

Sudan University of Science and Technology

College of Engineering

School of Electronics Engineering



CONTROL OF ROBOTIC ARM OVER FPGA

A Research Submitted In Partial Fulfillment for Requirements
of the Degree of B.Sc. (Honors) in Electronics Engineering

Prepared By:

- 1. Moaied Abdelmoniem Rahamta Allah**
- 2. Ala Eldien Abd Alaziz Mohammed Taha**
- 3. Tarig Sanhoury Abo Elgasim Mohammed**
- 4. Samer Alwaly Ibrahim Idris**

Supervised By:

Dr. Sami Hassan Omer Salih

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

قال تعالى:

(اللَّهُ نُورُ السَّمَاوَاتِ وَالْأَرْضِ مِثْلُ نُورِهِ كَمِشْكَاةٍ فِيهَا مِصْبَاحٌ
الْمِصْبَاحُ فِي زُجَاجَةٍ الزُّجَاجَةُ كَأَنَّهَا كَوْكَبٌ دُرِّيٌّ يُوقَدُ مِنْ شَجَرَةٍ
مُبَارَكَةٍ زَيْتُونَةٍ لَا شَرْقِيَّةٍ وَلَا غَرْبِيَّةٍ يَكَادُ زَيْتُهَا يُضِيءُ وَلَوْ لَمْ
تَمْسَسْهُ نَارٌ نُورٌ عَلَى نُورٍ يَهْدِي اللَّهُ لِنُورِهِ مَنْ يَشَاءُ وَيَضْرِبُ
اللَّهُ الْأَمْثَالَ لِلنَّاسِ وَاللَّهُ بِكُلِّ شَيْءٍ عَلِيمٌ)

صدق الله العظيم

سورة النور، ايه رقم ٥٣

With all love, we dedicate this work to our pressures mothers and fathers, may Allah give them peace and health.

Acknowledgement

First of all ,we are so thankful to Allah (subhanahu wa ta'al) for giving us the ability to finish our project and give us the success to complete this work.

We would also like to give our great thankful and appreciation to our supervisor *dr. Sami* for all the guides, advices, help, assistance and guidance that he gave to us to complete our project, and the patience with us all time of work.

Also, we would like to thank all of our electronic engineering's doctors and teachers for encouraging, explaining and standing next of us when we need them, especially in our hard times and solving our problems that faced us.

Last but not least, we would like to give our biggest thankful and appreciation to our parents for all the support, encouragement and efforts during our entire life until we reach this point.

Abstract

Robotics is a key technology in the modern world nowadays. Robots are become the most important elements in any manufacturing process in the most factories because of the high accuracy, safety that robots provides it to human and the stability of work with less errors, that's why robots takes the main role of almost every factory now days.

Most factories around the world began to turn their work from single-purpose factory to multi-purpose factory. Multi-purpose production provides many types of products, thus, the sales and profits will increase, but, it also requires a lot of equipments and robots, therefore, the cost will be high for the construction.

The main aim of our project is to control the movement of a material handling arm using FPGA controller with VHDL code. the advantages of FPGA over other controllers is that FPGA controller can provides the multi-purpose function that factories needs to reduce the cost of construction. FPGA is a fast and efficient system with high accuracy, parallel data processing and stability in performance, that's why we prefers it from another controllers.

المستخلص

علم الإنسان الآلي من التقنيات الرئيسية في عالمنا الحديث، الإنسان الآلي أصبح من أهم العناصر الأساسية في أي عملية تصنيع تحدث في معظم المصانع، وذلك بسبب ما يمتاز به من الدقة العالية وكما يوفره للإنسان الأمان والبعد من الخطر، أيضا يمتاز الإنسان الآلي بالإستقرارية في العمل مع عدد بسيط جدا من الأخطاء، لهذا أصبح في زمننا أصبحت تأخذ الأدوار الأساسية في المصانع.

أكثر المصانع حول العالم بدأت بالتغيير من مصنع ذو غرض ومنتج واحد إلى مصنع متعدد الأغراض والمنتجات، بوجود أكثر من منتج في مصنع واحد فإن هذا يؤدي إلى زيادة المبيعات وزيادة الفائدة، ولكن أيضا مثل هذه المصانع تتطلب الكثير من المعدات والاجهزة وبالتالي تكلفة عالية جدا.

الهدف من هذا المشروع هو التحكم في حركة ذراع الي باستخدام (متحكم مصفوفة البوابات القابلة للبرمجة). يمتاز عن غيره بالمتحكمات انه يمكن ان يوفر خاصية تعدد الأغراض وهي نفسها التي تحتاجها المصانع الحديثة لتقليل التكلفة. وهو متحكم ذو سرعة ودقة عالية، يمكنه تحليل البيانات بالتوازي (خاصية تعدد الأغراض) ولهذا تم تفضيله على باقي المتحكمات.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	الاستهلال	I
	Dedication	II
	Acknowledgement	III
	Abstract	IV
	المستخلص	V
	Table of contents	IX
	List of Figures	XII
	List of Tables	XIII
	List of Symbols	XIV
	Abbreviations	IXIV
1	INTRODUCTION	
	1.1 Preface	2
	1.2 Problem statement	4
	1.3 Proposed solution	4
	1.4 objectives	4
	1.5 methodologies	5
	1.6 Thesis outlines	5

2	CONTROL AND ROBOTS	
	2.1. Background	7
	2.1.1. Control system	7
	2.1.2. Types of Control System	8
	2.1.2.1. Open Loop Control System	9
	2.1.2.2. Closed Loop Control System	10
	2.1.2.3 Feedback Loop of Control System	11
	2.2 Robot	11
	2.3 Servomotor	12
	2.3.1 Types of Servo Motors	12
	2.4 Related Work	14
3	FPGA	
	3. FPGAs	17
	3.1 FPGAs in industry	17
	3.2 Comparisons between the controllers and FPGAs	19
	3.2.1 FPGAs versus DSPs	19
	3.2.2 FPGAs versus Microcontroller	20
	3.2.3 FPGA versus ASIC	21
	3.3 Controlling of servo motors by FPGAs	22
	3.3.1 Servo motors	22
	3.3.2 Generating PWM	23

	3.4 PWM counters in our project	26
4	IMPELEMENTATION	
	4. Implementation	28
	4.1. PC Software	28
	4.2. Hardware environment	30
	4.2.1 Kit A	30
	4.2.2 Kit B	31
	4.2.3 The arm	34
	4.3 Software environment	35
	4.3.1 Procedure for Designing in Xilinx ISE	35
5	CONCOLUSION & RECOMMENDATIONS	
	5.1. Conclusion	37
	5.2 Recommendation	37

LIST OF TABLES

TAPLE NO.	TITLE	PAGE
3 -1	Comparison between FPGA and controllers	20

LIST OF FIGURE

FIGURE NO.	TITLE	PAGE
2-1	open loop system	
2-2	closed loop system	10
2-3	positions of servo motor depending on clock input	13
3-1	servo motor	23
3-2	duty cycle of PWM	25
3-3	principle PWM	26
4-1	GUI main frame	29
4-2	GUI motor speed meters frame	30
4-3	ST112A - kit A	31
4-4	ST112B - kit B	32
4-5	Control signal from the GUI	33
4-6	Robot arm	34

LIST OF SYMBSLS

D - **duty cycle**

τ - **on time**

T - **total time**

ABBREVIATIONS

ISO	international organization for standardization
PWM	pulse width modulation
CPLD	complex programmable logic device
VHDL	VHISC hardware description language
FPGA	field programmable get array
VLSI	very large scale integration
ASIC	application specific integrated circuit
CNC	computer numerical control
RTL	register transfer level
UCF	user constraints file

CHAPTER ONE
INTRODUCTION

CHAPTER ONE

1. Introduction

1.1 Preface

Industry is the production of goods or services within an economy. Manufacturing industry became a key sector of production and labor in European and North American countries during the Industrial Revolution, upsetting previous mercantile (was an economic theory and practice, dominant in Europe from the 16th to the 18th century) and feudal economies. This occurred through many successive rapid advances in technology, such as the production of steel and coal.[1]

The industrial revolution led to the development of factories for large scale production, with consequent changes in society.[4] Originally the factories were steam-powered, but later transitioned to electricity once an electrical grid was developed. The mechanized assembly line was introduced to assemble parts in a repeatable fashion, with individual workers performing specific steps during the process. This led to significant increases in efficiency, lowering the cost of the end process. Later automation was increasingly used to replace human operators. This process has accelerated with the development of the computer and the robot.[2]

An industrial robot is defined by ISO 8373 as an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes. The field of robotics may be more practically defined as the study, design and use of robot systems for manufacturing .[1]

Industrial robots are programmed to faithfully carry out specific actions over and over again (repetitive actions) without variation and with a high degree of accuracy. These actions are determined by

programmed routines that specify the direction, acceleration, velocity, deceleration, and distance of a series of coordinated motions. Other robots are much more flexible as to the orientation of the object on which they are operating or even the task that has to be performed on the object itself, which the robot may even need to identify.[2]

The most type of robots that used in factories is the robotic arms. A robotic arm is a type of mechanical arm, usually programmable, with similar functions to a human arm; the arm may be the sum total of the mechanism or may be part of a more complex robot. The links of such a manipulator are connected by joints allowing either rotational motion (such as in an articulated robot) or translational (linear) displacement.[3]

The end effectors, or robotic hand, can be designed to perform any desired task such as welding, gripping, spinning etc., depending on the application. For example, robot arms in automotive assembly lines perform a variety of tasks such as welding and parts rotation and placement during assembly.[6]

A material handling robot is a type of robotic arms that moves, selects or packs products. These robots are also used to automate feeding or disengaging parts or tools from one location to another. Manufacturers raise the safety level of their facilities. Material handling is usually tedious, dull and sometimes injury-inducing. Material handling robots get human workers out of the tedium and into more fulfilling jobs in the company.

1.2 Problem statement

In Industrial environment human interaction may be limited due to hazard or unsafe areas. Such as nuclear reactors, or any area that contain a harmful rays or high temperature it could be very dangerous for a human to do the job and a serious injuries could happen.

Also when high precise manipulation is required human interaction will fail.

Thus, robots arms are introduces to modern factories. Most firms are designed to produce various products. So, a multi-purpose robots are utilized in order to shrink the CAPEX and OPEX .

This required a multipurpose controller as well. To provide maximum flexibility while assure/maintain high precision .

1.3 Proposed solution:

FPGA can be the ideal solution because it a multi-purpose controller, so it will be easy to change the function of the arm.

Also FPGA characterized by high response and it's more stable than other controllers.

1.4 Objectives

- To design a graphical user interface that let the user send his instructions to the robotic arm and the way of movement that he wants.
- To add a wireless transmission system that sends the control signals with standard IEEE 802.11.

- To design a system that receive the control signals and calculate the frequency of the PWM and send it to the servo motors .

1.5 Methodology

Microsoft visual basic will be used to design a PC software that will send the control signal via universal serial port (USB) to CPLD controller.

VHDL code will be written to program the CPLD controller, this controller will receive control signals by USB port then send it to a FPGA controller via a wireless transmission. The standard that will be used in wireless transmission is IEEE 802.11/g .

VHDL code will be written to program the FPGA , this controller will receive control signals. Then calculate the frequencies of the PWM depending on control signals which will be send to servo motors .

1.6 Thesis outlines

Chapter one : Introduction

Chapter two : control and robot

Chapter three :FPGA

Chapter four: implementation

Chapter five: Conclusion and Recommendations

CHAPTER TWO

OVERVIEW OF CONTROL AND ROBOTS

CHAPTER TWO

2. Control and robots

2.1 BACKGROUND

2.1.1 Control System

A control system is a device, or set of devices, that manages, commands, directs or regulates the behavior of other devices or systems. Industrial control systems are used in industrial production for controlling equipment or machines.[7]

As the human civilization is being modernized day by day the demand of automation is increasing accordingly. Automation highly requires control of devices. In recent years, control systems plays main role in the development and advancement of modern technology and civilization. Practically every aspects of our day-to-day life is affected less or more by some control system. A bathroom toilet tank, a refrigerator, an air conditioner, a geezer, an automatic iron, an automobile all are control system. These systems are also used in industrial process for more output. We find control system in quality control of products, weapons system, transportation systems, power system, space technology, robotics and many more. The principles of control theory is applicable to engineering and non-engineering field both.[8]

The main feature of control system is, there should be a clear mathematical relation between input and output of the system. When the relation between input and output of the system can be represented by a linear proportionality, the system is called linear control system. Again when the relation between input and output cannot be represented by

single linear proportionality, rather the input and output are related by some non-linear relation, the system is referred as non-linear control system.[8]

2.1.2 Types of Control Systems

There are various types of control system but all of them are created to control outputs. The system used for controlling the position, velocity, acceleration, temperature, pressure, voltage and current etc. Are examples of control systems. Let us take an example of simple temperature controller of the room, to clear the concept. Suppose there is a simple heating element, which is heated up as long as the electric power supply is switched on. As long as the power supply switch of the heater is on the temperature of the room rises and after achieving the desired temperature of the room, the power supply is switched off. Again due to ambient temperature, the room temperature falls and then manually the heater element is switched on to achieve the desired room temperature again. In this way one can manually control the room temperature at desired level. This is an example of manual control system. This system can further be improved by using timer switching arrangement of the power supply where the supply to the heating element is switched on and off in a predetermined interval to achieve desired temperature level of the room. There is another improved way of controlling the temperature of the room. Here one sensor measures the difference between actual temperature and desired temperature. If there is any difference between them, the heating element functions to reduce the difference and when the difference becomes lower than a predetermined level, the heating elements stop functioning[9].

Both forms of the system are automatic control system. In former one the input of the system is entirely independent of the output of the system. Temperature of the room (output) increases as long as the power supply switch is kept on. That means heating element produces heat as long as the power supply is kept on and final room temperature does not have any control to the input power supply of the system. This system is referred as open loop control system[9].

But in the latter case, the heating elements of the system function, depending upon the difference between, actual temperature and desired temperature. This difference is called error of the system. This error signal is fed back to the system to control the input. As the input to output path and the error feedback path create a closed loop, this type of control system is referred as closed loop control system.

Hence, there are two main types of control system. They are as follow:

1\Open loop control system

2\Closed loop control system

2.1.2.1 Open Loop Control System

A control system in which the control action is totally independent of output of the system then it is called open loop control system. Manual control system is also an open loop control system. Fig - 1

shows the block diagram of open loop control system in which process output is totally independent of controller action[10]

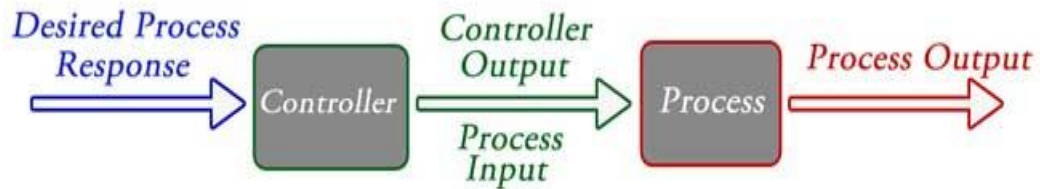


FIGURE 2-1: Examples of Open loop

2.1.2.2. Closed Loop Control System

Control system in which the output has an effect on the input quantity in such a manner that the input quantity will adjust itself based on the output generated is called closed loop control system. Open loop control system can be converted in to closed loop control system by providing a feedback. This feedback automatically makes the suitable changes in the output due to external disturbance. In this way closed loop control system is called automatic control system. Figure below shows the block diagram of closed loop control system in which feedback is taken from output and fed in to input.[10]

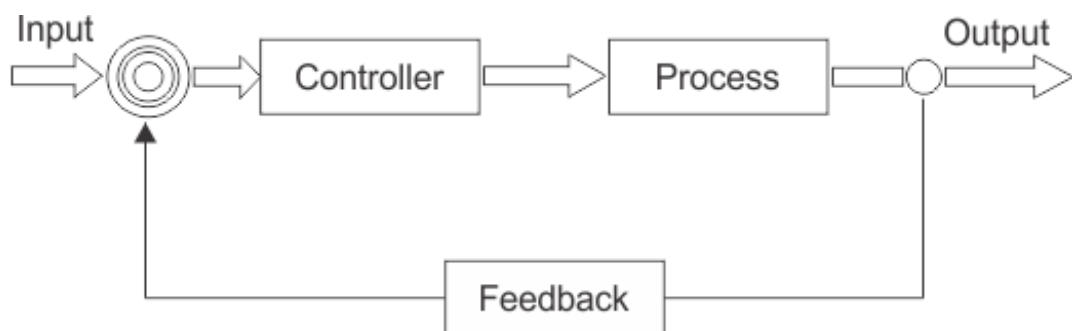


FIGURE 2-2: Examples of closed loop 1

2.1.2.3. Feedback Loop of Control System

A feedback is a common and powerful tool when designing a control system. Feedback loop is the tool which take the system output into consideration and enables the system to adjust its performance to meet a desired result of system[10].

In any control system, output is affected due to change in environmental condition or any kind of disturbance. So one signal is taken from output and is fed back to the input. This signal is compared with reference input and then error signal is generated. This error signal is applied to controller and output is corrected. Such a system is called feedback system. Figure below shows the block diagram of feedback system [11].

2.2. Robot

A robotic arm is a type of mechanical arm, usually programmable, with similar functions to a human arm; the arm may be the sum total of the mechanism or may be part of a more complex robot. The links of such a manipulator are connected by joints allowing either rotational motion (such as in an articulated robot) or translational (linear) displacement.[12][13] The links of the manipulator can be considered to form a kinematic chain. The terminus of the kinematic chain of the manipulator is called the end effector and it is analogous to the human hand.

The end effector, or robotic hand, can be designed to perform any desired task such as welding, gripping, spinning etc., depending on the application. For example, robot arms in automotive assembly lines

perform a variety of tasks such as welding and parts rotation and placement during assembly. In some circumstances, close emulation of the human hand is desired, as in robots designed to conduct bomb disarmament and disposal [14].

2.3. Servomotor

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. [15] It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors..Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.

Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing. [15]

2.3.1. Types of Servo Motors

There are two types of servo motors: AC and DC. AC servos can handle higher current surges and tend to be used in industrial machinery. DC servos are not designed for high current surges and are usually better suited for smaller applications. Generally speaking, DC motors are less expensive than their AC counterparts. These are also servo motors that have been built specifically for continuous rotation, making it an easy way to get your robot moving. They feature two ball bearings on the

output shaft for reduced friction and easy access to the rest-point adjustment potentiometer.[16]

Servos are controlled by sending an electrical pulse of variable width, or pulse width modulation (PWM), through the control wire. There is a minimum pulse, a maximum pulse and a repetition rate. A servo motor can usually only turn 90° in either direction for a total of 180° movement. The motor's neutral position is defined as the position where the servo has the same amount of potential rotation in the both the clockwise or counter-clockwise direction. The PWM sent to the motor determines position of the shaft, and based on the duration of the pulse

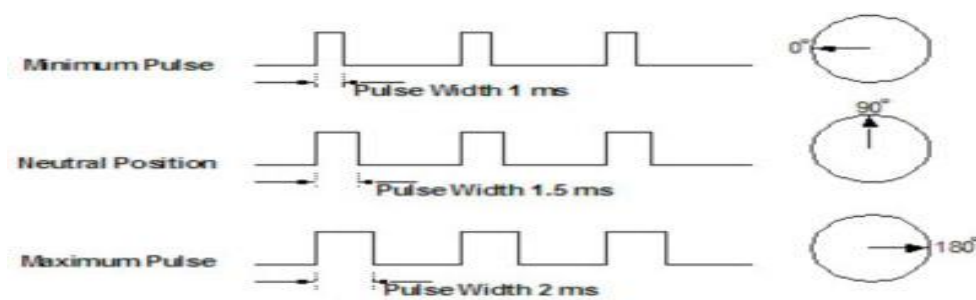


FIGURE 2-3: positions of servo motor

sent via the control wire the rotor will turn to the desired position. The servo motor

expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position. Shorter than 1.5ms moves it to 0° and any longer than 1.5ms will turn the servo to 180° .

When these servos are commanded to move, they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo

can exert is called the torque rating of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position[17].

2.4. RELATED WORK

There were many literatures in controlling a robot arm. In 2009, Spartan II FPGA is used to control the movement of three stepper motors that created a small robot arm with three joints, the aim of project is to change the degrees and the position of the small arm with control signal sent from FPGA chip by using digital PWM signals, each stepper motor is connected with a hardware circuit to supply the motor with enough power to run [18].

.In august 2010 , a field programmable get array FPGA is used to control five axes robot arm that performed pick and replace operation , digital PWM is used to control the speed an position of dc motors , it was manual control operation by interfacing the small robot arm with the FPGA hardware circuit which contains number of bush buttons to change the speed and the position of the small arm [19].

In November 2010, a robot arm is controlled by personal computer PC using graphical user interface GUI , a robot arm with six degrees of freedom are interfaced with hardware microcontroller circuit that received the control signal from a personal computer PC to control the speed and position of several servo motors [20].

In February 2013, in order to improve the functions of the robot arms, a multipurpose robotic arm which can perform operations with higher accuracy with 6 degrees of freedom is controlled by Spartan II FPGA by Shri Lakshmi. The improvement of the functions of the robot appears in the automation which applied in the VHDL code in the FPGA chip [21].

In September 2014, a simple robot arm with three degrees of freedom is been controlled by FPGA chip, an educational robotic arm that contains three servo motors receives control signal from the FPGA chip to control the movement of the servo motors in order to control the position of the arm, also digital PWM are used to change the speed and the position of the servo motors. [22]

CHAPTER THREE
OVERVIEW OF FPGA

CHAPTER THREE

3. OVERVIEW OF FPGA

FPGA technology allows developing specific hardware architectures within a flexible programmable environment. This specific feature of the FPGAs gives designers a new degree of freedom comparing to microprocessor implementations, since the hardware architecture of the control system is not imposed a priori. However, in many cases, the development of this architecture is rather intuitive and not adapted to the implementation of more and more complex algorithms. Thus, in order to benefit from the advantages of the FPGAs and their powerful CAD tools, the designer has to follow an efficient design methodology. Such a methodology rests on three main principles: the control algorithm refinement, the modularity and the best suitability between the algorithm to implement and the chosen hardware architecture. These three concepts are detailed thereafter. [23]

When designing industrial electronics circuits, several criteria have to be considered. Some of the most significant are: the cost, the power consumption (essential in the case of embedded systems), the application performance and above all, the suitability for the chosen hardware technology to match the requirements of the algorithm to implement. [23]

3.1 FPGAs in industry

FPGA technology is now considered by an increasing number of designers in various fields of application such as wired and wireless telecommunications image and signal processing where the always

more demanding data throughputs take advantage of the ever increasing density of the chips. Still, more recently, other application fields are in growing demand, such as medical equipment robotics automotive and space and aircraft embedded control systems . For these embedded applications, reduction of the power consumption thermal management and packaging reliability and protection against solar radiations are of prime importance. Finally, industrial electrical control systems are also of great interest because of the ever increasing level of expected performance while at the same time reducing the cost of the control systems .[24]

FPGAs have already been used with success in many different electric system applications such as power converter control (PWM inverters power factor correction multilevel converters matrix converters soft switching and STATCOM) and electrical machines control (induction machine drives SRM drives motion control multi-machines systems Neural Network control of induction motors Fuzzy Logic control of power generators speed measurement). This is because an FPGA-based implementation of controllers can efficiently answer current and future challenges of this field. Amongst them, we can quote:

- Decrease of the cost for at least three reasons: the use of an architecture based only on the specific needs of the algorithm to implement, the application of highly advanced and specific methodologies improving implementation time also called "time to market", and the expected development in VLSI design that will allow integrating a full control system with its analog interface in a single chip, also called System-on-a-Chip (SOC); [25]

- Confidentiality, a specific architecture, integrating the know-how of a company, is not easily duplicable; [25]

- Embedded systems with many constraints as in aircraft applications, like limited power consumption, thermal consideration, reliability and Single Event Upset (SEU) protection;[26]

- Improvement of control performance. For example, execution time can be dramatically reduced by designing dedicated parallel architectures, allowing FPGA-based controllers to reach the level of performance of their analog counterparts without their drawbacks (parameter drifts, lack of flexibility). Besides, an FPGA-based controller can be adapted in run-time to the needs of the plant by dynamically reconfiguring it.[26]

3.2 Comparisons between the controllers and FPGAs



Technology	Performance/Cost	Time until running	Time to high performance	Time to change code functionality
ASIC	Very High	Very Long	Very Long	Impossible
Custom Processor/DSP	Medium	Long	Long	Long
FPGA	Low-Medium	Short	Short	Short
Generic	Low-Medium	Short	Not Attainable	Short

Table 3 - 1: Comparison

3.2.1 FPGAs versus DSPs

Digital signal processors are a specialized form of microprocessor, while FPGAs are a form of highly configurable hardware. In the past, the use of digital signal processors was nearly ubiquitous, but with the needs of many applications outstripping the

processing capabilities of digital signal processors (measured in millions of instructions per second (MIPS)), the use of FPGAs is growing rapidly. Currently, the primary reason most engineers choose use a FPGA over a digital signal processors is driven by the application's MIPS requirements. Thus, the comparison between digital signal processors and FPGAs focuses on MIPS comparison, which, while certainly important, is not the only advantage of an FPGA. Equally important, and often overlooked, is the FPGA's inherent advantage in product reliability and maintainability. [28]

The large electronic design automation (EDA) industry continually drives the development of FPGA and ASIC test and verification tools. It does not have a comparable counterpart in the software development world .This may change, as the industry realizes the enormous costs and challenges in software verification, but for now, the practical software solution is to keep downloading the latest patch.[28]

Most engineering managers understand that the rate of software updates to remedy bugs far exceeds the rate of comparable FPGA updates in nearly all cases. It is expected and normal to roll out bug fixes on embedded software on a regular basis. With the availability of both low-cost and high-end DSP optimized FPGA devices, extensive IP cores, availability of high-level design entry methods, and the inherent robustness of the design and verification process, FPGAs will increasingly be the preferred choice for implementing DSP.[28]

3.2.2 FPGAs versus Microcontroller

As for the difference between a microcontroller and a FPGA, you can consider a microcontroller to be an ASIC which basically processes

code in FLASH/ROM sequentially. You can make microcontrollers with FPGAs even if it's not optimized, but not the opposite. FPGAs are wired just like electronic circuits so you can have truly parallel circuits, not like in a microcontroller where the processor jumps from a piece of code to another to simulate good-enough parallelism. However because FPGAs have been designed for parallel tasks, it's not as easy to write sequential code as in a microcontroller.[27]

For example, typically if you write in pseudo code "let C be A XOR B", on a FPGA that will be translated into "build a XOR gate with the lego bricks contained (lookup tables and latches), and connect A/B as inputs and C as output" which will be updated every clock cycle regardless of whether C is used or not. Whereas on a microcontroller that will be translated into "read instruction - it's a XOR of variables at address A and address B of RAM, result to store at address C. Load arithmetic logic units registers, then ask the ALU to do a XOR, then copy the output register at address C of RAM". On the user side though, both instructions were 1 line of code. If we were to do this, THEN something else, in VHDL we would have to define what is called a Process to artificially do sequences - separate from the parallel code. Whereas in a microcontroller there is nothing to do. On the other hand, to get "parallelism" (tuning in and out really) out of a microcontroller, you would need to juggle with threads which is not trivial. Different ways of working, different purposes.[29]

3.2.3 FPGA versus ASIC

Field Programmable Gate Arrays (FPGAs) and Application Specific Integrated Circuits (ASICs) provide different values to designers, and they must be carefully evaluated before choosing any one over the other. Information abounds that compares the two technologies.

While FPGAs used to be selected for lower speed/complexity/volume designs in the past, today's FPGAs easily push the 500MHz performance barrier. With unprecedented logic density increases and a host of other features, such as embedded processors, DSP blocks, clocking, and high-speed serial at ever lower price points, FPGAs are a compelling proposition for almost any type of design.[30]

The FPGA design flow eliminates the complex and time-consuming floor planning, place and route, timing analysis, and mask / re-spins stages of the project since the design logic is already synthesized to be placed onto an already verified, characterized FPGA device. However, when needed, Xilinx provides the advanced floorplanning, hierarchical design, and timing tools to allow users to maximize performance for the most demanding designs.[31]

3.3 Controlling of servo motors by FPGAs

3.3.1 Servo motors



Figure 3-1: servo motor

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration.[5] It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.

Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

As the name suggests, a servomotor is a servomechanism. More specifically, it is a closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is some signal, either analogue or digital, representing the position commanded for the output shaft.

The motor is paired with some type of encoder to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller. If the output position differs from that required, an error signal is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signal reduces to zero and the motor stops.

PWM controls the angle of the rotation as we will see .

3.3.2 Generating PWM

Pulse width modulation (PWM) is a powerful technique for

controlling analog circuits with a processor's digital outputs. PWM is employed in a wide variety of applications, ranging from measurement and communications to power control and conversion .[32]

The applications of PWM are used in control systems where the PWM signals were used to convert unregulated DC voltage to regulated or variable DC voltage at the output. [33]

A PWM signal is not constant, the main parameter is a dutycycle D that is a part of PWM period and describes the proportion of on time to regular interval. The equation (1) describes the duty cycle as the following:

$$D = \frac{\tau}{T} \quad (1)$$

Where: $0 \leq D \leq 1$

Thus the output signal is calculated in equation (2):

$$output = D \times input = \frac{t_{on}}{t_s} \times input \quad (2)$$

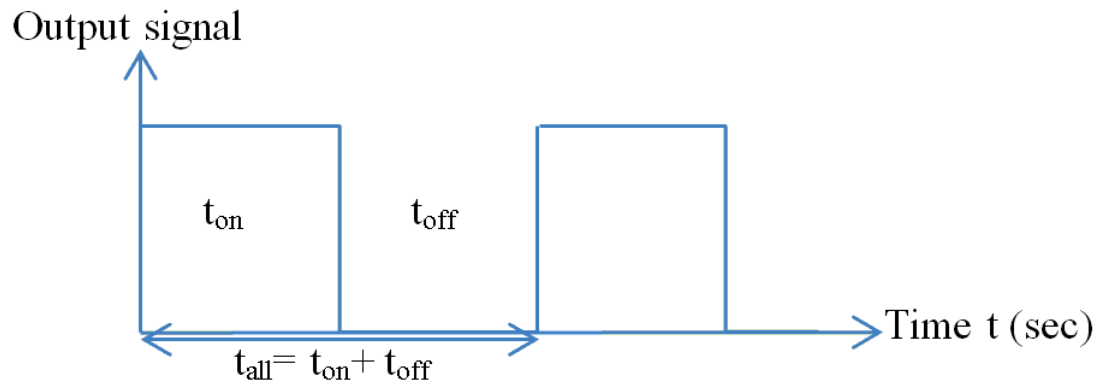


figure 3-2 : duty cycle of PWM

The principle of PWM work is required to design PWM. Figure 3-3 illustrates the principle of work :

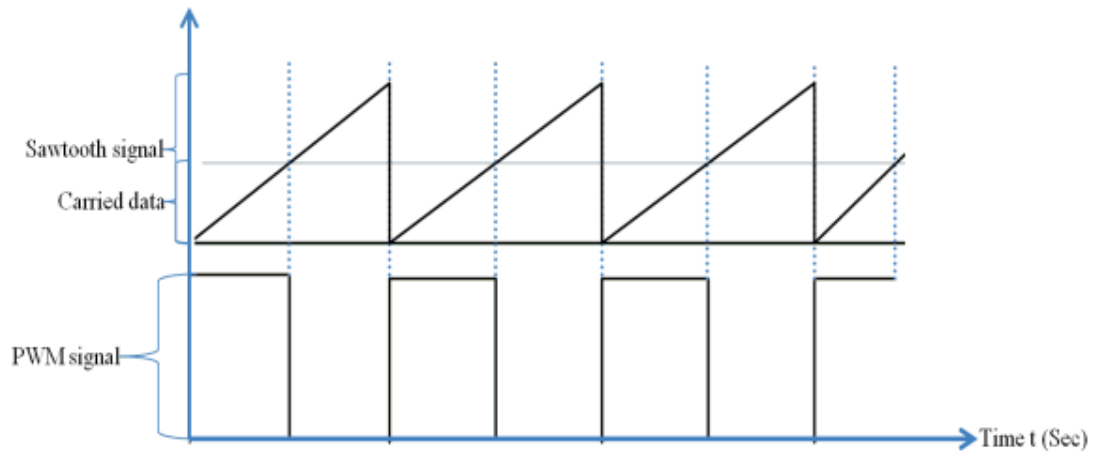


Figure 3 - 3: principle PWM

Figure 5 illustrates the architecture of PWM counter. This counter is used to generate the signal using 8bits.

As figure 3 – 3 shown the design of processor that generates the signals, which give the output of PWM, requires a comparator that compares between two values. The first value represents the sawtooth signal and the second value represents the data that is entered.

In generating PWM by FPGA we use the counters . first we take not of the frequency of the oscillator that supply the main clock of the FPGA .

Next we choose the frequency of the PWM and calculate the Number of divisions of the main clock .[34]

The code of this counter is written using VHDL. The counter increases one bit every CLKDV rising and it counts for example we use 8 bits from $(00000000)_2$ to $(11111111)_2$. The maximum value is $(255)_{10}$, when PWM counter arrives to this value, it returns to zero.

There must be a comparator that The output of comparator will be PWM generator. When the counter reach a particular value for example the output will be 1 and when it reach another particular value the output will be 0 .this changes of the output create the PWM with a specific frequency.

In our project we use four counters to divide a 8MHZ into variable values of PWM from 1ms minimum to 3ms maximum

3.4 PWM counters in our project

At first we generate a 1 MHz by a counter with a 19 bits . the counter increase by one every clock cycle . And when it reaches all ones it will reset so it will be all zeros . the 1MHz PWM we took it from bit number 2 of the counter.

So now we have a clock with a 1MHz frequency we use it like the Previous counter to generate 0.09 ms .

And then we generate 0.081 ms . the last frequency which we use it to generate our PWM values from . Therefore to generate 1 ms we use a loop that count 13 times .And for the 2 ms we count 25 .

As we will see later the number of count will be sent from the program in the PC .

CHAPTER FOUR

IMPLEMENTATION OF THE SYSTEM

CHAPTER FOUR

4. Implementation of the system

Our project contains of two parts: a pc software and a hardware components .in this chapter we will discuss all aspects related to designing of the system.

4.1 PC Software

Microsoft visual basic 2010 has been used to design a control unit to send control signals from pc to the wireless transmitter kit .

The control unit let the user choose whether he wants a manual or a automatic control of the arm. In the manual control there are six buttons : up, down, left, right, open and close. The first four buttons controls the direction of the arm ,open and close controls the end effector of the arm.

In the automatic mode the arm do a sequence of instructions that has been programmed previously .

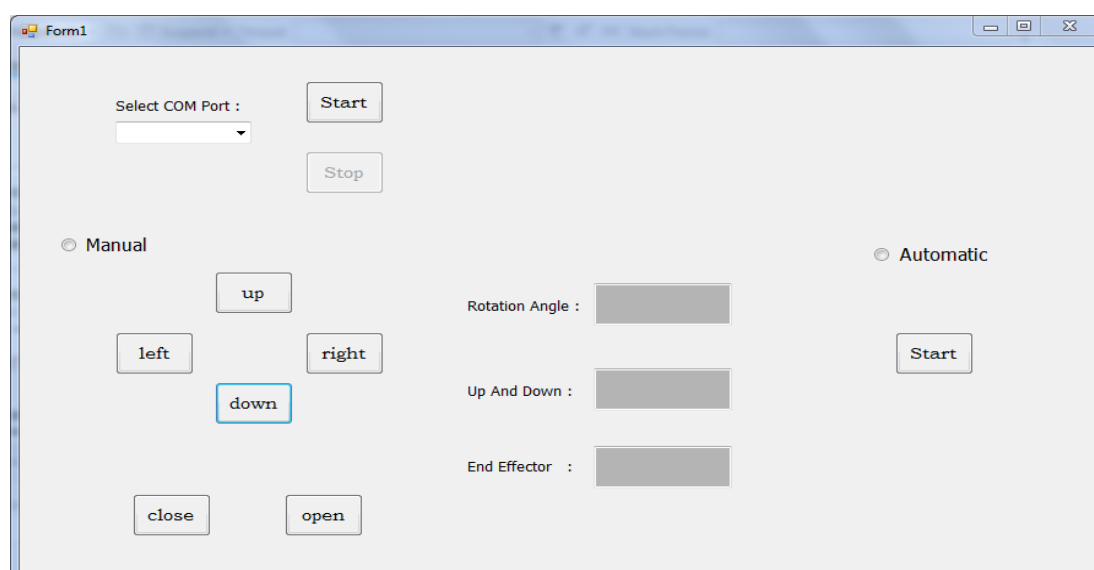


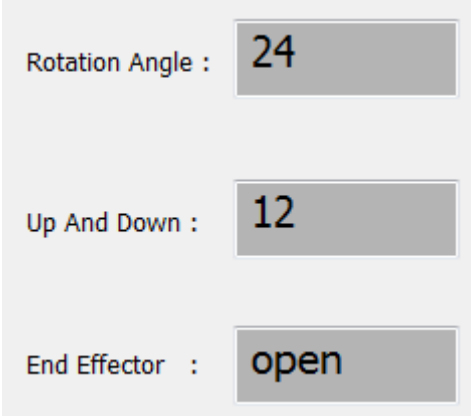
Figure 4-1: PC software window

Universal serial port(USB interface) has been used to transfer the control signal from the pc to the wireless transmitter kit.

Control signal contains of 8 bits from (7 down to 0). Bits number seven and six has been used to select the intended servo motor. If "10" that refers to rotation servo (left and right) , "01" refers to up and down servo and "11" refers to the end effector servo.

The rest of the bits has been used to send the angle of the intended servo . It contains a number from "001111" to "111111" in binary. When this number arrives to the wireless receiver kit it determines the frequency of the PWM that controls the intended servo as we will see later.

Also the software show the user what the current angles of the three motors.



Rotation Angle :	24
Up And Down :	12
End Effector :	open

Figure 4-2: GUI servo angles

The user cannot do anything until he select the specific COM port which is connected with the transmitter kit . the combo box contains the names of the COMs.

```
myPort = IO.Ports.SerialPort.GetPortNames()  
ComboBox1.Items.AddRange(myPort)
```

And when the user press (start) button the specific COM will open and now it's ready to transferring control signals.

4.2. Hardware environment

In our project we have three sections that related to hardware environment : kit A , kit B and the arm.

4.2.1 Kit A

We can name it wireless transmitter kit . it contains of a Xilinx CPLD controller and wireless transmitting unit. We can summarize its characteristics as below :

- Xilinx family : CPLD, XC95108PC84
- Device density : 2400 gates, 108 macro cells
- On board : +5V, 3.3V, 2.5V supply
- On board : 8 MHz crystal
- Configuration method : Boundary scan
- I/Os : 63
- USB port



Figure 4-3: ST112A - kit A

A VHDL code has been written to program the CPLD. The control signal comes to the kit by the USB interface. And then this control signal will be send by wireless .

The stander that we used in the wireless transmission is IEEE 802.11.

In order to program the CPLD JTAG interface must be connected from the LPT in the PC. Programming will be with the ISE software this will be discuss later.

4.2.2 Kit B

This kit contains two sections. The first one is receiving control signals by the wireless module. It receives the 8 bits that the user sent it from the PC software. And the second section is to generating the PWM and sends it to the motors. It contains of a Xilinx Spartan 2 controller and wireless module. We can summarize the characteristics as below:

- Xilinx family : Spartan 2, XC2S50PQ208
- Device density : 50 K gates, 1728 logic cells
- On board : +5V, 3.3V, 2.5V supply
- On board : 8 MHz crystal
- Configuration method : slave serial
- Number of I/Os : 136
- USB port



Figure 4-4: ST112B – Kit b

A VHDL code has been written to receives the 8 bits and save it in a variable.

And in the second section in order to generating a PWM we divide the main clock from 8 MHz to 1 MHz and then to 0.081ms. this frequency is suitable for generate our PWM on the way we want.

Then we make a counter depending on the final frequency that we generated and we have an output . the output become 1 when the counter start and become 0 depending on the control signal. When the counter equal to the specific number which was sent in the control signal the output become 0. So we with this process we generate a PWM with a particular frequency.

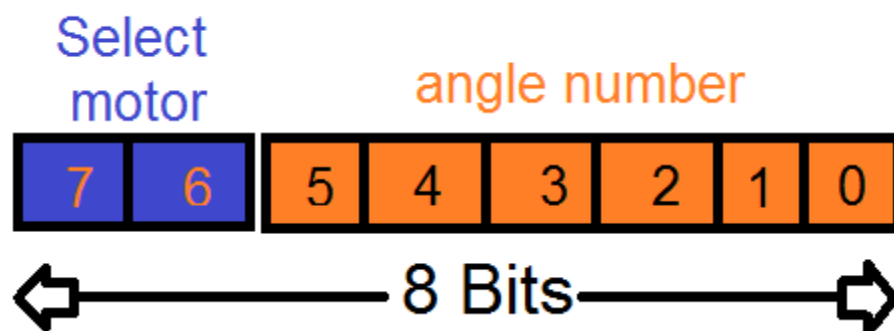


Figure 4-4: control signal from the GUI

There are three counters such as that has discussed. Any one generate a PWM with a different frequency.

The first two bits (MSB and 6) selects the intended motor as we said previously.

Range of the angle number is between 13 (001101)B up to 63 (111111)B. From the datasheet of the servo motor : 1ms gives us 0° and when we calculate this :

$$\frac{\text{wanted frequency}}{\text{counter frequency}} = \frac{1 \text{ ms}}{0.081} = 12.3 \approx 13$$

And also from datasheet 2ms gives us 90° so the calculation become :

$$\frac{\text{wanted frequency}}{\text{counter frequency}} = \frac{2 \text{ ms}}{0.081} = 24.6 \approx 25$$

So the step that we used become:

$$25 - 13 = 12$$

$$\frac{90^\circ}{12} = 7.5^\circ$$

so the angle increase by 7.5° degree therefore the maximum angle will be 226° when the angle number become 63.

4.2.3 The arm

Contains of two joints any of them has a servo motor and an end effector also has a servo to handle materials. It must connect to 220v power supply and it have a regulator to adjust the suitable voltage for driving the motors.

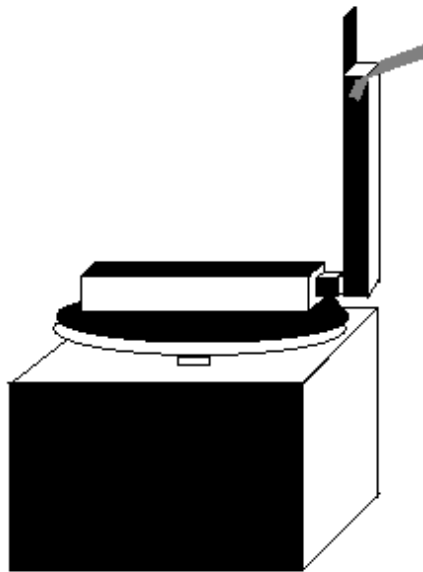


Figure 4-5: the robot arm

4.3 Software environment

Both FPGA and CPLD which we used in our project need a software environment to be programmed. Xilinx ISE 8.1i software has been used to do the configurations.

Integrated Synthesis Environment (ISE) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

4.3.1 Procedure for Designing in Xilinx ISE

Step one : opening the software and from file selecting new project and then choosing a name for it and then pressing on "Next"

Step two : Selecting the Family, Device and Package of the chips.

- Family : Spartan2 (For kit B)
: XC9500 CPLD (For kit A)
- Device : XC2S50 (For kit B)
: XC95108(For kit A)
- Package : PQ208 (For kit B)
: PC84 (For kit A)

Step three : assign new sources whether VHDL or UCF.

Step four : summary will appear and then clicking "Finish". And then adjusting the impact properties.

Step five : double click on Impact and choosing whether boundary scan (for kit A) or slave serial (for kit B). and then programming the controller.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONES

CHAPTER FIVE

5. CONCLUSION AND RECOMMENDATIONES

5.1 Conclusion

We designed the PC software by Microsoft visual basic and determined the values of the control signals. Then we tested the output from the USB port and we found that it was the correct values that we wanted.

After that we wrote the VHDL codes which we used it in the configurations of the controllers. ISE Xilinx software environment had been used in the configurations.

firstly we tested the wireless transmission and we made sure that it was working in proper way.

Secondly we tested the PWM generating it's was working in proper way also. We tested manual and automatic control.

So we made sure that all system was working as it was supposed to be. And there was a very fast response.

5.2 Recommendations

We recommend for the future to adding an intelligent system. It can be detect the locations of materials without been programmed by the user.

The control can be close-loop by adding a feedback that to adjust the output and making sure all angles are correct.

And decrease the step of the servos less than 7 degree to be more accurate in handling materials.

REFERENCES

Reference

- [1]Data Sheet For TS-53 Servo's ELECTRIC Motor BY: Darren Sawicz
- [2] Francis, Robert J., Jonathan Rose, and Zvonko G. Vranesic. Field programmable gate arrays. Vol. 180. Springer, 1992..
- [3] ISO Standard 8373:1994, Manipulating Industrial Robots Vocabulary .[2] An Automatic Block-Setting Crane". Meccano Magazine (Liverpool UK: Meccano) 23 (3): 172. March 1938.
- [4] Taylor, Griffith P. (1995). Robin Johnson, ed. The Robot Gargantua. Gargantua: Constructor Quarterly.
- [5]. Ruskin J (1995) Selected writings. JM Dent-Everyman, London
- [6]. Moran ME (2003) Robotic surgery: urologic implications. J Endourol 17(9):695–708
- [7]. Cepolina F, Challacombe B, Micheline RC (2005) Trends in robotic surgery. J Endourol 19(8):940–951
- [8]. Birkett DH (2002) Robotics. Surg Endosc 16:1257–1258
- [9]. Ellis JJ (2000) Founding brothers: the revolutionary generation. Vintage Books, New York, p 18
- [10]. Rosheim ME: Leonardo's programmable automaton. A reconstruction.
- [11]. Nuland SB (2000) Leonardo da Vinci. Penguin Books, New York
- [12]. Capello S, Moran ME, Belarmino J, Firoozi F, Kolios E, Perrotti M (2005) The da Vinci robot. 23rd world congress on endourology. J Endourol 19(1):A133
- [13]. Wood G (2002) Edison's eve. A magical history of the quest for mechanical life. Anchor Books, New York

- [14]. Dorf, R. C., and R. H. Bishop, "Modern Control Systems," 9th ed., Prentice Hall, 2001.
- [15]. Franklin, G. F., J. D. Powell, and A. Emami-Naeimi, "Feedback Control of Dynamic Systems," 3rd ed., Addison-Wesley, 1994.
- [16]. Kuo, B. C., "Automatic Control Systems," 7th ed., Prentice Hall, 1995.
- [17]. Gao, Z., Y. Huang, and J. Han, "An Alternative Paradigm for Control System Design," Presented at the 40th IEEE Conference on Decision and Control, Dec 4–7, 2001, Orlando, FL.
- [18] meshram U, bande, p, dwaramwar, P .A, harkare ,R.R, signal processing and communication technologies, IMPACT 0'9, nagbur, india, 2009.
- [19] FPGA Based Five Axis Robot Arm Controller, UrmilaDevendra Meshram1 & R.R. Harkare2, International Journal of Electronics Engineering, india, august 2010,
- [20]Muhammed Zulhimi Mansoor, pc based small arm control, University Malaysia Pahang, Malaysia, November 2010
- [21] FPGA Based Robotic Arm With Six Degrees of Freedom, Lakshmi Pravalika, Saveetha Engineering College, Chennai, Tamilnadu, India, February 2013.
- [22] Robot arm controller using FPGA, SudanUniversity of science and technology, Sudan, September 2014.
- [23] Eric Monmasson, SENIOR MEMBER, IEEE, MarcianCirstea, SENIOR MEMBER, IEEE, FPGA Design Methodology for Industrial Control Systems[2] FPGA vs. DSP Design Reliability and Maintenance

,2007 Michael Parker, Sr. Technical Marketing Manager, IP and Technology Product Marketing, Altera Corporation .

[24] MSP430 Microcontroller Basics by John H. Davies 2008 .

[25] FPGA vs. ASICXilinx Inc 2007..

[26]Generating PWM Signals With Variable Duty From 0% to 100%

Based FPGA SPARTAN3ANbyZiadNouman, BohumilKlima, Jan Knobloch

[27] Wiśniewski, Remigiusz . Synthesis of compositional microprogram control units for programmable devices. University of Zielona Góra, 2009.

[28] Maxfield, Clive. The design warrior's guide to FPGAs: devices, tools and flows. Elsevier, 2004

[29] Trimberger, Stephen, ed. Field-programmable gate array technology. Springer, 1994.

[31] Trestian, Ionuț, et al. "FPGA-based computation of the inductance of coils used for the magnetic stimulation of the nervous system." (2007).

.

[32] H. Koca, M. Doğan and M. C. Taplamacıoğlu, Robot Manipülâtörün Kartezyen-Özgül Denetimi, J. Fac. Eng. Arch. Gazi Univ. vol 23, No 4, pp. 769-776, 2008.

[33] G.Mccomb and M.Predko , "Robot Builder's Bonanza",2006 ,pp.393-413 .

[34] M. S. Lateef, “Design and Implementation of FPGA Based Adaptive Neural Controller for Intelligent Robot Navigation”, 2011, Msc. thesis, Engineering College, University of Mosul, Iraq.