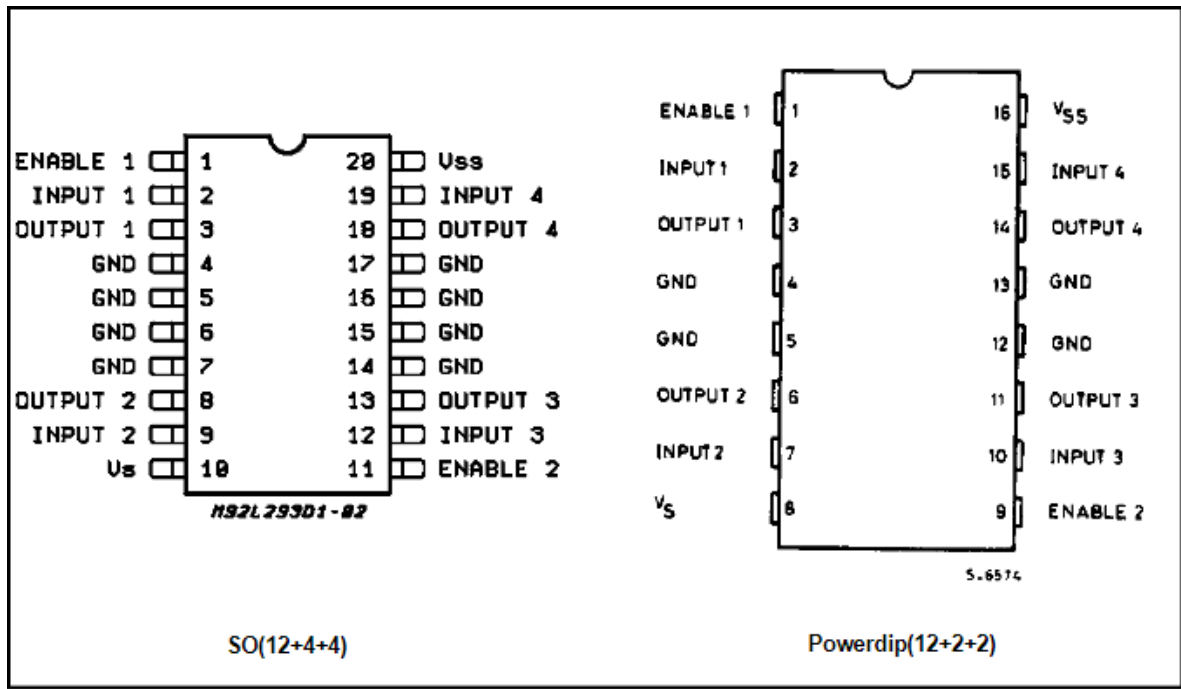


Figure 31: Broche De Connexions

4. Description du schéma de commande

Le schéma représente un montage d'un système de commande d'un moteur pas à pas bipolaire en Utilisant le Microprocesseur 8086 sur PROTEUS. Le montage nous permet de changer le sens de rotation et de varier la vitesse du moteur pas à pas (voir figure 32).

4.1. Variation de la vitesse :

Dans notre simulation, la variation de la vitesse du moteur peut être réalisée de deux manières différentes

- la première méthode consiste à changer le pas du moteur (demi/complet) en utilisant un switch. Cette méthode varie la vitesse du moteur en changeant la matrice de commande du pilot L293D

```

00000110
11111100
11111001
11111011

Matrice du pas complet
    
```

```

00000011
11111101
11111001
11111011
11110011
11110111
11110111

Matrice de demi-pas
    
```

- la seconde méthode consiste à varier la résistance du potentiomètre qui va varier la valeur du retard entre les pas du moteur. On peut avoir 11 valeurs différentes du potentiomètre en utilisant le convertisseur ADC, la plage de la valeur de retard est entre 06FFH et 185CH.
- En utilisant les deux méthodes de variation de la vitesse en même temps, nous pouvons obtenir 22 vitesses différentes du moteur.

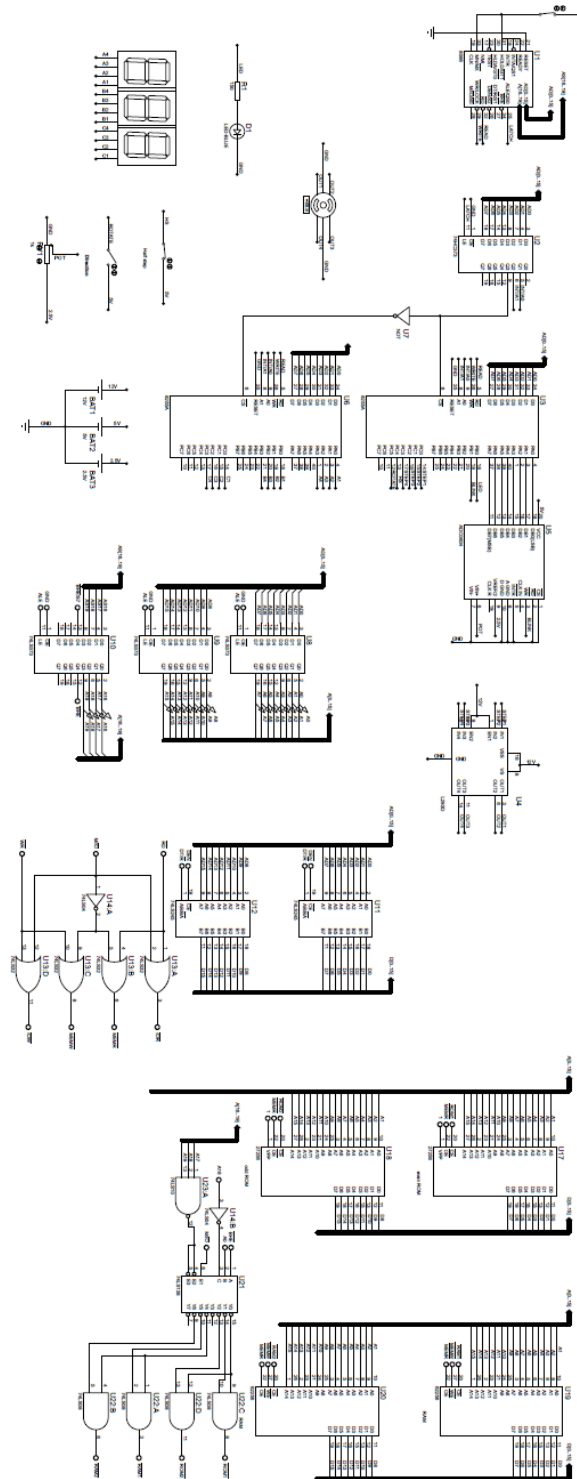


Figure 32 : schéma de commande d'un moteur pas à pas

4.2. L’Affichage :

Dans notre circuit nous avons deux types d’affichages.

Une LED qui vas indiquer le sens de rotation. Cette LED est connecté avec le premier périphérique 8255 dans la broche PB0.

Le deuxième mode d’affichage consiste de trois afficheurs 7 segments connectés avec le deuxième périphérique 8255 dans la broche PA0–PA3, PB0–PB3, PC0–PC3 respectivement. Cette afficheur est charge d’afficher la vitesse de rotation du moteur.

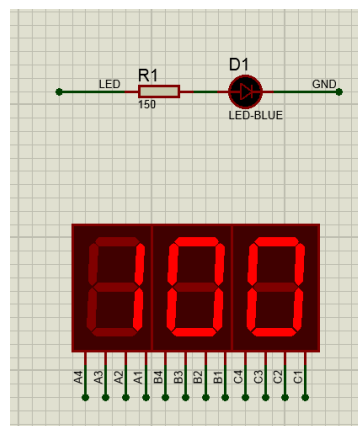


Figure 33 :L’Affichage

5. Description du programme :

La programmation du microprocesseur est réalisée par le langage assembleur. Le programme complet de la simulation est donné par:

```

MODEL SMALL
DATA
  PAA      EQU 10H
  PAB      EQU 12H
  PAC      EQU 14H
  CWA      EQU 16H
  PBA      EQU 08H
  PBB      EQU 0AH
  PBC      EQU 0CH
  CWB      EQU 0EH
  DE       DW 0H
  HDIR     DB 00H
  RHS      DB 30H
  DSPL     DW 0H

FSTEP  DB 00000011B,
        88888008B,

```

```

        88880088B ,
        88880880B,
        88888800B,
        88888008B ,
        88880088B ,
        88880880B

HSTEP DB 00000001 B,
        88888800 B,
        88888808 B,
        88888008 B,
        88888088 B,
        88880088 B,
        88880888 B,
        88880880 B

II      DW 0
STACK 10H
CODE
.STARTUP
    MOV AL ,10010000B
    OUT CWA , AL
    MOV AL ,10000000B
    OUT CWB , AL
MAIN PROC
    CALL GETPTO
    CALL GETVIT
    CALL MARCH
    CALL GETAFF
    CALL AFF
    CALL SLEEP
    JMP  MAIN
    RET
MAIN ENDP

MARCH PROC
    TEST HDIR , 1
    JNZ a
    LEA SI , FSTEP
    JMP b
a:  LEA SI , HSTEP
b:  TEST HDIR , 2
    JNZ CCW
CW: ADD SI,II
    MOV AL ,[SI /
    OUT PORTAC , AL
    CMP II , 7
    JZ n
    INC II
    RET
n : MOV II , 0
    RET
CCW: ADD SI , II

```

```

MOV AL , [SI]
OUT PORTAC , AL
CMP II,0
JZ m
DEC II
RET
m: MOV II,7
RET
MARCH ENDP

GETVIT PROC
IN AL , PAA
MOV BL , 35
MUL BL
ADD AX , 06FFH
MOV DE , AX
MOV BL , HDIR
SHR BL , 1
MOV AL , BL
OR AL , 010B
OUT PAB , AL
MOV CX , 00FFH
Loop convert
AND AL , 00000001B
OUT PAB , AL
RET
GETVIT ENDP

GETPOT PROC
IN AL , PAC
AND AL , RHS
MOV BX , 16
DIV BX
MOV HDIR , AL
RET
GETPOT ENDP

SLEEP PROC
PUSH CX
MOV CX, DE
delayloop: LOOP delayloop
POP CX

RET
SLEEP ENDP

AFF PROC
MOV AX , DSPL
MOV BX , 10
MOV DX , 0

DIV BX
XCHG AL , DL

```

```

OUT PBC , AL
XCHG AL , DL
MOV DL , 0
DIV BX
XCHG AL , DL
OUT PBB , AL
MOV AL , DL
DIV BL
MOV AL , AH
OUT PBA , AL
RET
AFF ENDP

GETAFF PROC
MOV AX ,DE
MOV BX ,64H
MUL BX
MOV BX ,185CH
DIV BX
MOV DSPL ,80H
SUB DSPL ,AX
RET
GETAFF ENDP
.EXIT
End

```

Le programme est constitué de sept fonctions :

GETPTO : cette fonction configure le convertisseur analogique numérique et réalise l'acquisition de la tension du potentiomètre qui représente la référence de la vitesse de rotation.

GETVIT : cette fonction utilise la valeur du potentiomètre acquise par l'ADC pour calculé le retard dans les pas du moteur qui détermineront son vitesse de rotation.

MARCH : cette fonction est responsable de faire tourner le moteur dans les deux sens de rotation ainsi que de changer la vitesse de rotation.

GETAFF: cette fonction est chargée du calcul de la vitesse de rotation.

AFF : cette fonction utilise la valeur de la vitesse de rotation calculer précédaiant pour l'afficher dans les trois afficheurs 7 segment.

SLEEP : cette fonction va arrêter tous le circuit pour une période de temps

MAIN : c'est la fonction principale qui représente une boucle sans fin qui va appeler tous les autres fonctions du programme.

6. Résultats de simulation

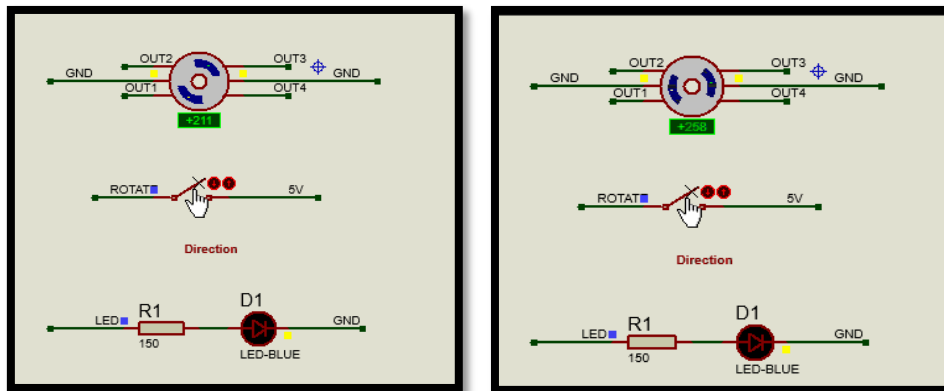


Figure 34 : Changer le sens de rotation

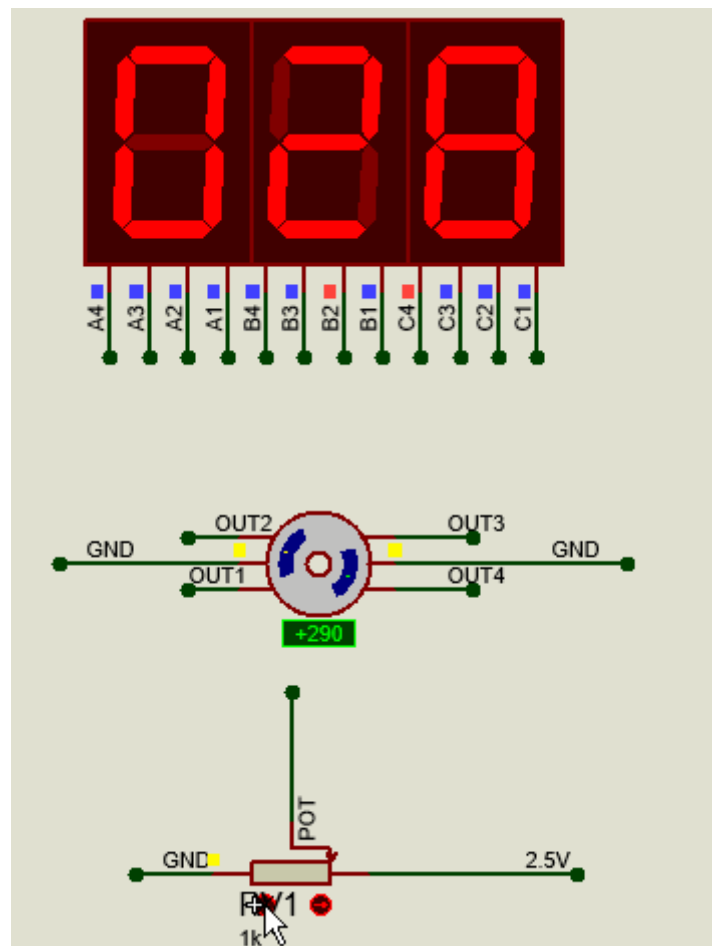


Figure 35 : Changer la vitesse de rotation

7. Conclusion

Dans ce chapitre nous avons réalisé une simulation sous ISIS PROTEUS pour commander un moteur pas à pas en utilisant le Microprocesseur 8086. La simulation nous permet d'obtenir 22 vitesses différentes du moteur dans deux sens de rotation différents. Ainsi que l'affichage du sens et vitesse de rotation.



Conclusion Générale

Conclusion Générale

Dans ce travail, nous avons pu réaliser une simulation de commande d'un moteur pas à pas architecturé autour du microprocesseur Intel 8086. Ce travail nous a permis de mettre en valeur une partie importante des connaissances théoriques que nous avons accumulées pendant toute la durée de nos études. Il nous a permis également d'acquérir une expérience nouvelle dans l'électronique pratique et la programmation en langage Assembleur. Au cours de la réalisation de notre projet, nous avons utilisé des logiciels de conception, simulation et programmation des circuits électroniques. En effet, nous avons utilisé l'Assembleur pour la programmation du microprocesseur, PROTEUS pour le dessin du circuit électrique et l'implantation des composants et pour la simulation de manière interactive de notre projet.



BIBLIOGRAPHIE

- [1] Bernard Multon - Historique des machines électromagnétiques - Revue 3E.In°3 - juin 1995
- [2] Kercha Mébarka,14/12/2005,Commande par mode de glissement d'un Moteur pas à pas à Aimant Permanent, Mémoire de magister, Faculté des Sciences de l'Ingénieur Département d'Electrotechnique, UNIVERSITE DE BATNA, 76p.
- [3] MENNAI.Noussaiba , LOUNIS.Chahrazed , REHAB BEKOUCHE.Aya,2017/2018,Les moteurs pas à pas (MPAP),Energies renouvelables et protection de l'environnement,ESTI Annaba,16p.
- [4] : **Patrice Oguic**,*Moteurs pas-à-pas PC*, France, éditionDunod, deuxième édition
- [5] Kharchoufa Youness,2017,Commande d'un moteur pas-à-pas par un microcontrôleur,LICENCE SCIENCES ET TECHNIQUES Génie électrique,FST- FES.41P.
- [6] RANDRIAMIARISOA Haritiana Léonce ,2011,Etu de Et commande de moteur PAS A PAS PAR ordinateur pour un positionnement sur le plan, Génie Mécanique et Productique - Génie Electrique, UNIVERSITE D'ANTANANARIVO ECOLE SUPERIEURE POLYTECHNIQUE, 38p.
- [7] : Hachette Technique, Claude Divoux , « Guide du technicien en électrotechnique, éd »,1999
- [8] : « Systèmes électromécaniques » ; Haute Ecole d'ingénierie et de Gestion Du Canton du Vaud,
CD/SEM/Cours/Chap07 . M. Correvon.
- [9] : **Electronique et loisir**, *Février 2000,numéro 9*
- [10] : **G. Pinso**, *Physique Appliquée Moteurpas àpas—C35/ 1*
- [11] M. Correvon,LES MOTEURS PAS À PAS, Systèmes électromécaniques,
CD\SEM\Cours \Chap7.doc ,60p.
- [12] benbekhti hamza,2020, Étude de réalisation de système de commande d'une machine cnc,Mémoire de fin d'étude En vue d'obtention du Diplôme de Master électronique,Centre nivercitaire belhadj bouchaib ain temouchant,58p.
- [13] TRANSDUCTEURS ELECTROMAGNETIQUES ,Auteur Marcel Jurer
- [14] Melle Hamdou Halima,COMMANDE D'UN MOTEUR PAS A PAS AVEC UNE CARTE D'ACQUISITION,l'obtention du diplôme de MAGISTER EN AUTOMATIQUE,UNIVERSITE D'ORAN ES-SENIA INSTITUT DE MAINTENANCE ET DE SECURITE INDUSTRIELLE,138p.



[15] M. DALMAU, Les Microprocesseurs, IUT de Bayonne, SD,105p.

[16] HAGGEGE, 2003, cours de microprocesseur, ISET Rades,84p.

[17] Cours Microprocesseurs et API, partie II: Intel 8086,13p.

[18] Mer. ATOUI Hamza,2007,Programmation Et Interfaçage De Microprocesseur 8086,Support de cours,Centre De Formation **HBM Corporation**,79p.

[19] A. Oumnad,Microprocesseur 8086,SD,53p.

[20] DOC ; Unit-4 Application of Microprocessor & Intel 8086 ;19p.

[21] <https://www.alldatasheet.fr/>

Résumé

The stepper motor constitutes an electromechanical converter intended to transform the electrical signal into mechanical displacement (angular or linear). These engines have developed due to the appearance of microprocessors. The 8086 microprocessor is a complex integrated circuit belonging to the VLSI (Very Large Scale Integration) family capable of sequentially and automatically performing sequences of elementary operations. Each microprocessor recognizes a set of instructions known as the manufacturer's fixed instruction set. Intel's 8086 processor is the basis of today's Pentium processors. Successive (PC) processors were in fact built little by little by adding additional instructions and functions to each processor, but each time retaining the specificities of the previous processor. In this study we practiced Control of a Stepper motor using the 8086 Microprocessor on PROTEUS (Proteus is an electronic CAD software suite) There are many control systems dedicated to the stepper motor. Nowadays, the most used control is that by the microcomputer thanks to the interfaces which make it possible to establish the communication between the central unit and the peripherals.

Mots clé : Le moteur pas-à-pas ; Le microprocesseur ; PROTEUS.

Summary

Le moteur pas-à-pas constitue un convertisseur électromécanique destiné à transformer le signal électrique en déplacement (angulaire ou linéaire) mécanique. Ces moteurs se sont développés dû à l'apparition des microprocesseurs. Le microprocesseur 8086 est un circuit intégré complexe appartenant à la famille des VLSI (Very Large Scale Intégration) capable d'effectuer séquentiellement et automatiquement des suites d'opérations élémentaires. Chaque microprocesseur reconnaît un jeu d'instructions connu sous le nom de jeu d'instructions fixe du fabricant. Le processeur 8086 d'Intel est à la base des processeurs Pentium actuels. Les processeurs successifs (de PC) se sont en effet construits petit à petit en ajoutant à chaque processeur des instructions et des fonctionnalités supplémentaires, mais en conservant à chaque fois les spécificités du processeur précédent. Dans cette étude on pratiquée Commande d'un moteur Pas à Pas en Utilisant le Microprocesseur 8086 sur PROTEUS (Proteus est une suite logicielle de CAO électronique) l existe de nombreux systèmes de commande dédiée au moteur pas à pas. De nos jours, la commande la plus utilisée est celle par le micro-ordinateur grâce aux interfaces qui permettent d'établir la communication entre l'unité centrale et les périphériques.

Keywords: The stepper motor, The microprocessor , PROTEUS

ملخص

يشكل محرك السائر محوّلًا كهروميكانيكيًا يهدف إلى تحويل الإشارة الكهربائية إلى إزاحة ميكانيكية (زاوي أو خطي). تم تطوير هذه المحركات بسبب ظهور المعالجات الدقيقة. المعالج الدقيق 8086 عبارة عن دائرة متكاملة معقدة تنتمي إلى عائلة VLSI تكامل واسع النطاق قادرة على تنفيذ تسلسل العمليات الأولية بشكل تسلسلي وتلقائي. يتعرف كل معالج دقيق على مجموعة من الإرشادات المعروفة باسم مجموعة التعليمات الثابتة الخاصة بالشركة المصنعة. يعد معالج Intel 8086 أساس معالجات Pentium الحالية. تم بناء المعالجات (PC) المتتالية شيئًا فشيئًا عن طريق إضافة تعليمات ووظائف إضافية لكل معالج ، ولكن في كل مرة يتم الاحتفاظ بخصائص المعالج السابق. في هذه الدراسة ، مارسنا التحكم في محرك متدرج باستخدام معالج دقيق 8086 على PROTEUS (Proteus) عبارة عن مجموعة برامج CAD إلكترونية) هناك العديد من أنظمة التحكم المخصصة للمحرك السائر. في الوقت الحاضر ، يعد التحكم الأكثر استخدامًا هو ذلك بواسطة الكمبيوتر المصغر بفضل الواجهات التي تجعل من الممكن إنشاء الاتصال بين الوحدة المركزية والأجهزة الطرفية.

كلمات المفتاحية: محرك السائر , المعالج الدقيق ,بروتايوس.