CONTROLLING OF A MECHANICAL SYSTEM HAVING THREE DIMENSIONAL MOVING CAPABILITY BY USING SERVO MOTORS, FPGA AND A PERSONAL COMPUTER

M. Sc. THESIS

IN

ELECTRICS AND ELECTRONICS ENGINEERING

UNIVERSITY OF GAZIANTEP

BY

ZAINAB MAHMOOD KHUDHUR

FEBRUARY 2016

Controlling of a Mechanical System Having Three Dimensional Moving Capability by Using Servo Motors, FPGA and a Personal Computer

M.Sc. Thesis

in

Electrics and Electronics Engineering

University of Gaziantep

Supervisor

Assoc. Prof. Dr. Vedat Mehmet KARSLI

by

Zainab Mahmood KHUDHUR

February 2016

REPUBLIC OF TURKEY
UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES
ELICTRIC AND ELECTRONIC ENGINEERING

Name of the thesis: Controlling of a Mechanical System Having Three Dimensional Moving Capability by Using Servo Motors, FPGA and a Personal Computer

Name of the student: Zainab Mahmood KHUDHUR

Exam date: 03/02/2016

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Metin BEDIR

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ergun ERCELEBI

Head of Department

This is to certify that we have read this thesis and that in our consensus/majority opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Vedat Mehmet KARSLI

Supervisor

Examining Committee Members	Signature
Assoc.Prof. Dr. Vedat Mehmet KARSLI	
Prof. Dr. Rauf MİRZABABAYEV	
Prof. Dr. Safer ALHILAL	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Zainab Mahmood KHUDHUR

ABSTRACT

CONTROLLING OF A MECHANICAL SYSTEM HAVING THREE DIMENSIONAL MOVING CAPABILITY BY USING SERVO MOTORS, FPGA AND A PERSONAL COMPUTER

KHUDHUR, Zainab Mahmood

M.Sc. in Electrical –Electronics Engineering

Supervisor: Assoc. Prof. Dr. Vedat Mehmet KARSLI

February 2016, 73 pages

CNC machines are controlled by a computer mounted on CNC. Fast interface tools are required for the synchronization between the PC and servo motor drivers. That is one of the important problems of CNC technology. The control of system based on FPGA together with a computer will contribute to studies on the subject and will increase the level of information on this subject.

The major goal of automation in manufacturing facilities is to integrate various operations in order to improve the productivity, increase the product quality and uniformity, minimize the cycle times and effort, and reduce the labor costs. Since 1940s, automation has accelerated because of the rapid advances in control systems for machines and in computer technology. CNC machines are imported mostly in our country. The studies on this subject will contribute to increase the national technology so the economy. This study will contribute to increase the speed, flexibility and control knowledge of the CNC machines in our country.

The communication between the computer and the FPGA is provided. The data is processed on the computer with a graphical program. This data can display on the computer screen in both graphical and numerical. These data could be used for statistical analysis. The selected FPGA is connected to the computer and operate together to generate required control signals for the servo motors. Controlling of the three axes of servo motor smoothly and flexibility using FPGA is achieved in this study.

Key words: *Servo Motor, FPGA, SCADA, Axis control*

ÖZET

ÜÇ BOYUTLU HAREKET EDEBILEN MEKANIK SISTEMIN SERVO MOTOR, FPGA VE KIŞISEL BILGISAYAR KULLANILARAK KONTROLÜ

KHUDHUR, Zainab Mahmood

Yüksek Lisans Tezi, Elektrik-Elektronik Mühendisliği

Danışman: Doç. Dr. Vedat Mehmet KARSLI

subat 2016, 73 sayfa

CNC makinaları CNC üzerine monte edilmiş bir bilgisayar tarafından kontrol edilir. Servo motor sürücüleri ve PC arasında senkronizasyon sağlanması için hızlı arayüze sahip cihazlar gerekmektedir. Bu durum CNC teknolojisinin en önemli problemlerinden biridir. Bilgisayarla birlikte FPGA'ya dayalı sistem kontrolü konuyla ilgili çalışmalara katkı sağlayacaktır ve bu konu hakkında bilgi düzeyini artıracaktır.
Üretim tesislerinde otomasyonun ana hedefi; verimliliği artırmak, ürün kalitesini ve tekdüzeliği artırmak, çevrim sürelerini ve denemelerini minimize etmek ve işgücü maliyetlerini azaltmak için çeşitli işlemleri birbirlerine entegre etmektir. 1940'lardan bu yana, otomasyon, bilgisayar teknolojisinde ve makinelerde yer alan kontrol sistemlerindeki hızlı gelişmelerden dolayı hızlanmıştır. CNC makinaları ülkemizde çoğunlukla ithal edilmektedir. Bu konudaki çalışmalar ulusal teknoloji ekonomisinin artması için katkı sağlayacaktır. Bu çalışma; ülkemizdeki CNC makinalarının hız, esneklik ve kontrol bilgisinin artmasına katkı sağlayacaktır.

Bilgisayar ve FPGA arasındaki iletişim sağlanacaktır. Veriler grafik programı kullanılarak bilgisayar üzerinden işlenir. Bu veriler bilgisayar ekranından hem grafiksel hem de numerik olarak görüntülenebilir. Bu veriler istatiksel analiz için kullanılabilir. Seçilen FPGA bilgisayara bağlanır ve servo motorlar için gereken kontrol sinyallerini üretmek için bilgisayarla birlikte çalıştırılır. Bu salışnada üç eksen servo motor kontrolü yumuşak ve esnek bir şekilde FPGA kullanılarak yapılmıştır.

Anahtar Kelimeler: *Servo Motor, FPGA, SCADA, Eksen kontrolü*

To My Parents

ACKNOWLEDGMENTS

First and most importantly I must thank the Most Merciful, Almighty ALLAH for making everything possible.

I would like to thank my supervisor Doç. Dr. Vedat Mehmet KARSLI for his support and guidance throughout the study.

Also I would like to express my sincere thanks to all member of my family for supporting me in every situation.

I would like to provide special thanks to my father Mr. Mahmood KHUDHUR for providing moral and intellectual support to me in my study.

I would like to express my thank Mr. Mehemet Taner KARSLI from university of Harran for helping me in practical work and taking his thesis as a mainly reference in this study.

And I would like to thank Mr. Rizgar Ahmed from university of Sulaimany for helping me in this study.

CONTENTS

LIST OF FIGURES

LIST OF TABLE

LIST OF SYMBOLS/ABREVIATIONS

CNC : Computer based Numeric Control

FPGA : Field Programmable Gate Array

IC : Internal Circuit

SCADA : Supervisory Control and Data Acquisition

EDA : Electronic Design Automation

DC : Direct Current

AC : Alternative Current

PID : Proportional Integral Derivative (feedback control)

TF : Transfer Function

DAC : Digital Analog Converter

PC : Personnel computer

PMDC : Permanent magnet direct current

DTC : Direct Torque Control

DSP : Digital Signal Processor

ASIC : Application Specified Integrated Circuit

MOSFET : metal-oxide-semiconductor field effect transistor

J : moment of inertia of the rotor

b : damping (friction) of the mechanical system

K :(back-) electromotive force constant

R : electric resistance

L : electric inductance

Vb :Back electromotive force (emf)

ω : angular velocity

HDL : Hardware Description Language

VHDL : Very High–Speed Integrated Hardware Description Language

JTAG : Joint Test Action Group

AS : Active Serial

TCP/IP : Transmission Control Protocol / Internet Protocol

RTU : Remote Terminal Unit

PWM : Pulse Width Modulation

UART : Universal Asynchronous Receiver/Transmitter

CHAPTER 1

INTRODUCTION

The brushless three-phase servomotor has become the main choice made by electric drive and motion control system designers due chiefly to its higher torque and speed ability for its size likened to stepper motors and brush commutated servomotors [1,2,3]. The servo motor is an electrical and mechanical device that motor motion is controlled by the electrical input signal. It is designed for system feedback in a closed loop [4]. Fast rate of taking samples with lower power consumption are provided by FPGA- established motion controller. FPGA-based motion control suggesting benefits such as a high speed, complicated application, and small power consumption [5].

The main topic of this study is to introduce a servo motors system through controlling them with FPGA device and given the importance of these motors in modern industry systems like motion control of CNC machines system.

Since their initiation, computer numerical control (CNC) machines have progressed to become accustomed to varying industrial and technological requests. Dominant issue that led the original progress of the CNC is needed to increase efficiency of machines. The requirement of machining ever reduced mechanisms and to conform to harder standards that bound the amount of deficiencies in these modules needs CNC machines not increase only their accuracy and speed but also to keep in height efficiency. For covering these requirements, new age group machines have to implement observing schemes to permit independent self-optimization that prepared over the incorporation of control progression approaches into the machine device control construction [6].

The controlling of servo motor system is described in this thesis. Mathematical modeling of servo motor is derived in transfer function with Laplace transform. For stabilizing the servo motor system different type of controllers are designed utilizing the root-locus technique. Proportional-integral-derivative (PID) controller technique is used to control output speed of servo motor. Then the simulation results are presented using MATLAB Simulink.

Multi-axes high-speed CNC machine needs in height-speed servo loop control to rejoinder a quick effort of machine. For this in height-speed system, FPGA is viable because of parallel processing. Even there are more controlled axes, servo loop time is constant. Thus, it is proper for high-speed control system and parallel function sections application to do procedure concurrently [7].

In the experimental setup and components, the physical structure of 3-axis CNC machine controlling system using servo motor, FPGA controller and PC is illustrated. Types and mechanism of servo motor are explained. Parts of CNC machine including limit switches, inductive sensors and linear module are described briefly in this study.

FPGA is programmed using VHDL code for implementing the control algorithm to generate three suitable 8-bit PWM signals to drive the motors to desired position. Pulse Width Modulation is extensively used in power electronics as a controller in power diversion and motion control and servo motors has a significant effect on industrial applications.

There are two common motor control modes are used in this study using FPGA. Position command is sent by a series of pulse train signals, and the position of each axis is feedback through the encoder attached on the motor to the driver to control the motor position.

The control results of experimental setup are presented. In the following the overview of chapters are summarized.

1.1 Overview of the chapters

This thesis is organized into six chapters:

Chapter 1: This chapter gives a simple detail of the work, the main functions of the thesis and the organization of the study.

Chapter 2: This chapter views the literature related to the work and the place of this work in the literature.

Chapter 3: Mathematical model and controller design for servo motors are demonstrated in this chapter.

Chapter 4: This chapter explains components that are used for CNC machine.

Chapter 5: In this chapter software of motion control system including servo motors driver software ASDA_soft and Quartus II with VHDL code are explained**.**

Chapter 6: system work and controlling system results are obtained.

Chapter 7: This chapter contains the conclusion and discussing the future work.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

There are many who make research and studies on the machines and methods of control these machines, with some used to control the motors of all types like stepper motors or servo motors, etc., as well as some of them made their studies about CNC machines, and that because of the extent of the importance of these machines in public numerous industries and applications. Researchers are focused on studying how to control these different machines using several methods of control, including: PD, PI, PID, pole placement, sliding mode techniques and FPGA controller device, etc… to improve the accuracy and efficiency of the controlled systems. This chapter touches on some of these studies.

2.2 Literature Related With the Work

Maciej Petko, Tadeusz Uhl (2001).This paper presents the idea for the design of control system for complex mechatronics. This idea is based on the default models of controllers using the simulation model. The distinction between the two different types of simulation is used in this paper; real-time simulation and off-line simulation. This includes simulation model design system components; treat all these mechanical and electrical components, control and Alkdk as important during the design process. They presented computer aided design approach of mechatronic system. They studied the concept of design and built-in control unit, based on the solution FPGA. Software in the context of the application of the methodology presented is expressed and tested. The specific design of a flexible robot arm control system case study is presented [9].

Thomas James and Coggins Marcian Nicolae Cirstea (2005) they obtained in their paper Field Programmable Gate Array (FPGA) operation features of a novel guiding Control and Data Acquisition (SCADA) Remote Telemetry Unit (RTU), advanced as a System on a Programmable Chip (SoPC). Transmitton Ltd. is a manufacture of electronic control tools for the railroad and development control productions. The corporation tried to make their production more modern by evolving a different range of fast flex for their control system, in cooperation with Anglia Polytechnic University. These (EDA) modern Electronic Design Automation methods are presence in employment, depending on Hardware Description Languages (HDL) and pointing FPGAs for application. Benefits such as capability in-house design, privileges for company's IP, saving cost, obtain high speed of the circuit [10]**.**

Kung-Lin Ho and Chiu-Keng Lai (2005) they mainly pointed at design motion control system for multi-axis CNC machines. To realize the function of multi-axis motion control they used a hardware- based DDA (Digital Differential Analyzer), which is implemented by the FPGA chip. They used FPGA because of its flexibility and it's capability for integrate the whole motion control system. Collaborating FPGA with interface card of self-designed PCI and software of real-time system permits a lower cost for completing the whole control system scheme [8].

Jingchuan, Yue Xu, Dong Taiyong Wang and Yanju Zhao (2008) they obtained in this paper a structure of CNC (computer numerical control) system which is based on entrenched equipment and implanted-real-time-operational system. ARM and DSP was used for designing the structure hardware system which is of dual-CPU model. This assembly is used for controlling construction procedure. FPGA module recognizes system’s supple industrial and reconfiguration. Because of difficulties of single task software was building using an operating system of entrenched real time and software-layer-design system improves stability of CNC system and task-message’s real time. Finally, possibility of this technique is specialized through requests in NC machine [11].

Hong-Tzong Yau, Ming-Tzong Lin, Meng-Shiun Tsai and Hao-Wei Nien (2008) To realize the determination of in height-speed and precision motion control, this paper suggests a (FPGA) field programmable gate array based motion controller with real-time look-forward function for (CNC) computer numerical control machines. Compared to a single PC or DSP system they offered motion controller below PC-

FPGA design by using a two-step interruption system to decrease the computational load of non-constant normal B-spline (NURBS) interpolation and look-forward procedures through parallel processing design. The PC implements look-ahead algorithms and first-step interpolation, the FPGA collects the facility from the PC and achieves second stage interruption and servo control algorithm. Computing act for NURBS interpolation through parallel computing and develop the following act by look-forward role and two-stage interpolation process ,can be improves by numerical simulations and experimental results using an X-Y table specify that the high-level integration applied on the PC-FPGA construction [12].

Kariyappa B. S and Dr. M. Uttara Kumari (2008) in this paper a controlling of AC Servomotor speed using sinusoidal PWM system based on Xilinx Field Programmable Gate Array (FPGA) was presented. Xilinx FPGA is a programmable logic scheme industrialized by Xilinx these kinds of devices deem as an effectual hardware for fast prototyping. It is utilized to produce 50 Hz sine wave, the triangular wave and the sinusoidal PWM signals. Sinusoidal pulse width is used for control motor speed. The control system has been obtained using Xilinx FPGA SPARTAN XC3S400 and verified using SM115 typical of (AC) servomotor. The product offers a controllable speed with suitable dynamic and static acts [13].

Hariprasad S. A., R. Nagaraj and Kariyappa B. S. (2009) this paper presented, that by using Field Programmable Gate Array (FPGA) a new method for control position of AC Servomotor is obtained. This FPGA controller is used to produce direction and the digit of pulses essential to rotate for assumed angle. It sends pulses as a square wave, the number of pulses controls the angle of rotation and frequency of square wave controls the speed of rotation. The suggested control system has been understood using XILINX FPGA SPARTAN XC3S400 and tested using MUMA012PIS model (AC) servomotor. The results of Experiments proved the control of AC Servo motor position efficiently [14].

C. Nwankpa, T. Chagnon, P. Vachranukunkiet, J. Johnson and P. Nagvajara (2009) obtained the primary computational results based on a steward computer connected to a (FPGA) Field Programmable Gate Array. Load stream designs represented by these results which are implemented a realistic size of the systems of power. The reason behind this work is to reduce computational time that dependable on power system process. Machinists work with a variability of control system logical packages is pointed to ensure that the data real-time processing is being transmitted from the SCADA system through net and telemetry. In this paper the results are likened to achievement processes which gained from forming these procedures on benchmark power networks. Explanations additional explain that algorithm-specific hardware can offer in assured cases up to a command of level speedup over the software program of the same process organization on Pentium-based PCs [15].

Tianmiao Wang, Hongxing Wee, Youdong Chen, Yong Zou and Kai Sun, (2009) they proposed in this paper that by using FPGA technology a different whole strategy and application of a System on Chip resolution for CNC system is obtained. It contributes all-purpose processor, an Axis Control core, a PLC core, a control of Motion core, a Wishbone accepting bus and interface of a standard exterior connection on FPGA. The considered CNC system established on SOC (system on chip) has the benefits of in height speed and firm. They prove accomplished alternate to evolving CNC systems [16].

Enrico Nobile, Salvatore Pontarelli, Simone Teofili and Claudio Greco (2010) their paper is about an FPGA based architecture for complex rule matching with stateful inspection of multiple TCP connections. They showed a creative construction for series matching. It is concerned with an FPGA application and, differs from other comparable mechanism, it is mostly appropriate for procedures identical in several streams. The paper offering the industrialized design capable to proficiently achieve different streams deliberates how to enhance the strategy to bind the number of FPGA logic properties and displays the gained results [17].

Luis Alberto Vera-Salas , Roque Alfredo Osornio-Rios Aurelio Dominguez-Gonzalez , Sandra Veronica Moreno-Tapia , Rene de Jesus Romero-Troncoso and Ion Stiharu (2010) they explained in this paper a new generation machines like CNC (computer numerical control) that have to implement detecting strategies by integrating several sensors. The use of smooth sensors is required because in the most requirements the online processing of the variables is important. The effect of the exertion is the improvement of a wireless network stage of reconfigurable intelligent sensors for CNC machine demands which observing with the measurement necessities of new generation CNC machines. They put under test in the network four different smart sensors and their identical signal processing systems are realized in a Field Programmable Gate Array (FPGA)-based sensor bulge [6].

Soroush Shirali, Shahab Ensafi and Mahsa Naseri (2010) presented design RTU Hardware for SCADA Systems Using FPGA. Remote Terminal Units (RTUs) Are critical components in SCADA systems are answerable for data gaining in the sub-stations, and the achievement of the orders the dominant position. In this project they offer a new device RTU design the basic functions of RTU. By this process they will have totally parallel computing that increases the RTU occupations speed called non delay RTU. Without using processor the hazard of system crashes will be reduced and will enhance the dependability of RTU. The design consists of four main units, containing analog inputs, digital inputs, communication interface and digital output. These units have their specific modules in their buildings. This design is consist of the FPGA with the option of complete different rearrange so that it can produce enough I / O using the correct number of these units and RTU be appropriate for use in any sub-stations [18].

Kamonwan Tanta-ngai and Theerapong Fongjun (2011).This paper presented controlling High-Speed CNC grinding machine using Field programmable Gate Array (FPGA) and that because of multiple servo loop is much needed in CNC milling machine controller, the rapid processing important for high-speed axis CNC control. Due to the structure of the FPGA that can be processed at the same time, it can instantaneously control frequent movement at one time. Hence, even more axis are controlled servo loop time will be constant. In this project a PID- feed forward with anti-windup is applied [7].

Hao Zhu, Jianmin Zuo, Hua Yan and Xinghua Zhu (2011).Showed that with the intention of realize (SHT) software hardening technology of the entire computer numerical control (CNC) system, which is essential that communications module harden created on field programmable gate array (FPGA) so this paper sets forward the technical structure of SHT unit serial communications, which adjust the appropriate transmission rate and improve the steadiness and viability of the communication procedure in CNC mechanism. The project implemented on Verilog hardware explanation language (Verilog HDL) in the Quartus II 9.0 advance setting, established on the FPGA chip of Altera Company CYCLONE II series. To decrease the happening likelihood of asynchronous reset metastable state, the dual buffer circuits of asynchronous reset and synchronous liberation occupied in the serial communication unit. There are four altered rates for the baud rate that can be established in module for serial communication unit. The results indicate that the projected system is realistic and possible [19].

Li Zhu and Yi_min Yang (2011).This paper described two dimensional graphics CNC machine program and simulation so Windows XP is used as the operational system, and Visual C++ 6.0 software is used as the progressing tool. The resolution is completing the simulation of NC machining development with computer program. This subject explains the standards of programming and creates a number of two-dimensional graphics models. System can well define the scheme of motion control, servo motor drive and its application [20].

Xiaoxia Li, Baosheng Wang, Mulan Wang, and Xinghua Zhu (2011).They suggested in their paper that if all the tasks are implemented by software, the identical fulfillment time will touch the rapidity of the CNC system and that because many of difficult computational complications have to be predomination in the Computer Numerical Control (CNC) system, for example tool reparation calculation, speed disintegration, contour interpolation, position control and so on. This study puts forward a plan to software inurement performance established on FPGA (Flied Programmable Gate Array), and established to calculate the liberated mathematics units for the CNC system to develop the feed rate of the machine tool. Hardware Description Language (VHDL) is approved in Quartus II 9.0 advance setting, established on FPGA chip of Altera Company CYCLONE II series. Also presented some practical algorithms related to the operation of complex arithmetic operations in detail, such as the adding operation, subtracting, multiplication and division of fluctuating-point numbers. The results demonstrate that the projected strategy calculations for the unit as possible and realistic methods, which can also improve the movability and the possibility of applying for fluctuating-point calculation model in the CNC system [21].

Dr. Alex Noel Joseph Raj and Shri Lakshmi Pravalika (2013).They made clear in this project how to build robotic arm with a multi-purpose operations that can operate in a higher accuracy by 6 degrees of liberty. Because of using FPGA for control and processing the arm will carry a higher speed of operation and less delay in treatment. This will improve the speed of the process and reduce the response time when they are used in applications such as Bionic arm when compared to current methods [22].

Ahmed BELKHEIRI, Said AOUGHELLANET, Mohammed BELKHEIR (2013). Explain how to design and implement the revised strategy in the third phase of conventional technology SPWM Built on Cyclone II ALTERA FPGA use of different voltage and frequency AC supplies with high precision to cover the range required of V and F, with customizable features SPWM such as modulation index and frequency carrier wave, and modifying the frequency and time signal delay (dead-time). The design access is realized using Quartus II software through diagram and VHDL explanation language tools. Using Altera DE2 progress and instruction board the offered design has been employed and verified. And realized behavioral simulation and test results display that successfully offered SPWM signal generation strategy is working suitably and it can limit the use of logic elements (LE) [23].

Rajendrasinh Navalsinh Bariya, Chetanya Sharma and Prof. Kausal Doshi (2013).Indicated design and operation process of the idea around the axis motion control CNC machine built on the G and M codes by using FPGA. G and M codes are the plurality part of the motion act of CNC machine and with the additional expression motion curves and track mode and fulfillment [5].

Dipali B. Bhatt and Hemant R. More (2014).Illustrated in this paper that several productions increasing the use of the device CNC (computer Numerical control) day after day. So should control the movement of the accuracy of these machines. In many productions use of CNC (Computer Numerated Control) machine energies increasing day by day. So it should control the movement of these machines precisely. CNC machine motion control becomes challenging difficult. Monitor the movement of the CNC machine apply a policy unit movement control 3-axis basis of field-programmable gate array (FPGA). Because of the movement of a straight line to the machine CNC basic movement, this project offers Linear interpolation technique to estimate a straight line movement within any area of 3-axis. As control of hardware interpolation is realized by logic devices for example FPGA circuit in the controller, so this process can evade a great quantity of difficult calculations, which suggests that the controller has a high real-time execution of this method [24].

Azfar Khalid and Amjad Nawaz (2014).They mentioned that as DC motor common components used in mechanical systems, in approximately every manufacturing. In the paper, focusing on the design machine device and precision control system, DC motors are also used as the engine for all main steps. Use of high-end closed sophisticated CNC or control of open construction and sensors will run multifaceted CAM procedures. For amount sensor as a system of feedback in manufacturing machine CNC tools linear or rotating encoders and Glass scale are commonly used. Smart control design is essential to activate elastic CNC machines tool structures to realize high precision. To make these systems at low rate, it is required to construct the machine tool system with low number of sensors .In this paper, followed by an intelligent controller DC motor speed sensor less ambience in order to evade expensive hardware. This is a method to estimate the random estimate DC motor angular rate and current without the use of any sensor response rate in closed loop act. In this elementary and cost operative method, they have utilized the Kalman Filter (KF) to control the speed of the DC motor, where it is applied to a random time variable system. KF found strong compared to exterior turbulences, and has the capability to prophesy such states and restrictions which are not easy to be measure through sensors deployed. The employed system habits the state estimate method by investigating the currents in the windings and re-builds the states. In this way, the machine tool can become high accuracy using a low-cost alternative control [25].

2.3 Summary

From the above discussion, we can say that there are some various ways for motion control of CNC machines some of them are controlling motors using different types of controller modes such as Kalman Filter (KF), DSP, FPGA, etc... These techniques have helped to improve the performance of control of the machines according to their facilities. SCADA system is very important for some researches to make communication between PC and the controller devices by sending data from PC to the controller device.

CHAPTER 3

MATHMETICAL MODEL AND CONTROLLER DESIGN FOR SERVO MOTORS

3.1 Introduction

Brushless DC motor is a three-phase synchronous ac motor with position transducer inside the engine to move the position of the motor shaft to drive the amplifier for the resolution of controlling the current easing in the three stages of the motor windings [26].

A basic concept of motion control devices has not varied much in the last 50 years. The simple causes for using servo systems in disparity to open loop systems contain the necessity to advance passing reply times, decrease the steady state errors and decrease the sensitivity to load parameters.

In this chapter the analysis of electric servo drive motors are presented. The motor calculations show the existence of two time constants, mechanical time constant and electrical time constant. Usually the commercial specifications of servo motor record these times constant for the motor only with no load inertia attached to the motor shaft. It is essential to identify the actual value of the time constants under real load states, because these two time constants are part of the motor illustration that used in servo analysis [26].

Also transfer function of servo motors is explained. PID controller is used in this chapter to improve the performance and efficiency of the system.

3.2 Mathematical Models for Servo Motors

As mentioned above the block diagram of an AC servo motor is similar to the block diagram of DC servo motor.

Servo motor circuit diagram is illustrated in Figure 3.1

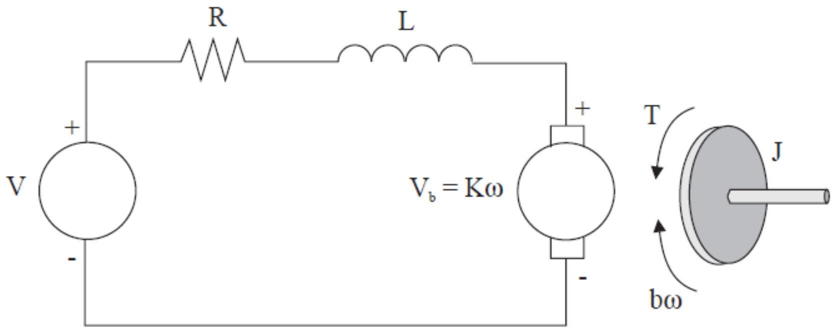

Figure 3.1 Servo motor circuit diagram illustration

As shown in Figure 3.2 amplifier can use in the operational mode modeled as amplifier output current, *I* which is directly proportional to the input voltage *V* the proportionality factor K_a

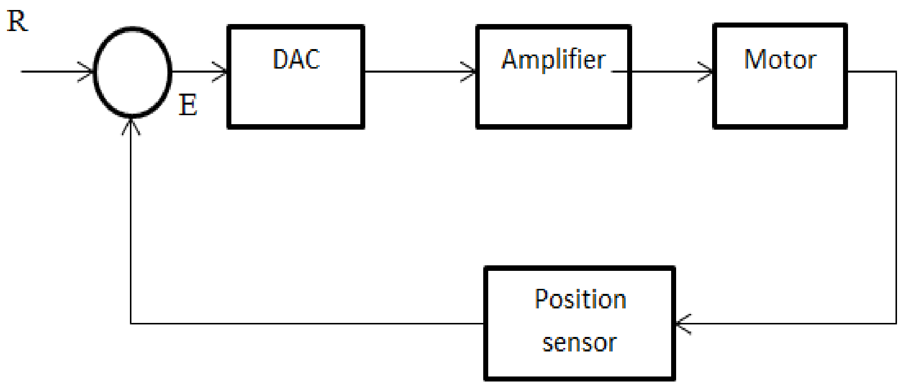

Figure 3.2 Block diagram of motor and amplifier

$$I = K_a V \tag{3.1}$$

Or the amplifier is modeled as voltage gain:

$$U = K_v \tag{3.2}$$

Motor torque equation:

$$T = Ki \tag{3.3}$$

Where *i* is armature current, K is constant factor.

Back electromotive force (emf), V_b, is associated to the angular velocity (ω):

$$V_b = K\omega = K\frac{d\theta}{dt} \tag{3.4}$$

We can drive the following equations from Figure 3.1 depending on the Newton's law joint with the Kirchhoff's law:

$$J\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} = Ki \tag{3.5}$$

$$L\frac{di}{dt} + Ri = V - K\frac{d\theta}{dt}. \tag{3.6}$$

Where,

J: moment of inertia of the rotor.

b: damping (friction) of mechanical system.

K: (back-) electromotive force constant.

R: electric resistance.

L: electric inductance.

3.3 Transfer Function

By using Laplace transform, equations (5) and (6) written as:

$$Js^2\theta(s) + bs\theta(s) = KI(s)\,, \tag{3.7}$$

$$LsI(s) + RI(s) = V(s) - Ks\theta(s), \tag{3.8}$$

Where (s) indicates the Laplace operator. from (8) we can find I (s):

$$I(s) = \frac{V(s) - Ks\theta(s)}{R + Ls}, \tag{3.9}$$

And by substitute it in (7) to obtain

$$Js^2\theta(s) + bs\theta = K\frac{V(s) - Ks\theta(s)}{R + Ls}. \tag{3.10}$$

The equation for DC motor is shown in the block diagram in Figure 3.3.

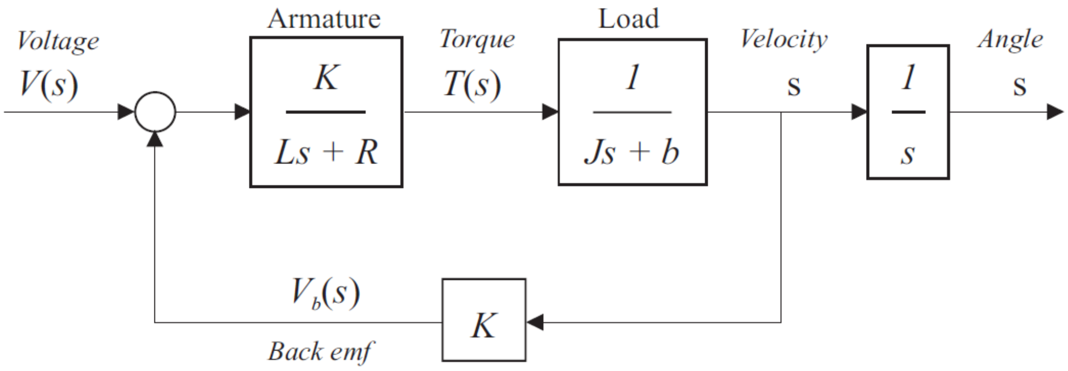

Figure 3.3 A block diagram of servo motor

From equation (10), the transfer function from the input voltage, $V\ (s)$, to the output angle, θ, is express as:

$$G_a(s) = \frac{\theta(s)}{V(s)} = \frac{K}{s[(R+Ls)(Js+b)+K^2} \quad (3.11)$$

From the block diagram in Figure 3.3, we can see the transfer function from the input voltage, $V\ (s)$, to the angular velocity, ω, is:

$$G_v(s) = \frac{\omega(s)}{V(s)} = \frac{K}{(R+Ls)(Js+b)+K^2} \quad (3.12)$$

Encoder provides the position, suppose an incremental encoder generates N pulses per revolution, that the encoder generates output.

K_f proportionality factor, K_f equals the number of units of feedback per one radian of rotation.

$$K_f = \frac{4N}{2*pi} \quad (3.13)$$

Another type of position sensor is the one of binary representation. The model for this sensor is total number of positions per revolution:

$$K_f = \frac{2^n}{2*pi} \quad (3.14)$$

Servo motor system design using Matlab Simulink is shown in Figure 3.4.

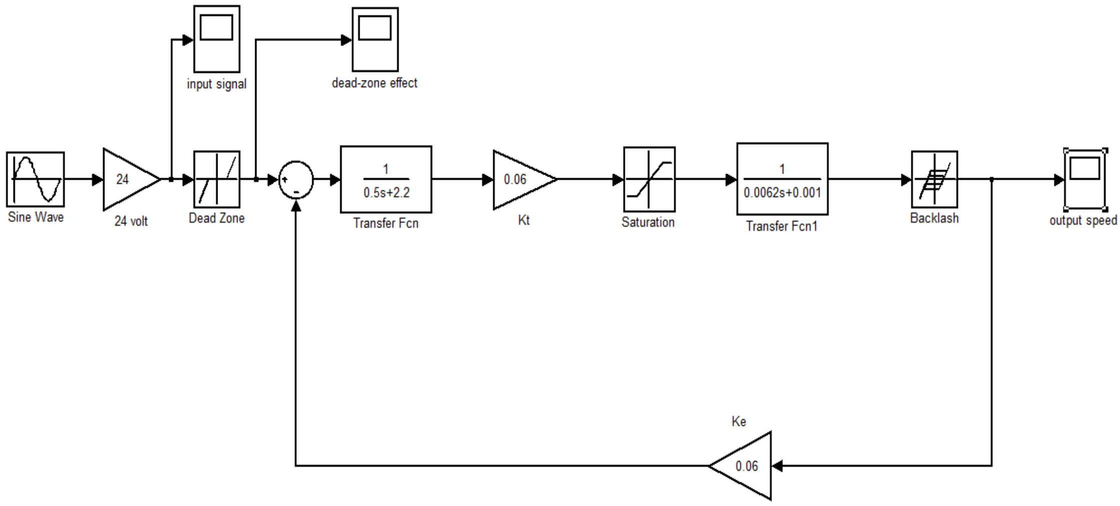

Figure 3.4 A Servo Motor System with Matlab Simulink

Mechanical time constant for DC servo motor is:

$$t_m = \Sigma R_a J_T / KK_T \quad \text{(sec)} \tag{3.15}$$

Electrical time constant for DC servo motor is:

$$t_e = L_a / \Sigma R_a \quad \text{(sec)} \tag{3.16}$$

Mechanical time constant for AC servo motor which is mentioned to as a brushless dc motor is:

$$t_m = \Sigma R_{PHASE} J_{TOTAL} / KK_{TOTAL} \quad \text{(sec)} \tag{3.17}$$

And electrical time constant is:

$$t_e = \frac{\text{Total inductive path}}{\text{Total resistive path}} = L_{L-L} / \Sigma R_{m\,(L-L)} \quad \text{(sec)} \tag{3.18}$$

Where,

J_T: Total inertia of motor armature plus load (lb-in-sec^2).

K: Voltage constant of motor (v/rad/sec).

K_T: Torque constant of motor (lb-in/A).

L_a: Winding inductance of motor (Henries).

R_a: Armature resistance (ohms).

Figure 3.5 shows the servo motor output speed SIMULINK result.

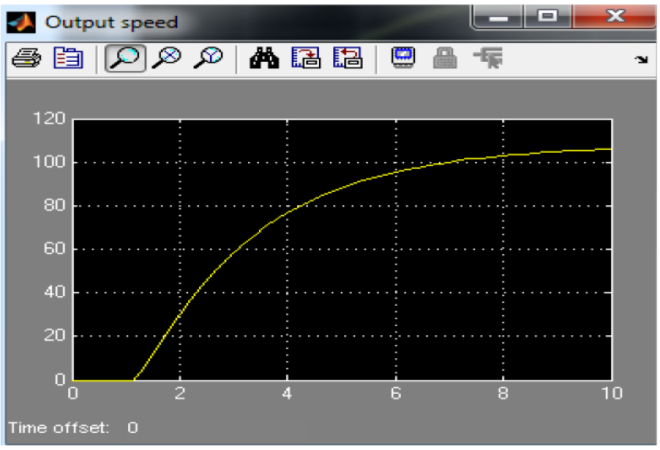

Figure 3.5 Speed output signal of servo motor

3.4 PID Controller

Proportional Integral Derivative (*PID*) controller considered the most overall type of controller used in dynamic systems.The main advantage of this controller is that it does not need a specific logical model of the system that is being controlled. For this cause, *PID* controllers have been commonly used in process control, robotics, industrial, transportation, automation, and interestingly in real-time preparation of simultaneous errands in multi-tasking requests. [27]

PID general equation is:

$$u(t) = K_p(error(t)) + K_i \int(error(t))\, dt + K_d\, d/dt\, (error(t)) \qquad (3.19)$$

Hence, transfer function of controller is:

$$C(s) = \frac{U(s)}{E(s)} = K_p + K_i / s + K_d\, s = (K_d\, s^2 + K_p\, s + K_i) / s \qquad (3.20)$$

Where u is the controller output and K_p, K_d, K_i are the proportional, derivative and integral gains.

Servo motor system with PID controller is designed using Matlab Simulink as shown in Figure 3.6.

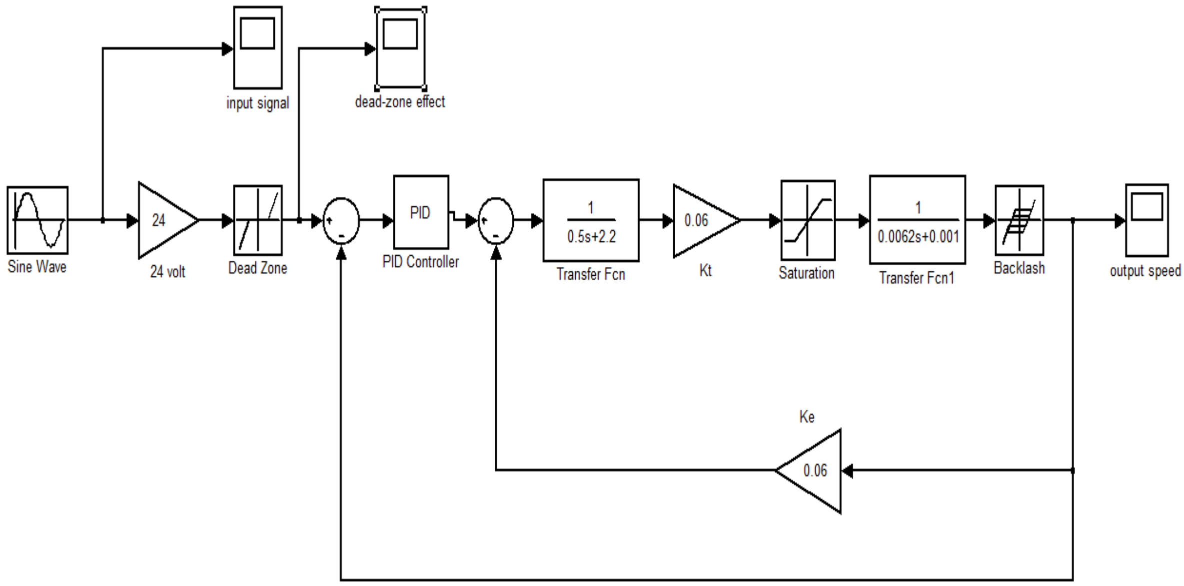

Figure 3.6 PID Controller SIMULINK diagram

The root locus function of the system and step response is obtained using Matlab simulink:

Figure 3.7 shows root locus for servo motor.

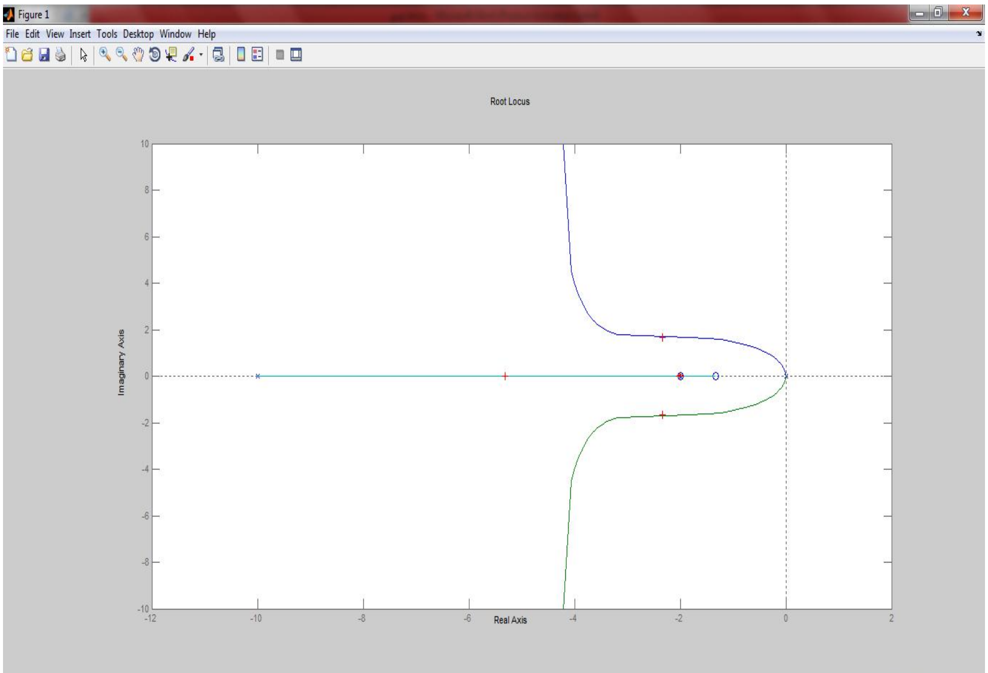

Figure 3.7 Root locus for servo motor

The rlocus and rlocfind purposes are utilized to choice the overall gain of the PID controller, such that the controller is stable and has the preferred location of the poles (within the defined ratio among the *Kp*, *Ki* and *Kd* constants). If the design is not reasonable, this ratio can be improved.

Servo motor closed-loop step response with a PID controller is obtained as shown in Figure 3.8.

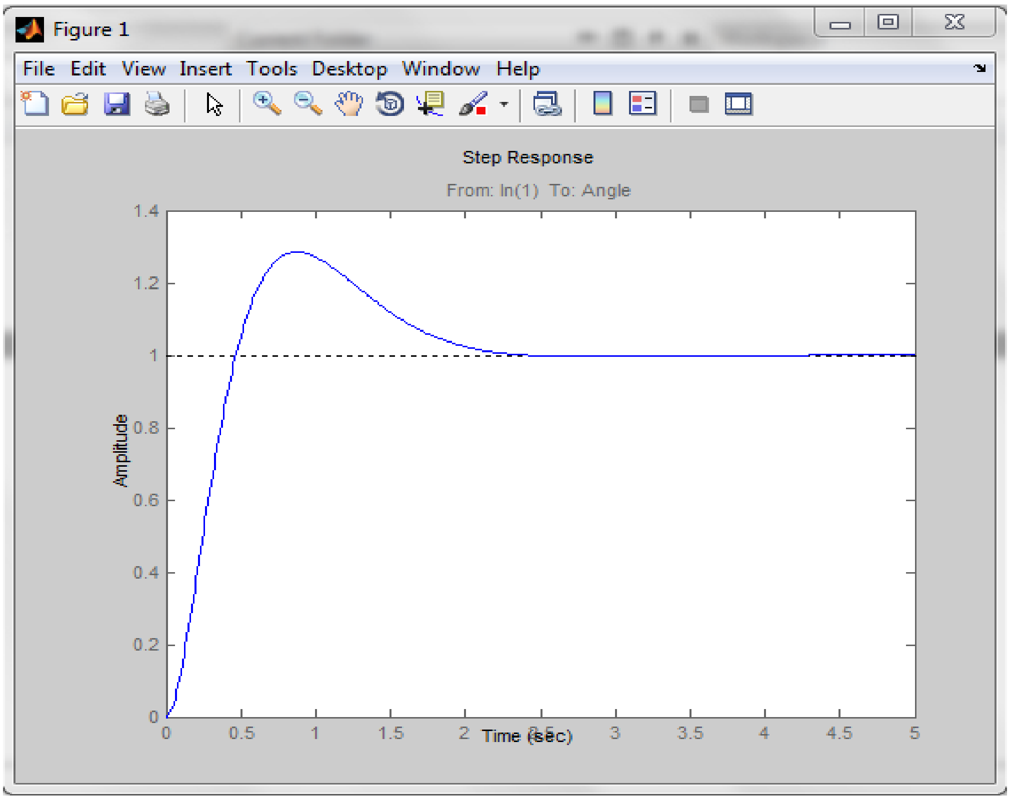

Figure 3.8 Closed-loop step response with a PID controller.

In Figure 3.9 we can see output speed signal of servomotor with PID controller.

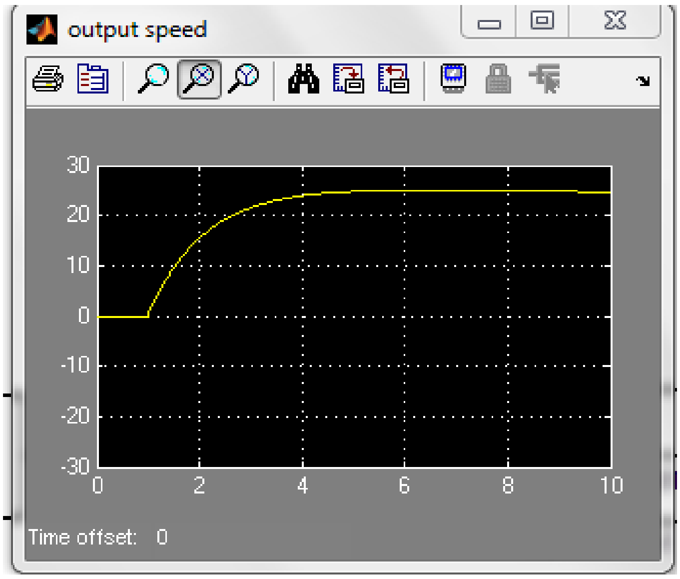

Figure 3.9 Output speed with PID controller

3.5 Summary

Transfer function of the system is obtained by analyzing the electrical circuit for motor using Laplace Transform and block diagram for system which based on transfer function obtained was designed. Root Locus approach is introduced for controller design.PID controller is used to help make the system more efficient and stabilizing; PID (Proportional, Integral, Derivative) control have parameters (Kp.Ki,Kd) in its section that make the motor moves in more stable situation as it shown in Figure 3.7 and 3.8 . These are the three fundamental coefficients to a mathematical algorithm that intelligently recalculates and delivers the power needed by the motor 8,000 times per second. The input to the PID control is the instantaneous desired position minus the actual position, be it at rest, or part of an ongoing trajectory. This difference is called the position error.

The Proportional parameter of the PID control creates a simple spring constant. The further the shaft is rotated away from its target position, the more power is delivered to return it.

CHAPTER 4

EXPERIMENTAL SETUP AND COMPONENTS

4.1 Introduction

In this chapter, the experimental setup and its components are introduced in details. Figure 4.1 shows a schematic of over all of the system.

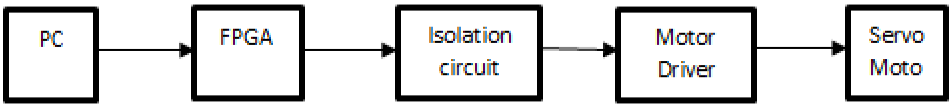

Figure 4.1 Schematic of over all of the system.

In experimental work the servo motors get data of PWM control signal from FPGA which is connected to interface circuit (isolation circuit) to protect FPGA from high voltage level because FPGA woks on 3.3v. Motion control algorithim is designed inside FPGA device using QuartusII software which written by VHDL (**V**ery **H**igh–Speed Integrated Hardware **D**escription **L**anguage) code. In this experimental study, the process of on-screen monitoring and control of FPGA based positioning operations were performed. The main objective of the set-up system is to control the axial movement on X-Y-Z axes, the transaction to be performed on the coordinates entered by the user.

4.2 Operation of the System

System work includes control of the 3-Axis machine with servo motor using FPGA technology by generating PWM control signal. Interface circuit consists of isolation and Mosfet driver circuit that isolate between power voltage level and FPGA voltage level and driving the Mosfet. FPGA is a field-programmable gate array which is an integrated circuit (IC) that can be programmed in the field after production.

4.3 Servo motors

The servo motor is an electromechanical device that the electrical input controls the position of the motor. Servo motor's Shaft can be placed in a precise angle through sending a signal. AC servo motors have remained commonly utilized in industrialized areas and different methods have been prepared to achieve high performance motion control. This can actually be used in many position control systems which vulnerability to exterior troubles such as friction [28].

DC servo motors have been used commonly in industry, computers, industrial equipment's, numeric control machines, and speed control of alternators, control mechanism of full involuntary regulators as the first starter, starting systems fast and properly [29].

Different size of servo motors is illustrated in Figure 4.2.

Figure 4.2 Different size of servo motors

With consecutively improving the precision and efficiency of the numerical controllers, digital control procedures have prevailed over all other analog parts counter. The benefits of digital controllers are: [28]

- Re-configurability
- Saving electrical selections
- Less external passive modules
- Fewer delicate to high temperature variant
- In height effectiveness

Classification of servo motors are shown in Figure 4.3

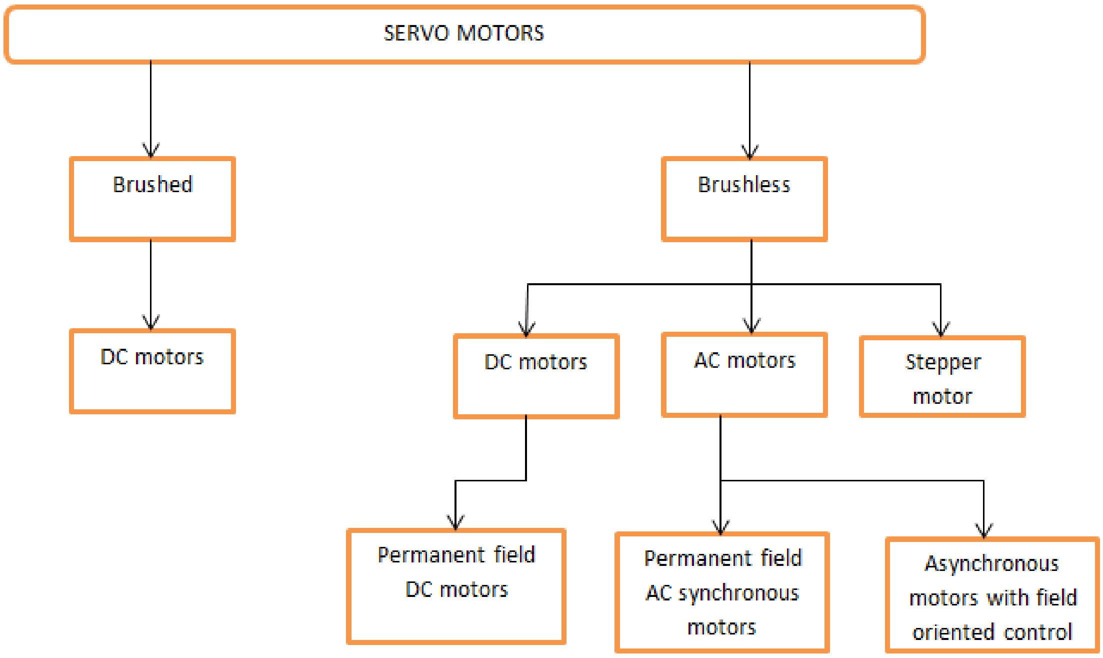

Figure 4.3 Classification of servo motors

In Figure 4.4 PMBL servomotor sections are illustrated

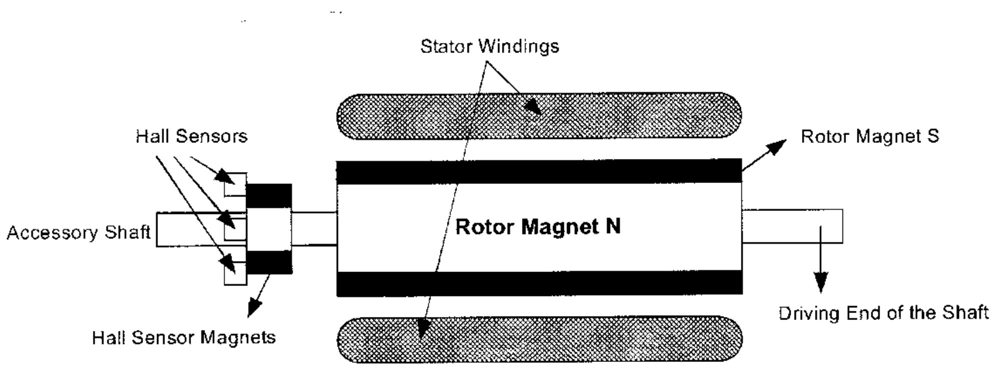

Figure 4.4 PMBL servo motor sections

BRUSH LESS DC (BLDC) / AC SERVO MOTORS come in single-phase, 2 phase and 3-phase configurations.

- **Stator**

Most BRUSH LESS DC (BLDC) / AC SERVO MOTOR have three stator winding connected in star fashion. Each of these windings is constructed with numerous coils interconnected to form a winding. One or more coils are placed in the slots and they are interconnected to make a winding. Each of these windings is distributed over the stator periphery to form an even numbers of poles. There are two types of stator windings variants: trapezoidal and sinusoidal motor. This differentiation is made on basis of the interconnection of coils in the stator windings to give the different types of back Electromotive Force (EMF) [30].

- **Rotor**

The rotor is made of permanent magnet and can vary from two to eight pole pairs with alternate North (N) and South (S) poles; unlike a brushed DC motor, the commutation of a BLDC / AC Servo is controlled electronically. To rotate the BLDC / AC Servo motor, the stator winding should be energized in a sequence. It is important to know the rotor position in order to understand which winding will be energized following the energizing sequence. Rotor position is sensed using Hall Effect sensors embedded into the stator [30].

4.3.1 Mechanism of servo motor

As the name proposes, a servomotor is a servomechanism. Further exactly, it is a closed-loop mechanism that utilizes position feedback to control its motion and final position. Digital or analogue signals are the input to its control which demonstrating the position ordered for the output shaft. The motor is mutual with several sort of encoder to bid speed and position feedback. The measured output position is complemented to the command position, the external input to the controller. An error signal is produced if the output position varies from that desired, which after that reasons the motor to rotate in either direction, as required to fetch the output shaft to the suitable position. As the positions procedure, the error signal decreases to zero and the motor stops. Basic servomotors use position-only sensing through a potentiometer and bang-bang control of their motor; the motor continuously rotates at full speed (or is stopped). This kind of servomotor is not commonly used in industrial motion control, but it processes the basis of the low-cost and simple servos used for radio-controlled models. Additional classy servomotors measure both the position and also the speed of the output shaft. They might besides control the speed of their motor, instead of continuously running at full speed. Both of these improvements, typically in combination with a PID control system, permit the servomotor to be carried to its commanded position more rapidly and more accurately, with less overshooting [31].

4.3.2 Servo motors control modes

Motor controller is the real device that energizes and de-energizes the circuit to the motor so that it can start and stop.

• Motor controllers may contain some or all of the following motor control functions:

- starting, stopping, over-current protection, overload protection, reversing, speed changing, jogging, plugging, sequence control, and pilot light indication.

- Controllers range from simple to complex and can offer control for one motor, group of motors, or supplementary tools for instance heaters, brakes, solenoids, clutches, or other signals.

In the following sections, position, speed and torque control modes will be mentioned.

4.3.2.1 Position control mode

A rotation detector (encoder) is attached on the motor and feeds the rotation position/speed of the motor shaft back to the driver. The driver estimates the error of the pulse signal or analog voltage (position command/speed command) from the controller and the feedback signal (current position/speed) and controls the motor rotation so the error becomes zero. Closed loop control method is realized with a driver, motor and encoder, thus the motor can carry out highly precise positioning processes.

- An END signal is achieved that interconnects the achievement of the positioning process.

- An alarm can be output if there is an irregularity such as an overload making it probable to interconnect tools irregularities [32].

Servo motor position control system is shown in Figure 4.5.

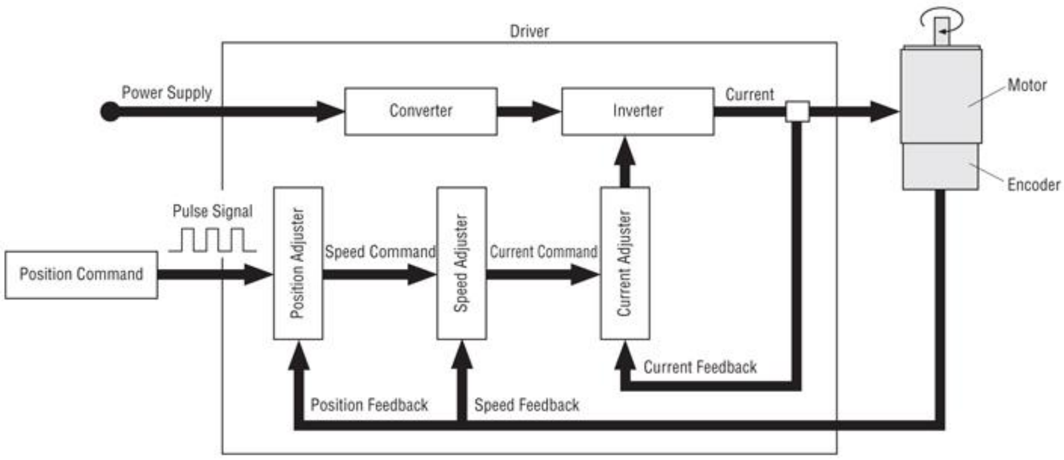

Figure 4.5 Servo motor position control system [33]

4.3.2.2 Speed Control mode

The purpose and the rendering essential for these motors are wide-ranging. When concentrating care on segment of the speed control of the motor market, stepping and servo motors are controlling their speed with a pulse sequence, while the induction motor and the brushless DC motor control speed with an outside resistor and/or DC voltage [34].

In Figure 4.6 the servo motor speed control system is illustrated.

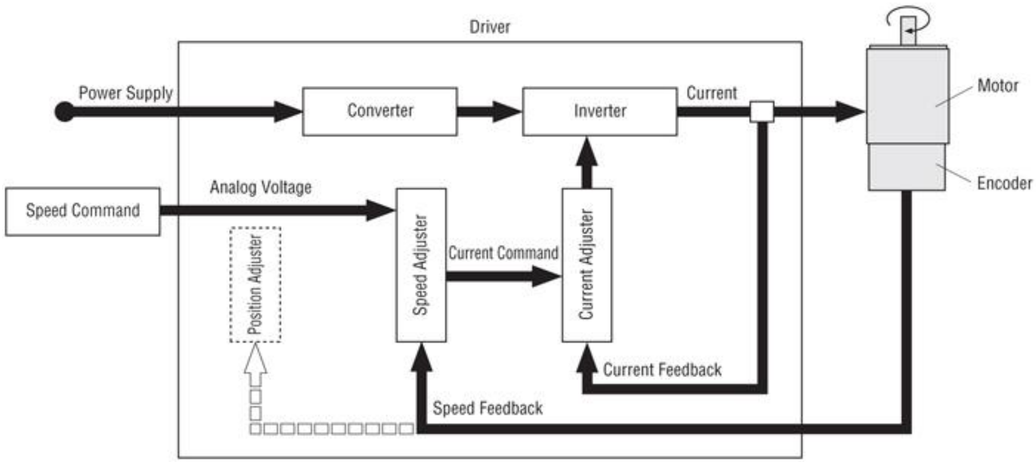

Figure 4.6 Servo motor speed control system [33]

4.3.2.3 Torque Control mode

Researches on PMSM such as direct torque control technique DTC have been enhanced also as the technology gets improved. DTC has several benefits such as high torque at low speeds, in height speed sensitivity and faster torque control. The central impression in DTC is to use the motor flux and torque as simple control variables, equal to the DC energies. The information concerning the rotor position is essential to achieve the field alignment procedure of the flux-vector drive appropriate to imitate the magnetic operational settings of a DC motor. This info should be achieved by feeding the speed of rotor and angular position back using a pulse encoder. Encoders are expensive and they append extra complication to the general system [35].

Servo motor torque control system is shown in Figure 4.7.

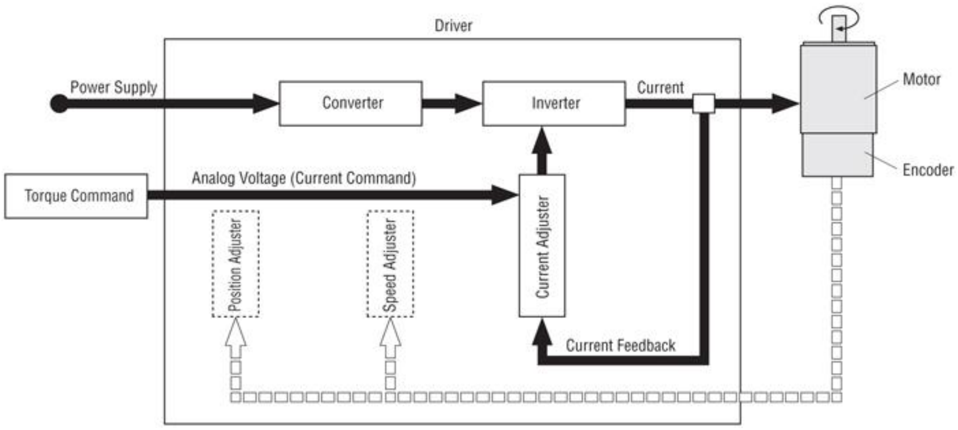

Figure 4.7 Servo motor torque control system [33]

4.3.3 Advantage and disadvantage of servo motors

Advantages of Servo Motors are: [36]

- High torque intermittent
- High torque toward inertia ratio
- High speeds
- Effort fine for rapidity control
- Obtainable in all sizes
- Less noisy

Servo motors have some disadvantages which are:

- Costly more than stepper motors
- Cannot work in open loop - feedback is obligatory
- Necessitate tuning of control loop parameters

• More conservation because of brushes on brushed DC motors

So, there are a lot of types of motors depending on the manufacture, DELTA servo motors are used in this project.

4.3.4 Delta servo motors

The improvement and cost effectually of permanent magnet manufacturing equipment for servo motor, high power and high execution semiconductor power procedures is enhanced incessantly as the techniques of microprocessor unit becomes more progressive. Consequently, AC servo motor and AC servo control system have become the basic procedures to accomplish the automatic control technology in the field of the present industry. Delta servo system is established on Delta's strength in manufacturing and electronic equipment and advanced for altered clienteles' necessities of different request machine implements. ASDA series servo drives are providing with an advanced digital signal processor (DSP) which denotes high-speed performance of control circuit loop. Further, the additional benefits of Delta ASDA series containing ease motor operation, gain tuning and software analysis / monitor function also offer high-speed and high-accuracy motion control for a wide range of developed mechanization implementation [37].
Types of Delta servo motors and drivers are shown in Figure 4.8.

Figure 4.8 Delta servo motors and drivers

Since the monetary resources required for obtaining new motors over gain were limited, the motor at hand were employed. The utilized servo motors are those of the medium and top segments of Delta Company. The motor models in system on the X and Y-axes are the servo motors of 2 Kw, coded

ECMA-K11320ES, and these ones, technologically, rank at the top segment of Delta Company. These motors have brakes and when servo motor becomes 'on' logic bit is '1', and then these brakes get on the active mode. Instead, the servo motor model used on the Z axis is a 0.75 Kw- AC servo motor coded ECMA-C30807FS. This one is older model than that used on the axes of X and Y. The brake of this motor is normally on active mode. The motors, which used, on the vertical axis mentioned to as Z-axis are too much load, the control over the braking system of these motors used on this axis cannot be logically performed. The brake is unloaded by giving 24 volts to the brake cables. Characteristic of the servo motors used are given in table 4-1 [38].

Table 4-1 Characteristics of Servo Motors

Axis	**Input/Entry Volt**	**Current**	**Power**	**Torque**	**Speed**
X	380 V	11.01 A	2.0 kW	9.55 N.m	2000 Rpm
Y	380 V	11.01 A	2.0 kW	9.55 N.m	2000 Rpm
Z	220 V	5.1 A	0.75 kW	2.39 N.m	3000 Rpm

4.3.5 Servo Motor Drivers

Two types of servo driver from Delta Company are used, one of them is ASDA-A2 servo series driver which is driving the motor for X and Y axes, other one are ASDA-B series driver that drive the motor for Z-axis.

4.3.5.1 ASDA-A2 series drivers

Delta hurled the high-feature motion control ASDA-A2 series servo motors and servo drives in 2009 to meeting the urgent needs of motion control implementation in developed mechanization and placate the requests of high-accuracy positioning control demanded by device inventors and structure integrators. These drivers have some of characteristics: Superlative Vibration Suppression, High Precision Control, Private Built-in Electronic Cam (E-CAM), Full Closed-loop Control (Capable of reading second feedback signals) and elastic Internal Position Mode (Pr Mode) [37].

4.3.5.2 ASDA-B series drivers

The B series drivers are a digital servo system with standard functions. It offers built-in Speed / Position / Torque control types and ropes pulse facility input and analog signal control. Parameters excision and process is prepared by elective handheld digital keypad. The gain tuning function permits informal process and the internal memory blocks reduce the parameters upload and download time. The B series is a cost-operative and acceptable servo system that plugs the most elementary and functional requests of manufacturing automation field [37]. In table 4-2, some features of ASDA-A2 and ASDA-B series drivers are given comparatively [38].
In short, as for touching on the features and capacities of the drivers: While ASDA-B type servo driver works with 220 volts, the power of the motor is also supplied from here. The motor used in the practice is a single-phase one, but in the models of this type of driver with more than 1.5KW power, 3-phase motor can also be used. The driver has 9 pieces of inputs and outputs in total. The encoder resolution of the driver is 2500 "pulses per revolution (PPR)". On the other hand, with ASDA-A2 type drivers, 3-phase servo motors with 2KW can be driven. These drivers have totally 13 outputs-inputs as 8 inputs and 5 outputs [47].

The encoder resolution sensitivity of ASDA-A2 drivers, one of the top segment drivers of Delta Company, is 1.280.000 PPR. This means that one cycle of a servo motor shaft can be scaled up to the number 1.280.000 [47]. These sensitivities are at quite critical levels for the practice/study performed. Nonetheless, there is a matter of inertia within the system due to the fact that the linear movement modules are smaller than the motor sizes.

Table 4-2: features of ASDA-A2 and ASDA-B series drivers

Function/Feature	**ASDA-B**	**ASDA-A2**
Input Phase (AC 220V)	1/3 phase (<= 1.5Kw) 3 phase (2Kw)	1/3 phase (<= 1.5 Kw) 3 phase (2~3Kw)
Encoder Resolution	2500 PPR	1.280.000 PPR
Setting Operation	External Keypad	LED display (5 buttons)
Max. Input Pulse	200Kpps (OC) 500Kpps	200Kpps (OC) 500Kpps
Frequency	(Line driver)	(LD) 4Mpps (line receiver)
Speed Responsiveness	250 Hz	650 Hz
Absolute Encoder	None	Option

4.4 FPGA

Currently, the FPGA has gotten more consideration before because of the development of VLSI technique. The benefits of the FPGA contains their programmable hard-wired performance, shorter design cycle, fast time-to-market, low power ingesting, implanting processor and higher compactness for the application of the digital system. FPGA offers a cooperation among the special-purpose ASIC (application specified integrated circuit) hardware and general-purpose processors [39].

In this study Altera DE2-115 cyclone IV FPGA device is used to control 3-axis servo motors system depending on programming the control algorithm with VHDL code.

4.4.1 Altera DE2-115 cyclone IV

Figure 4.9 Altera DE2-115 Developments and Education Board is illustrated.

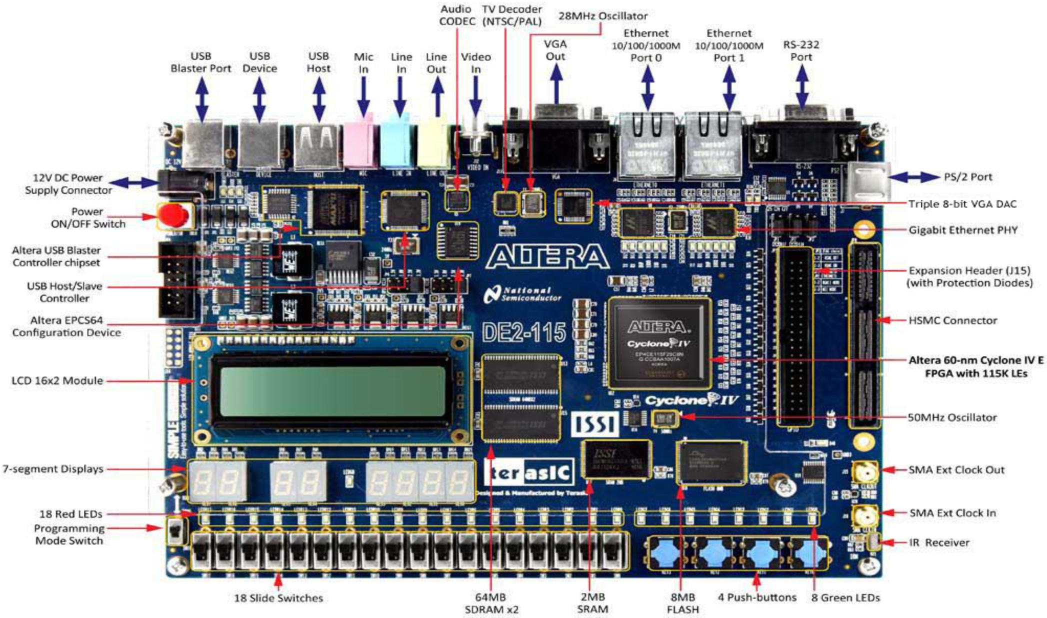

Figure 4.9 Altera DE2-115 Developments and Education Board

Behaviors of the DE2-115 board are:[40]

- USB Blaster (on board) for software design; both JTAG and Active Serial (AS) programming ways are reinforced.
- Altera Serial Structure device – EPCS64.
- Altera Cyclone® IV 4CE115 FPGA device.
- Two 64MB SDRAM.
- 2MB SRAM.
- 8MB Flash memory.
- SD Card socket.
- 18 Slide switches.
- 4 Push-buttons.

- 24-bit CD-quality audio CODEC with line-in, line-out, and microphone-in jacks.
- 18 Red user LEDs.
- VGA DAC (8-bit high-speed triple DACs) with VGA-out connector.
- 9 Green user LEDs.
- 50MHz oscillator for clock sources.
- 2 Gigabit Ethernet PHY with RJ45 connectors.
- TV Decoder (NTSC/PAL/SECAM) and TV-in connector.
- RS-232 transceiver and 9-pin connector
- USB Host/Slave Controller with USB type A and type B connectors
- IR Receiver
- PS/2 mouse/keyboard connector
- One 40-pin Expansion Header with diode protection
- 2 SMA connectors for external clock input/output
- 16x2 LCD module
- One High Speed Mezzanine Card (HSMC) connector

4.4.2 FPGA Architecture

The classic FPGA contains the following constituents [41]:

1. Programmable Logic blocks

2. Interconnection Resources

3. Input /output (I/O) blocks

The common diagram of an FPGA is shown in Figure 4.10.

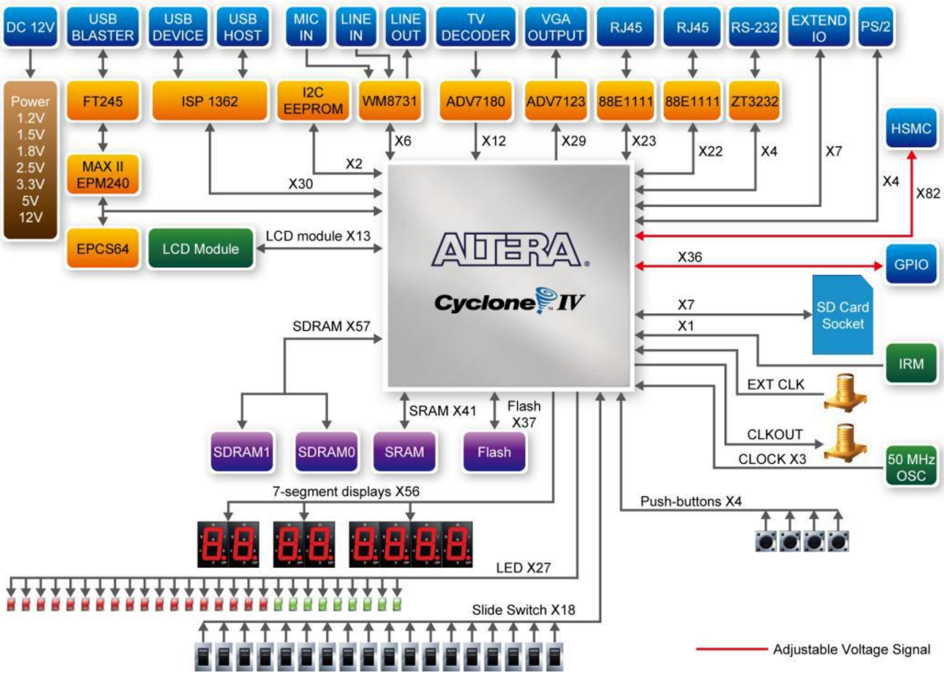

Figure 4.10 FPGA block diagram

4.5 Isolation Circuit

Isolation and MOSFET driver circuit consist of chiefly two-part. Isolation part is used for isolate power voltage levels and FPGA voltage level. In this study Optocoupler (Toshiba 6N137) is used for isolation. 6N137 contains a high emitting photo diode and a receiver circuits. Output signal of the 6N137 is inverted because its output has a nand-gate [42].

Figure 4.11 shows internal schematics and pin configuration of the 6N137.

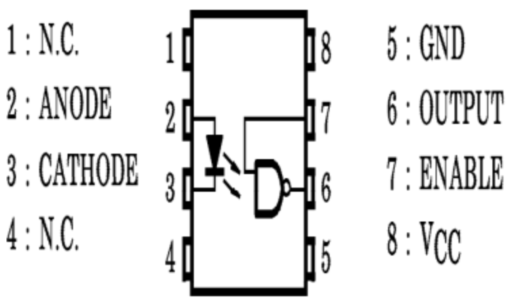

Figure 4.11 Internal schematics and pin configuration of the 6N137

Second part of the circuit is Mosfet driver. MICREL MIC4422 is a non-inverting Mosfet driver. The MIC4422 takes any logic input from 2.4V to VS without external speed-up capacitors or resistor networks. It has matched rise and fall times (25ns), low delay time 30ns typically, high capacitive load drive 47,000pF [43].

Pin configuration of the MIC4422 is shown in Figure 4.12.

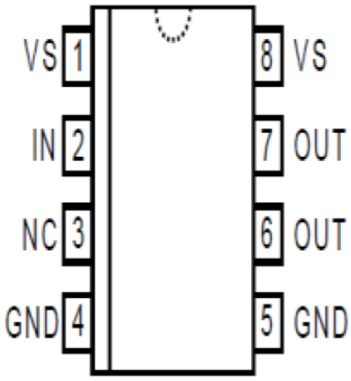

Figure 4.12 Pin configuration of the MIC4422

MIC4422 can be used for Switch Mode Power Supplies, Pulse Transformer Driver, Motor Controls, Line Drivers, Class-D Switching Amplifiers, Driving MOSFET or IGBT Parallel Chip Modules, Local Power ON/OFF Switch and Pulse Generators applications.

Interface circuit (isolation and Mosfet driver circuit) is shown in Figure 4.13.

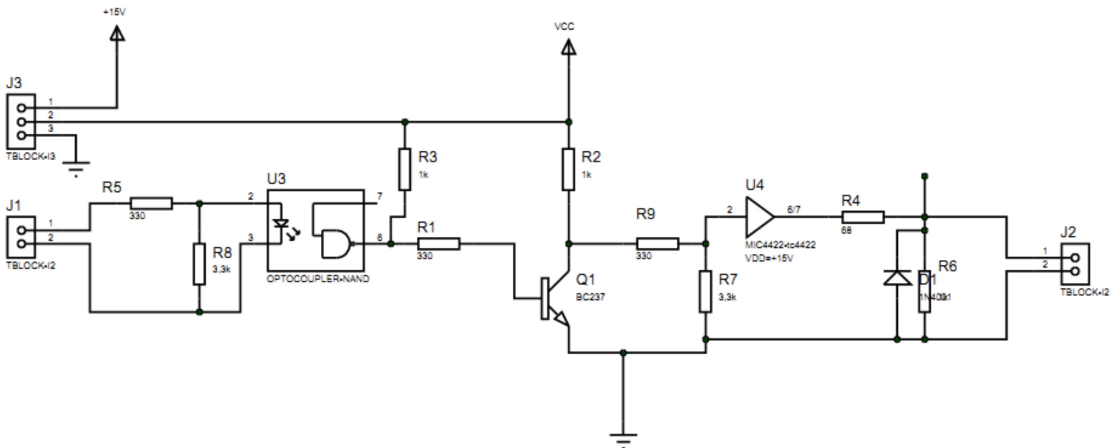

Figure 4.13 Isolation and Mosfet driver circuit [44]

4.6 RS232 Serial Port

RS-232 serial port was once a standard feature of a personal computer, used for contacts to modems, data storage, printers, uninterruptible power supplies, and other marginal devices. Nevertheless, RS-232 is disadvantaged by large voltage swing, low transmission speed and high standard connectors. In present personal computers, USB has displaced RS-232 from most of its marginal interface parts. Several computers do not originate ready with RS-232 ports and must utilize either an external USB-to-RS-232 converter or an internal development card with one or more serial ports to connect to RS-232 peripherals. However, RS-232 devices are quiet used specifically in manufacturing devices, networking tools and technical devices [31].

4.7 Summary

Experimental setup and components that used in this study are explained in this chapter. The schematic diagram of the system is shown in figure 4.1 and the operating of system is explained. Kinds of servo motors and servo motors control modes have been clarified. Also the system is including limit switch, inductive sensor. For controlling the whole system FPGA (Altera DE2-115 cyclone IV board) technology is used .FPGA characteristics and structure is obtained. Isolation and MOSFET driver circuit is constructed protect FPGA from high power voltage level. RS232 serial port is used to send data from PC to FPGA.

CHAPTER 5

SOFTWARE FOR MOTION CONTROL SYSTEM

5.1 Introduction

Software is very important to realize desired control algorithms. This chapter presents the features of software that used in system including servo motor driver software ASDA_soft and QuartusII software using VHDL code. Servo motors parameters are entered in ASDA_soft for motor drivers. Control algorithm is designed using Quartus II software and then downloading the designed program to FPGA board.

5.2 Servo Motor Driver Software Tools ASDA_Soft

Parameters of servo driver such as torque, speed, number of cycles, acceleration and deceleration time must be entered to the servo motor driver by using ASDA_Soft. These parameters are definite according to the desired way the motor should operate. A number of parameters changed by the user because servo driver can respond to the user requests and becomes programmable. Delta Company has developed a program called ASDA-soft in order to adjust and written the parameters in the servo driver.
The employed servo motor has mainly 4 principal operating modes.

1. External speed mode

2. Torque mode

3. Internal position mode

4. External position mode

In this study, all the servo drivers were used by modifying the external position control mode. In the external position control mode, the information of position, direction, and speed which is coming from an external source of pulse is processed in the servo driver and the servo motor is allowed to reach the targeted position. In this technique, the driving process is mainly carried out by using the specific inputs, such as “Pulse” and/or “Pulse/” and “SIGN” and/or “SIGN/” contained in the CN1 connector of the servo driver [38].

The image of servo operating mode selection screen required for a practical study by using the ASDA_Soft program is shown in Figure 5.1.

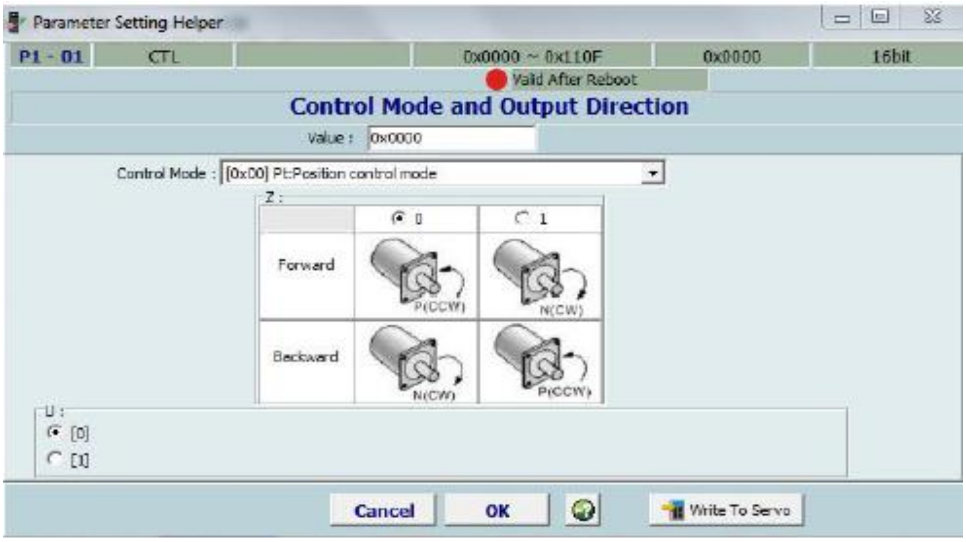

Figure 5.1 Servo motor driver operating mode select screen

The gain parameters of servo drivers with about 450 parameter adjustments, the input-output functions, the direction controls (adjustments) of forward and backward limit switches, and the sensitivity controls is shown in Figure 5.2.

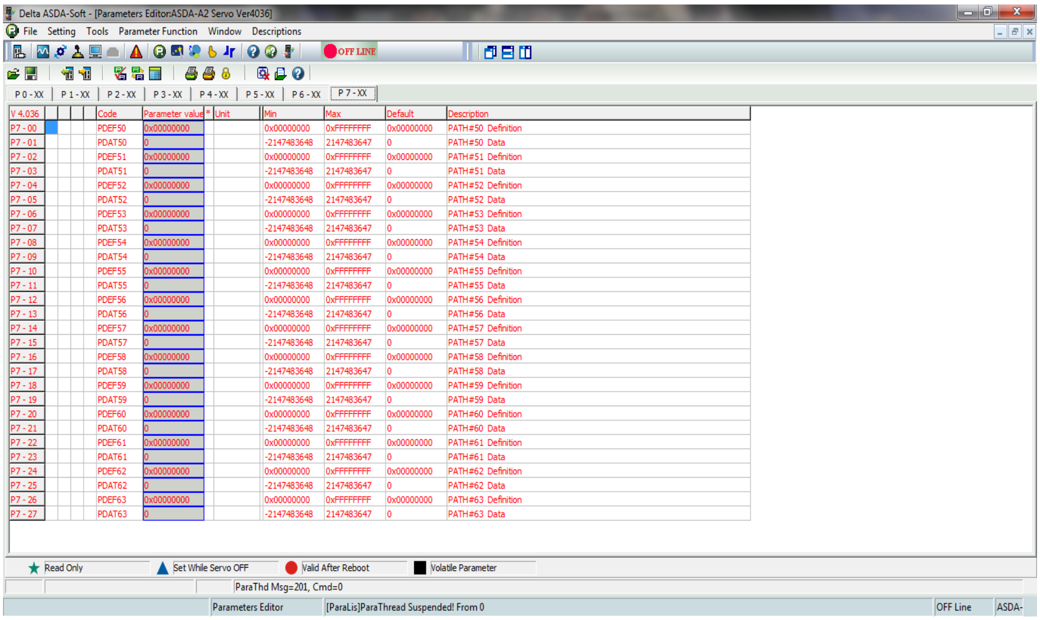

Figure 5.2 Servo motor driver parameter change screen

Further, the sensitivity adjustments of the servo driver are carried into position by calculating them at the desired resolution. The sensitivity of the designed system within the movement field on the tray was written on the servo driver by doing the calculation according to the equation 5.1 and by adjusting the involved P1-44 and P1-45 parameters.

Revolution of motor mile formula is given as:

$$Revolotion\ of\ motor\ mile = \frac{\text{Applied Pulse Frequency}}{\text{Encoder resolulation} \times 4} \times \frac{P1-44}{P1-45} \times 60 \quad (5.1)$$

5.3 Programming in FPGA

The FPGA chip programmed using hardware description language (HDL), which consist of two sorts of the languages, very high description language (VHDL) and Verilog language. VHDL is a hardware description language for relating digital designs. It designed from a government programmed in the advance of Very High Speed Integrated Circuits. VHDL is like a common programming language with additions to model both synchronized and sequential flows of operation and the idea of delayed assignment of values [45].

VHDL (**V**ery **H**igh–Speed Integrated Hardware **D**escription **L**anguage) is one of the most common hardware description languages. With VHDL, we can program the FPGA and simulate the program and then downloading it to the FPGA.

There are two types of codes in VHDL [46]:

- **Synthesizable Codes**: are the codes which can be employed as hardware on FPGAs. All VHDL codes are not synthesizable. So FPGA inventers shall be careful while selecting the codes
- **Non-synthesizable Codes**: are the codes used for simulation intent and cannot be applied on FPGAs.

VHDL has four design units:

- **Entity**: In this part all inputs and outputs of the section are cleared.
- **Architecture**: Functions between inputs and outputs are clear in this part.
- **Configurations**: Architecture and Entity are related.
- **Packages**: A collection of statements that can be used in more than one design.

This study offers complete information that will help to understand how to create a FPGA design and run it on DE2-115 development board using QuarusII software.

The standard FPGA design flow starts with design entry using graphics or a hardware description language (HDL), for instance Verilog HDL or VHDL. In this step, a digital circuit that is implemented inside the FPGA can be created. The flow then proceeds through compilation, simulation, programming, and confirmation in the FPGA hardware.

Figure 5.3 shows the FPGA design flow block diagram.

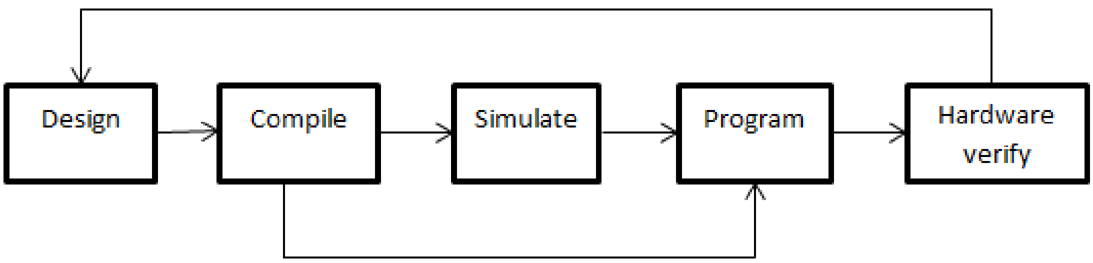

Figure 5.3 Design flow block diagram

Designing FPGA project with Quartus II software is explained step by step below:

_ creating a project

To start working on a new project first a new design project must be clear. Quartus II software makes the programmer's work informal via given that backing in the form of a wizard. Creating a new project as follows:

1. The project is getting start by opening Quartus II 10.0 main software screen as shown in Figure 5.4.

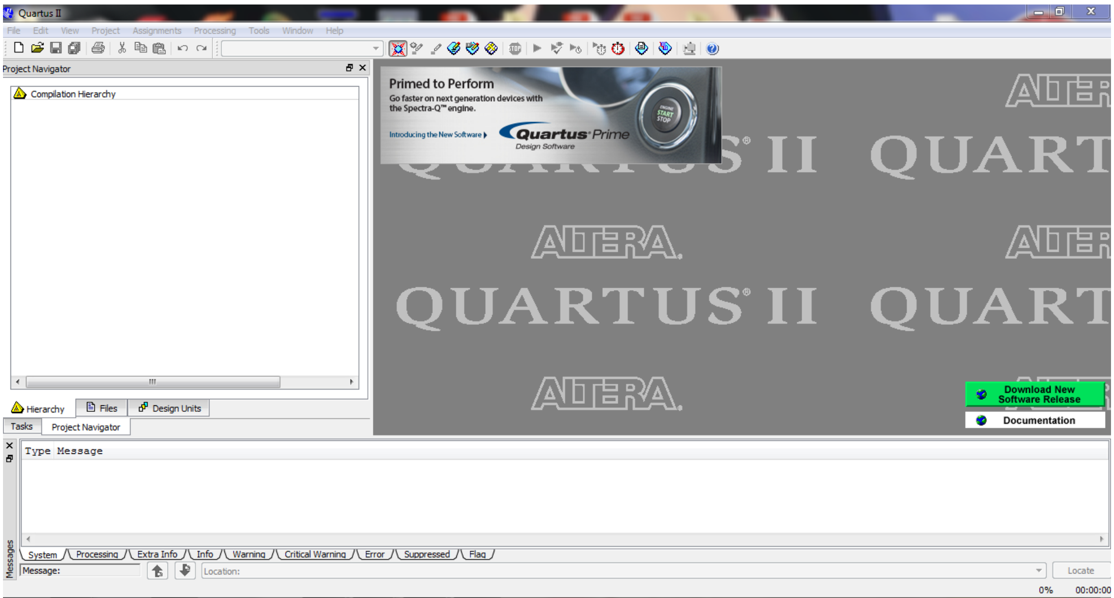

Figure 5.4 The main Quartus II display.

then select new project wizard from the file command to scope the window in Figure 5.5, which refer to the ability of this wizard.

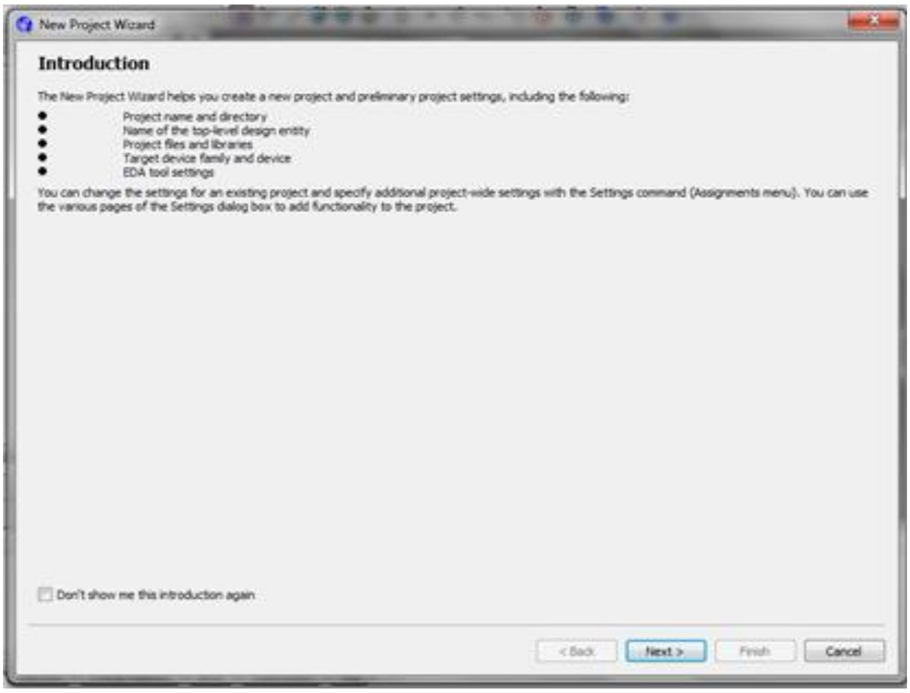

Figure 5.5 Tasks performed by the wizard.

2. Press **Next** from the window of new project wizard as shown in Figure 5.6 directory, name and top-level entity is entered in their fields.

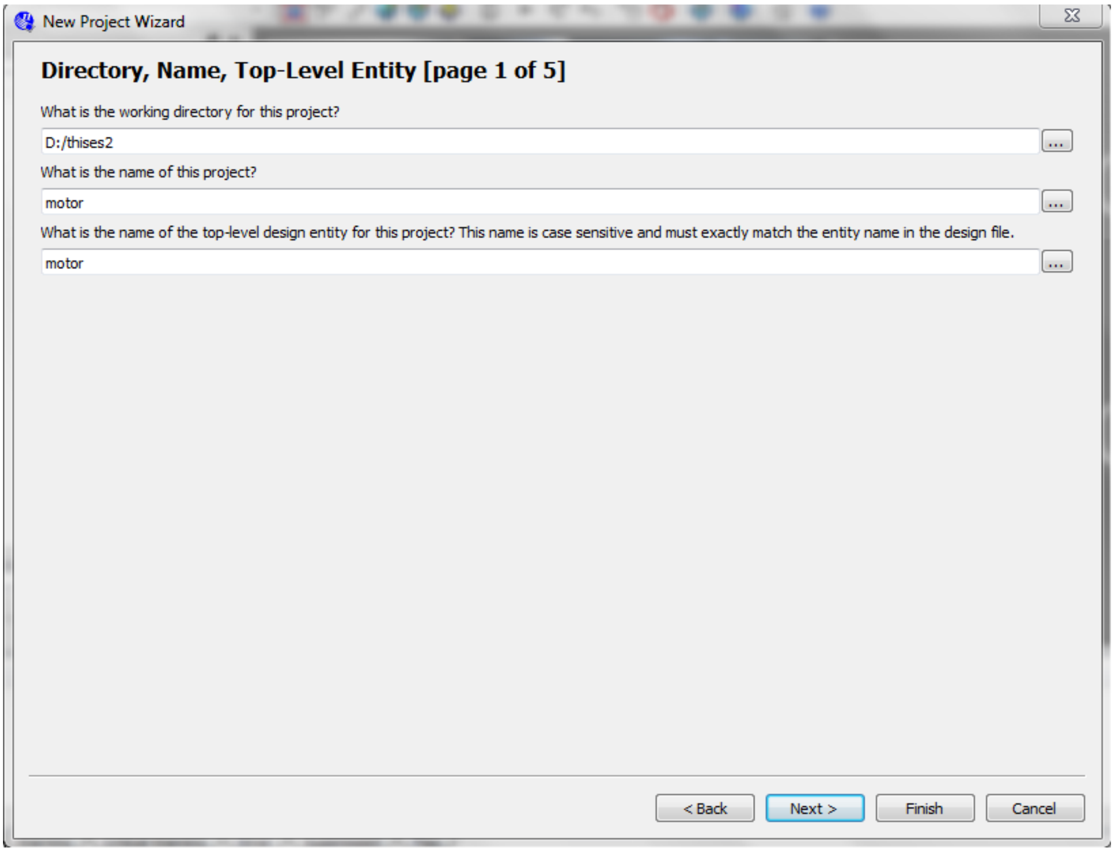

Figure 5.6 Creation of a new project.

3. Press **"Next** "to get window shown in Figure 5.7 The wizard makes it easy to require which existing files (if any) should be comprised in the project.

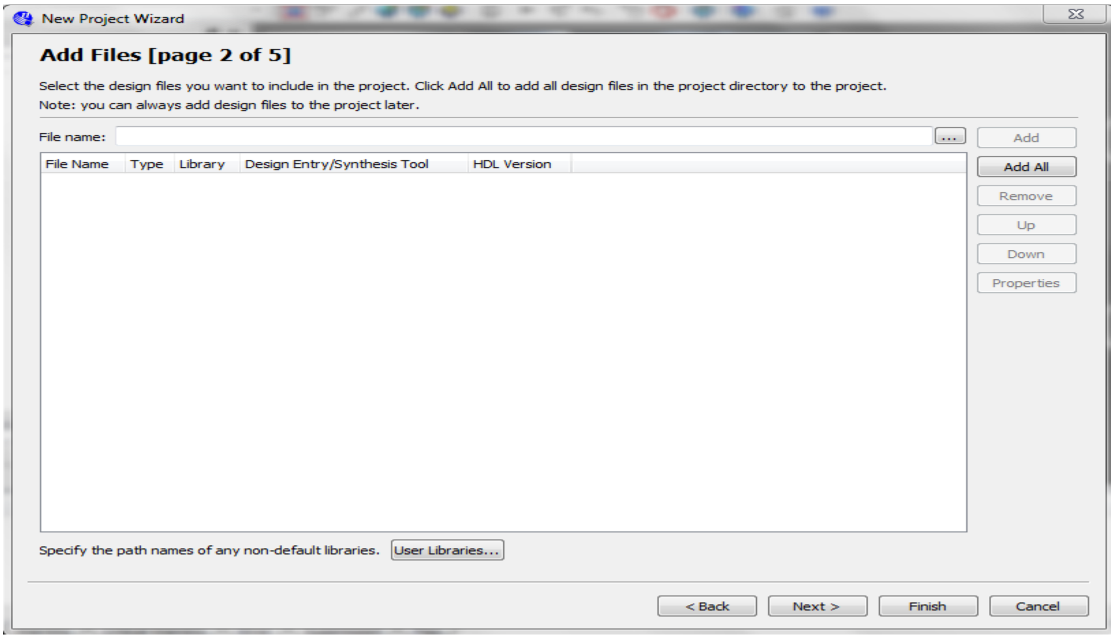

Figure 5.7 The wizard can include user-specified design files.

Click **Next**, which leads to the window in Figure 5.8.

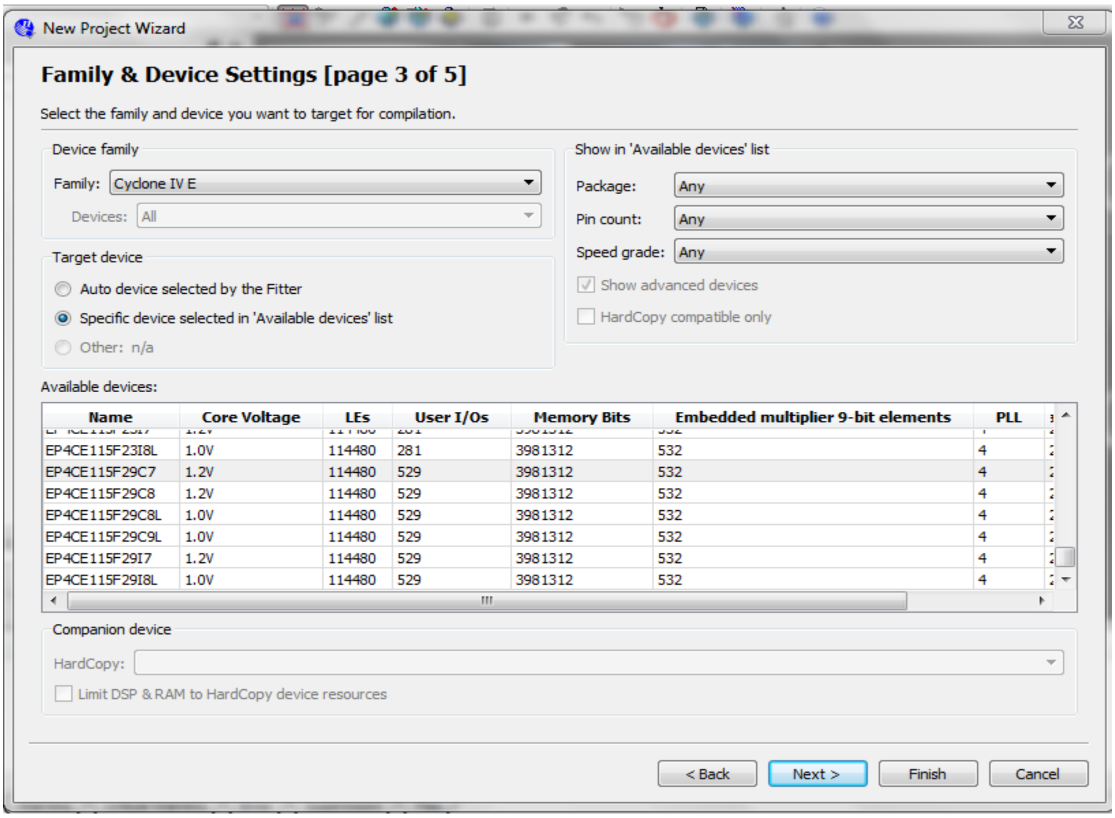

Figure 5.8 Family & device settings

4. Specifying the kind of device that the designed circuit will be applied on it. From the list of offered devices, the device called EP4CE115F29C7 is chosen which is the FPGA used on Altera's DE2-115 board. Press "**Next**" which leads to the window shown in Figure 5.9.

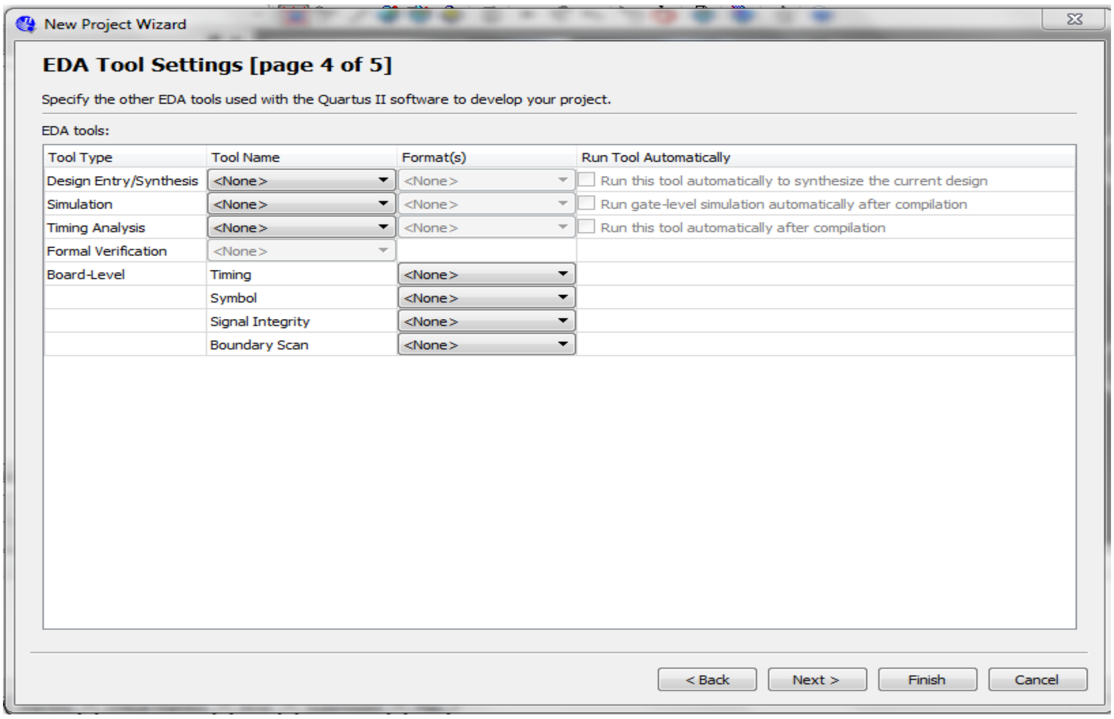

Figure 5.9 EDA tools settings.

5. The user can require any third-party tools that should be used. EDA tools are a commonly used term for CAD software for electronic circuits, where the shortening views for Electronic Design Automation. Press **"Next**". A summary of the selected settings performs in the screen shown in Figure 5.10. Then press **Finish**.

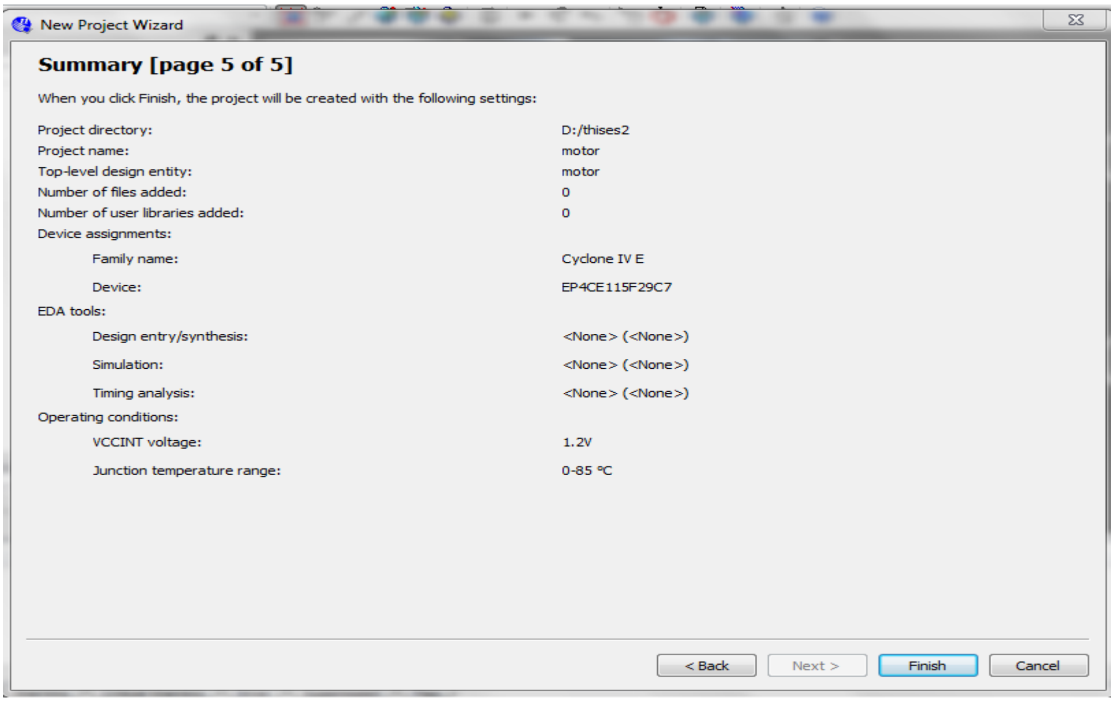

Figure 5.10 Summary of the project settings.

_ Design entry using VHDL code

VHDL file window is opened by choosing **file › new** the window in Figure 5.11 is appear and **VHDL file** is selected to open the VHDL cod window as appeared in Figure 5.12 then the system program in VHDL code is insert to the window and selecting **save** to save the file type as VHDL file.

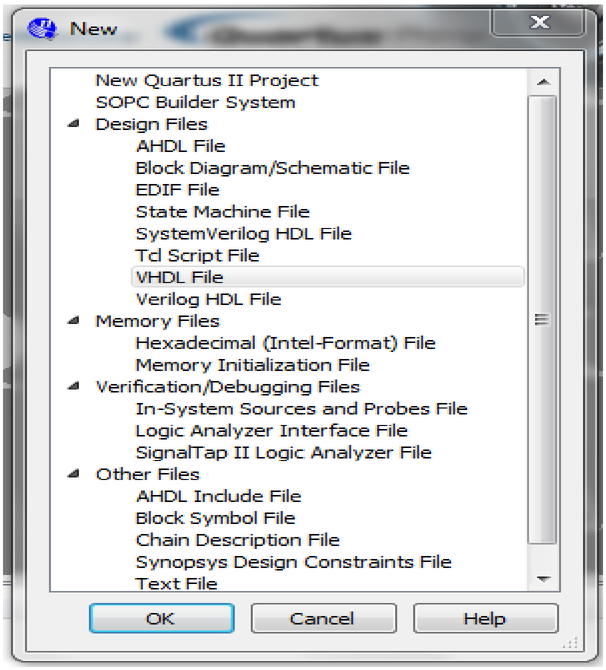

Figure 5.11 Choose to prepare a VHDL file.

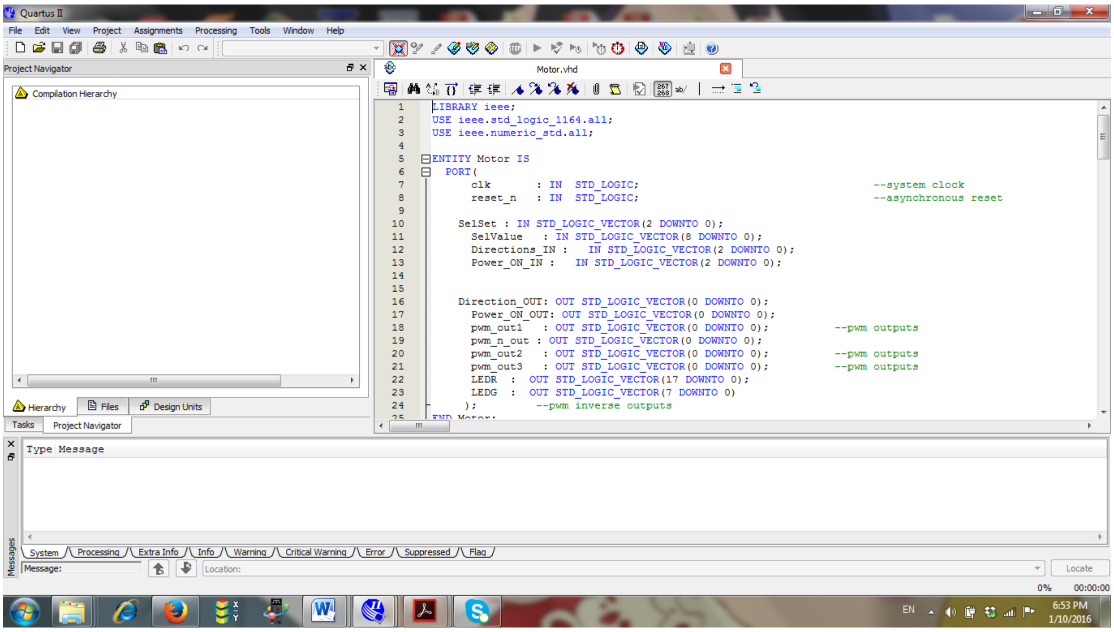

Figure 5.12 VHDL code window display

_ Compiling the designed system

The VHDL code in the file motor.vhd is handled by some Quartus II tools that investigate the code, synthesize the circuit, and generate an operation of it for the objective chip. Compiler is the application program that controls Quartus II tools. The compiler is running by selecting Processing > Start Compilation, or by clicking on the  toolbar icon. Report window of the compilation progress is appearing on the left side of the Quartus II display. Successful (or unsuccessful) compilation is specified in a pop-up box as shown in Figure 5.13[46].

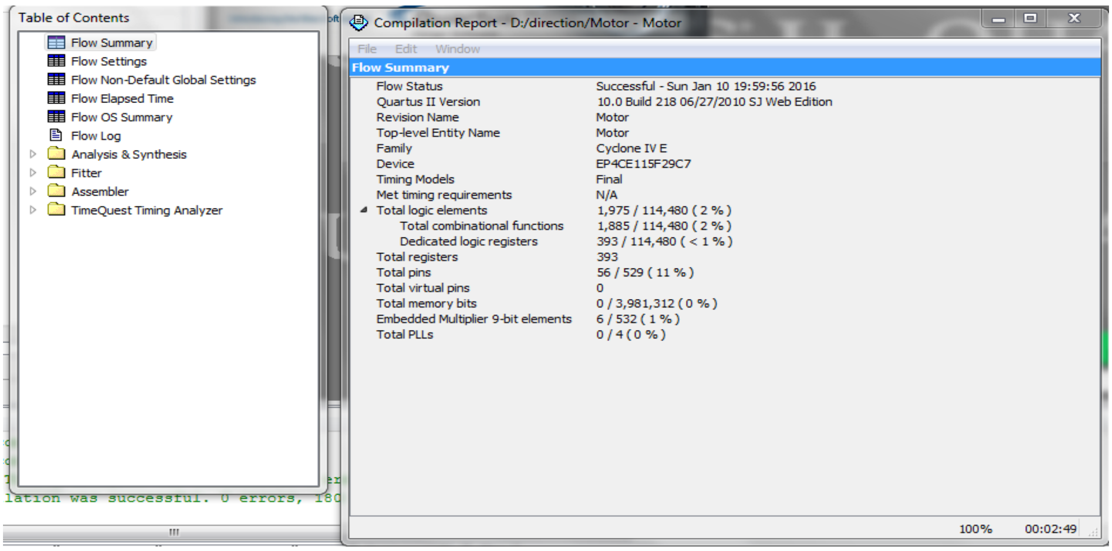

Figure 5.13 Display after a successful compilation.

One of the messages will report that the compilation is successful and there are no errors, if the VHDL design file is correct.

_ Assigning the circuit inputs and outputs to precise pins on the FPGA

Through the compilation above, the Quartus II Compiler was allowed to select any pins on the selected FPGA to work as inputs and outputs. However, the DE2-115 board has hardwired connections among the FPGA pins and the other components on the board. Pin assignments are prepared by using the Assignment Editor. Select **Assignments** > **Pins** to get to the window in Figure 5.14. Below Group select Pin. Recompile the circuit, which make the compilation with the right pin assignments.

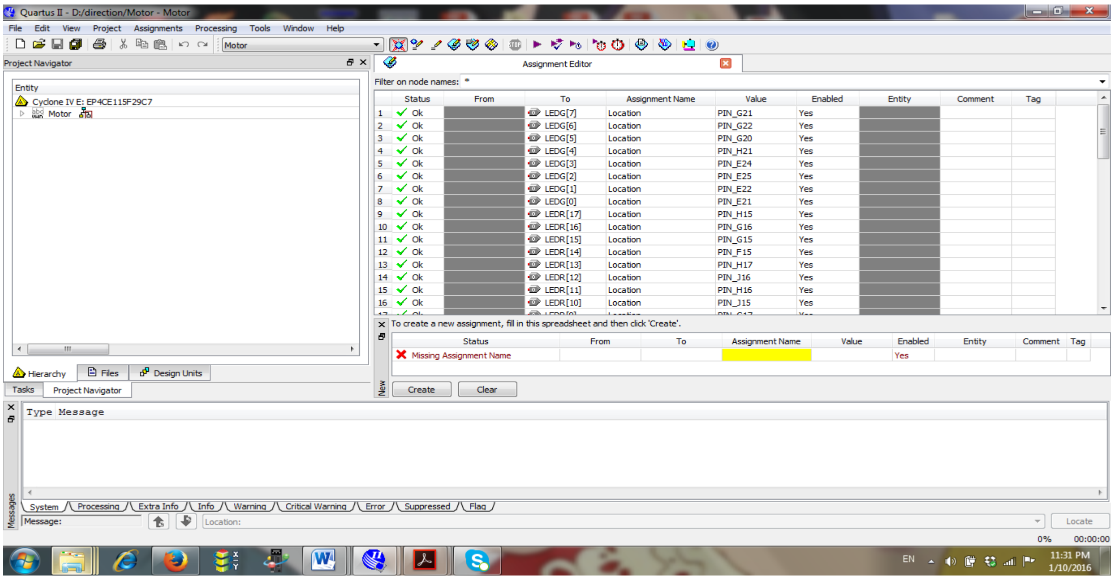

Figure 5.14 The Assignment Editor window

_ Simulating the designed system

Modelsim software can be used to simulate the performance of a designed system.

_ Programming and configuring the FPGA chip on Altera's DE2-115 board

The FPGA device should be programmed and configured to implement the designed circuit. Quartus II Compiler's Assembler module produces the configuration file. There are two different ways that permits the configuration to be done, known as JTAG and AS modes.

In the JTAG (Joint Test Action Group) mode, the configuration data is overloaded directly into the FPGA device. This group clears a simple way for testing digital circuits and loading data into them, which became an

IEEE standard. It will recall FPGA configuration if the power remains turned on but when the power is turned off the configuration information will be lost. The second way is by using Active Serial (AS) mode. In this case, a configuration device that contains some flash memory is used to save the configuration data. Quartus II software sets the configuration data into the configuration device on the DE2 board. Then, the data is downloaded into the FPGA upon power-up or reconfiguration. Consequently, if the power is turned off and on, the FPGA need not be configured by the QuartusII software. We can choice between these two modes by the RUN/PROG switch on the DE2 board. The RUN position chooses the JTAG mode, while the PROG position chooses the AS mode [46].

JTAG mode is used in this study.

The configuration data is moved from the host computer (which runs the Quartus II software) to the board using USB-Blaster driver, Plug in the 12-volt adapter to offer power to the board. USB cable is used to connect the leftmost USB connector (the one nearby to the power switch) on the DE2-115 board to a USB port on a computer that runs the Quartus II software.

5.4 SCADA Software WinTr

It is possible to use several programs to enter the coordinates of the 3-axis motor from the computer, including the SCADA program; all the coordinates to be entered by the user must be determined by taking the zero point as a basis. The target position and speed info determined by the user are entered into the section where it reads position into the part indicated. With the help of the ‘Start Move’ button on the screen, the involved commands of FPGA are performed. Consequently, the single spot movement is completed and the user is informed on the SCADA screen simultaneously. The X, Y, Z coordinates of 3-axses servo motors can enter using the main screen and sending data from it to FPGA.

5.5 Summary

Digital systems is very effective than analog systems because of easy modification. Control of the system is applied using digital controllers. Software is very important to realize desired digital control algorithms. To operate system some program are utilized to write program and to load the written program in to FPGA device The algorithm is written using VHDL code used for simulating system with desired controllers

CHAPTER 6

SYSTEM WORK AND RESULTS

In this chapter the connections of motion control system that used in this study is explained. Generating PWM signal for controlling 3-axis servo motors is obtained using VHDL code for programming control algorithm. UART is used for RS232 for sending data to FPGA and make the control system more fast and flexibility.

Figure 6.1 shows the servo motor driver connections and wiring scheme for position control mode.

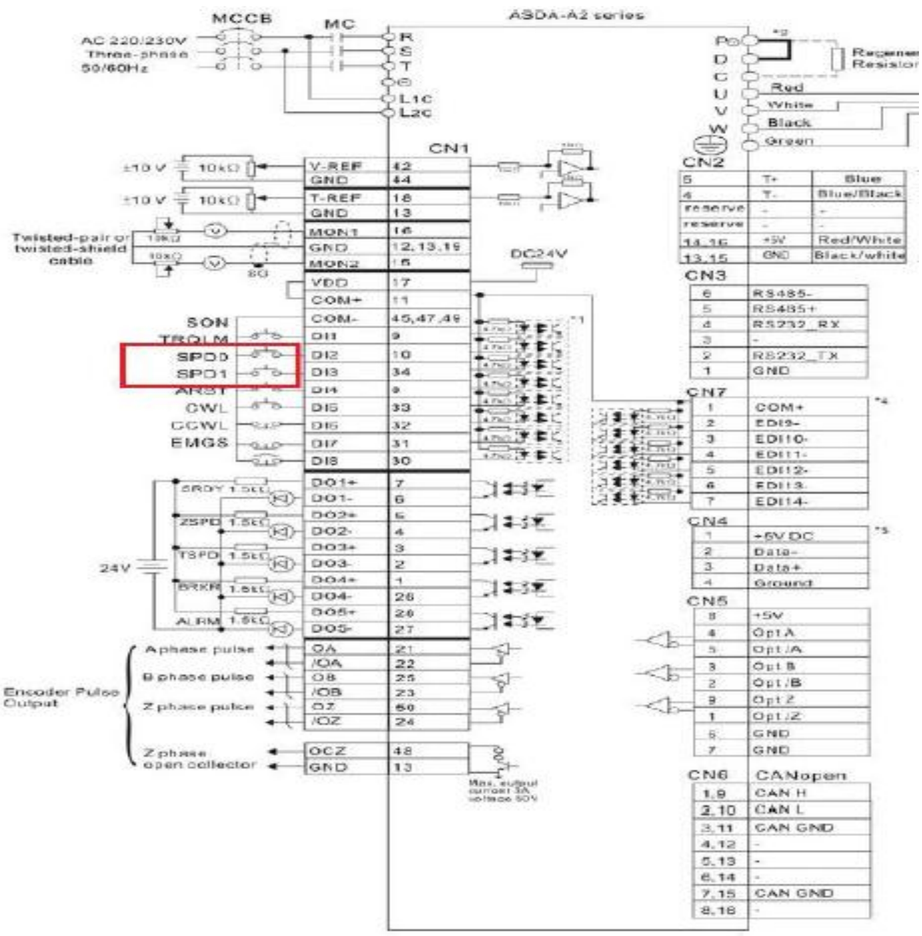

Figure 6.1 The servo motor driver connections and wiring scheme for position control mode [38].

‘CN1’connector is the servo driver input-output connector. CN1 contains internal power pulse inputs, supply output, inputs-outputs, the ‘sign’ referred to directional inputs, etc. pins. The inputs and outputs can be assigned to different tasks by being customized. This connector for the ASDA-A2 driver, in spite of having 50 pins, is of a very easily deformable structure. There are separate connection schemes according to every mode. There are outputs on which a pull-down needs to be done. In order to make all these connection adjustments modular, a terminal block connector card, which lets for and enables other connections to be made, was created after this study. This connector is different from ASDA-B type drivers’ input-output connector. ASDA-B type servo driver is used 25- pin classical serial ports connectors. Since the connectors of ASDA-A2 and B type drivers are different, 2 different cards were created [47].

In Figure 6.2-a, the terminal block card to be used in the ASDA-A2 driver, and in Figure 6.2-b, the terminal block card for the ASDA-B driver are seen.

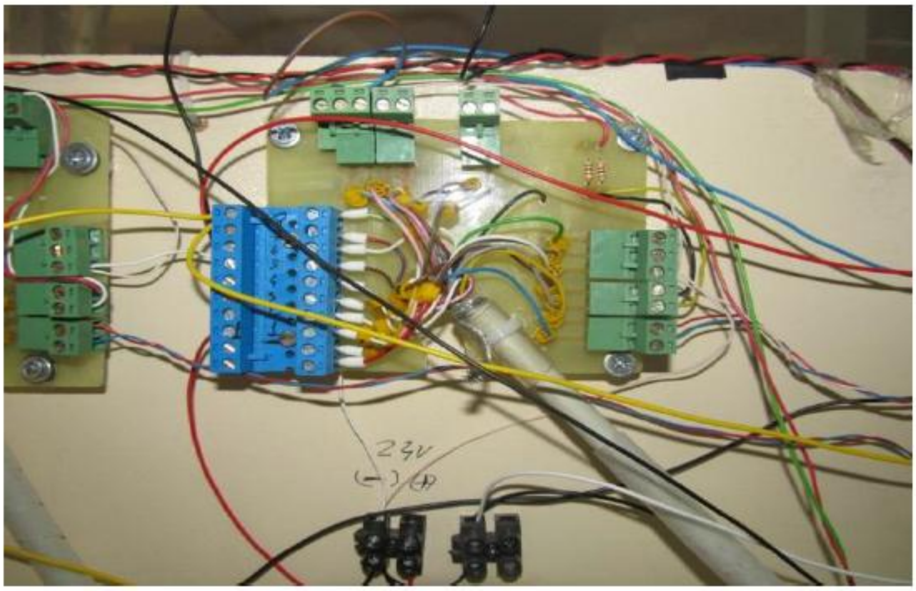

Figure 6.2a The terminal block card used in the ASDA-A2 driver [47].

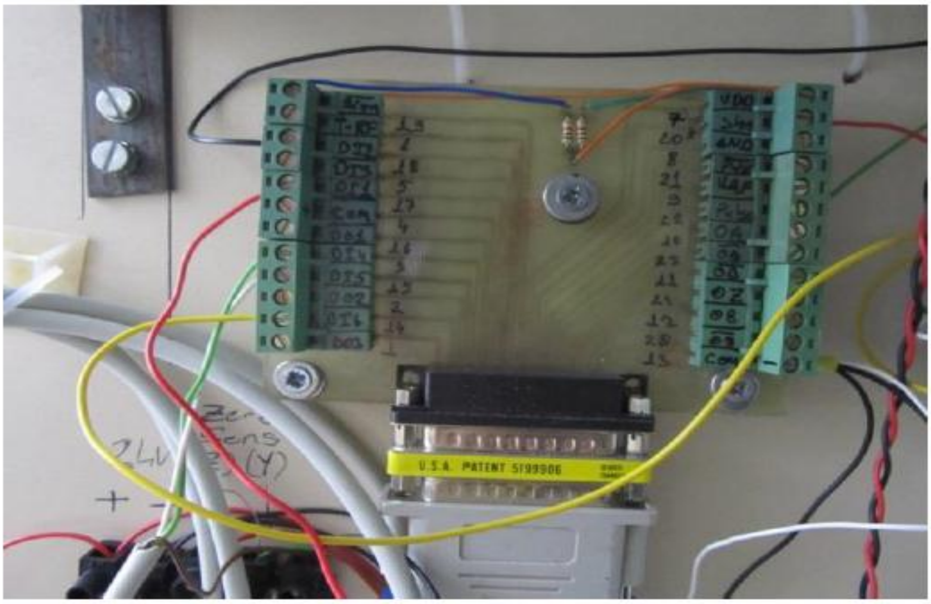

Figure 6.2b Terminal block card used for the ASDA B driver [47].

The feed of the 0.75 kw-motor used in the study was done with 1 phase and the sustenance of the other two 2kW-motors was achieved with 3 phases. There are two limit switches on each axis as forward-backward ones. The limit sensors were directly connected to the input ports of servo drivers fitting to the involved axis. The zero sensors, on the other hand, were connected to the input ports of FPGA.

After connecting the motors driver and whole system, the driving process will be done through the square waves applied to the pulse inputs of the driver. The coming frequency of square waves controls the speed of the motor. In the same way, the number of these square waves arriving at a certain frequency determines the distance to be proceeded by the motor [38, 48]. The servo motor is allowed to move towards the desired position in this way**.**

PWM strategy plays an essential part in reducing the switching losses and harmonics in these adapters, especially in the field of applications in three phases. In the past two decades, PWM different strategies, control systems, and realization techniques have been advanced [49, 50].

A PWM signal is not constant, the chief parameter is a duty cycle *D* that is a part of PWM period and defines the amount of on time to regular interval [51]. The equation (6.1) refers to the duty cycle as the following.

$$D = \frac{\tau}{T} \tag{6.1}$$

Where: $0 \leq D \leq 1$

Thus the output signal is calculated in equation (6.2):

$$output = D \times input = t_{on} / t_s \times input \tag{6.2}$$

Figure 6.3 shows the duty cycle of PWM.

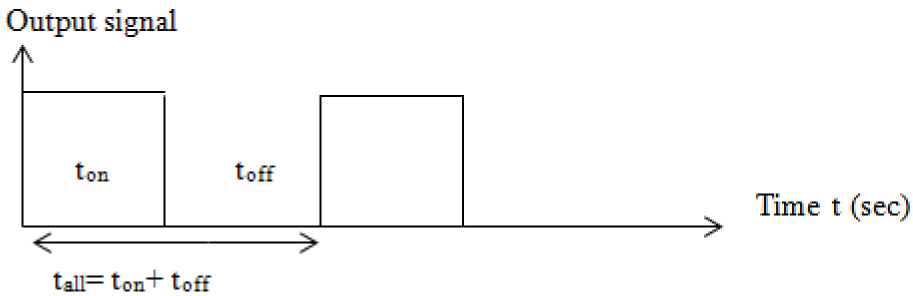

Figure 6.3 PWM with duty cycle

The principle work of system the input data is entered to FPGA as a frequency and duration of PWM signal using 18 slide switches (0-17) in binary numbers and one push-buttons switches is used to reset the program, so in QuartusII software the PWM signal programmed with VHDL code by making motor speed (motor frequency) is equal to the PWM frequency as mentioned before.

When the user enters the distance and motor speed from motor driver software the frequencies are equally depending on time of reach algorithm which is programmed in QuartusII software the generating impulses are stopped and this means the motor is arrived to desired position. And the algorithm of time to reach that used in this case is:

$$Timeofreach = \frac{distance * 10000}{motor\ speed(motor\ frequency)} \tag{6.3}$$

$$Motorspeed(motorfrequency) = PWMfrequency \tag{6.4}$$

Each push-button switch offers a high logic level when it is not pressed, and offers a low logic level when depressed. Since the push-button switches are denounced, they are suitable for using as clock or reset inputs in a circuit.

VHDL code is downloaded to Altera DE2-115 cyclone IV FPGA board. PWM signal is generated to control servo motor. Table 6-1 shown the experimental results and

Figure 6.4 shows the PWM generator flow chart; Figure 6.5 shows the PWM signal using modelsim simulation.

Table 6-1: The result of experiment

Motor frequency	PWM frequency In binary	Period in binary	Time of motor rotation
1000 Hz	110000001111101000	111000001111101000	1 Sec
5000Hz	11001001110001000	11101001110001000	5 Sec
10000Hz	110010011100010000	111010011100010000	10 Sec
15000Hz	110011101010011000	111011101010011000	15 Sec
200.000Hz	0100 00110101000000	0101 00000000001100	200 Sec

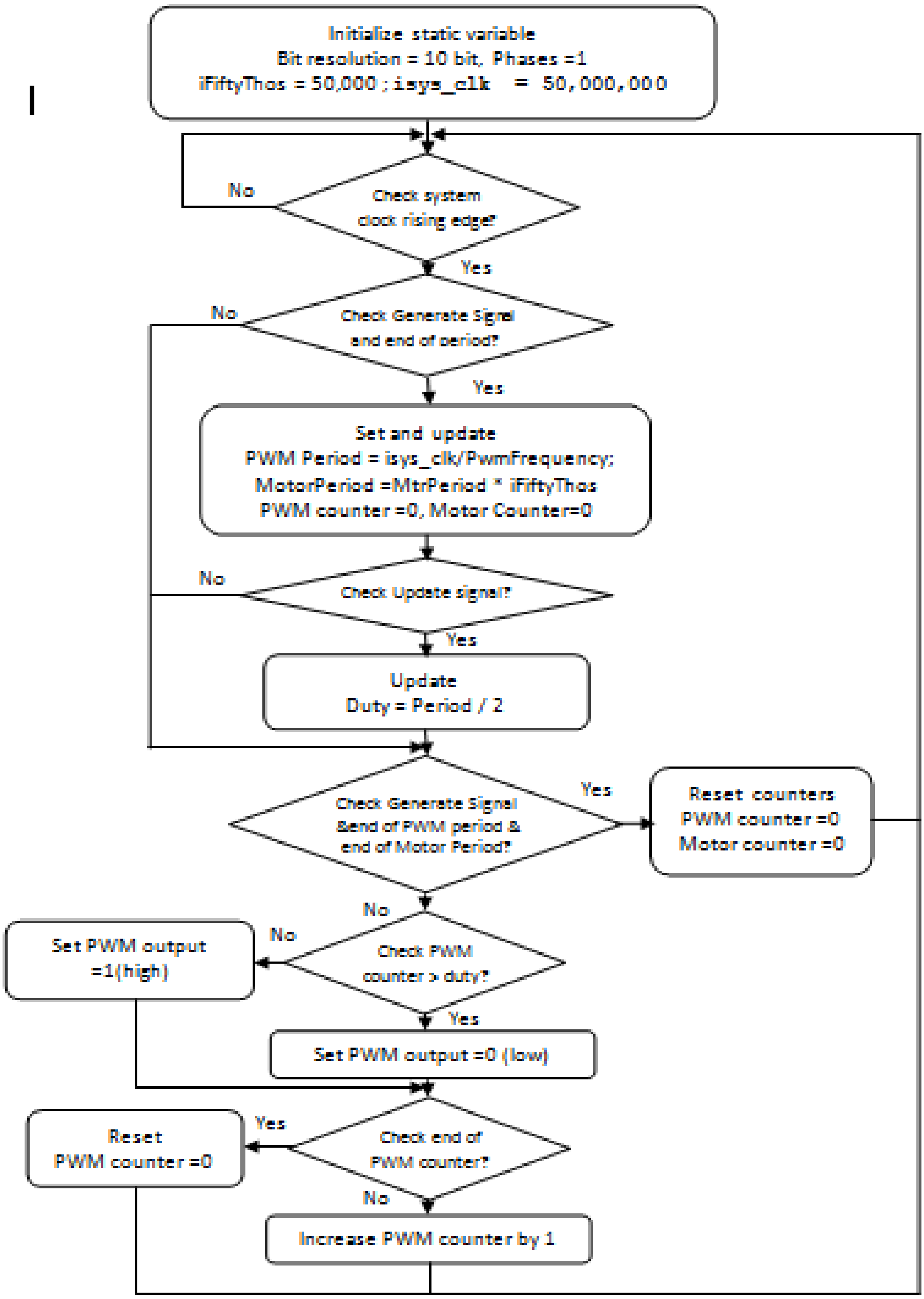

Figure 6.4 PWM signal generator flow chart

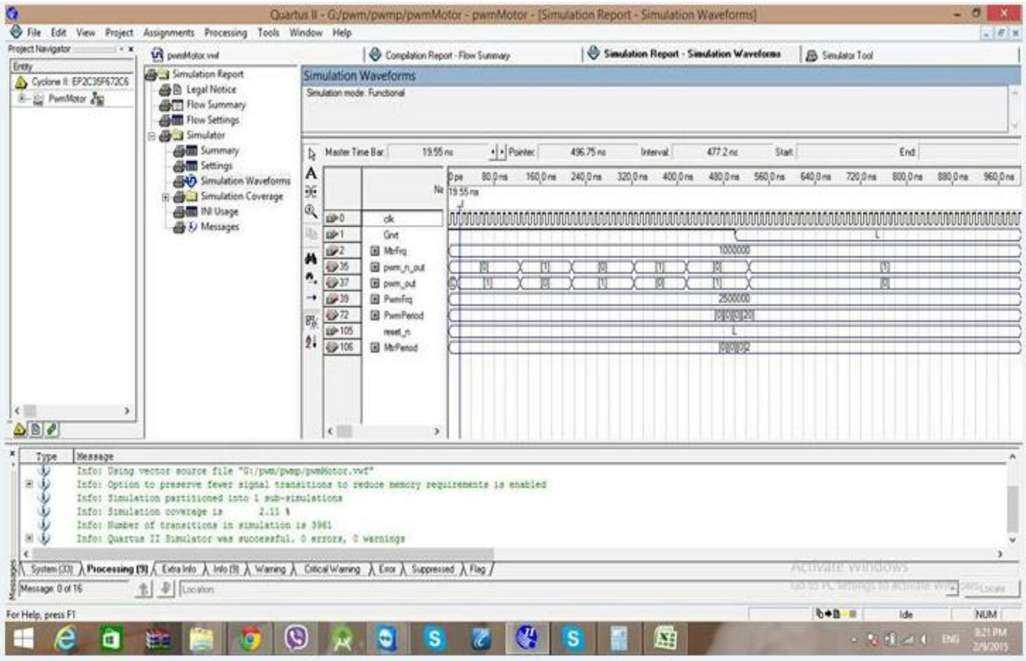

Figure 6.5 PWM signal in modelsim simulation

Data and control signals coming from FPGA for each motor, such as pulse, sign (motor direction data) and the movement lock logic data (servo on) of the servo motors were connected to the inputs and outputs of the involved axis drivers. Isolation circuit is used to isolate the power voltage level and FPGA voltage level because FPGA works in 3.3V. The input-output parameter modifications of the drivers were adjusted through the ASDA_Soft program and were installed in the servo drivers by utilizing the USB and RS232-USB convertors. The inputs-outputs entity inside the system is controlled by FPGA, and these data can be concurrently monitored by the operator on the SCADA screen. The commands given by the user through the software, SCADA in the PC environment were communicated over RS232 port over FPGA. After given commands, FPGA generates the control pulse that will allow the servo motors to move to the desired position as explained in motor flow chart in Figure 6.6. This signal is transmitted to the ‘’pulse’’ pins of the servo motor drivers. The operating clock of the designed FPGA controller is 50 MHz.

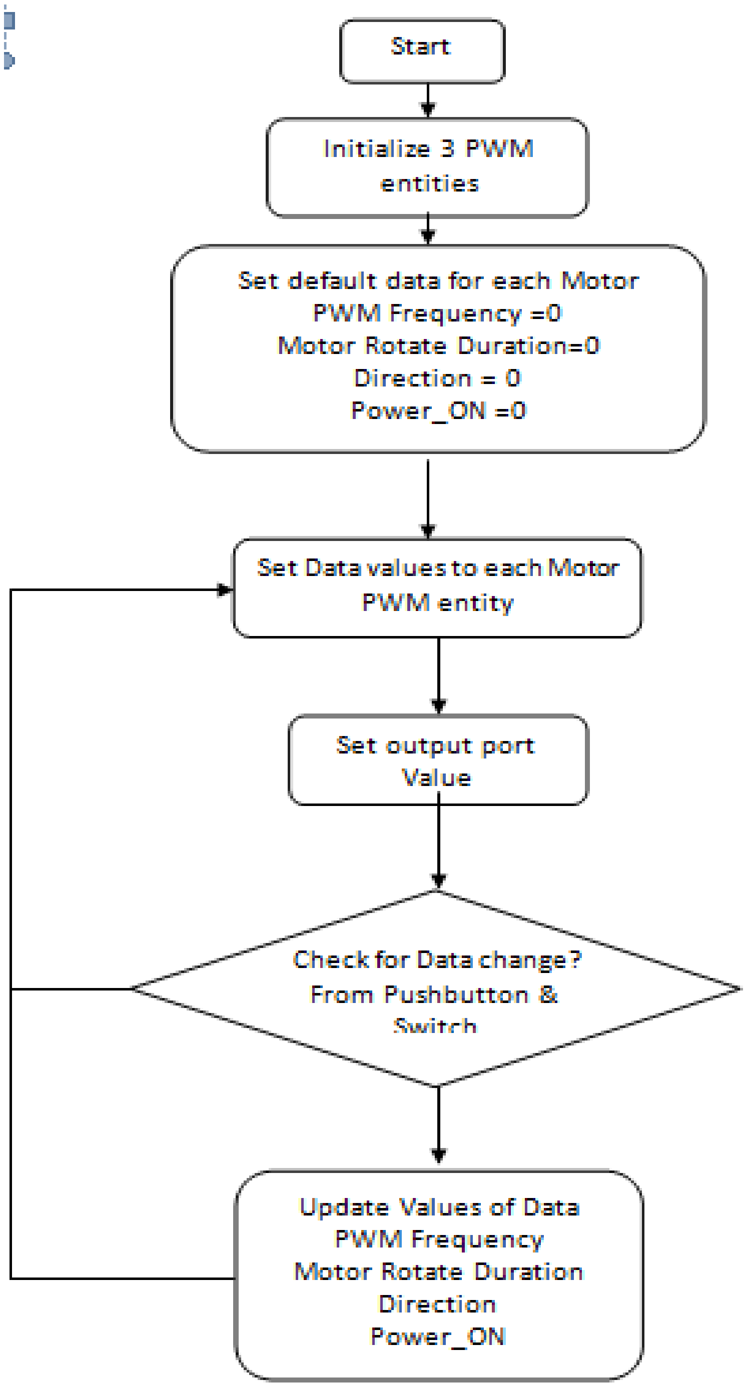

Figure 6.6 Motors controller flow chart

UART (Universal Asynchronous Receiver/Transmitter) is a computer hardware design that translates data from parallel to serial forms. UART is used in this study to program RS232 serial port in FPGA controller device with VHDL code. The UART serial communication section is separated into three sub-modules: the baud rate generator, receiver module and transmitter module, shown in Figure 6.7. Thus, the application of the UART communication module is actually the realization of the three sub-modules.

The baud rate generator is used to create a local clock signal which is should be higher than the baud rate to control the UART receive and transmit; The UART receiver module is used to receive the serial signals at RXD, and convert them into parallel data; The UART transmit module vagaries the bytes into serial bits according to the basic frame format and transmits those bits through TXD [52] this UART making the controller with FPGA quite fast.

Baud Rate Generator is actually a kind of frequency divider. The baud rate frequency factor can be calculated according to a given system clock frequency and the required baud rate. The calculated baud rate frequency factor is used as the divider factor.

Assume that the system clock is 50MHz, baud rate is 9600bps, and then the output clock frequency of baud rate generator should be 1* 9600Hz. Therefore the frequency coefficient (M) i.e. counts value of the baud rate generator is: M =50MHz/1*9600Hz=5208 When the UART receives serial data, it is very critical to determine where to sample the data information. The ideal time for sampling is at the middle point of each serial data bit [52].

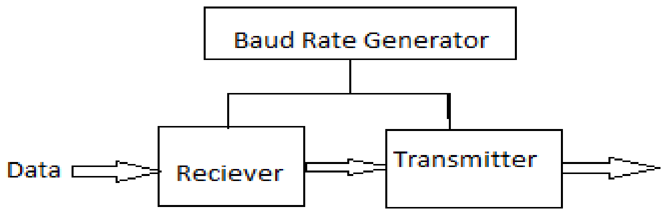

Figure 6.7 UART Module

Simulation result for UART using VHDL code is obtained in Figure 6.8 below:

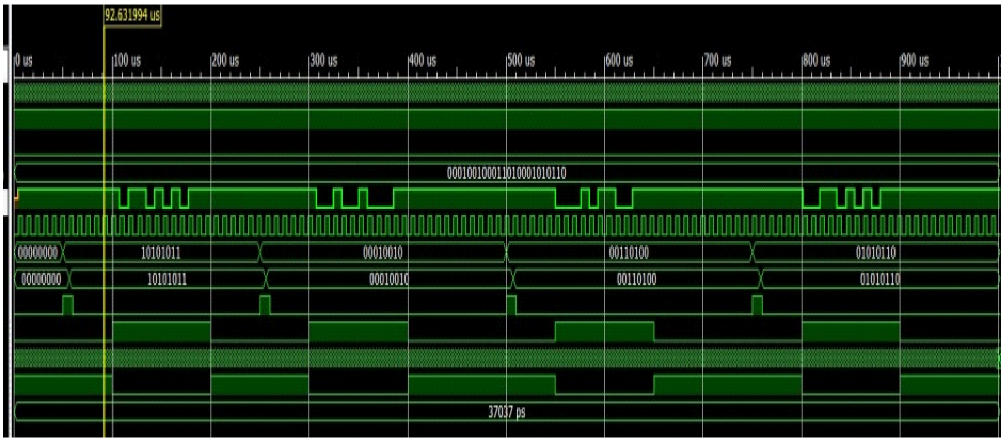

Figure 6.8 UART simulation result with modelsim program of FPGA

The pulse signals arriving at servo drivers are put into process in the digital signal processor (DSP) of the driver. The error that happens according to the operating mode of the driver (torque, internal position, external position, speed) is determined by the information coming from the motor encoder. The motors are controlled by the control signals processed in DSP and these signals are generated by the PID controller. An internal proportional-integral-derivative (PID) controller adjusts the rotor position using the received angle information.

The FPGA herein uses Cyclone IV-EP4CE115, which is the product of Altera cooperation. In addition, a Nios II processor is embedded into FPGA to establish an SoC environment. The motion control IC comprises a Nios II embedded processor and multiple position/speed/current controller IP for multi axis machine as shown in Figure 6.9. The Nios II processor is utilized to perform the motion trajectory computation [53].

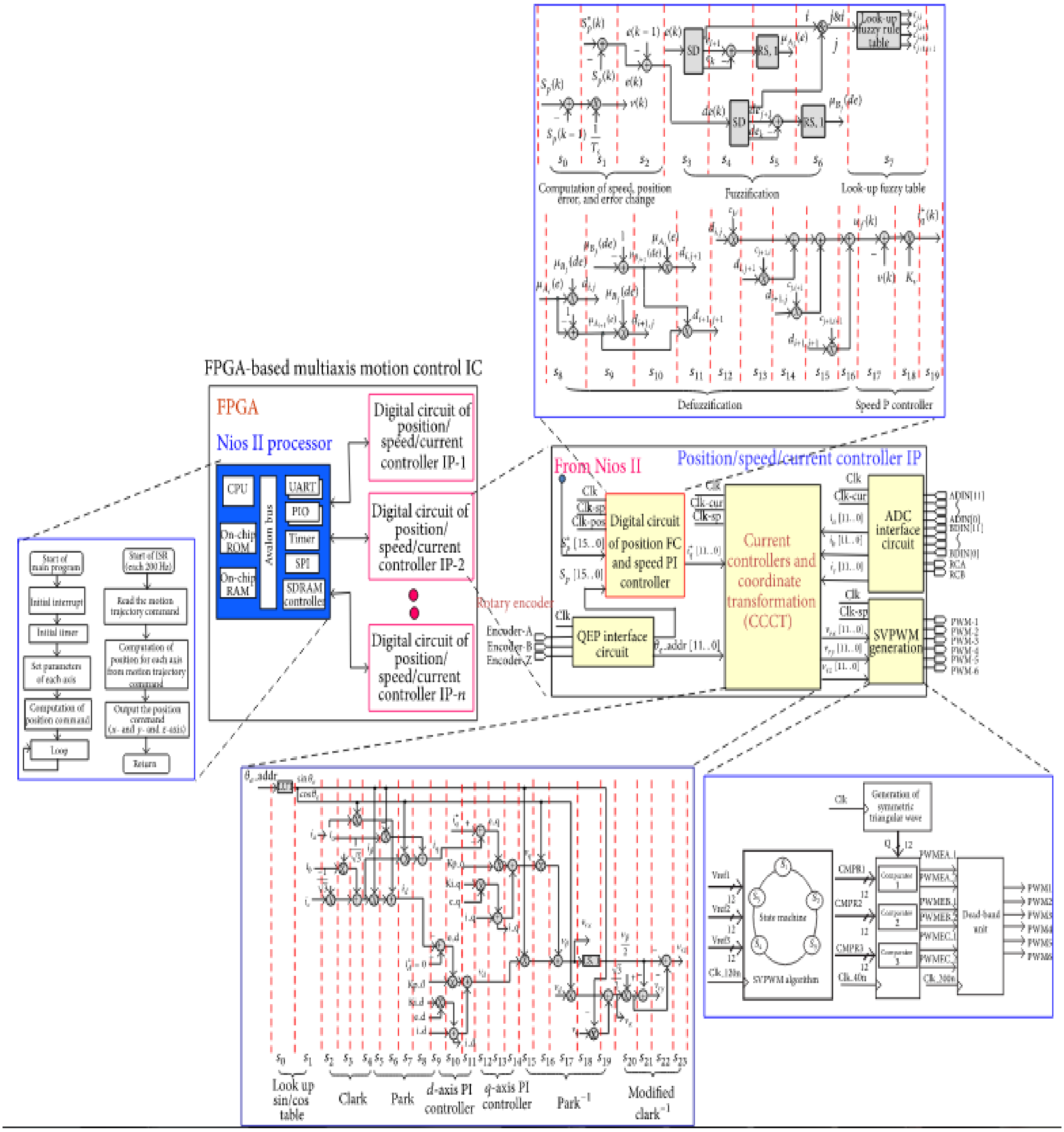

Figure 6.9 Internal circuit design of the proposed FPGA-based multi-axis motion control IC.

The Simulink/ModelSim cosimulation architecture for the three-axis motion control system is shown in Figure 6.11. Each axis system is driven by one servo motor. The EDA Simulator Link for ModelSim executes the co-simulation using VHDL code running in ModelSim program. Matlab m-code is used to develop the program of the motion trajectory [53]. By using DSP in Matlab combination with FPGA the Figure 6.10 is simulated and results in Figure 6.11 is obtained.

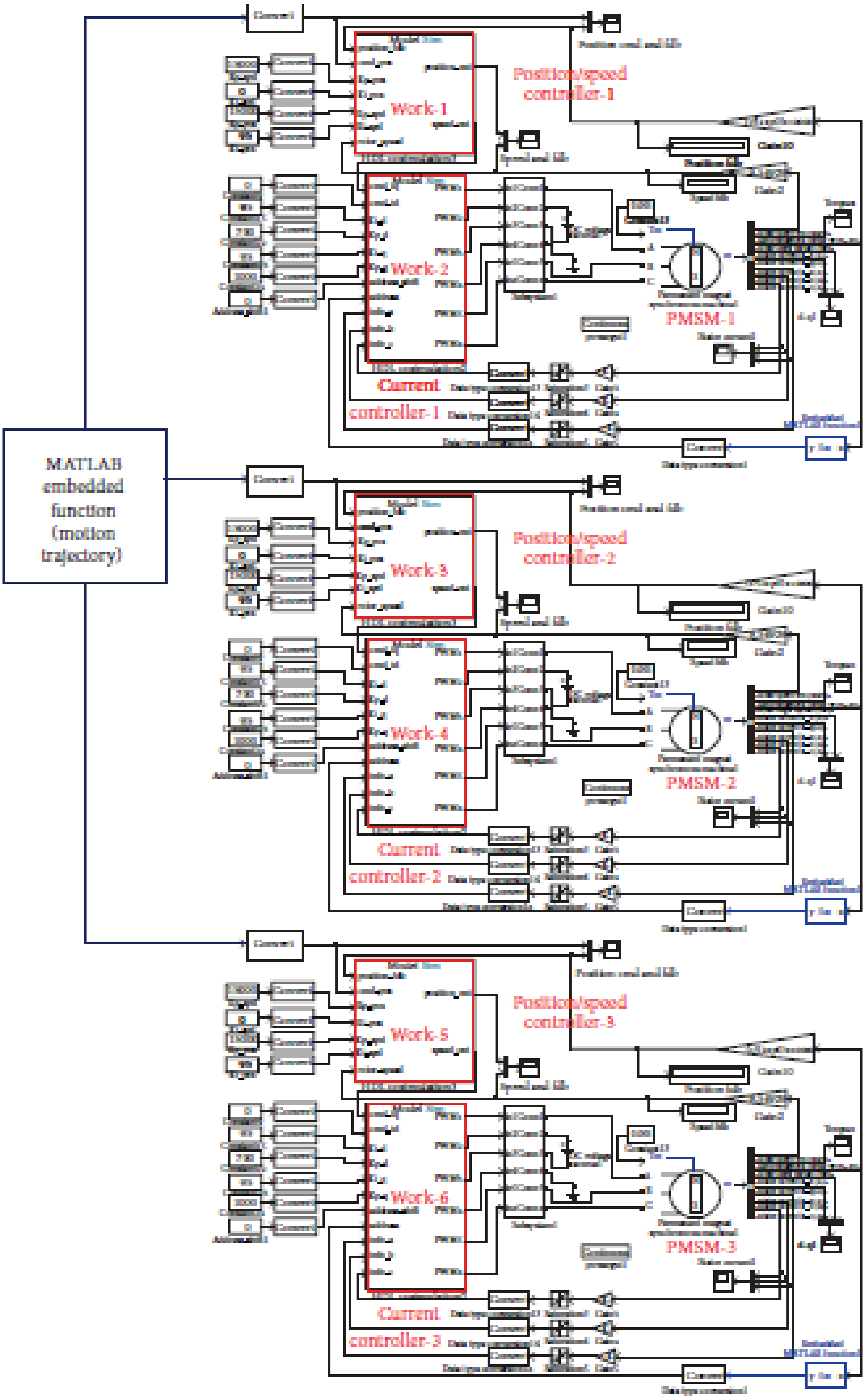

Figure 6.10 Simulink/ModelSim cosimulation architecture for three-axis motion control system [53].

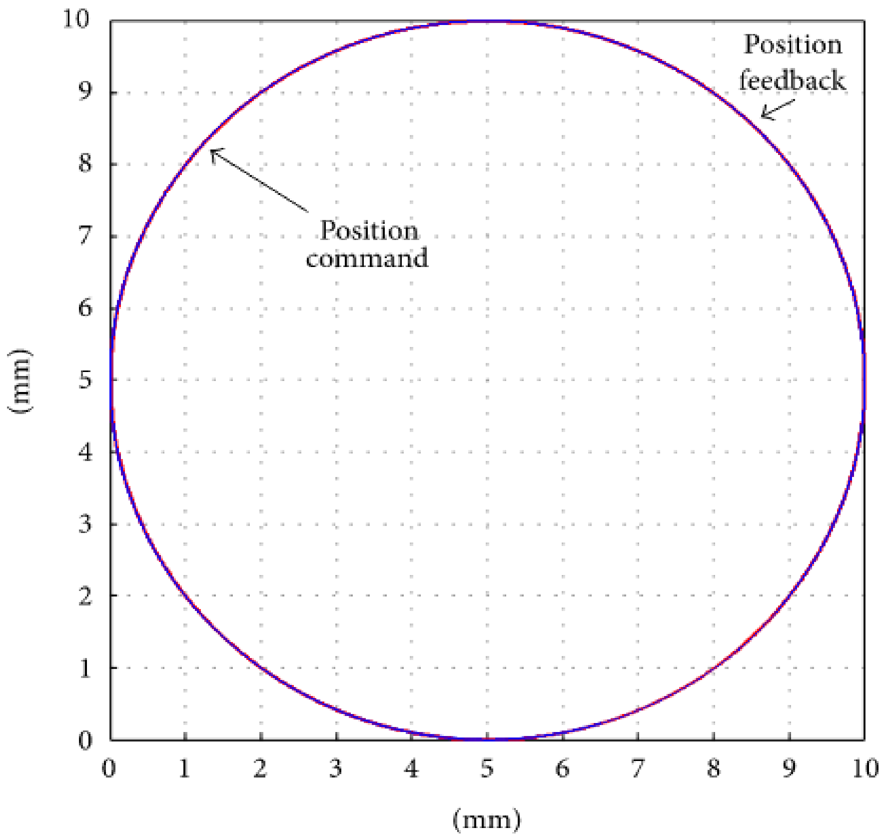

Figure 6.11 Circular motion trajectory tracking

The experimental setup results with 3-axis motion control are shown in Figure 6.12 and interface circuit (isolation circuit) designed is shown in Figure 6.13.PWM control signal is obtained in Figure 6.14 below:

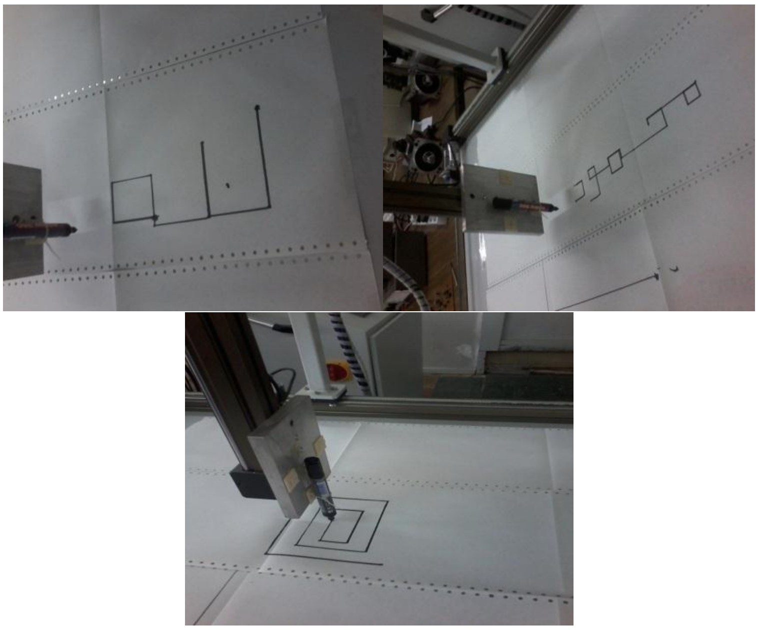

Figure 6.12 3-axis servo motors drawing result

Figure 6.13 FPGA and interface circuit

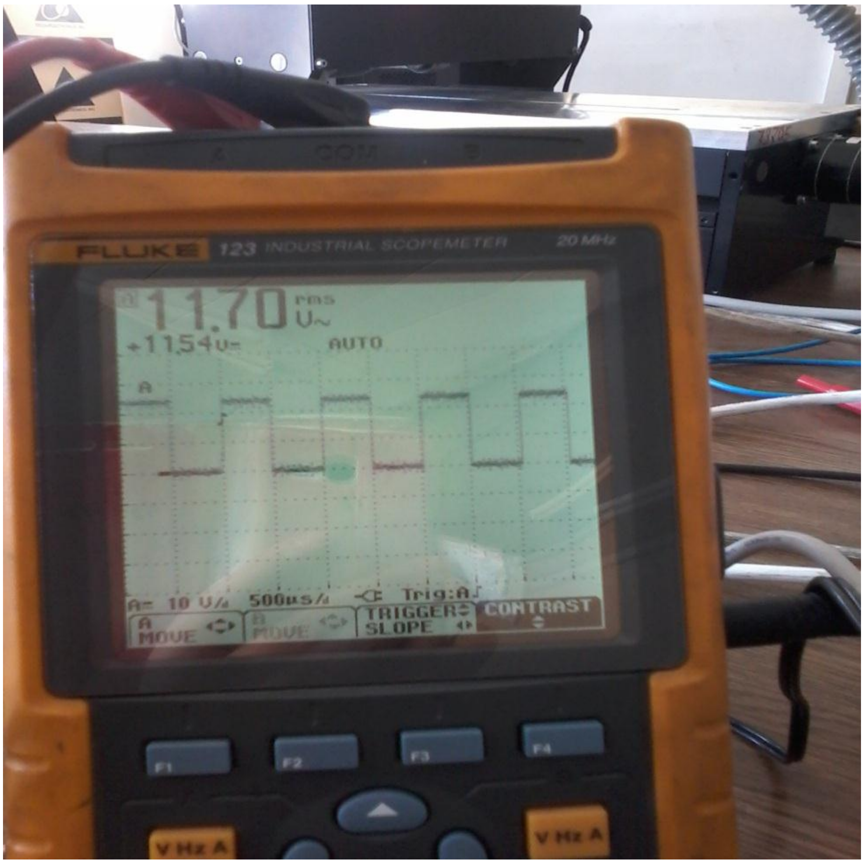

Figure 6.14 PWM output signal

Summary

Three axis mechanical systems with servo motors are controlled with FPGA controller. PWM signal is obtained using the modelsim simulation for FPGA to control motors. Using the switch in FPGA device, PWM signal is controlled. Control algorithm is done by using Very High Description Language (VHDL) code which is written using Quartus II software. Results and flow charts of the system are implemented. RS232 serial port is used for sending data from PC to FPGA and it has been programed using UART (Universal Asynchronous Receiver/Transmitter) is used for translating data from parallel to serial and it make the controller more faster. Finally, the work in this study is summarized as follows.

(a) The functionalities required to build a multi axis servo controller and motion trajectory planning can be integrated in one FPGA chip.

(b) The computation of key algorithms for PWM in servo drive can be completed by FPGA implementation.

(c) The VHDL code of the proposed multi axis servo controller IP has been effectively verified from ModelSim/ Simulink co-simulation results which shown in Figure 6.9, 6.10.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

The concept of digital control began in the forties of the twentieth century in response to a need in the advanced manufacturing techniques to run complex machines. Digital control technique is simply to control the types of digital machines application. Programmable digital control is used for motion control machines like CNC machine, robots, and other kind of machines that use the modern techniques.
Servo motor is one of the kinds of special motors (special machine) it used in control position control so it is sometimes called a control motor. Servo motor system model mathematically presented, output speed of servomotor is obtained and depending on parameter which obtained mathematically and controlled using PID controller.

3-axis servo motor is controlled practically using FPGA controller.

The FPGA is in the end a chip containing the inputs and outputs. Inputs are defined by use the command to the output port to show the output signal. This process is done by programming the FPGA device by a special programming language is VHDL (Very High Description Language).

Control algorithm is written by VHDL code and PWM control signal is obtained using QuartusII program. The stage of system control processing is achieved step by step using Quartus II software.

SCADA program is designed for making system modular. The benefits of system controlling using FPGA are high speed, low power conception, complex functionality and making the system more flexibility.

7.2 Future work

For future study it will be good to control 3D printer CNC machine with FPGA which need a lot of number of position which is better to saving these data FPGA memory by using SD card. And also 6-axsis robot motion controller can be controlled by FPGA because it's very modern, multi-purpose and effective technology.

REFERENCES

[1] **Ji,** J. K., and **S.** K. SUI.(1993). DSP-based self-tuning iP speed controller for rolling mill DC drive. *Proc. IECON ‘93 Conf.,* pp. 2276-2281.

[2] Dhaouadi, R., and N. Mohan. DSP-based control of **a** permanent magnet synchronous motor with estimated speed and rotor position. *EPE, 91*.

[3] Iwasaki, M., and N. Matsui. (1990). DSP-based high performance speed control system of vector control IM with load torque observer. *Proc. Int. Power Electronics* ***Conf.,*** pp. 436-441.

[4] O. A. Jasim, A. Z. Mansoor and M. R. Khalil. (2011). Position Control of DC Servo Motors Using Soft-Core Processor on FPGA to Move Robot Arm .*International Electrical Engineering Journal (IEEJ)*, Vol. 2 No. 3, pp. 555-559 ISSN 2078-2365.

[5] Rajendrasinh Navalsinh Bariya, Chetanya Sharma and Prof. KausalDoshi. (2013). 3-Axis Motion Control of CNC Machine based on G-Code, M-Code using FPGA and Also Apply Bezier Curve. *IJSRD - International Journal for Scientific Research & Development*, Issue 3, Vol. 1, 2321-0613.

[6] Sandra Veronica Moreno-Tapia, Luis Alberto Vera-Salas, Roque Alfredo Osornio-Rios, Aurelio Dominguez-Gonzalez, Ion Stiharu and Rene de Jesus Romero-Troncoso .(2010). A Field Programmable Gate Array-Based Reconfigurable Smart-Sensor Network for Wireless Monitoring of New Generation Computer Numerically Controlled Machines. *Sensors*, 7263-7286.

[7] Theerapong Fongjun and Kamonwan Tanta-ngai. (2011). Improving Efficiency of FPGA Position Control System. *Telecommunications*

and Information Technology (ECTI) Association of Thailand – Conference.

[8] K Ho, C Lai . (2005). PC-based Real-time Motion Controller Design using the FPGA Chip. *Proceedings of 2005 CACS Automatic Control Conferenceconference***.**

[9] Maciej Petko, Tadeusz Uhl. (2001). Embedded controller design – mechatronic approach. *Second Workshop on Robot Motion and Control.*

[10] Thomas James Coggins Marcian Nicolae Cirstea. (2005). FPGA Implementation of a fastflex Supervisory Control and Data Acquisition (SCADA) Remote Telemetry Unit. *IEE.*

[11] Yue Xu, Taiyong Wang, Yanju Zhao, Jingchuan Dong. (2008). Design of reconfigurable CNC system based on Embedded Technology. *Proceedings of the 7th World Congress on Intelligent Control and Automation, IEE.*

[12] Ming-Tzong, Lin Hong-Tzong Yau, Hao-Wei Nien and Meng-Shiun Tsai. (2008). FPGA-Based Motion Controller with Real-time Look-ahead Function, *Proceedings of the 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Xi'an, China.

[13] Kariyappa B. S, Dr. M. Uttara Kumari. (2008). FPGA Based Speed Control of AC Servomotor Using Sinusoidal PWM. *IJCSNS International Journal of Computer Science and Network Security,* VOL.8 No.10.

[14] Kariyappa B. S., Hariprasad S. A., and R. Nagaraj. (2009). Position Control of an AC Servo Motor Using VHDL & FPGA. *World Academy of Science, Engineering and Technology* ,Vol:3.

[15] C. Nwankpa, J. Johnson, P. Nagvajara, T. Chagnon and P. Vachranukunkiet. (2009). FPGA Hardware Results for Power System Computation. *IEEE.*

[16] Youdong Chen, Hongxing Wee, Kai Su, Tianmiao Wang, Yong Zou. (2009). System-on-Chip (SOC) Design for CNC System. *IEEE International Symposium on Industrial Electronics (ISlE 2009) Seoul Olympic Parktel*, Seoul, Korea.

[17] Claudio Greco, Enrico Nobile, Salvatore Pontarelli, Simone Teofili. (2010). An FPGA based architecture for complex rule matching with stateful inspection of multiple TCP connection. *IEEE* .

[18] Soroush Shirali, Shahab Ensafi, Mahsa Naseri. (2010). RTU Hardware Design for SCADA Systems Using FPGA. *International Conference on Computer Applications and Industrial Electronics (ICCAIE 2010),* Kuala Lumpur, Malaysia.

[19] Jianmin Zuo, Hao Zhu, Xinghua Zhu, Hua Yan. (2011). FPGA-based Communication Module in Numerical Control System with Hardening Technology. *International Conference of Information Technology, Computer Engineering and Management Sciences, IEEE.*

[20] Li Zhu, Yi_min Yang, (2011). A Programming and Practice on the simulation about CNC machining on two-dimensional graphics, *IEEE.*

[21] Mulan Wang, Baosheng Wang, Xinghua Zhu, Xiaoxia Li, (2011). Development of FPGA-based Arithmetic Module in CNC System. *International Conference on Mechatronic Science, Electric Engineering and Computer,IEEE,* Jilin, China.

[22] Shri Lakshmi Pravalika, Dr. Alex Noel Joseph Raj, (2013). FPGA Based Robotic Arm With Six Degrees of Freedom. *International Journal of Innovations in Engineering and Technology (IJIET),* Vol. 2 Issue 1, ISSN: 2319 – 1058.

[23] Ahmed BELKHEIRI, Said AOUGHELLANET, Mohammed BELKHEIRI. (2013). Reconfigurable three-phase SPWM

implementation on DE2 FPGA. *PRZEGLĄD ELEKTROTECHNICZNY*, ISSN 0033-2097.

[24] Dipali B. Bhatt , Hemant R. More. (2014). CNC machine controller using FPGA. *International Journal of Review in Electronics & Communication Engineering (IJRECE)*, Volume 2 - Issue 1, ISSN 2321-3159.

[25] Azfar Khalid, Amjad Nawaz. (2014). Sensor less Control of DC Motor using Kalman Filter for Low Cost CNC Machine. *International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE)*, *IEEE*, Islamabad, Pakistan.

[26] George W. Younkin. Electric servo motor equations and time constant. *IEEE*.

[27] Y.F. Chan, M. Moallem, W. Wang, (2004). Efficient Implementation of *PID* Control Algorithm using FPGA Technology. *43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas.

[28] Kariyappa B. S., Hariprasad S. A., and R. Nagaraj, (2009). Position Control of an AC Servo Motor Using VHDL & FPGA. *World Academy of Science, Engineering and Technology, International Scholarly and Scientific Research & Innovation* ,Vol:3.

[29] Dipraj, Dr. A. K. Pandey. (2012). Speed Control of D. C. Servo Motor By Fuzzy Controller. *International Journal Of Scientific & Technology Research*, volume 1, issue 8, ISSN 2277-8616.

[30] http://www.indiadiesel.com/bldc_const.htm.

[31] www.wikipedia.com .

[32] http://www.orientalmotor.com.

[33] http://www.orientalmotor.com/technology/articles/servo-motor-features.html

[34] Kazuya SHIRAHATA. Speed Control Methods of Various Types of Speed Control Motors.

[35] Selin Ozcira and Nur Bekiroglu, Direct Torque Control of Permanent Magnet Synchronous Motors, *Yildiz Technical University, Turkey*.

[36] www.ni.com NI tutorial.

[37] www.deltaww.com.

[38] Delta high resolution AC servo drive for network communication applications ASDA series user manual

[39] Ying-Shieh Kung, Member IEEE, Ming-Shyan Wang and Chung-Chun Huang. (2007). Digital Hardware Implementation of Adaptive Fuzzy Controller for AC Motor Drive. *The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pp.10208-1213, Taipei, Taiwan.

[40] Terasic DE2-115 User Manual www.terasic.com.

[41] V. Sornam Viswanathan. (2005). Embedded Control Using FPGA. Interdisciplinary Programme in Systems and Control Engineering.

[42] 6N137 datasheet, Toshiba, Available via URL: http://www.semicon.toshiba.co.jp/eng/product_detail/opto/coupler/1260156_13597.html.

[43] MIC4422, 9A-Peak Low-Side MOSFET Driver, Micrel, Available via URL: http://www.micrel.com/_PDF/mic4421.pdf

[44] Ali Osman ARSLAN. (2012). Control of Single Axis Electromagnetic Levitation System.

[45] Xie Weikun and Zhang Huibin Zhang Qiuli. (2013). Testing FPGA Devices on an Automatic Test Equipment. *Proceeding of the IEEE International Conference on Information and Automation*, Yinchuan, China.

[46] Quartus II software tutorial.

.

[47] MEHMET TANER KARSLI. 2014. Controlling Of a Mechanical System Having Three Dimensional Moving Capability by Using Servo Motors, PLC, and A Personal Computer

[48] Tuning ASDA Series Servo Systems 2010.

[49] J. Holtz. (1992). Pulse width modulation-A survey. *IEEE Trans. Ind. Electron*, vol 39, no. 5, pp. 410-420.

[50] Yang Gui-jie, Sun Li, Cui Nai-zheng and Lu Yong-ping. 2001. Study on method of the space vector PWM. *Proceedings of the CSEE*, China, vol.21, pp.79-83.

[51] Ziad Nouman, Bohumil Klima, Jan Knobloch. (2013). Generating PWM Signals With Variable Duty From 0% to 100% Based FPGA SPARTAN3AN. *Electro revue*, VOL.4, NO.4, ISSN 1213-1539.

[52] Ms.Neha R. Laddha, Prof.A.P.Thakare. (2013). Implementation of serial communication using UART with configurable baud rate. *International Journal on Recent and Innovation Trends in Computing and Communication*, Volume: 1 Issue: 4 ISSN 2321 – 8169.

[53] Ying-Shieh Kung, Jin-Mu Lin, Yu-Jen Chen and Hsin-Hung Chou. (2014). ModelSim/Simulink Cosimulation and FPGA Realization of a Multiaxis Motion Controller. *Hindawi Publishing Corporation,* Volume 2015, Article ID 202474, 17 pages.

APPENDEX 1

PWM VHDL code

```
؛ LIBRARY ieee

؛USE ieee.std_logic_1164.all

؛USE ieee.numeric_std.all


ENTITY PWM IS

)PORT

clk      : IN  STD_LOGIC;                              --system clock

reset_n   : IN  STD_LOGIC;                             --asynchronous reset

Gnrt      : IN  STD_LOGIC ;

MtrPeriod : IN unsigned(31 DOWNTO 0 ) ;

PwmFrq : IN unsigned(31 DOWNTO 0 )  ;

pwm_out   : OUT STD_LOGIC_VECTOR(0 DOWNTO 0);          --pwm outputs

pwm_n_out : OUT STD_LOGIC_VECTOR(0 DOWNTO 0 )

) ;                                                    --pwm inverse outputs

END PWM;


ARCHITECTURE logic OF PWM IS

SIGNAL half_duty : unsigned(31 DOWNTO 0):=x"00000000";                  --number of clocks in 1/2
duty cycle

signal GnrtSignal : STD_LOGIC := '0' ؛

signal isys_clk : unsigned(31 downto 0) := x"02FAF080";-- 50_000_000

SIGNAL iFiftyThos : unsigned(15 downto 0):="1100001101010000" ؛

SIGNAL MtrCnt  :  unsigned(31 DOWNTO 0):=x"00000000" ؛

SIGNAL iMtrPeriod  : unsigned(31 downto 0 )

SIGNAL iPwmPeriod  : unsigned(31 downto 0  );

SIGNAL iPwmCntr : unsigned(31 downto 0);
```

```
SIGNAL GnrtChk : STD_LOGIC :='0';

SIGNAL GnrtChng : STD_LOGIC :='0';

SIGNAL Update : STD_LOGIC :='1 ';

BEGIN

PROCESS(clk, reset_n)

BEGIN

IF(reset_n = '0') THEN                                --asynchronous reset

MtrCnt <= (OTHERS => '0');                              --clear counter

iPwmCntr <= (OTHERS => '0');

iPwmPeriod <= (OTHERS => '0 --

pwm_out <= (OTHERS => '0');                             --clear pwm outputs

pwm_n_out <= (OTHERS => '0');                           --clear pwm inverse outputs

GnrtSignal <= '0 ';

GnrtChng <='0 ';

Update <= '1';

GnrtChk <= '0 ';

ELSIF(clk'EVENT AND clk = '1') THEN                     --rising system clock edge

GnrtChng <= '0 ';

if(Gnrt = '1' and GnrtChk = '0') then

GnrtChng <= '1';

iPwmPeriod <=
isys_clk/PwmFrq;                                        -- update PWM period from new frequency

iMtrPeriod <=
Resize(MtrPeriod * iFiftyThos,32);

iPwmCntr <= ( others
=> '0'

MtrCnt <= ( others  );
=> '0'

end if ;

GnrtChk <= Gnrt ;

if(Gnrt = '1' and Update='1' ) then         -- if generate active and allow update

 --GnrtChng <='1''

-- Update <= '0';                           -- previlge updating
```

```
--iPwmPeriod <= isys_clk/PwmFrq;         -- update PWM period from new frequency
 --iMtrPeriod <= Resize(MtrPeriod * iFiftyThos,32);
 --end if؛
 --if ( GnrtChng ='1' and Update = '1') then      -- activate generate PWM after checking  GnrtChng
signal
GnrtChng <='0';
 half_duty <= iPwmPeriod/2;                             --update half duty Period
 GnrtSignal <= '1';
   Update <= '0';
      end if  ؛
     if ( GnrtSignal ='1' and MtrCnt  >= iMtrPeriod-1 and iPwmCntr = iPwmPeriod-1) then
     GnrtSignal <='0';
      MtrCnt <=(others => '0');
      iPwmCntr <=(others => '0');
     Update <='1؛'
      end if    ؛
       if( GnrtSignal ='1') then
        MtrCnt <= MtrCnt+1     ؛
      IF(iPwmCntr = iPwmPeriod - 1 ) then -- - i*period/phases) THEN             --end of period reached
       iPwmCntr <= (others => '0');                                        --reset counter
      ELSE                                                                --end of period not reached
        iPwmCntr <= iPwmCntr + 1;                                         --increment counter
      END IF؛
       IF(iPwmCntr > half_duty) THEN                                   --phase's falling edge reached
        pwm_out(0) <= '0';                                              --deassert the pwm output
        pwm_n_out(0) <= '1';                                             --assert the pwm inverse output
      else                                                               --phase's rising edge reached
        pwm_out(0) <= '1';                                               --assert the pwm output
        pwm_n_out(0) <= '0';                                         --deassert the pwm inverse output
      END IF؛
       end if؛
      END IF؛
```

```
END PROCESS;
 END logic;
```

APPENDIX 2

Motors VHDL Code

```
LIBRARY ieee؛

USE ieee.std_logic_1164.all؛

USE ieee.numeric_std.all؛

ENTITY Motor IS

 PORT)

    clk       : IN  STD_LOGIC;                                --system clock

    reset_n   : IN  STD_LOGIC;                                --asynchronous reset

    SelSet : IN STD_LOGIC_VECTOR(2 DOWNTO 0);

    SelValue   : IN STD_LOGIC_VECTOR(8 DOWNTO 0);

    Directions_IN :   IN STD_LOGIC_VECTOR(2 DOWNTO 0);

    Power_ON_IN :   IN STD_LOGIC_VECTOR(2 DOWNTO 0);

    Directions_OUT: OUT STD_LOGIC_VECTOR(2 DOWNTO 0);

    Power_ON_OUT: OUT STD_LOGIC_VECTOR(2 DOWNTO 0);

    pwm_out1   : OUT STD_LOGIC_VECTOR(0 DOWNTO 0);         --pwm outputs

    pwm_n_out : OUT STD_LOGIC_VECTOR(0 DOWNTO 0);

    pwm_out2   : OUT STD_LOGIC_VECTOR(0 DOWNTO 0);         --pwm outputs

    pwm_out3   : OUT STD_LOGIC_VECTOR(0 DOWNTO 0);         --pwm outputs

    LEDR   :OUT STD_LOGIC_VECTOR(17 DOWNTO 0);

    LEDG                :OUT STD_LOGIC_VECTOR(7 DOWNTO 0(

   );                    --         pwm inverse outputs

END Motor؛


ARCHITECTURE Mtr OF Motor IS
```

```
component  PWM is
            port)
              clk       : IN  STD_LOGIC;                              --system clock
     reset_n   : IN  STD_LOGIC;                              --asynchronous reset
     Gnrt      : IN  STD_LOGIC؛
     MtrPeriod : IN unsigned(31 DOWNTO 0);
     PwmFrq : IN unsigned(31 DOWNTO 0);
     pwm_out   : OUT STD_LOGIC_VECTOR(0 DOWNTO 0 (
                                    );
end component؛
            Signal iMtrFrq , iMtrFrq2, iMtrFrq3,cnttest :   unsigned(31 DOWNTO 0) := (others => '0');
            Signal iPwmFrq :   unsigned(31 DOWNTO 0):= (others => '0');
            Signal sys_clk        : INTEGER := 50_000_000; --system clock frequency in Hz
Signal PW1,PW2,PW3 : STD_LOGIC_VECTOR(0 DOWNTO 0);
Signal Pr : STD_LOGIC_VECTOR(31 DOWNTO 0);
 Signal iGnrt      :   STD_LOGIC     ؛
 begin
LEDG(0) <=  reset_n؛
LEDG(1) <= PW1(0);
LEDG(2) <= PW2(0);
LEDG(3) <= PW3(0);
LEDR <= Pr(17 downto 0);
pwm_out1 <= PW1؛
pwm_out2 <= PW2؛
pwm_out3 <= PW2؛
Directions_OUT <= Directions_IN؛
Power_ON_OUT <= Power_ON_IN؛
PWM1: PWM
port map              (
              clk      => clk,                             --system clock
     reset_n   => reset_n,                              --asynchronous reset
```

```
      Gnrt => Power_ON_IN (0),
      MtrPeriod => iMtrFrq،
      PwmFrq => iPwmFrq،
      pwm_out    => PW1
                        );
PWM2: PWM
port map             (
              clk      => clk,                          --system clock
      reset_n   => reset_n,                             --asynchronous reset
      Gnrt => Power_ON_IN (1),
      MtrPeriod => iMtrFrq2،
      PwmFrq => iPwmFrq،
      pwm_out    => PW2
                        );
 PWM3: PWM
port map             (
              clk      => clk,                          --system clock
      reset_n   => reset_n,                             --asynchronous reset
      Gnrt => Power_ON_IN (2،(
      MtrPeriod => iMtrFrq3،
      PwmFrq => iPwmFrq،
      pwm_out    => PW3
                        );
-iPwmFrq(13 downto 0) <= "00000000000101";
--iMtrFrq(13 downto 0) <= "01001110001000";
  --PROCESS(clk, reset_n(
  --BEGIN
   --IF(clk'EVENT AND clk = '1') THEN                         --rising system clock edge
    --cntTest <= cntTest +1؛
    --if ( cntTest > x"1DCD6500") then
--                         iGnrt <= '1';
```

```
--                    cntTest <= x"00000000"؛
--          else
     --Gnrt <= '0';
     --end if؛
--end if؛
--end process؛
process (SelSet,SelValue) is
begin
   case SelSet is
                    when "000"=>
                              iPwmFrq   <= unsigned("0000000000000000000000" & SelValue(8 downto 0) & "0")؛
                    when "001"=>
                              iPwmFrq(17 downto 9) <= unsigned(SelValue(8 downto 0));
                    when "010"=>
                              iMtrFrq  <= unsigned("0000000000000000000000" & SelValue(8 downto 0)& "0")؛
                    when "011"=>
                              iMtrFrq(17 downto 9) <= unsigned(SelValue(8 downto 0));
                    when "100"=>
                              iMtrFrq2 <= unsigned("0000000000000000000000" & SelValue(8 downto 0)& "0")؛
                    when "101"=>
                              iMtrFrq2(17 downto 9) <= unsigned(SelValue(8 downto 0));
                    when "110"=>
                              iMtrFrq3 <= unsigned("0000000000000000000000" & SelValue(8 downto 0)& "0")؛
                    when "111"=>
                              iMtrFrq3(17 downto 9) <= unsigned(SelValue(8 downto 0));

end case؛
end process؛
end Mtr;
```