People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering
Department of Electronics

Final Year Project Report Presented in Partial Fulfilment of the Requirements of the Degree of

'MASTER'

In: **Computer Engineering**

Title:

FPGA Based Microstepping Controller

Presented By:

- **Rahal Chahira**

under supervision of :

Dr. A. Benzekri

2022/2023

Dedication

"

I dedicate this work to :

My tender ***Mother Fatiha*** *who represents for me the source of tenderness and the example of devotion who has never ceased to encourage me, who has done more than a mother can do so that his children follow the right path in their life.*

Mmy dearest ***father Mohamed*** *No words can express the love, respect that I have for him. Nothing in the world beats the efforts he made day and night for my education and well-being. This work is the fruit of its sacrifices.*

My beloved husband ***Amine*** *and my princess* ***Sami*** *whom has been a blessing to me.Your encouragement and support has always been a breath of fresh air for me.*

My brothers, ***Houssem*** *,* ***Nadjib*** *and* ***Abd erraouf*** *who I adore deeply and my dear sister* ***Nour****.*
My aunt, ***Malika****, my uncle* ***Ibrahim*** *and my cousin****Abd elmadjid*** *whom were always there to support me when I needed them .*

THANKS.

"

-Chahira

ABSTRACT

This project describes the design, simulation and implementation of an FPGA-based microstepping driver to control a dual H-Bridge to properly commutate a bipolar permanent magnet stepper motor for precise-position tracking applications. The kernel of the driver is a microstepping mode algorithm implemented in two ROMs as look-up tables. This engine is used to generate the appropriate pulse width modulation (PWM) signals to control the current levels in the motor's windings. Because the current patterns in the windings closely resemble sine waves with 90° phase shift, we used a sinusoidal (sine/cosine) approximations function to build the look-up table to drive the motor's windings. The digital driver is developed with the Very high speed integrated circuit Hardware Description Language (VHDL). The driver is synthesized using Quartus® II, the Intel®- FPGA software development suite tools, and targeted at an FPGA of the Cyclone-II family. Computer simulations are carried on Quartus II simulator. The results show the effectiveness and merit of this design process by testing several fractions of a full step (1/2, 1/4 and 1/8). In addition, the real-time applicability of this driver is exemplified on a permanent magnet bipolar stepper motor.

ACKNOWLEDGMENT

First of all, i would like to first thank God Almighty for granting me the strength and will to accomplish this work.

A big thank to my family for their unconditional love, their advice and their moral and support which allowed me to successfully complete the studies i wanted.

I thank my supervisor **Dr.Benzekri** for his monitoring, his advice and his discussions which helped me during my dissertation.

I also thank my freind **Radia Si Serir** for her advice and suggestions and for encouraging this work.

Finally, I would like to thank the members of the jury for accepting the evaluation of this work and all their comments and criticisms.

ABBREVIATIONS

FPGA	*Field Programmable gate array*
PM	*Permenant magnet*
VR	*Variable reluctance*
HSM	*Hybrid stepper motor*
VHDL	*Very high speed integrated circuit Hardware Description Language*
PCB	*Printed Circuit Board*
PWM	*Pulse Width Modulation*
ROM	*Read Only Memory*
IC	*Intergeated Circuit*
LUT	*Look Up Table*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

A stepper motor is a brushless, synchronous electric motor that converts digital pulses into mechanical shaft rotation. A motor that takes electrical pulses and changes them into mechanical movement. It is one of the few motors that is essentially digital in nature. A device is digital in nature when it can respond to digital information, a series of high and low voltages. Stepper motors are used in various industrial applications where low power is required and low speed, torque, fast dynamics and precise positioning are important factors, for example Stepper motors provide a means for precise positioning and speed control without the use of feedback sensors.

The importance and usage of the electric stepper motor has increased immensely during the past two decades. Stepper motors, for example, are widely used in modern electronic typewriters, word processors, and other computerized products and in medical applications for accurate medicament dosage with peristaltic pumps or pipettes and as motion control actuators in dialysis equipment [1].

They are important components of industrial robotic systems, large and small, and in other machines where repeatable positioning control is required. Stepper motors for most applications must be designed to work reliably over hundreds of thousands to millions of cycles. Stepper motors are used in applications that require precise control of the position of the motor shaft. Typical applications include pen positioning, rotary and indexing table control, robotic arm movement control, laser positioning, fax machine operation, and printer control [2].

Stepper motors are widely used for incremental motion control in such applications as computer peripherals, X- Y positioning tables, medical instruments, and specialized precision machinery requiring modest power. Some of the advantages of steppers include reduced maintenance owing to lack of brushes ; simplicity of operation owing to their open-loop design rather than expensive, elaborate feedback circuitry ; and adaptability to digital electronic devices.

The advantages of stepper motors are low cost, ruggedness, construction simplicity, high reliability, no maintenance, wide acceptance, no “tweaking” to stabilize, and no feedback components needed ; they are inherently fail-safe and tolerate most environments. Steppers are simple to drive and control in an open-loop configuration. They provide excellent torque at low speed, up to 5 times the continuous torque of a brush motor of the same frame size or double the torque of the equivalent brushless motor. Frequently, a gearbox can be eliminated. A : stepper-driven system is inherently stiff, with known limits to the dynamic position error [3].

Disadvantages of the stepper motor include no flexibility in step resolution, resonance effects, relatively long settling times, and rough performance unless a micro stepper is used. Undetected position loss may result in open-loop systems. Steppers consume current regardless of load conditions, and they tend to run hot. Steppers tend to be noisy, especially when operated at high speeds [3].

Microstepping was invented by Larry Durkos, a mechanical engineer of the American Monitor Corporation. Microstepping is actually a sine cosine driving in which the winding current approximates a sinusoidal waveform. This allows stopping and holding a position between two standard step positions and provides smoother operation in low speeds. PWM current control is usually combined with a nonlinear digital-to-analog converter which allows to control the motor current.

The basic principles of microstepping are described by Akdogan[17]. This work is focused on simulation and demonstration of microstepping technique, providing the fundamental mathematical background for calculating the current flow in both phases.

The most important benefit to be gained by microstepping is increased positioning resolution. The reduction in torque ripple, especially important at lower motor speeds. Microstepping minimizes the natural resonance problems that all stepper motors have.
Scope This project presents the design, simulation and implementation of an FPGA-based microstepping controller to control a dual H-Bridge to properly commutate a bipolar permanent magnet stepper motor for precise-position tracking applications. a step motor driver design with the aim of achieving micro-stepping features. Step motors are analyzed and different driving modes including several possible control approaches are discussed. An electronic board for drive and control of a bipolar step motor has been developed and used in experiments to compare different driving strategies with respect to their electrical and mechanical performances.

This project describes the design, simulation and implementation of an FPGA-based microstepping driver to control a dual H-Bridge to properly commutate a bipolar permanent magnet stepper motor for precise-position tracking applications. The kernel of the driver is a microstepping mode algorithm implemented in two ROMs as look-up tables. This engine is used to generate the appropriate pulse width modulation (PWM) signals to control the current levels in the motor's windings.

Related work

The initial step motor studies dates back to 1920's [1]. A three phase variable reluctance step motor was introduced as a position control device, despite the fact that the identical motor structure was known and operated as an electromagnetic engine in the 19th century [2]. During this time, permanent magnets were first used in step motor structures, and step motor torque and step precision were gradually increased through inventions [3]. The General Electric Company created, patented, and produced the hybrid step motor that is still in use today in 1952 [4]. Step motor technology advanced in the 1960s and 1970s as a result of advances in computer technology.

Furthermore, because step motors are discrete, using them has grown easier and more preferred as microprocessors become more widely used.

Mechanical switches operated by hand were the first instances of devices used to excite step motors. Later, more advanced and independent driver alternatives have been used thanks to the development of switching devices such thyratron gas tubes. These switches were gradually replaced by thyristors and transistors. With the appearance of the integrated circuits, the implementation of logic circuit of drivers became simpler with lower costs.

Because of their affordability, ease of use, and inexpensive driver solutions, step motors are appealing and frequently utilized in the industry. There are several ways to integrate a driver's power converter block ; different manufacturers offer partially or fully on-chip solutions. ST Microelectronics' L297–L298 integrated circuits are among of the most well-known driver options.

Step motors can also be driven by extremely complex single chip solutions that are currently available on the market. Examples of this kind of drivers are the Allegro A3981 [5], A4980 [6], and AMIS-30523 [7]. Discrete driver solutions are still often utilized even though on-chip solutions provide straightforward assembly and a fully working drive circuitry with a small number of external components.

For the majority of integrated driver integrated circuits, drive control logic circuitry possibilities are also restricted and unchangeable. On the other hand, a PCB level driver that uses a microcontroller device to manage the driver logic provides a versatile setting for optimizing different driver algorithms. The processor, which generates the driver logic, is the main part of a PCB level step motor driver. Step motor drivers frequently use DSPs (Digital Signal Processors) as processors in addition to microcontrollers.

In recent years, step motor driver solutions based on Field Programmable Gate Arrays (FPGA) have also been introduced. An FPGA-based open-loop step motor driver is proposed by Le and Jeon [8]. The reasoning behind FPGA-based solutions is because they are compatible with Application Specific Integrated Circuit (ASIC) conversion, which may make them preferable even though they are more expensive and difficult to build.

A step motor driver solution realized with a new generation PIC family is introduced in an application note from Microchip [9].

Motivation

My personal motivation when choosing a topic for my Master's project was my interest in FPGA technology and the advantages of synthesized hardware over the classic microcontrollers or any other device used to perform the operations involved in the control of Alternating Current Motor methods. My intention was to find a useful application where FPGAs could have a real improvement of performance and time response. This is why, while defining the experimental part of the project, a good option was the implementation of a motor control method in VHDL.

Outline

This report contains four chapters. Chapter 2 introduces step motors which are the mechanical components to be driven with the driver circuitry and provides a discussion on step motor control methods. Chapter 3 highlights The proposed driver circuit and discusses experimental work and their results. Finally, Chapter 4 puts forward conclusive remarks and future directions.

Chapitre 1

THEORITICAL BACKGROUND

This chapter deals with the fundamental ideas behind step motor theory. The discussion is opened with the basic operation principle of step motors in the first section. Different types of step motors in use and different winding structures are mentioned in this chapter. Afterwards, several methods that are used to excite and operate step motors are discussed. Microstepping method, a primary focus of this project, is presented in chapter. An overview that covers different aspects related to FPGAs is mentioned.

STEP MOTOR THEORY

1.1 Basic operation of stepper motor

Angular displacement of step motors is possible with moving active parts with magnetic force which is induced with discrete electrical pulses.

A basic type of step motor, called the hybrid step motor, consists of a permanent magnet, toothed rotor made from two sections, or “cups,” that are opposite in polarity. The electromagnetic stator is also toothed. This section explains the principles of motion inside a 4-phase hybrid stepper motor in order to make an introduction to the basic concepts of step motors. Different types of step motors are discussed and for simplicity, a 3-phase hybrid stepper motor is used to explain different excitation modes in the following sections in more detail.

Basic structure of a hybrid step motor is shown in Fig. I.1. The rotor is in two sections and has 50 teeth on each section.The half-tooth displacement between the two sections is retained. The stator has eight, each with five teeth, making a total of 40 teeth then the step angle is 1.8°. Visualize that a tooth is placed in each of the gaps between the stator poles, in which case there would be a total of 48 teeth, two less than the number of rotor teeth.

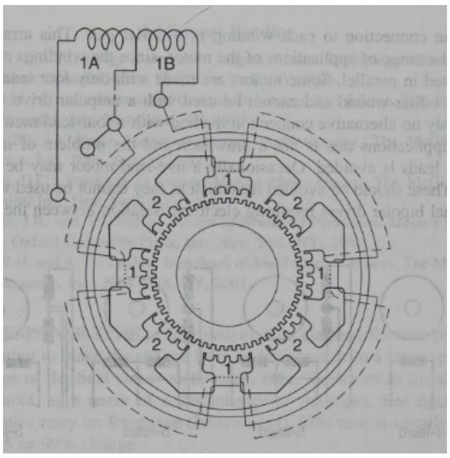

Fig. I.1 : Cross sectional diagram of a 4-phase hybrid step motor.

The stepper motor uses the theory of operation for magnets to make the motor shaft turn a precise distance when a pulse of electricity is provided [11]. The shaft of a stepper motor rotates at fixed angles when it receives an electrical pulse. The electrical pulse magnetizes the motor's stationary field, called the stator field. The magnetic stator field moves the permanent magnet. The permanent magnet is called the rotor or armature. The rotor steps forward to align with the stator's magnetic field (N to S and S to N) and stops movement until another stator field is energized [14]. If we manually rotate the magnet without energizing any coils, we get the 'notched' feeling whenever a relatively larger magnetic force is generated, because of the alignment of the permanent magnet with the core of the electromagnets. This force is termed 'detent torque' [16].

Fig. I.2 when no power is applied to the motor, the residual magnetism in the rotor magnets will cause the rotor to detent or align one set of its magnetic poles with the magnetic poles of one of the stator magnets. When the rotor is in a detent position, it will have enough magnetic force to keep the shaft from moving to the next position. This is what makes the rotor feel like it is clicking from one position to the next as you rotate the rotor by hand with no power applied [11].

Fig. I.2 when power is applied, it is directed to only one of the stator pairs of windings, which will cause that winding pair to become a magnet. One of the coils for the pair will become the north pole, and the other will become the south pole. When this occurs, the stator coil that is the north pole will attract the closest rotor tooth that has the opposite polarity, and the stator coil that is the south pole will attract the closest rotor tooth that has the opposite polarity. When current is flowing through these poles, the rotor will now have a much stronger attraction to the stator winding, and the increased torque is called holding torque [11].

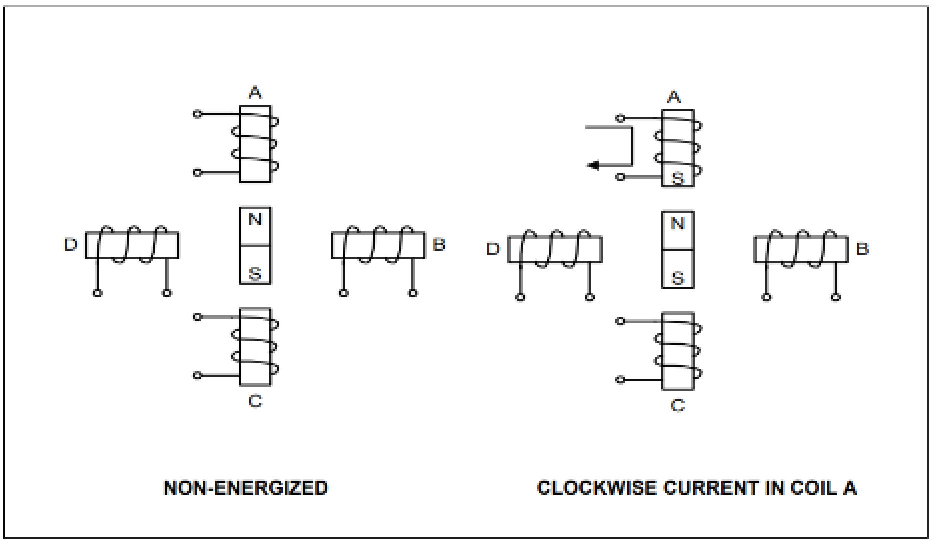

Fig. I.2 : Non energized and Clockwise current in coil.

If rotor and stator teeth were aligned at 12 :0'clock, they would also be aligned at 6 o'clock. But at 3 and 9 o'clock the teeth would be misaligned. However, due to the displacement between the sets of rotor teeth, alignment will occur at 3 o'clock and 9 o'clock at the other end of the rotor [12].
In practice, the windings are arranged in sets of four and wound such that diametrically

opposite poles are the same. Thus, referring to Fig. I.1, the north poles at 12 and 6 o'clock attract the south-pole teeth at the front of the rotor and the south poles at 3 and 9 o'clock attract the north-pole teeth at the back. By switching current to the second set of coils, the stator field pattern rotates through 45°, but to align with this new field, the rotor only has to turn through 1.8°. This is equivalent to one-quarter of a tooth pitch on the rotor, giving 200 full steps per revolution [12].

Note that there are as many detent positions as there are full steps per revolution, namely, 200. The detent positions correspond with rotor teeth being fully aligned with stator teeth. When power is applied to a stepper drive, it is usual for it to energize in the zero-phase state in which there is current in both sets of windings. The resulting rotor position does not correspond with a natural detent position, so an unloaded motor will always move by at least one-half step at power on. Of course, if the system were turned off other' than in the zero-phase state, or the motor is moved in the meantime, a greater movement may be seen at power-up [12].

For a given current pattern in the windings there are as many stable positions as there are rotor teeth (50 for a 200-step motor). If a motor is desynchronized, the resulting position error will always be a whole number of rotor teeth, or a multiple of 7.2° : A motor cannot "miss" individual steps. Position errors of one or two steps may be due to noise, spurious step pulses, or a controller fault[10].

1.2 Stepper motor parameters

1.2.1 Step Angle

Step angle is defined as the angle at which the rotor of a stepper motor moves when one pulse is applied to the input of the stator. It is expressed in degrees.

One electrical period of stepper motor corresponds to 4 full-step of mechanical movement. That is, 360 degrees of electrical angle doesn't same the stepper motor's rotation angle. One full electrical period finishes atter four full mechanical step, which corresponds to 90 degrees of electical angle [17].

Mechanical angle represents the step angle of the step. In the full step mode of a 1.8° motor, the mechanical angle is 1.8°. In the 10 micro step mode of a 1.8° motor, the mechanical angle is 0.18º. An electrical angle is defined as 360° divided by the number of mechanical phases and the number of micro step. In the full step mode of a 1.8° motor, the electrical angle is 90°. In the 10 micro step excitation of a 1.8° motor, the electrical angle is 9º [18].

$$\theta = |\frac{(N_s - N_r) * 360^{\circ}}{(N_s * N_r)}| \quad (1.1)$$

where :
Nr= number of rotor teeth
Ns= number of stator teeth

1.2.2 Resolution

The resolution or the step number of a motor is the number of steps it makes in one revolution of the rotor. The smaller the step angle, the higher the resolution of the positioning of the stepper motor [19].

$$Resolution = \frac{step\, angle}{steps\, per\, revolution} \quad (1.2)$$

$$Resolution = \frac{number\, of\, steps}{number\, of\, revolutions} \quad (1.3)$$

1.2.3 Steps per each revolution

$$Steps\, per\, each\, revolution = \frac{360}{step\, angle} \quad (1.4)$$

$$Revolution\, per\, second = \frac{(steps\, per\, second)}{(steps\, per\, revolution)} \quad (1.5)$$

Step angle	Steps per revolution
0.9°	400
1.8°	200
3.6°	100
7.2°	50
15°	24

TAB. I.1 : Steps per revolution correspond to each step angle.

1.2.4 Stepper motor speed

RPM (revolutions per minute) is the measure used to describe the speed of an electric motor [21].

$$RPM = (\frac{steps\, per\, second}{steps\, per\, revolution}) * 60 \quad (1.6)$$

However, a slightly different formula (1.7) is used to calculate the maximum speed of a stepper motor [20].

$$W_{max} = \frac{V}{(2Li_{max} * spr)} \quad (1.7)$$

where :
W_{max} = Maximum speed of the stepper motor
V= actual applied voltage
i_{max}= maximum current
L= winding inductance

1.3 Types of stepper motors

This section lists and describes different types of step motors according to their magnetic structure and how magnetic flux triggers the motion of their rotors. The first type of step motor is the permanent magnet motor, which utilizes permanent magnets to perform electromechanical rotation. The second type is variable reluctance motors are discussed further and several different variants are given . Finally, the hybrid step motors combine mechanical and electromagnetic properties of the former two types to achieve higher torque within the same physical volume.

1.3.1 Permenant magnet (PM) stepper motor

Permanent magnet step motors have cylindrical magnet rotors and single pole electrical phases. When a phase is excited, magnetic flux occurs inducing north or south pole on a single stator teeth, which in turn aligns the permanent magnet rotor inside. Illustration of a permanent magnet step motor is shown in Fig. I.3.

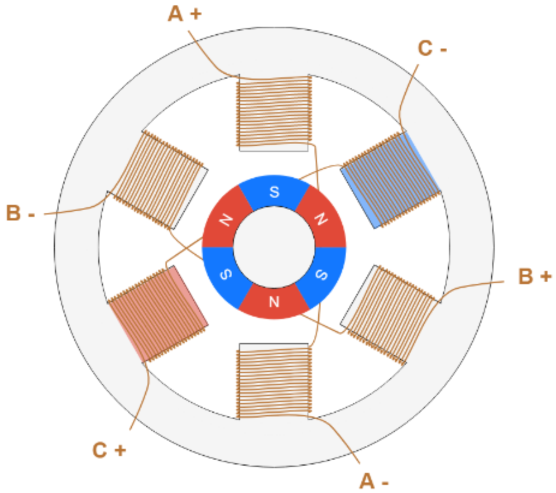

Fig. I.3 : Permenant magnet stepper motor.

The tin-can, or "canstack," motor shown in Fig. I.4 is perhaps the most widely used type in commercial, nonindustrial applications.It is essentially a low-cost, low-torque, low-speed device ideally suited for use in computer Peripherals [12].

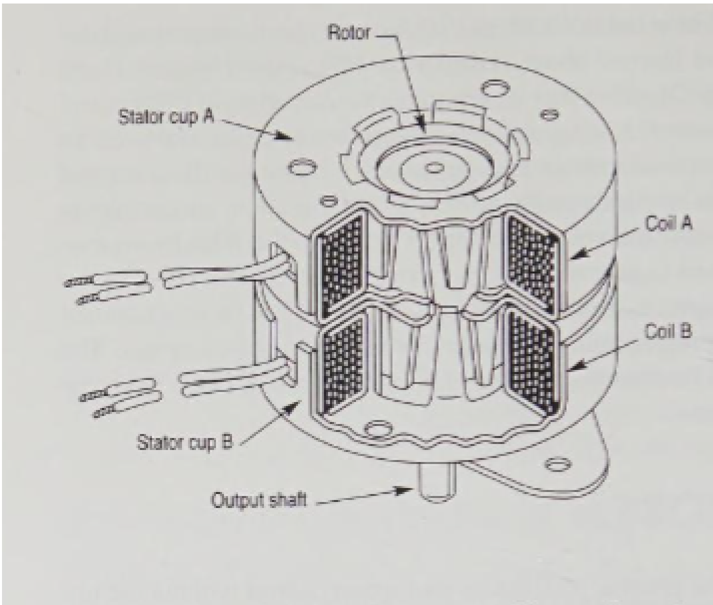

Fig. I.4 : Cutaway diagram of a permanent magnet stepper motor. (Courtesy of Parker Compumotor Division)

The main feature of the permanent magnet motor is that a permanent magnet is used for the rotor, which means that no brushes are required. The drawback of this type of motor is that it has relatively low torque and must be used for lowspeed applications [11].

1.3.2 The variable reluctance (VR) stepper motor

The variable-reluctance motor is a type of motor where the rotor is not cylindrical but looks like bars with several teeth on it. The rotor teeth are made of soft iron. In the non-energized condition, there is no magnetic flux in the air gap, as the stator is an electromagnet and the rotor is a piece of soft iron [22] Thus, the rotor spins freely without detent torque [12]. Variable reluctance motors are normally constructed with three or five stator windings, as opposed to the two windings in the PM motors [22].

The amount of torque for this type of motor is still small, so it is generally used for small positioning tables and other small positioning loads. Since this type of motor does not have permanent magnets, it cannot use the same type of stepper controller as other types of stepper motors [11].

Variable-reluctance motors are seldom used in industrial applications. These motors are not sensitive to current polarity and thus require a different driving arrangement compared to other types [12].

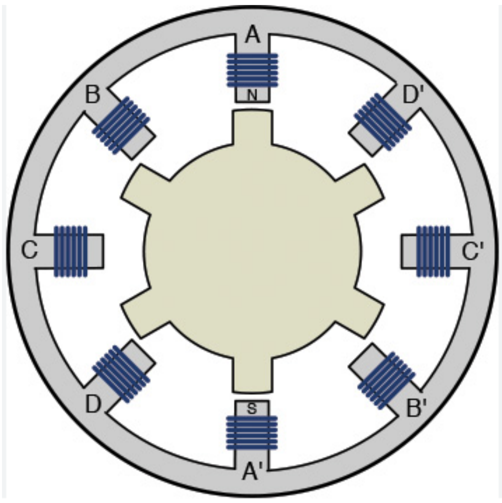

Fig. I.5 : Variable reluctance stepper motor.

1.3.3 Hybrid Stepper Motor(HSM)

Construction of permanent magnet motors becomes very complex below 7.5 degrees step angles. Smaller step angles can be realized by combining the variable reluctance motor and the permanent magnet motor principles. Such motors are called hybrid motors [16]. it is the most widely used because it gives much smaller step angles, as small as 0.9 degrees per step [16]. hybrid motors combines features of both the variable reluctance and the permanent magnet stepper motor in that it combines a permanent magnet core and a toothed rotor [23].

A typical hybrid motor is shown in Fig. I.6. The stator construction is similar to the permanent magnet motor, and the rotor is cylindrical and magnetized like the PM motor with multiple teeth like a VR motor. The teeth on the rotor provide a better path for the flux to flow through the preferred locations in the air gap. This increases the detent, holding, and dynamic torque characteristics of the motor compared to the other two types of motors[16].

the stator construction of Permanent magnet motors and hybrid motors is very similar, a common control circuit can easily drive both types of motors [16].

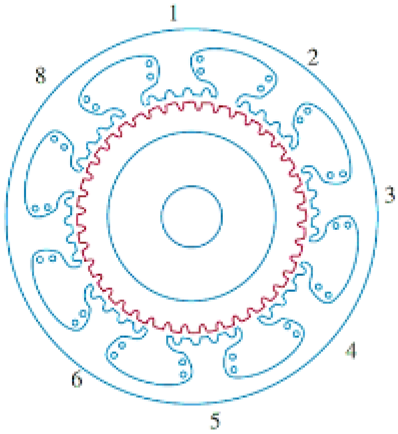

Fig. I.6 : Hybrid stepper motor.

1.4 Step Motor Winding Types

1.4.1 Unifilar Winding

Unifilar, as the name implies, has only one winding per stator pole. Stepper motors with a unifilar winding will have 4 lead wires [24].

1.4.2 Bifilar Winding

Most motors are described as being bifilar wound, which means there are two identical sets of windings on each pole. Two lengths of wire are wound together as though they were a single coil [12].

1.5 Unipolar and Bipolar Winding Drive

Two leads on each of the four coils of a stepper motor can be brought out in different ways.

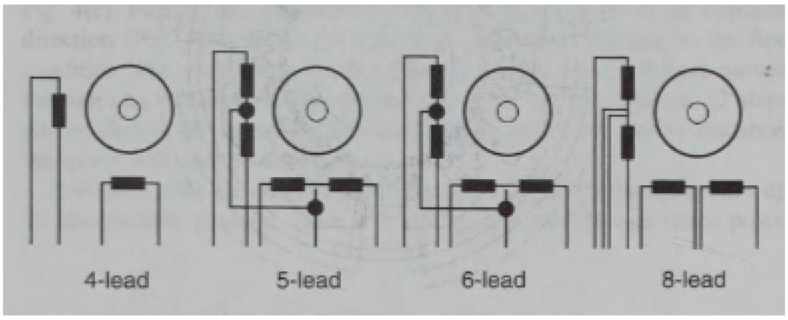

FIG. I.7 : motor lead configurations

If the wires are brought out with one or two center tap(s), it is called a unipolar motor and if the coil ends are brought out as shown in Fig. I.18 , then the motor is called a bipolar motor.

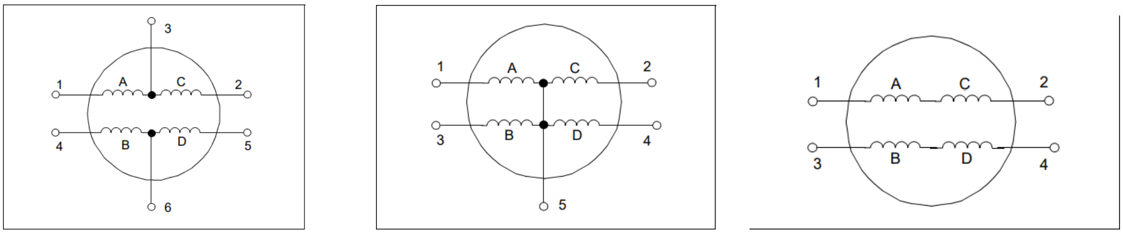

FIG. I.8 : Unipolar and bipolar lead configuration

1.5.1 Unipolar Winding Drive

The stepper motor is called unipolar stepper motor because current flows in only one direction in each winding and will energize its coils at a single polarity. It is also called bifilar because it contains two coils whose polar orientation is opposite each other while wrapped on the same core with a center tap. The unipolar stepper is also called four-phase because it has four windings to energize. The unipolar stepper motor requires only that a driver switch one polarity of current to its windings [22].

Unipolar stepper motors has either 5 or 6 wires to connect. They are most commonly found with step angle 1.8° per step (200 steps per full rotation) or 3.6° per step (100 steps per full rotation) or 7.5° per step (48 steps per rotation) [22].

1.5.2 Bipolar Winding Drive

The bipolar stepper motor has two sets of windings alternating positions. The bipolar stepper motor moves by energizing its coils first one polarity and then reversing the polarity as the rotor is turned, hence the designation bipolar [22].

The bipolar stepper motor is so named because each winding is energized in both directions. It's called unifilar because each pole has a single winding and it's also called two-phase because it has two separate windings [22].

Bipolar stepper motors are stronger and faster than unipolar stepper motors of the same size because a bipolar stepper motor has half the number of windings(a single winding that isn't center tapped). However, a bipolar stepper motor requires that the driver be able to switch the polarity of the stator windings, and this makes the bipolar driver a more complex circuit [22].

The bipolar controller must be able to reverse the polarity of the voltage across either coil, so current can flow in both direction. And, it must be able to energize these coils in sequence. This circuit is called an H-bridge. Two such circuits are needed to drive both coils of bipolar stepper motor, and is commonly called a “dual Hbridge” [25].

1.6 Full-Bridge Converter

The full-bridge converter is a topology that allows winding current to flow in both directions and it is the typical driving circuitry for bipolar step motors. Full-bridge converter has two legs as shown in Fig. I.9. Each leg contains two transistors that can pull each wire high or low and act as switches and two antiparallel diodes, also called catch diodes. The center points of the two legs are the output ports of the bridge and connected to the phase of stepper motor. Full-bridge circuitry is also called H-bridge [26].

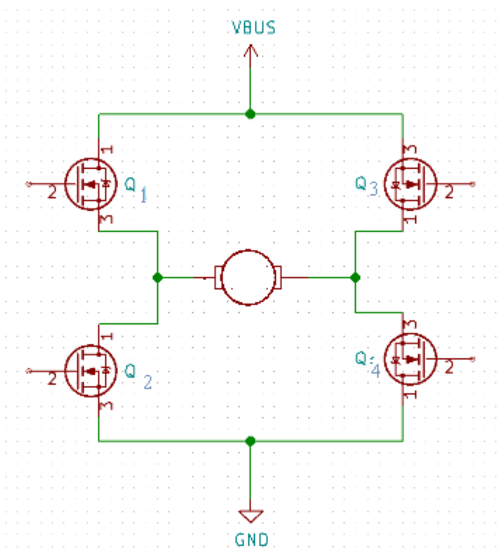

Fig. I.9 : H-bridge for the first winding.

One technique to drive the bridge is turning Q1 and Q4 on while turning Q2 and Q3 off as the first state, and in the second state turning Q2 and Q3 on while turning Q1 and

Q4 off. In full-bridge terminology, this type of driving is referred as bipolar driving mode.

This H-bridge will control the current through one of the windings. Since there are two windings, we need to double this circuit.

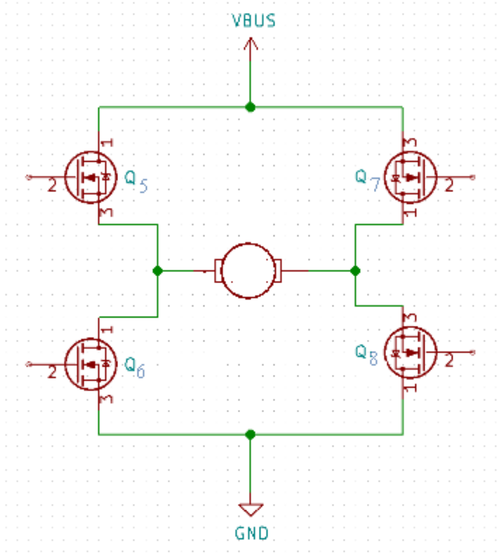

Fig. I.10 : H-bridge for the second winding.

Q1 and Q2 (or Q3 and Q4)must never be close at the same time. If that happens, a direct path between the power supply voltage and ground which is of very low resistance is created, which ultimately results in a short circuit. A short circuit can lead to destruction of the H-bridge, or some other component in the circuit.

The basic work principle of the H-bridge to rotate the stepper motor in the case of bipolar motor is : if Q1 and Q4 are turned on, the left side of the winding will be connected to the power supply, and the right to the ground while Q2 and Q3 are turned off. The current flows through the winding and make the first step to start rotating the the stepper motor.

if Q5 and Q8 are turned on, the left side of the winding will be connected to the power supply, and the right to the ground while Q6 and Q7 are turned off. The current flows through the winding and make the second step to rotate the the stepper motor.

If Q2 and Q3 are on while Q1 and Q4 are turned off, the winding will be energized in the opposite direction that means the winding will be powered in the reversed direction and third step will be done.

If Q6 and Q7 are on while Q5 and Q8 are turned off, the winding will be energized in the opposite direction that means the winding will be powered in the reversed direction and fourth step will be done.

the sequence will be repeated to cause the motor run.

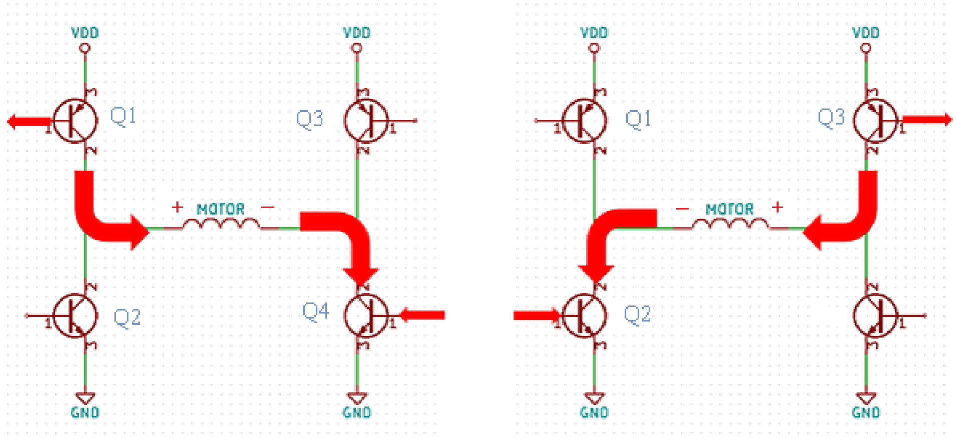

FIG. I.11 : The winding current Flow in forward and reverse direction.

1.7 Step Motor Excitation Modes

There are four stepping modes of driving the stepper motor. The stepping mode refers to the sequence pattern in which stator coils are energized.

1.7.1 Wave Drive Excitation

Wave stepping involves turning on one winding at a time where only a single phase winding is excited at a time [23]. All phases are excited one by one in an alternating sequence as represented in Fig. I.12.

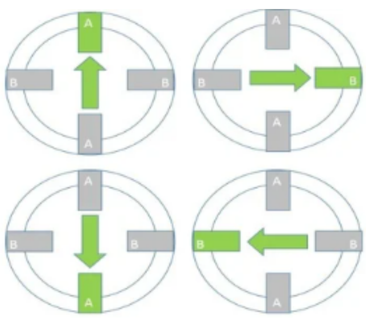

FIG. I.12 : Wave drive excitation.

This operation is called full-step excitation since it causes rotation in full natural steps.

Tab. I.2 gives the winding energizing step patterns for unipolar stepper motor in the case of wave step excitation.

	A	B	C	D
1	On			
2		On		
3			On	
4				On

TAB. I.2 : Step pattern of wave excitation using unipolar winding drive.

Tab. I.3 gives the winding energizing step patterns for bipolar stepper motor in the case of wave step excitation.

A_1		**A_2**	**B_1**	**B_2**
1	-	+		
2			+	-
3	+	-		
4			-	+

TAB. I.3 : Step pattern of wave excitation using bipolar winding drive.

The excitation sequence for wave excitation is given in Fig. I.13.

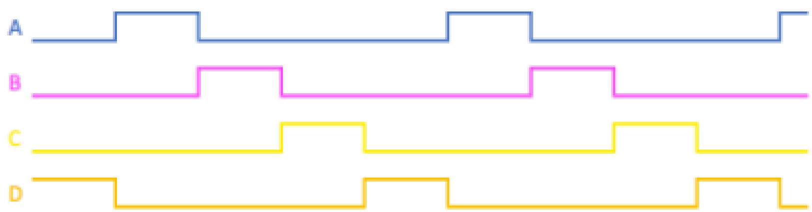

FIG. I.13 : wave drive or single coil excitation.

1.7.2 Two-Phase On Excitation

Two-phase on energizes two windings at a time, and the rotor turns to align its permanent magnet at a point halfway between the two energized windings as illustrated in Fig. I.14. Two-phase uses twice the power of wave stepping because two windings are turned on [22]. This operation is also called full-step excitation since it causes rotation in full natural steps.

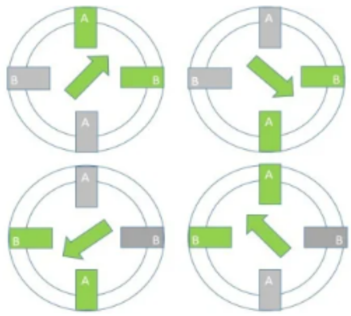

FIG. I.14 : Two-pahse on excitation.

If the two coils are energized simultaneously, the rotor takes up an intermediate position, since it is equally attracted to two stator poles.

Greater torque is produced under these conditions because all the stator poles are influencing the motor. The motor can be made to take a full step simply by reversing the current in one set of windings. This causes a 90° rotation of the stator field, as before. In fact, this would be the normal way of driving the motor in the full-step mode, always keeping two windings energized and reversing the current in each winding alternately.

Tab. I.4 gives the winding energizing step patterns for unipolar stepper motor in the case of two-phase on step excitation.

	A	B	C	D
1	On	On		
2		On	On	
3			On	On
4	On			On

TAB. I.4 : Two-phase on drive excitation using unipolar winding drive.

Tab. I.5 gives the winding energizing step patterns for bipolar stepper motor in the case of two-phase on step excitation.

	$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{B}_1$	$\mathbf{B}_2$
1	-	+	-	+
2	-	+	+	-
3	+	-	+	-
4	+	-	-	+

TAB. I.5 : Two-phase on drive excitation using bipolar winding drive.

The excitation sequence for two-phase on excitation is given in Fig. I.15.

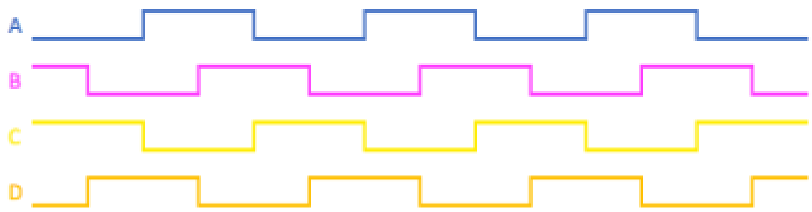

FIG. I.15 : Two-phase On Excitation

1.7.3 Half-Step Excitation

In half-step excitation, the actual step angle is divided into half and all rotor positions from both wave drive excitation and two-phase on excitation are offered. Higher angular precision but less torque are the effects of this process.

By alternately energizing one winding and then two (Fig. I.16), the rotor moves through only the half actual angle at each stage, and the number of steps per revolution will be doubled.

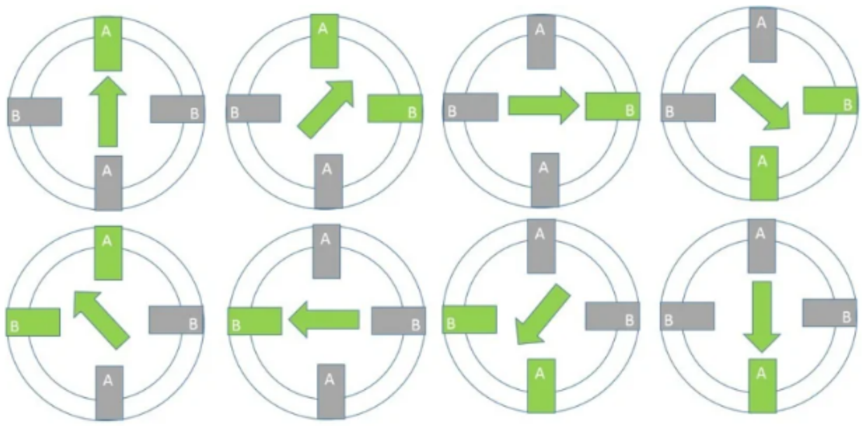

FIG. I.16 : Half step excitation.

Most industrial applications make use of this stepping mode. This mode results in much better smoothness at low speeds, and less overshoot and ringing occur at the end of each step.

Tab. I.6 gives the winding energizing step patterns for unipolar stepper motor in the case of half step excitation.

	A	B	C	D
1	On			
2	On	On		
3		On		
4		On	On	
5			On	
6			On	On
7				On
8	On			On

Tab. I.6 : Half-step drive excitation using unipolar winding drive.

Tab. I.7 gives the winding energizing step patterns for bipolar stepper motor in the case of half step excitation.

	$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{B}_1$	$\mathbf{B}_2$
1	-	+	-	+
2			-	+
3	+	-	-	+
4	+	-		
5	+	-	+	-
6			+	-
7	-	+	+	-
8	-	+		

Tab. I.7 : Half-step drive excitation using bipolar winding drive.

The waveform of the excitation sequence for two-phase on excitation is given in Fig. I.17.

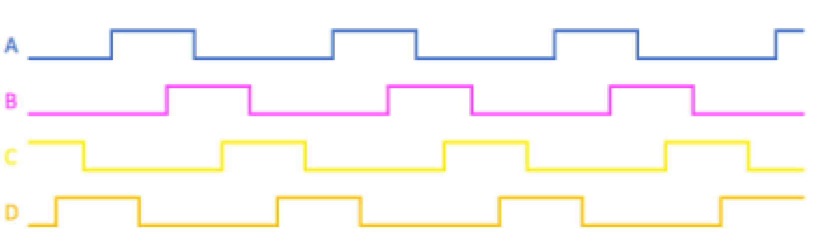

Fig. I.17 : Excitation sequence waveform of half step excitation.

1.8 Microstepping Mode

The main topic of this project is the microstepping excitation mode. It will be noted from the prior discussion that energizing both windings with equal current produces intermediate step position halfway between the one-phase on positions.if two phase current are

Unequal, the rotor position will be shifted toward the stronger pole. This effect is utilized in the microstepping drive, which is an electronic technique that divides a step motor's natural step into numerous little steps by proportioning the current flowing through the motor's windings to provide additional position between the poles of the rotor [27].

Accurate microstepping places increased demands on the accuracy of current control in the drive, particularly at low current levels. A small phase imbalance that may be barely detectable in a halfstep drive can produce unacceptable positioning errors in a microstep system [27].

The amount of resolution (number of steps) can be increased by manipulating the current that the controller sends to the motor during each step [28].

Subdivision of the basic motor step produces a series of intermediate step positions between the one phase-on points. It is clearly desirable that these intermediate positions are equally spaced and produce approximately equal torque when the motor is running[27].

Some applications are required smaller angle than full step angle. Half-step and micro-step driving modes were developed with this aim. Thus, the stepper-motor runing modes include full-step, half-step and micro-stepping modes. Half-step modes has 1/2 rate of full-step. Microstepping modes have 1/4, 1/8, 1/16, 1/32 rates of full-step with stepping rate equal to 800, 1600, 3200 steps per a revolution, respectively.

The current can be varied in one winding with a sine function of an angle θ and in the other winding with a cosine function of $'\theta'$ as shown in Fig. I.18.

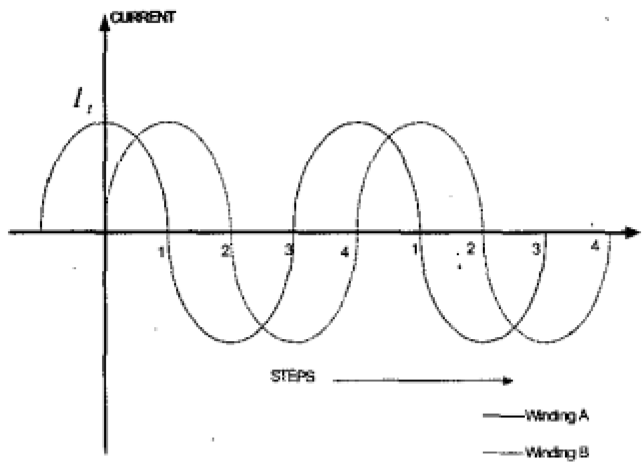

FIG. I.18 : Current applied to the stator winding.

The total current applied to the windings is

$$I_{\text{stat}} = \sqrt{(I_t \sin\theta)^2 + (I_t \cos\theta)^2} \tag{1.8}$$

$$I_{\text{stat}} = I_t * \sqrt{\sin^2\theta + \cos^2\theta} \tag{1.9}$$

Here, I_{stat} represent the final stator current. θ is electrical phase angle. I_t is rated maximum winding current, I_a and I_b are the current of stator winding A and B, respectively, so that,

$$I_a = I_t^* \sin\theta \tag{1.10}$$

$$I_b = I_t^* \cos\theta \tag{1.11}$$

In practice, current that in one winding is constant in half of full step and it changes as a function of $\sin\theta$, in other winding. Then, the final stator current is,

$$I_{\text{slat}} = \sqrt{I_t^2 + (I_t^* \sin\theta)^2} = I_t\sqrt{1 + \sin^2\theta} \tag{1.12}$$

In microstepping mode the amount of current flowing through the windings in each step have to be controlled at discrete levels.

The reference signals used to control the discrete current levels in each of phase windings can be obtained from the sinusoidal approximations to the static torque-displacement (T-θ) characteristic curve [27].

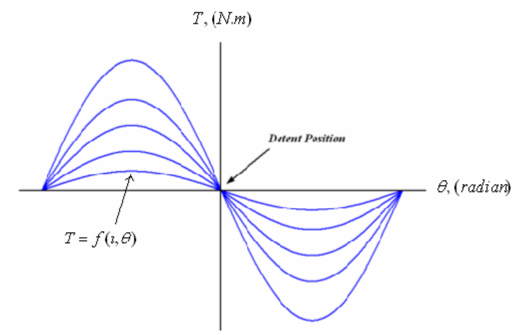

Fig. I.19 : Linear approximation to T-θ curves.

The MCRs for microstepping controllers typically have static torque around a detent position and a nonlinear relationship between the angular displacement θ and the current i.
The static torque (T) of a two-phase stepper motor can expressed as a function which is given by

$$T = f(i, \theta) \tag{1.13}$$

When the torque is zero, the normal detent positions take place at an angle of nθ_s, where n is 1, 2, 3, etc. and s is a regular step size.

Current references for each phase winding must be computed from (1.1) if sub-detent positions are required at an angle of n θ_s/m, where m is the number of microsteps.

As illustrated in Fig. I.20, the function can be approximated by a sinusoidal approximation of the T-θ curve in Fig. I.19.

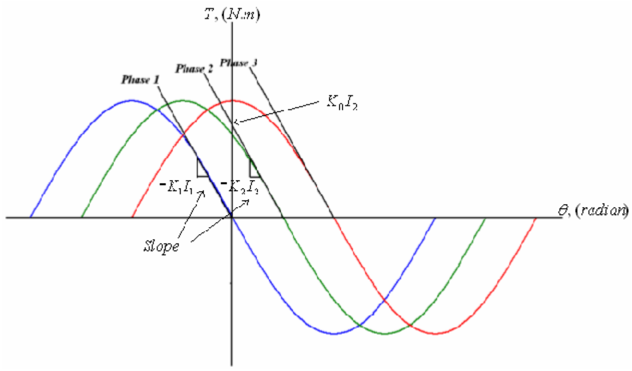

FIG. I.20 : Linear and sinusoidal approximation to T-θcurve.

Using the sine approximation to $T - \theta$ curves, the following description of each phase torque can be given.

$$T_1 = -T_0 I_1 \sin\theta$$
$$T_2 = T_0 I_2 \sin(\theta + \theta_s) \tag{1.14}$$

T0 is peak static torque. by setting $T_1 + T_2 = 0$ at any microstep detent position, we can obtain

$$-T_0 I_1 \sin\theta + T_0 I_2 \sin(\theta + \theta_s) = 0 \tag{1.15}$$

$$\frac{l_2}{l_1} = \frac{\sin\theta}{\sin(\theta + \theta_s)} \tag{1.16}$$

We arrive to

$$\frac{I_2}{I_1} = \frac{\tan\theta}{\tan\theta\cos\theta_s + \sin\theta_s} \tag{1.17}$$

The restoring torque at any microstep detent must be equal to T1 or T2 in order to maintain the T- θ curve characteristic for microstep mode.

$$T_0 I_2 \sin(\theta' + \theta_s) - T_0 I_1 \sin\theta' = T_0 \sin\left(\frac{\theta_s}{m}\right) \tag{1.18}$$

Where

$$\theta' = \theta - \frac{\theta_s}{m} \text{ and m} = 1, 2 \ldots, \text{ etc.} \tag{1.19}$$

The combination of (15) and (16) can leads to

$$I_1 = \frac{\sin\left(\frac{\theta_S}{m}\right)}{\frac{\text{lann}\,\theta}{\tan\theta\cos\theta_s + \sin\theta_s}\sin(\theta' + \theta_s) - \sin\theta'} \tag{1.20}$$

For different ministep detent positions, current of phase 1 and current of phase 2 can be found from (1.18) and (1.20).

Fig. I.21 shows the effect of dividing the 'electrical angle' into smaller, equal angles, and the corresponding current is given to the stator windings [28].

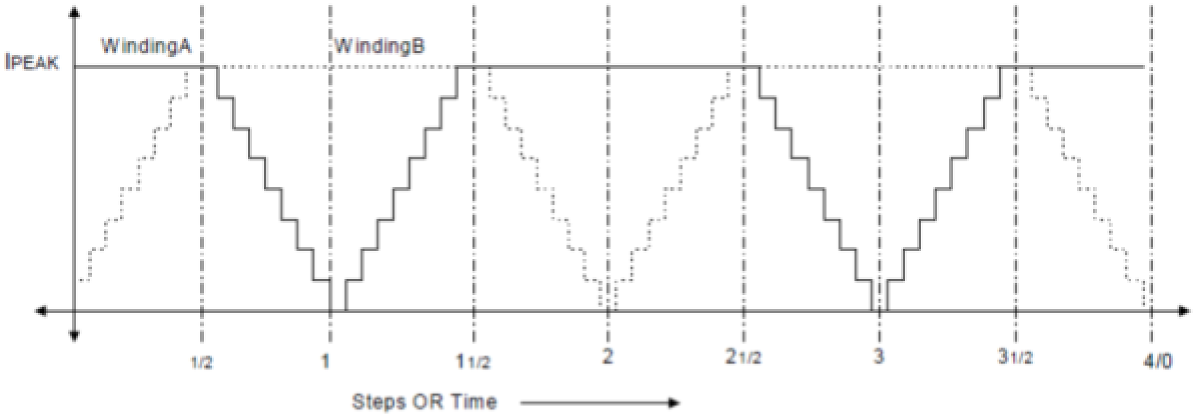

Fig. I.21 : Current flow in stator windings.

1.8.1 Look Up Table

This approach is the fastest and most flexible way to implement any arbitrary calculation.

An LUT [29] is a set of memory locations which contains some precalculated value. The memory locations of a sine LUT store duty value which are calculated using one of the following methods.
1/ Using sine mathematical relations at different angle :

$$\mathrm{LUT(n)} = \mathrm{A} * \sin\theta \tag{1.21}$$

where n is memory word length used for LUT and determines the resolution of the synthesizer. The angular resolution $\Delta\theta$ for an LUT with memory words equal to L is given by [2]

$$\Delta\theta = \theta(\mathrm{i}) - \theta(\mathrm{i}-1) = \frac{2\Pi}{\mathrm{L}} \tag{1.22}$$

where i varies from 0 to $\mathrm{L}-1$. When input clock frequency is fixed, varying L produces sinusoid of different frequency. Angle θ at any sample instance can be expressed as

$$\theta(\mathrm{i}) = \Delta\theta * \mathrm{i} \tag{1.23}$$

Substituting θ (i) into (1), we get

$$\mathrm{LUT(n)} = \mathrm{A} * \sin\frac{2\Pi}{\mathrm{L}} * \mathrm{i} \tag{1.24}$$

2/Using cosine mathematical relation 2:

$$D(step\_number) = \cos(\frac{(step\_namber * \pi)}{(number\_of\_steps + 1)}) * (2^{(bits\_rsolution)} - 1) \tag{1.25}$$

Using relation 2 [30], the following duty cycle values were calculated for a 16 microsteps per full step sequence using an 8-bit resolution PWM waveform :

Step Number	D	Step Number	D
1	251	9	24
2	238	10	70
3	217	11	114
4	188	12	154
5	154	13	188
6	114	14	217
7	70	15	238
8	24	16	251

TAB. I.8 : Duty cycle values for microstepping.

3/using software

To increase the accuracy of the stored values, it is preferable to do it in MATLAB. In case microstepping with m= 8, by typing these commands :
np=32;
A=255;
t=linspace(0,1-1/np,np) ;
sin_table = (round(sin(2*pi*t)*A))

```
0    50    98   142   180   212   236   250   255   250   236   212   180   142    98    50
```

FIG. I.22 : Generated values for microstep 8.

The generated values will be converted to hexadecimal and stored in the appropriate address.

1.8.2 Pulse Width Modulation

The GPIO device provides a method for sending digital signals to external devices. This can be useful to control devices that have basically two states : on and off. In some situations, it is useful to have the ability to turn a device on at varying levels.

One way that this can be accomplished is through pulse modulation. The basic idea is that the computer sends a stream of pulses to the device. The device acts as a low-pass filter, which averages the digital pulses into an analog voltage.

By varying the percentage of time that the pulses are high versus low, the computer can control how much average energy is sent to the device.

The percentage of time that the pulses are high versus low is known as the duty cycle. The duty cycle is calculated as below

$$Duty\,Cycle = \frac{ON\,Time}{Period} \tag{1.26}$$

The frequency of a PWM signal is constant but the time the signal remains high varies as shown in Fig. I.23.

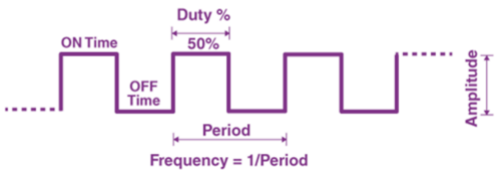

Fig. I.23 : PWM signal.

The frequency of PWM [31] determines how fast a PWM completes a period. The frequency of a pulse is shown in the figure above.
The frequency of PWM can be calculated as follows :

$$Frequency = \frac{1}{Time\,Period} \tag{1.27}$$

$$Time\,Period = \frac{1}{ON\,Time + OFF\,Time} \tag{1.28}$$

The output voltage of the PWM signal will be the percentage of the duty cycle. For example, for a 100% duty cycle, if the operating voltage is 5V then the output voltage will also be 5V. If the duty cycle is 50%, then the output voltage will be 2.5V.

If the resolution of PWM signal is n-bit, that is for a value of 0 there will be a duty cycle of 0% and for a value of (2^n) there be a duty cycle of 100%.

1.9 Field Programmable Gate Array

A field-programmable gate array (FPGA) is an integrated circuit and it belongs to a wide family of programmable logic components. An FPGA is defined as a matrix of configurable logic blocks (CLBs), linked to each other by an interconnection network, which is entirely reprogrammable [32]. The conceptual architecture of an FPGA device is shown in Fig. I.24.

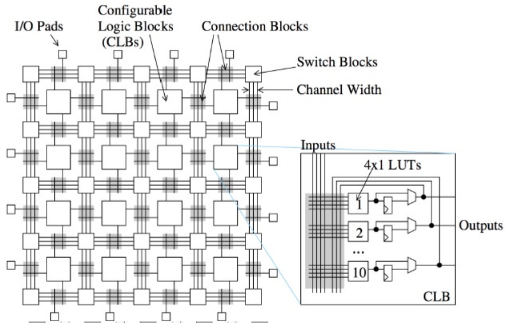

FIG. I.24 : Architecture of FPGA.

Lookup table (LUT) which contains storage cells that are used to implement a small logic function is the basic elements of CLB. If the LUT have k input then number of bits that can be configured are 2^k [33].

An FPGA can be described as ”field programmable” because it provides users the ability to modify the hardware to satisfy specific use case requirements after the manufacturing process using, for example, a hardware description language.

The first FPGA was introduced by Xilinx in 1985. Nowadays, most of the commercially available FPGAs are manufactured by Xilinx and Altera (Intel), which make up 90% of FPGAs market, but recently more companies have started developing their own FPGA boards [34].

1.10 Benefits of FPGA and its Applications

The comparison between FPGAs and modern microcontrollers [35]is one of the most debatable topics in the technological world.

Despite the fact that microcontrollers are relatively cheaper general-purpose devices that are capable of performing lots of computing, FPGAs are still a major player when it comes to extensive computing.

Microcontrollers can also be considered an ASIC implementation since they mainly offer a fixed architecture. This means that microcontrollers suffer from the same drawbacks as ASICs when compared to FPGAs. This is evident from the small number of input and output ports within Raspberry Pi and Photon. On the other side, FPGA offers an enormous number of I/O ports compared to these microcontrollers. Therefore, this is where the FPGA offers an edge over microcontrollers since it can be tailored to suit the exact needs of a specific application, unlike their counterparts which are limited by the manufactured ports and hence one can neither expand inputs and outputs if the application required more of them, nor save some resources if they are in excess.

Therefore, the major advantage of an FPGA over any ASIC or microcontroller is the fact that it allows developers to design and try out different implementations and configurations and assess their performance in the field before proceeding to the manufacturing stage which costs a tremendous amount of resources.

1.11 VHDL Coding

VHDL stands for "VHSIC (Very High-Speed Integrated Circuit) Hardware Description Language".It is a comprehensive language that allows a user to deal with design complexity.Design, and the data representing a design, are complex by the very nature of a modern digital system constructed from VLSI chip. VHDL is the first language to allow one to capture all the nuances of that complexity, and effectively manage the data and the design process. It arose out of the United States Government's Very High Speed Integrated Circuits (VHSIC) program, initiated in 1980.

In March 1986, the IEEE take on the effort of standardizing VHDL. The new language standard known us IEEE 1076. IEEE 1076 was approved in 1987.This standard is the current definition of VHDL and is the language adopted in this project [36].

1.12 DE2 Board

the Altera DE2 Educational and Development board was used in this experiment in order to check the flexible implementation with FPGA and to get the better and safely ways to use these specifications during any design implementations. The DE2 board has Altera Cyclone® II 2C35 FPGA device and Altera Serial Configuration device – EPCS16 that allow the user to implement a wide range of designed circuits, from simple circuits to various multimedia projects.
Fig. I.25 illustrates the layout of the board and indicates the location of the connectors and key components.

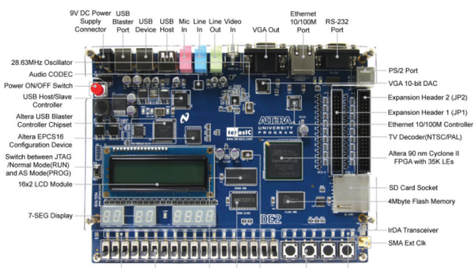

FIG. I.25 : The DE2 board.

The Altera DE2 Development and Education board's aim is provide the best means of guidance to learn about digital logic, computer architecture, and FPGAs. It exposes students and professionals to a wide range of topics by utilizing modern hardware and CAD tools. The board has an extensive set of features that make it suitable for use in a lab setting for college and university courses, as well as for a range of design projects and the creation of complex digital systems [37].

1.13 Quartus II Software

Intel Quartus II is programmable logic device design software produced by Intel ; earlier to Intel's procurement of Altera, the apparatus was called Altera Quartus Prime, prior to Altera Quartus II. Quartus Prime empowers the investigation and blends of Hardware description language (HDL) plans, which empowers the designer to compile their plans and perform timing examination.

As explained before is an FPGA programmable after the manufacturing process. The QuartusII software allows the user of the DE2 board to define the circuit in the FPGA using VHDL hardware description language or a schematic design entry method, in which a graphical diagram of the circuit can be drawn. Furthermore, the necessary pin assignments can be adjusted with the Quartus II software, to define in which way the circuit of the FPGA shall be connected to the other components on the board such as switches, LEDs or expansion headers. In the following we will briefly explain the different steps that have to be carried out for programming and configuring the FPGA on the DE2 board using the Quartus II software [39].

1.14 The RS332_082 5V Stepper Motor

This is a 4-phase Unipolar hybrid stepper motors from RS PRO are high performance DC motors, perfect for applications requiring precise movement or rotation in steps. They combine characteristics from both variable reluctance stepper motors and permanent magnet stepper motors, offering a versatile motor suitable for a range of applications.

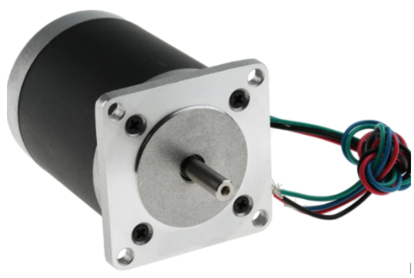

Fig. I.26 : Hybrid stepper motor.

Holding torque : 460 mNm

Stepping angle : 1.8°

Rated voltage : 5 V

1.15 The 4 Phase L298N Stepper Motor Driver PCB

The L298N integrated circuit is a high voltage (up to 46V), high current (total DC current up to 4A) dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepper motors. The block diagram of the L298N dual full-bridge driver is shown in Fig. I.27.

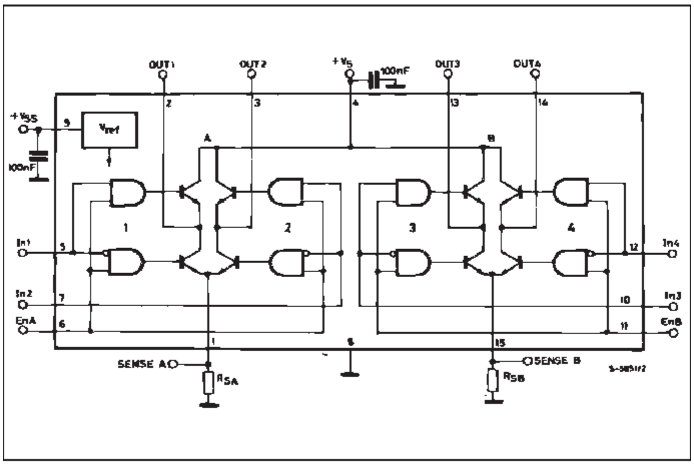

FIG. I.27 : Block diagram of L298N.

Two inhibit inputs, ENABLE A and ENABLE B, are provided to disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal, SENSE A and SENSE B, can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

This module has maximum of 5.5 V for logic control voltage, and maximum of 35 V for motor driving voltage and it is capable to supply maximum 2 Ampere of current per coil to the motor.

This is one of the most versatile of the small motor driver chips. It's designed to drive either two DC motors or one bipolar stepper motor. The L298N requires clamp diodes mounted externally for proper operation.

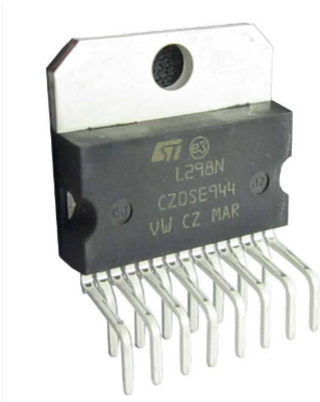

Fig. I.28 : L298N dual motor driver IC.

1.16 The SN74LS244N Three Octale Buffer

The SN74LS244N is an octal buffers and line drivers are designed specifically to improve both the performance and density of three-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters. The designer has a choice of selected combinations of inverting and non-inverting outputs, symmetrical, active-low output-control (G) inputs, and complementary output-control inputs. These devices feature high fan-out, improved fan-in, and 400-mV noise margin.

Fig. I.29 : SN74LS244N three octale buffer.

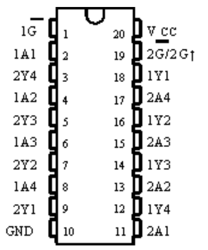

Fig. I.30 : Pin diagram of SN74LS244N.

1.17 1N4002 Diode

The 1N4002 diode belongs to the family of the 1N400x diode series, which are most commonly used in household electronic appliances. It allows the flow of current only in one direction, that is from anode terminal to cathode terminal just like a normal diode. It is referred to as a general-purpose rectifier diode used for rectification purposes. The 1N4002 diode is shown in the figure below. The grey strike on the diode is used to identify the cathode terminal.

Fig. I.31 : 1N4002 diode.

1.18 Capacitor

A bypass capacitor is just a capacitor intended to smooth out ripples (variations) and transients on the V+ power line. These bypass capacitors are usually .1 µF to .22 µF.

A capacitor is a device that resists a change in voltage. A bypass capacitor attached to your power bus will sit there at the usual Vcc voltage level. When a voltage spike appears, the capacitor will quickly absorb the energy, trying to keep the voltage constant.

When the voltage drops, the capacitor will discharge energy back into the circuit trying to keep the voltage up. So when your H-bridge spikes the V+ line, the bypass caps sitting there (usually .1 to .22 µF) will greatly reduce the effects of that current shunt.

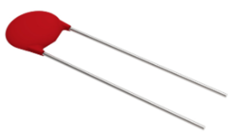

Fig. I.32 : Capacitor.

Chapitre 2

DESIGN AND IMPLEMENTATION

The microstepping mode algorithm implemented in ROM as look-up tables was used to generate the appropriate pulse width modulation (PWM) signals to control the current levels in the motor's windings. Because the current patterns in the win- dings closely resemble sine waves, we used a sinusoidal (sine/cosine) approximations function to build the look-up table to drive the motor's windings. The digital driver is developed with the Very high speed integrated circuit Hardware Descrip- tion Language (VHDL). The driver is synthesized using Quartus® II, the Intel®- FPGA software development suite tools, and targeted at an FPGA of the Cyclone-II family. Computer simulations are carried on Quartus II simulator.

2.1 The On-chip Hardware Design of Wave, Two-phase on and Half Step Controller

To visualize and compare different drive excitation mode to control stepper motor, we constructed this stepping motor drive system. Fig. II.1 depicts the block diagram of the stepper motor controller. The block diagram consists of the necessary input and output ports that will be used to communicate to and from the FPGA to generate the appropriate sequence to drive the four coils to achieve a visual representation for wave,two phase-on and half step modes.

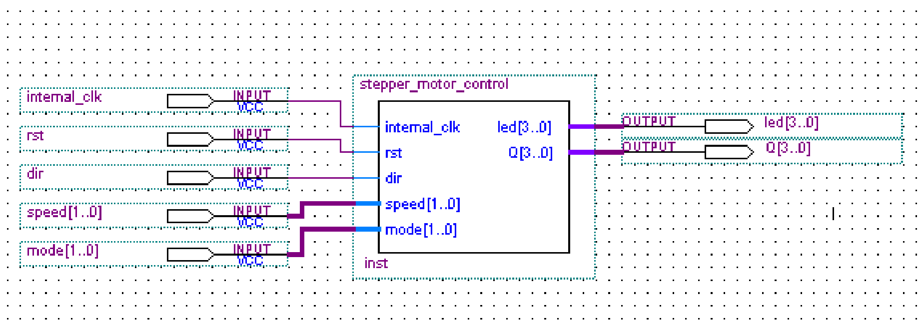

Fig. II.1 : Block diagram of stepper motor controller to drive the system in wave,two phase-on, half step modes.

we declared an entity that consist of :

```
entity stepper_motor_control is
port(
    internal_clk :in std_logic;--50 MHZ
    rst:in std_logic;
    dir : in std_logic;
    speed : in std_logic_vector(1 downto 0);
    mode : in std_logic_vector(1 downto 0);
    led :  out std_logic_vector(3 downto 0);
     Q :    out std_logic_vector(3 downto 0)
  );
  end stepper_motor_control;
```

FIG. II.2 : Entity declaration of the stepper controller that control the stepper motor in three different modes .

internal_clk takes the 50 MHZ system clock as an input. It comes from the internal oscillator of the DE2 board and it was used to control the timing of our stepper motor control sequence. rst is used to disable are enable the stepper motor. **dir** determines the direction in which the motor rotates. **speed** is used to adjust the speed of the stepper motor by controlling the frequency of at which we send step pulses to the motor driver. A clock divider was employed to configure the clock frequency to match the desired speed of the stepper motor. A higher pulse frequency results in a faster rotation, while a lower frequency results in a slower rotation. **mode** is an input signal that selects a particular operationg mode or configuration for the stepper motor. **Q** is an output ports for controlling each of the four phases of stepper motor. **led** are four leds on DE2 board that should represent the stepper motor's phases and display the pattern according to the sequence and selected mode.

The compilation of the VHDL file was successful. To test and verify the functional correctness of our design, simulation using Quartus® II simulator was done. The simulation waveforms result for the full step and half step excitation is reported in Fig. III.3 and Fig. II.4 .

Fig. II.3 depicts the pulse generated at each rising edge of the clock input. This pulses are the drive enable pulses Q[0], Q[1], Q[2] and Q[3] corresponds the stator pahse in programmed sequence. The pulses Q[0], Q[1], Q[2] and Q[3] are the simulation results obtained from simulator for clockwise rotation of motor. In order to reverse the direction of rotation we reverse the sequence of the pattern in the opposite direction. If the sequence is 1000, 0100, 0010, 0001 we switch it to 0001, 0010, 0100, 1000. As the stator switches the excitation phase in the time sequence, the stable position of the rotor gradually moves from left to right and then drives the rotor to rotate.

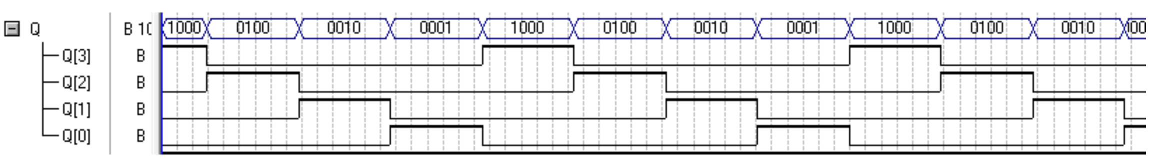

FIG. II.3 : Timing Sequence Diagram of a Stepping Motor Driven in The Full-step Mode

In Fig. II.4 the sequence of pulses was programmed to generate the half step excitation sequence. The drive enable pulses Q[0], Q[1], Q[2] and Q[3] are the simulation results obtained from simulator for clockwise rotation of motor. In order to reverse the direction of rotation we reverse the sequence of the pattern in the opposite direction. In our architecture the sequence was 1100, 0100, 0110, 0010, 0011, 0001, 1001, 1000 we switch it to 0011, 0010, 0110, 0100, 1100, 1000, 1001, 0001.

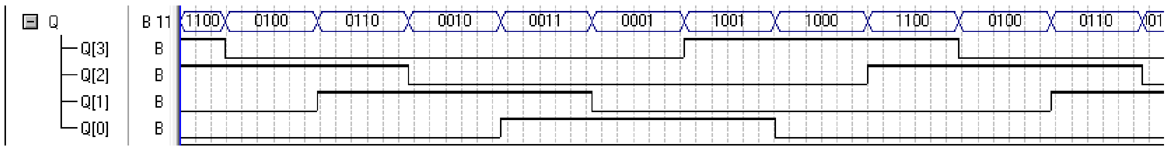

FIG. II.4 : Timing Sequence Diagram of a Stepping Motor Driven in The half-step Mode

After the simulation process, we assigned the selected pins as inputs or outputs according to design requirements.

2.2 The On-chip Hardware Design of Microstepping Controller

In VHDL there is no native support for trigonometric functions like sine and cosine, or for other complex math such as hyperbolic functions, square roots, etc. Calculations like these in digital logic can be slow and resource-hungry, and FPGA designers have different approaches for performing the calculations quickly and/or reducing the amount of necessary logic. Fig. II.5 illustrate the top-level file of the most important part in this project which is the microstepping controller.

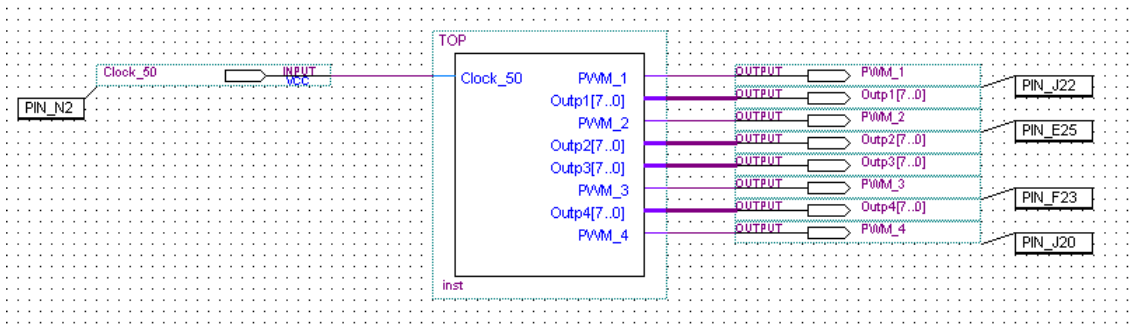

FIG. II.5 : The block diagram representation of the system.

In this figure, Clock_50 is used as an input port for the system clock signal.
PWM are four output ports to generate the pulse width modulation signal to control the amount of current in each phase. Outp are four 8-bit output ports , To provide values of n-resolution sine wave.

The TOP block diagram contains many other blocks diagram inside it, each representing an essential part of the microstepping controller. Each of them will be discussed deeply in the following subsection. The TOP level structure is the most important module that cotains all modules in the design. All the signals of comming modules will be mapped in the entity of TOP and assigned the input or output value based on the dataflow diagram.

2.2.1 ROM Module

The microstepping mode algorithm was designed as lookup tables(LUT's) implemented in read only memory(ROM) block. The count selects an address in the block, and the data contents at that address are the output. The process is explained step by step in the coming discussion. Fig. II.7 illustrate the block diagram of the ROM.

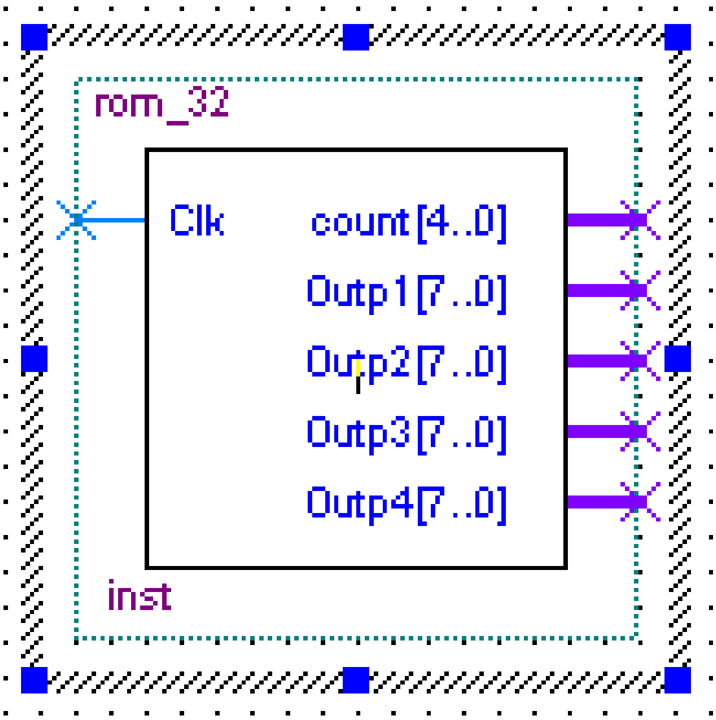

Fig. II.6 : Block diagram of ROM.

As discussed in chapter 1, ROM stores and retrieves data based on the value of an address signal. Storing data using addresses is refers to look-up tables. The Clk is used as an input clock signal to control the timing and operation of the ROM. For this purpose, a special module is used to make the necessary scaling.

The obtained values usnig method 3 in chapter 1 are converted into suitable 8 bits binary numbers and stored in look-up tables for each phase. The up counter is used for each LUT's addressing. Count is an input port that indicates the current address, ranging from 0 to N-1.

N is the number of addresses used to store the values. It depends on the used resolution m. In the case of m=2, each of LUT's have 8 addresses. For m=4, each of LUT's have 16 addresses. For m=8, each of LUT's have 32 addresses. For m=16, each of LUT's have 64 addresses.

The digital architecture of the stepper motor controller is going through simulation process before implemented into physical experiment. The simulation is to inspect the functionality of the stepper motor controller with a given input value. The literal text and analog waveform display of the outputs for m=8 can be shown as follow in Fig. II.7 and Fig. II.8, respectively. Fig. II.7 shows the waveform obtained from the simulator tool in Quartus 2 package. This waveform is in literal text display format.

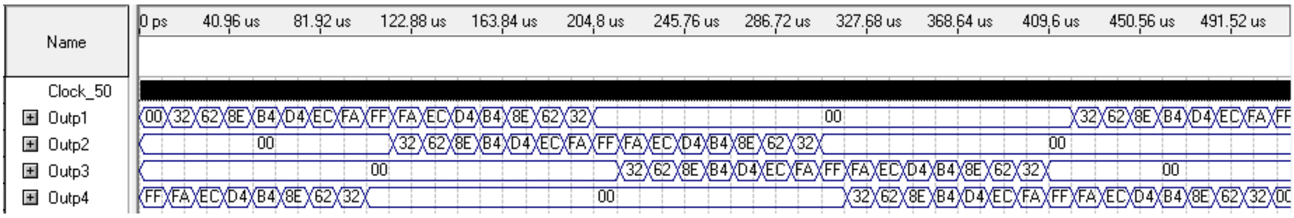

FIG. II.7 : Simulation result of microstep controller for m=8.

Fig. II.8 shows the outputs in analog waveform display format with step display style.

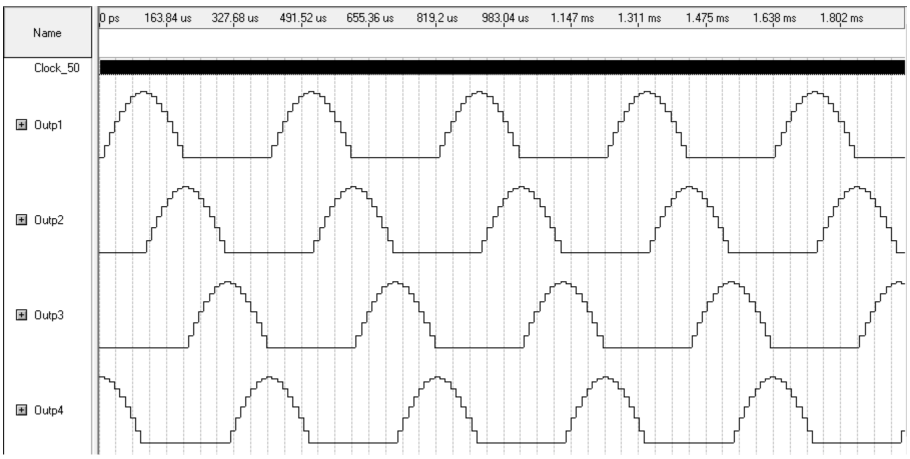

FIG. II.8 : analog waveform display format with step display style.

When driving stepper motors with full steps the output of the stepper motor driver looks like a square signal and produces rough movements. The bigger the micro-stepping the more the output signal looks like a sine wave and the stepper motor moves more smoothly. But there is a downside to this. With increasing micro-stepping value the torque drops a quite lot and if the value is too great it could happen that the motor can't produce enough torque to even turn. Usually 1/4, 1/8 or even 1/16 can produce satisfactory smooth movements while still producing enough torque.

2.2.2 PWM Module

a pulse width modulated signal is a square wave signal that has different duty cycles. It is a technique to provide a logic "1" and logic "0" for a controlled period of time. It is a signal source involves the modulation of its duty cycle to control the amount of power sent to a load [40] .

Fig. II.9 show the top level file of PWM Module.

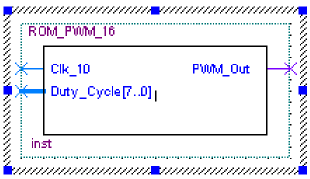

Fig. II.9 : Block Diagram of PWM.

In this project an 8 bit PWM resolution of a VHDL description is used. We start the description of the block diagram by specifying its external interface, which includes a description of its ports.
Clk_10 is an input port to control the timing and operation of the PWM. For this purpose, a special module is used to make the necessary scaling. **Duty_Cycle** is an output port from the ROM used as an input port in this design. **PWM_oUT** is an output port made to control the H-bridge circuits.

An 8 bit data register is used to store the data value. The data values that stores in the register will determine the duty cycle of the pulse width modulation output. An 8 bit up and down counter. Counts up or down based on the pulse width modulation **PWM_Out** signal and generates terminal count whenever counter reaches the maximum value or when it transists through zero .terminal count is used to automatically load the data value to generate different pulse width modulation out with different duty cycle. The duty cycle is computed using the formula :

$$Duty\,Cycle = \frac{Intger\,value\,of\,8 - bit\,word}{period} \tag{2.1}$$

The PWM period is declared as a constant. Since the PWM resolution is 8 bit, then the period is set to 255 which is the maximum 8 bit value. Thus the input duty can take values from 0 to 255. Thus duty cycle of the PWM signal can be varied from 0 to (255/256)*100 = 99.6 %.

In the Vhdl file of PWM, there are two processes to deal with. The first process is sensitive to Clk_10 signal. It checks whether the count has reached the end of the period. If so, it resets count to zero. Otherwise, it increments the count by 1.

The second process is sensitive to changes in the count signal. It compares the count with Duty_Cycle value. If the count is less than the Duty_Cycle, it sets PWM_Out to 1, indicating the ON state of the PWM signal. Otherwise, it sets PWM_Out to 0, indicating the OFF state of the PWM signal.

To summarize, the PWM module is used to generate a PWM signal on PWM_Out where the duty cycle is controlled by the Duty_Cycle input. The duty cycle determines the proportion of time the PWM signal is in the ON state within one period.
This PWM signal is used to control the stepper motor in microstepping mode, where the duty cycle corresponds to the microstepping level. The motor can be controlled based on the state of the PWM signal.

The resulting simulations for 8-bit resolution of PWM generation unit for different duty cycle is shown below in Fig. II.10 to Fig. II.11.

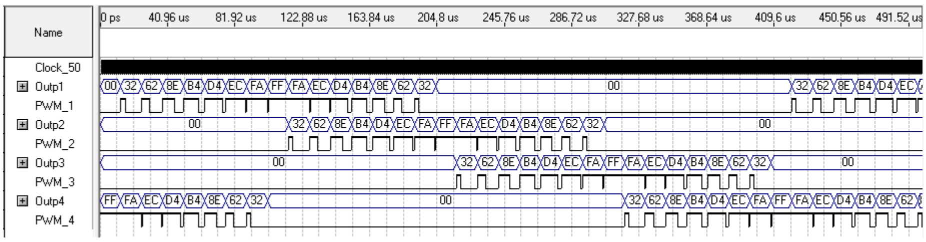

FIG. II.10 : Simulation result of designed system for m=8 in literal text display format.

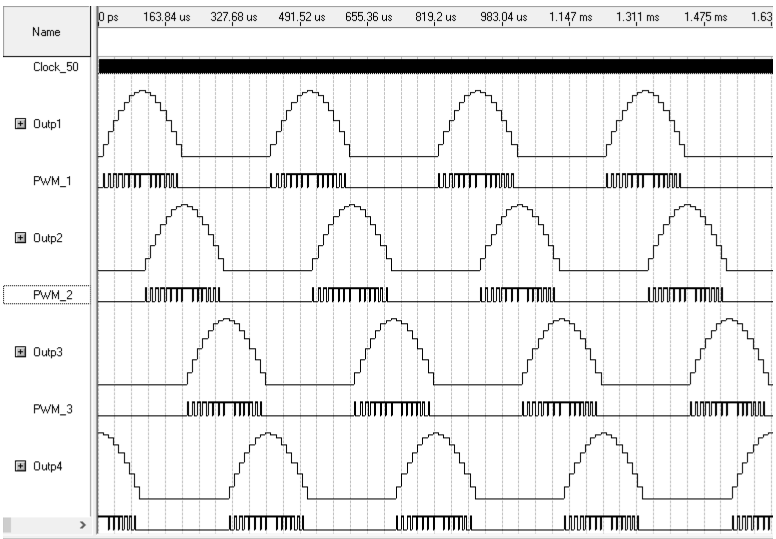

FIG. II.11 : Simulation result of designed system for m=8 in analog waveform display format.

When we zoom in Outp1 in Fig. II.12, we obtain :

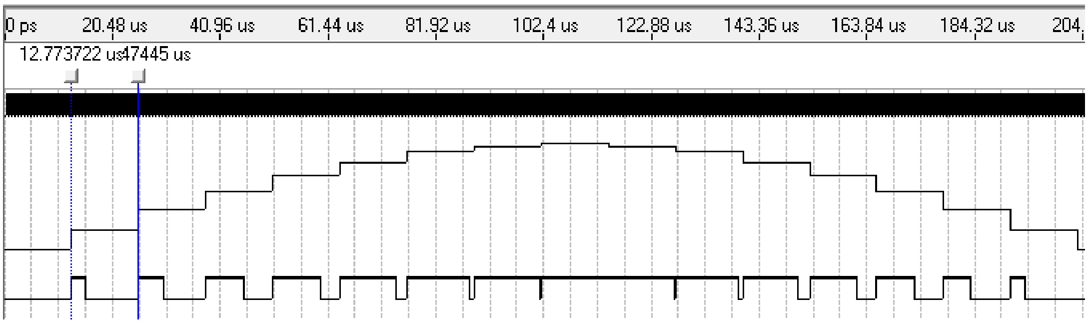

FIG. II.12 : Switch cycles for m=8.

This waveform is the analog waveform display with step display style for microstepping =8. For m=8, we must have 16 word. Thus, the waveform consists of 16 switching cycles. According to the graph, the time interval for 16 switching cycles is 204µs.

To get periodic time of switching cycle, we divided 204 s by 16 we get 12.75µs.

The periodic time for each switching cycle is the PWM period T_{PWM}. The PWM frequency is 1 over T_{PWM} wich equals to 78 KHZ.

the stored value in ROM at this interval is 32 in hexadecimal which represents 50 in decimal. The period is 255. Using equation (2.1) :

$$dutycycle = \frac{50}{255} = 19.6\%$$

The rotation of the motor depends on the number of pulses sent to the motor and the speed can be controlled by changing the frequency of the pulse signal.
The formula for the speed of step motor and used in the code :

$$Count = \frac{main\,clock}{desired\,frequency} - 1 \qquad (2.2)$$

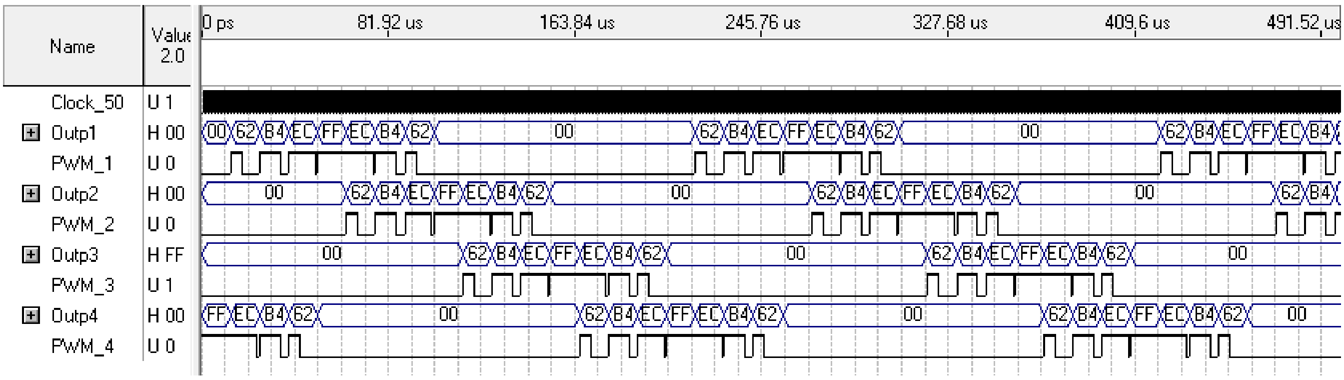

FIG. II.13 : Simulation result of designed system for m=4 in literal text display format.

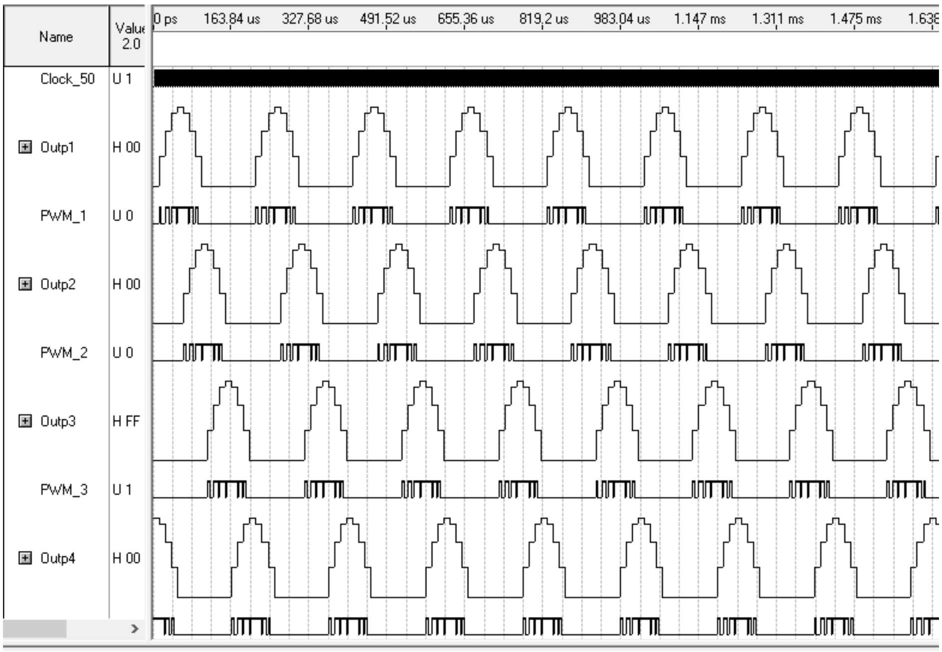

FIG. II.14 : Simulation result of designed system for m=4 in analog waveform display format.

2.3 the Off-chip Hardware Design Of Stepper Motor Controller

This section introduces the step motor driver circuit, which is designed and implemented as part of this project. This driver performs the bipolar micro-step excitation of four-phase hybrid step motors.

Generation and timing of electrical pulses to generate motor steps are accomplished in a functional unit which is called the digital control of the driver. These pulses are passed to IC driver, which shift the level of electrical inputs to operate a set of switches inside the full-bridge converters introduced in Chapter 1. The full-bridge circuit drives the current for the motor windings according to electrical pulse inputs.

The whole driver system is introduced in the first section to provide a general discussion on the problem. The section continues with the full-bridge converter in the second section, with the IC driver in the third section and with the digital control counterpart in the fourth section.

To evaluate and compare the benefits of microstepping, we constructed a stepping motor drive system, installed the system with an algorithm based on the proposed control method.

The stepping motor drive circuit is implemented on project board as seen in Fig. II.15. Also, an accompanying test setup composed of an ALTERA DE2 development and educational board.The driver is synthesized using Quartus® II, the Intel®-FPGA software development suite tools, and targeted at an FPGA of the Cyclone-II family.

four-phase unipolar hybrid step motors is used in the tests. The name of this motor is stepper motors Type 23 (RS stock no. 332-082). It offers a 1.8 degree step angle, holding torque with 460 mNM, detent torque with 30 mNm, resistance with 5 ohm, inductance with 10 mH and current with 1 mA for each phase.

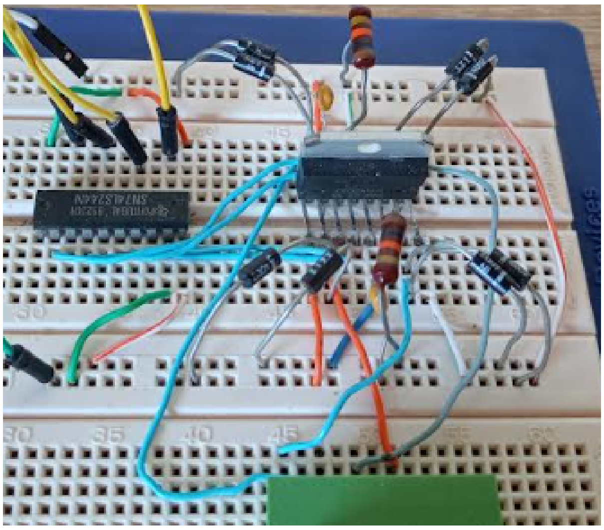

FIG. II.15 : Implemented step motor driver on project board.

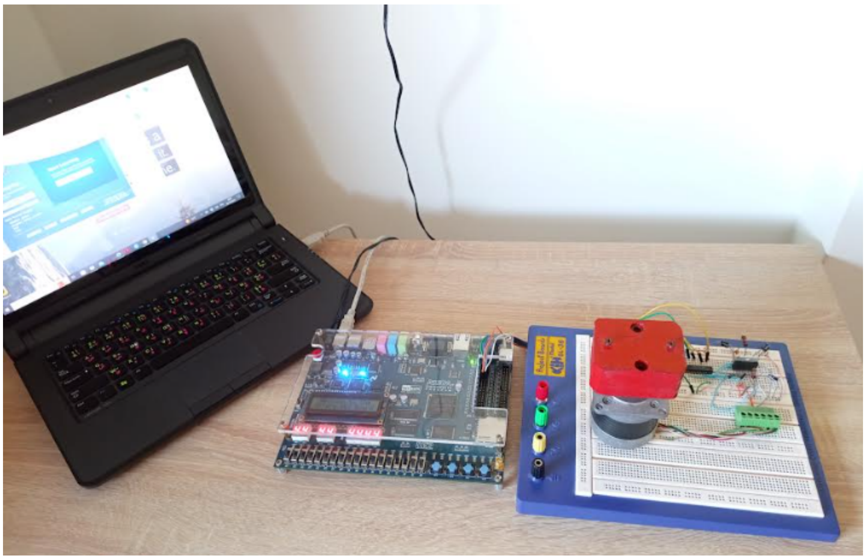

FIG. II.16 : Experimental setup : step motor on project board, driver in the right and the DE2 board in the right of PC controller interface which is seen in the left.

2.3.1 Stepper Motor Drive Circuit design

AC power and DC power can't drive stepper motor due to the special character of stepper motor. To attain the normal working state of stepper motor special equipment is needed. Therefore, SN74LS244N and L298N is used in the driven module to make the motor working.

L298N is applicable to high voltage, large current motor driver and belongs to productions of ST, it depend on two independent of H full bridge driver. It is dedicated to bipolar stepper motor.

In order to protect the outputs of the FPGA from the effects of voltage and a transistor current that are too high, the SN74LS244N was used. The SN74LS244N serves as an intermediary between the low-level digital signals from the FPGA and the inputs of the L298N motor driver, ensuring signal integrity, compatibility, and reliability. It can help protect the FPGA from voltage spikes or other electrical noise that may be generated by the motor driver and its connected motors.

The two kinds chips above are suitable for two phase bipolar and unipolar four phase stepper motor either. When the FPGA assigned direction and clock signal, driver module will receive commands, SN74LS244N pulse distributor will generate excitatory phase sequence automatically. Signal that can drive stepper motor will be produced by L298N's power amplifier. The advantage of the cooperation of L298N and SN74LS244N is that has low assembly costs and high reliability. So, the driven circuit is easy to control and design.

Each bridge in L298N is driven by means of four gates the input of which are In1 ; In2 ; EnA and In3 ; In4 ; EnB. The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

A non inductive capacitor, usually of 100 nF, must be foreseen between both Vs and Vss, to ground, as near as possible to GND pin. An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Shottky diodes would be preferred. As shown in the previous section, the compilation of our vhdl file was successful and the desired output was determined by the result of the simulation.

Next step is to assign the four output ports used to control the phases of stepper motor to the corresponding pins of the FPGA development board. The SN74LS244N operates on a 5V supply voltage. We Connected the VCC (Pin 20) of the SN74LS244N to the 5V power supply. The GND (Pin 10) and output enable 1G (Pin 1) of the SN74LS244N are connected to the ground (0V) of your system.

The SN74LS244N has 4 input lines(A1, A2, A3 and A4) and 4 output lines (Y1, Y2, Y3 and Y4). To connect FPGA with SN74LS244N, we connected the desired FPGA port to the corresponding input of the SN74LS244N. The 4 output lines (Y1, Y2, Y3 and Y4) are connected to the corresponding inputs of the L298N. The 4 output lines (Y1, Y2, Y3 and Y4) of L298N are connected to the corresponding inputs of the L298N. Fig. II.18 depicts the schematic diagram of a part of Stepper Motor Drive Circuit design used to drive a four-phase stepper motor.

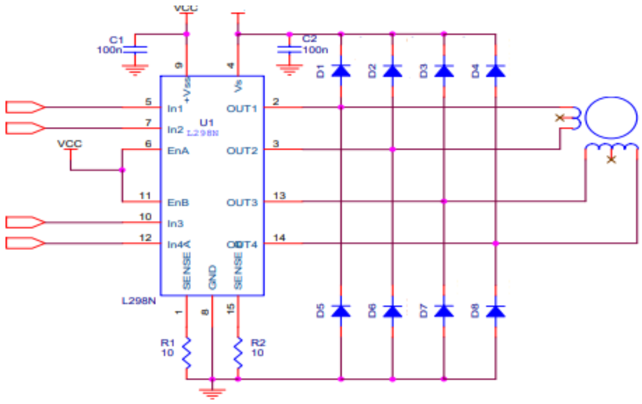

FIG. II.17 : Shematic diagram of L298N driver to control a four-phase bipolar stepper motor.

The driver implemented in this project uses PWM technique for driving the phase winding currents. PWM signals control the winding excitation of each phase where the duty cycle is proportional to the absolute value of the current waveform at a single micro-step. A full bridge circuit for each phase ensures that during the on state of the PWM, phase windings are energized and during the off state phase currents decay The FPGA generates PWM outputs which are generated by using the power control PWM module which simplifies generation of the needed gate signals for half-bridge switches.

Four pulse width modulation output ports of the FPGA, namely PWM_1, PWM_2, PWM_3 and PWM_4 were used to control a dual-H-bridge drive circuit. The two H-bridges drove the four phases of the stepper motor.

By driving the switches of each half-bridge of L298N driver with respective PWM signals, the duty-cycle of which varies with time according to a modulating signal.
The duty cycle of PWM pulse is provided by FPGA and is exerted to L298N driver input to control the switching process.

In this project we tested the funcionallity of the system using 1.8 four phase hybrid stepper motor with maximum speed 75 rpm at low frequencies which means that speed was low to visualize the different resolutions of stepper motor. In the stepper controller that can drive the stepper in wave, two-phase on and half step, we used a switch that can control stepper motor in four different speeds. For example if the frequency is 3HZ, 12HZ, 47HZ, 190HZ the speeds will be 0.9 rpm, 3.6 rpm 14 rpm, 57 rpm, respectively.

In stepper motor that uses microstepping controller for m=8, if we select the frequency at 4 hz the switching cycle for each data word will be 16 ms which is the PWM period T_{PWM}. The PWM frequency is 1 over T_{PWM} which equals to 62 HZ.

CONCLUSION

Implementing controls in an FPGA simplifies the system at a hardware level and allows scaling and extending it much more easily than if it was made with real hardware. A notable advantage is the fact that an FPGA directly controls signals instead of bits, so that unlike a microcontroller, the delays are much smaller.

Basic principles of stepper motor, and there different types and comparison of stepper motor are presented. Various modes of excitation namely full step, half step and micro step have been discussed.

The project Implementation and verification of a hardware-based controller for a bipolar stepper motor on an FPGA collects the result of a period of documentation and research in the field of stepper motor, especially regarding their control methods. a brief overview of FPGAs and stepper motors is also done. practical development of the work, the hardware description is made using ROM as LUT's and PWM signals was generated to control the current levels in the motor's windings. The microstepping mode algorithm have been developed, implemented, simulated and successfully verified with the VHDL language in Quartus II.

The driver implemented in this thesis uses PWM for driving the phase winding currents. PWM signals control the winding excitation of each phase where the duty cycle is proportional to the absolute value of the current waveform at a single microstep. A fullbridge circuit for each phase ensures that during the on state of the PWM, phase windings are energized and during the off state phase currents decay.

As a general conclusion, I consider the purposes of the project achieved and I want to show my personal satisfaction of having been able to enjoy, both in studying and working on the project.

Futur Work

Rapid extension and enhancement of HDL designs is facilitated by the flexibility and power of FPGAs. These paragraphs include descriptions of a few of the suggested enhancements.

One possible development of the project for the stepper motor control system would be to convert from an open-loop to a closed-loop version, monitoring the motor's phase current or connecting an encoder to the shaft. The effective torque of the motor, or the amplitude of the sine signals, could be changed by other components if these variables are able to be controlled independently of frequency. This would presumably result in increased performance, particularly from the motor, which would become dependable at lower speeds and with varying loads.

We can also improve the design's flexibility and modularity to facilitate future updates and simple interface with other control systems.

Optimizing the microstepping controller for high speeds and use higher-resolution encoders or sensors to provide more accurate feedback, allowing the controller to better keep track of the motor's position at high speeds.

Lastly but not least, by implementing error correction mechanisms to account for missed steps or position errors at high speeds and implement safety features, including emergency stop mechanisms, to ensure safe operation even at high speeds.

REFERENCES

[1] W. McClelland, "The application of the electricity in warships," Journal of the Instutition of Electrical Engineers, vol. 65, no. 369, pp. 866 - 869, September 1927.

[2] Kenjo Takashi, Stepping motors and their microprocessor controls. Kanagawa, Japan: Clarendon Press, 1984.

[3] C. L. Walker, "Improvements in and connected with electromagnetic step by step signaling and syncronous rotation," 137,150, 191.

[4] K. M. Feiertag and J.T. Donahoo, "Dynamoelectric machine," 2589999, 1952.

[5] Allegro, "A3981, Automotive, Programmable Stepper Driver," Data Sheet 2010.

[6] Allegro, "A4980, Automotive, Programmable Stepper Driver," Data Sheet 2010.

[7] ON Semiconductor, "AMIS-30523 CAN micro-stepping motor driver," Data Sheet.

[8] Ngoc Quy Le and Jae Wook Jeon, "An open-loop stepper motor driver based on fpga," in Proccedings of the International Conference on Control, Automation and Systems, Seoul, Korea, 2007, pp. 1322-1326.

[9] Sorin Manea, "Stepper Motor Control with dsPIC® DSCs," Microchip Technology Inc., Application Note 2009.

[10] J. T. Humphries, “Motors and controls: Humphries, James T., 1946- : Free download, Borrow, and streaming,” Internet Archive, https://archive.org/details/motorscontrols0000hump/page/86/mode/2up?q=motor (accessed Sep. 4, 2023).

[11] kissell thomas E, “Industrial Electronics: Applications for Programmable Controllers, instrumentation and process control, and electrical machines and motor controls: Kissell, Thomas E: Free download, Borrow, and streaming,” Internet Archive,

https://archive.org/details/industrialelectr0000kiss/page/544/mode/2up?q=stepper (accessed Sep. 6, 2023).

[12] glenn d. considine, "Van Nostrand's scientific encyclopedia: Free download, borrow, and streaming," Internet Archive, https://archive.org/details/vannostrandsscie0003unse/page/5208/mode/2up?view=theater&q=bifilar%2Bwindings (accessed Sep. 6, 2023).

[13] L. Przenioslo and M. Holub, "Development of microprocessor, time optimized stepper motor driving algorithm," International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 174-179, 2017.

[14] G. J. Rockis and glen A. Mazur, "DC Motors," in *Electrical motor controls for integrated systems*, 5th ed, vol. 769, Orland Park, pp. 282–283

[15] M. K. Petr Olivka, PWM nonlinearity reduction in Microstepping Unit Firmware, https://www.semanticscholar.org/paper/PWM-Nonlinearity-Reduction-in-Microstepping-Unit-Krumnikl-Olivka/4d3887daddcaca1e48381fac3d6258b6d3adf169 (accessed Oct. 28, 2023).

[16] P. Yedamale and S. Chattopadhyay, "AN822, Stepper motor microstepping with Pic18c452 - Microchip Technology," microchip, https://ww1.microchip.com/downloads/en/AppNotes/00822a.pdf (accessed Aug. 30, 2023).

[17] Akdogan, E., Topuz, V., Akbas, A., An education tool study on mechatronics : emulation of stepper motor driving systems by using a microcontroller based system interface, Mechatronics, 2004. ICM '04. Proceedings of the IEEE International Conference on, ISBN : 0-7803-8599-3

[18] G. Tekeste, "Chapter 2 Theory of Stepper Motor," Academia.edu, https://www.academia.edu/37092827/CHAPTER_2_THEORY_OF_STEPPER_MOTOR (accessed Oct. 28, 2023).

[19] RD class / ac machin es dr. Inaam Ibrahim - University of Technology, Iraq, https://www.uotechnology.edu.iq/dep-eee/lectures/3rd/Communication/machine/PART%201.pdf (accessed Oct. 28, 2023).

[20] "Stepper Motor Maximum Speed and power calculator," CalcTown, https://www.calctown.com/calculators/stepper-motor-maximum-speed-and-power-calculator (accessed Oct. 28, 2023).

[21] Stepper Motor Calculations | GlobalSpec, https://www.globalspec.com/pfdetail/motors/stepper-motor-calculations (accessed Oct.28, 2023).

[22] D. CLARK and M. OWINGS, "Building Robot Drive Trains : Clark, dennis : Free download, borrow, and streaming," Internet Archive, https://archive.org/details/buildingrobotdri0000clar (accessed Aug. 30, 2023).

[23] david allan bradley, *Power Electronics* , 2nd ed, 224 Vols. london: chapman & hall, 1995.

[24] "Unifilar Motor winding," Unifilar Motor Winding - Electronic Stepper Motor Tutorial - Circuits and Hobby Projects, https://www.hobbyprojects.com/stepper_motors/unifilar_motor_winding.html?no_redirect=true#google_vignette (accessed Oct. 29, 2023).

[25] Bipolar Hybrid Stepper Motor Control and drive with dedicated ICS L297 ..., https://doaj.org/article/636c8296e43146f4b750a39ce80e72f5 (accessed Oct. 28, 2023).

[26] Person, "How to drive a Stepper Motor," Key, https://forum.digikey.com/t/how-to-drive-a-stepper-motor/13412 (accessed Oct. 29, 2023).

[27] Muhammed F. Rahman and Aun-Neow Poo, "An application oriented test procedure for designing microstepping step motor controllers," IEEE Transactions on Industrial Electronics, vol. 35, no. 4, pp. 542-546, November 1988.

[28] Kenjo Takashi, Stepping motors and their microprocessor controls. Kanagawa, Japan: Clarendon Press, 1984.

[29] B. K. Upadhyaya, "Design and Development of VHDL Based Tuneable Sine Wave Generator Using DDS Technique: Extension to FPGA Implementation", NCC 2009, January 16-18, IIT Guwahati.

[30] R. Condit, Stepper Motor Control Using the PIC16F684, Technology Inc., 2004.

[31] https://ymerdigital.com/uploads/YMER2112DT.pdf

[32] E. Monmasson and M. Cirstea, ‘‘FPGA design methodology for industrial control systems A
review,’’ IEEE Trans. Ind. Electron., vol. 54, no. 4, pp. 1824–1842, Aug. 2007.

[33] S. Gandhare and B. Karthikeyan, "Survey on FPGA Architecture and Recent Applications," *IEEE Xplore*, Mar. 01, 2019. https://ieeexplore.ieee.org/abstract/document/8899550 (accessed Mar. 28, 2023).

[34] M. Cases Freixene, Implementation and verification of a hardware -based controller for a ..., https://upcommons.upc.edu/bitstream/handle/2117/134233/master.pdf?sequence=1 (accessed Sep. 19, 2023).

[35] M. Elnawawy, A. Farhan, A. A. Nabulsi, A. R. Al-Ali and A. Sagahyroon, "Role of FPGA in internet of things applications | IEEE conference ...," IEEE, https://ieeexplore.ieee.org/abstract/document/9001747/ (accessed Sep. 29, 2023).

[36] R. Lipsett, C. F. Schaefer, and C. Ussery, *VHDL: Hardware Description and Design*. Springer Science & Business Media, 2012.

[37] Altera, "DE2 Development and Education Board User Manual".

[38] [1] Dinh Bao Tuan Phan, CONTROL STEPPER MOTOR BY USING FPGA/VHDL-theseus, https://www.theseus.fi/bitstream/handle/10024/801476/Nilsone_Amanda.pdf?sequence=2 (accessed Sep. 28, 2023).

[39] ftp://ftp.altera.com/up/pub/Altera_Material/QII_9.0/Digital_Logic/DE2/Tutorials/
tut_quartus_intro_schem.pdf [2010-07-23]

[40] Angel Vpeterchev, "Digital pulse width modulation in power electronic circuits," July2002.