EMBEDDED POSITION CONTROL OF DC MOTOR USING FUZZY LOGIC WITH PI CONTROL

A PROJECT REPORT

Submitted by

S.Arunprasad (71206107002)

S.Divagar (71206107009)

P.Tamilazhagan (71206107053)

C.Vinodh kumar (71206107057)

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND INSTRUMENTATION ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641006

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2010

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report entitled **"EMBEDDED POSITION CONTROL OF DC MOTOR USING FUZZY LOGIC WITH PI CONTROL"** is the bonafide work of

Arunprasad. S (71206107002)

Divagar. S (71206107009)

Tamilazhagan. P (71206107053)

Vinodh kumar. C (71206107057)

who carried out the project work under my supervision.

SIGNATURE

Prof.R.ANNAMALAI M.E.,

HEAD,
Dept of Electronics & Instrumentation Engg.,
Kumaraguru College of Technology,
Coimbatore 641006.

SIGNATURE

Mrs.B.VEENAABIRAMI M.E.,

PROJECT GUIDE
Sr. Lecturer
Dept of Electronics & Instrumentation Engg.,
Kumaraguru College of Technology,
Coimbatore 641006.

The candidates were examined by us in the project viva - voce examination held on 16.04.2010

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

The completion of our project can be attributed to the combined efforts made by us and the contribution made in one form or the other by the individuals we hereby acknowledge.

We express our sincere thanks to Principal **Prof. S. RAMACHANDRAN**, who has been the back bone of all our needs.

We profusely thank **Prof. R. ANNAMALAI, M.E.,** Head of the Department of Electronics and Instrumentation Engineering for his encouragement, to bring out this project work successfully.

We take immense pleasure to record our heartfelt gratitude to our esteemed project coordinator **Mr. S. ARUN JAYAKAR, M.E.,** and project guide **Mrs. B. VEENA ABIRAMI**, **M.E.,** Sr. Lecturer, EIE Department, for their valuable guidance, timely help, constant encouragement and advice rendered throughout the project period for the successful completion of project.

We are also thankful to our faculty members of the Department of Electronics and Instrumentation Engineering, who have helped us in innumerable ways.

We also thank our parents without whom we could not have come so far and friends for their timely help that culminated as good in end.

ABSTRACT

This project deals with the position control of 'DC Servo Motor' using fuzzy logic with Proportional-Integral control.

The controllers used now are generally conventional controllers. The conventional controllers works good. But as the complexity of the problem increases it's more difficult to make things to work as desired.

Fuzzy logic is one of the latest developing areas with its various powerful tools. Many researches were going on in fuzzy because of its increasing application compared to the conventional Proportional-Integral-Derivative Control.

The fuzzy implementation is done using MATLAB, one of the powerful tools in design application. The Matlab simulation is done for conventional PID, fuzzy with PID and fuzzy with PI. The best suited controller is then implemented in the real time.

Our main objective in this project is to design a fuzzy tuned controller to reduce overshoot, settling time, oscillation and the steady state error in the system

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

NO.	SYMBOLS	ABBREVIATIONS
01	AC	Alternating Current
02	DC	Direct Current
03	PCM	Pulse Code Modulation
04	FPGA	Field Programmable Gate Array
05	VHSIC	Very High Speed Integrated Circuit
06	HDL	Hardware Descriptive Language
07	IC	Integrated Circuit
08	VLSI	Very Large Scale Integrated Circuit
09	PLD	Programmable Logic Device
10	IOB	Input Output Block
11	I/O	Input / Output
12	LUT	Look Up Table
13	CPLD	Configurable PLD
14	PCB	Printed Circuit Board
15	SRAM	Static Random Access Memory
16	DSP	Digital Signal Processing
17	ASIC	Application Specific IC
18	LC	Logic Cells
19	CLB	Configurable Logic Block
20	LCD	Liquid Crystal Display

CHAPTER 1

CHAPTER 1

INTRODUCTION

Direct Current (DC) servomotors are widely used in robot manipulator applications. Servomotors use feedback controller to control either the speed or the position or both, and the basic continuous feedback control is PID controller. In this project we are controlling the position of servomotor. The transfer function for the position control of the servo motor is obtained using the moment of inertia, damping ratio and the operating voltage. The analysis of controller performance for

- PID controller,
- Fuzzy with PID controller and,
- Fuzzy with PI controller on the system are made.

Using the transfer function model for position control of the servo motor the simulation is carried out. The simulation result is analysed and the best suited controller is selected. After selection of the controller, it is implemented in hardware to verify its real time operation.

1.1 BLOCK DIAGRAM

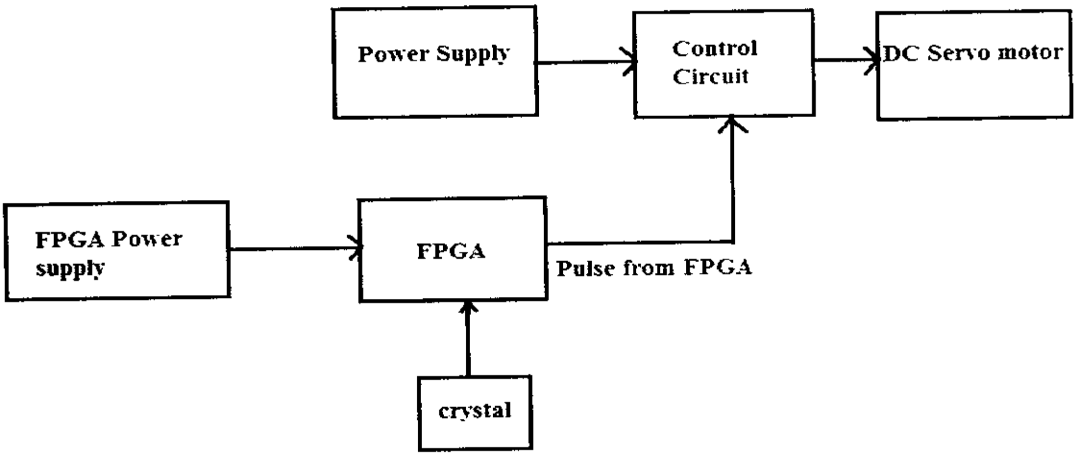


FIG1.1 BLOCK DIAGRAM

The Hardware Descriptive Language code for the best suited control is developed and it is downloaded to FPGA (Field Programmable Gate Array) through JTAG cable. The FPGA device is programmed using Verilog code to generate the gating pulses.

The FPGA produces the control pulse that controls the driver circuit. Servo motor rotates according the driver circuit output and its output is determined by the duration of the pulse or pulse width.

Based on pulse width, the control output is produced. The output of control circuit is then given to dc servo motor and the position corresponding to the control output is obtained.

1.2 ORGANIZATION OF THE REPORT

Chapter 1: Introduction to the project.

Chapter 2: Servo motor –An overview about construction, operating principle, and types of Servomotor.

Chapter 3:Conventional controller–Gives a brief idea about simulation of conventional PID.

Chapter 4: Fuzzy logic controller- simulation of fuzzy with PI control and fuzzy with PID control is implemented.

Chapter 5: FPGA- Gives the details about the FPGA and Xilinx.

Chapter 6: Hardware Description- gives the details about the power supply, timing diagram and algorithm.

Chapter 7: Conclusion.

Appendices: Gives the features of FPGA XC3S100E.

CHAPTER 2

CHAPTER 2

SERVOMOTORS

2.1 WHATS A SERVO?

A Servo is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft. As the coded signal changes, the angular position of the shaft changes. In practice, servos are used in radio controlled airplanes to position control surfaces like the elevators and rudders. They are also used in radio controlled cars, puppets, and of course, robots.

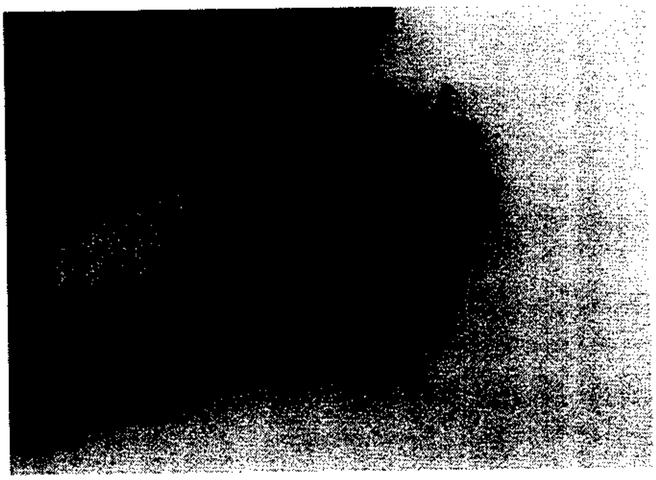


FIG 2.1A FUTABA S-148 SERVO

Servos are extremely useful in robotics. The motors are small, as shown in fig 2.1, have built in control circuitry, and are extremely powerful for their size.

A standard servo such as the Futaba S-148 has 42 oz/inches of torque, which is pretty strong for its size. It also draws power proportional to the mechanical load. A lightly loaded servo, therefore, doesn't consume much energy.

FIG 2.2 A SERVO DISASSEMBLED.

The servo motor has some control circuits and a potentiometer (a variable resistor, aka pot) that is connected to the output shaft. In the fig 2.2, the pot can be seen on the right side of the circuit board. This potentiometer allows the control circuitry to monitor the current angle of the servo motor. If the shaft is at the correct angle, then the motor shuts off. If the circuit finds that the angle is not correct, it will turn the motor the correct direction until the angle is correct. The output shaft of the servo is capable of travelling somewhere around 180 degrees. Usually, its somewhere in the 210 degree range, but it varies by manufacturer.

A normal servo is used to control an angular motion of between 0 and 180 degrees. A normal servo is mechanically not capable of turning any farther due to a mechanical stop built on to the main output gear. The amount of power applied to the motor is proportional to the distance it needs to travel. So, if the shaft needs to turn a large distance, the motor will run at full speed. If it needs to turn only a small amount, the motor will run at a slower speed. This is called proportional control.

The control wire is used to communicate the angle. The angle is determined by the duration of a pulse that is applied to the control wire. This is called Pulse Coded Modulation. The servo expects to see a pulse every 20 milliseconds (.02 seconds). The length of the pulse will determine how far the motor turns. A 1.5 millisecond pulse, for example, will make the motor turn to the 90 degree position (often called the neutral position). If the pulse is shorter than 1.5 ms, then the motor will turn the shaft to closer to 0 degrees. If the pulse is longer than 1.5ms, the shaft turns closer to 180 degrees.

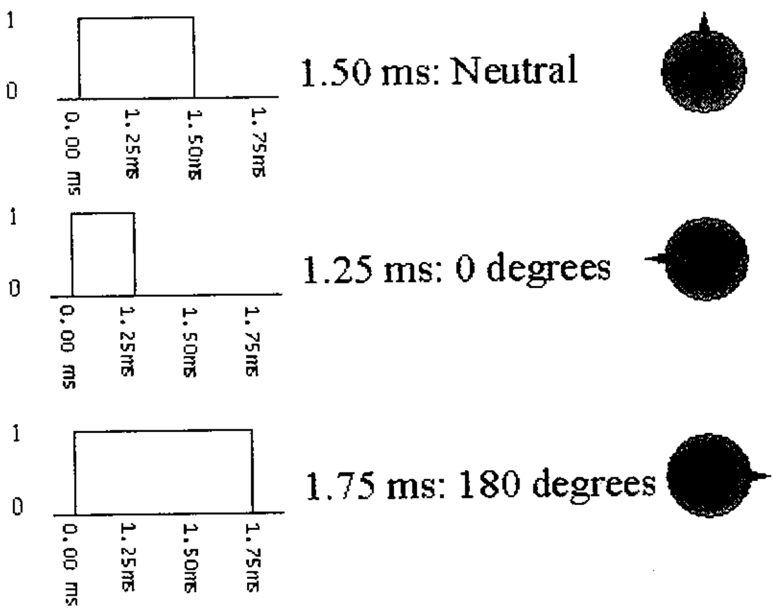


FIG 2.3 PULSE AND POSITION

As you can see in the picture, the duration of the pulse dictates the angle of the output shaft. Note that the times here are illustrative, and the actual timings depend on the motor manufacturer. The principle, however, is the same.

2.2 DC SERVO MOTORS

Dc servo motors are normally used as prime movers in computers, numerically controlled machinery, or other applications where starts and stops are made quickly and accurately. Servo motors have lightweight, low-inertia armatures that respond quickly to excitation-voltage changes. In addition, very low armature inductance in these servo motors results in a low electrical time constant (typically 0.05 to 1.5 ms) that further sharpens servo motor response to command signals.

Servo motors include permanent-magnetic, printed-circuit, and moving-coil dc servo motors. The rotor of a shell dc servo motor consists of a cylindrical shell of copper or aluminum wire coils which rotate in a magnetic field in the annular space between magnetic pole pieces and a stationary iron core.

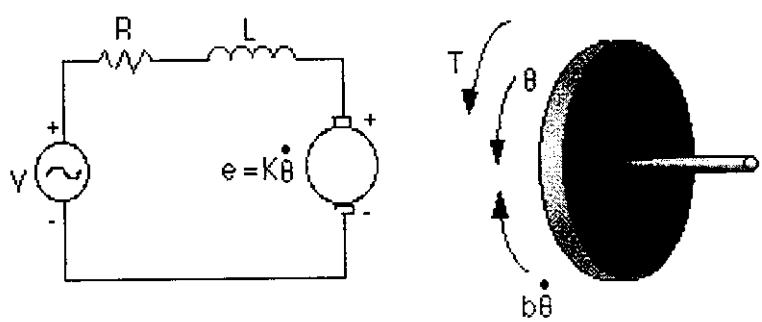


FIG 2.4 DC SERVO MOTOR

The servo motor features a field, which is provided by cast AlNiCo magnets whose magnetic axis is radial. Servo motors usually have two, four, or six poles.

Dc servo motor characteristics include inertia, physical shape, costs, shaft resonance, shaft configuration, speed, and weight. Although these dc servo motors have similar torque ratings, their physical and electrical constants vary.

Servo motors are used in closed loop control systems. The digital servo motor controller directs operation of the servo motor by sending velocity command signals to the amplifier, which drives the servo motor. An integral feedback device (resolver) or devices (encoder and tachometer) are either incorporated within the servo motor or are remotely mounted, often on the load itself. These provide the servo motor's position and velocity feedback that the controller compares to its programmed motion profile and uses to alter its velocity signal. Servo motors feature a motion profile, which is a set of instructions programmed into the controller that defines the servo motor operation in terms of time, position, and velocity. The ability of the servo motor to adjust to differences between the motion profile and feedback signals depends greatly upon the type of controls and servo motors used. See the servo motors Control and Sensors Product section.

Three basic types of servo motors are used in modern servo systems: ac servo motors, based on induction motor designs; dc servo motors, based on dc motor designs; and ac brushless servo motors, based on synchronous motor designs.

2.3 AC SERVO MOTORS

These servo motors are basically two-phase, reversible, induction motors modified for servo operation. Ac servo motors are used in applications requiring rapid and accurate response characteristics. To achieve these characteristics, these ac servo motors have small diameter, high resistance rotors. The ac servo motor's small diameter provides low inertia for fast starts, stops, and reversals. High resistance provides nearly linear speed-torque characteristics for accurate servo motor control.

An induction motor designed for servo use is wound with two phases physically at right angles or in space quadrature. A fixed or reference winding is excited by a fixed voltage source, while the control winding is excited by an adjustable or variable control voltage, usually from a servoamplifier. The servo motor windings are often designed with the same voltage/turns ratio, so that power inputs at maximum fixed phase excitation, and at maximum control phase signal, are in balance.

The inherent damping of servo motors decreases as ratings increase, and the servo motors are designed to have a reasonable efficiency at the sacrifice of speed-torque linearity. Induction type servo motors are available in fractional and integral horsepower sizes.

2.4 DC SERVO MOTOR SELECTION

The first selection approach is to choose a servo motor large enough for a machine that has already been designed; the second is to select the best available servo motor with a specific feature and then build the system around it; and the third is to study servo motor performance and system requirements.

The final servo motor system design is usually the least sophisticated that meets the performance specifications reliably. Servo motor requirements may include control of acceleration, velocity, and position to very close tolerances. This says that the servo designer must define the system carefully, establish the servo motor's performance specifications, determine critical areas, and set up tolerances. Only then will the designer be able to propose an adequate servosystem and choose a servo motor type.

CHAPTER 3

CHAPTER 3

CONVENTIONAL CONTROLLER

3.1 INTRODUCTION

Direct Current (DC) servomotors are widely used in robot manipulator applications. Servomotors use feedback controller to control either the speed or the position or both, and the basic continuous feedback control is PID controller. DC motors are designed to run on direct current electric power. DC motor is a machine that converts DC power into mechanical power. DC motors are bi-directional i.e. the rotor shaft can rotate both clockwise and anticlockwise depending on their polarities of the supply voltage. A wide range of speed control is possible using DC motors by adjusting the voltage or field current since the speed is directly proportional to its voltage and inversely proportional to the magnetic flux produced by the poles. Simulation of DC servomotor control was done using MATLAB/Simulink, and the analysis of controller performance, namely a PID controller, fuzzy with PID controller, and a fuzzy logic based PI controller on the system is made.

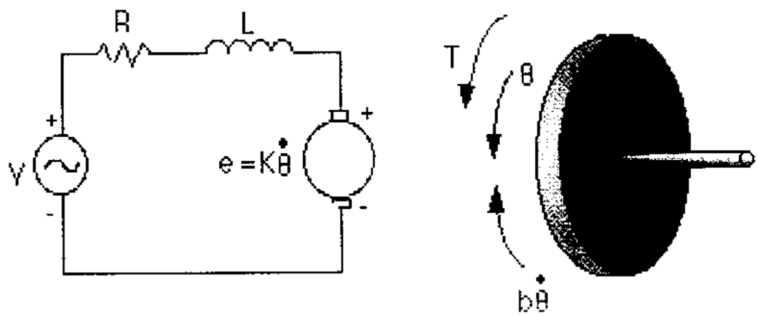


FIG 3.1 ELECTRIC CIRCUIT DIAGRAM OF DC SERVO MOTOR

With ref [1], the position control transfer function of dc servo motor is given by

$\theta(s)/V(s) = k/ (JLs^3+(Jr+bL)s^2+(br+k^2)s)$

Where

V = voltage or emf of the servo motor

θ = angular position of the servo motor in deg.

J = moment of inertia = 0.001kg.m^2/s^2

b = damping ratio = 0.1 Nms

k = constant = 0.01 Nm/A

r = resistance = 1 ohm

L = inductance = 0.5 H

By substituting these values in the above transfer function, we get

$\theta(s)/V(s) = 20/(s^3+102s^2+200.2s)$

3.2 CONVENTIONAL PID CONTROLLER

The PID control technique is widely used in many industries. It works on control feedback mechanism, where the output is compared to the input and the resulting error will be adjusted so that the output gets the desired value. It is made up of three main blocks namely Proportional, Integral and Derivative Block. The three blocks are dependent on each other.

PID controller is one of the most commonly used methods to solve the control problems. The controller has been implemented in many different application such as electronics, mechanical, pneumatic and computer technology.

PID controller is a simple implementation of feedback. Application on feed back principle has resulted in major breakthrough in control, instrumentation and communication systems. The PID controller is a key part of system for motor control.

A simple PID circuit is built using MATLAB, to control a motors position. By modeling the system, we obtain the transfer function for position control of dc servomotor. Transfer function of DC motor is used in the PID and the entire position control will be simulated using MATLAB.

The block diagram of position control of DC servo motor using conventional PID is shown in fig 3.2

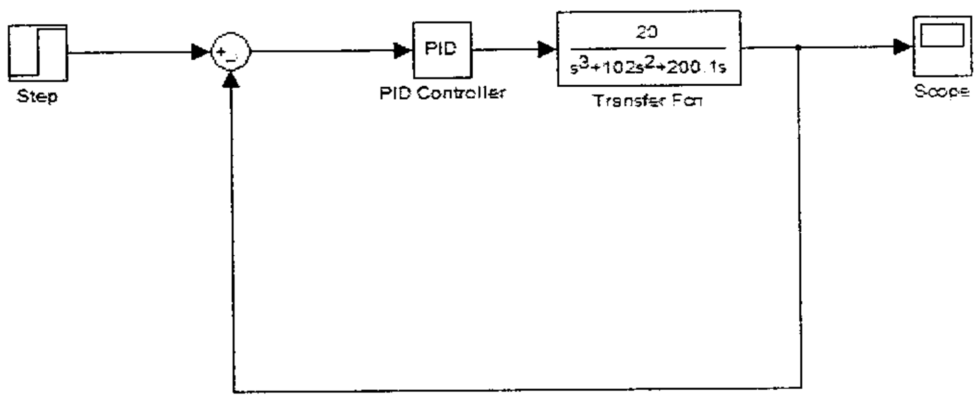


FIG 3.2 PID BLOCK DIAGRAM

The response of the system with PID controller obtained is shown in fig 3.3.

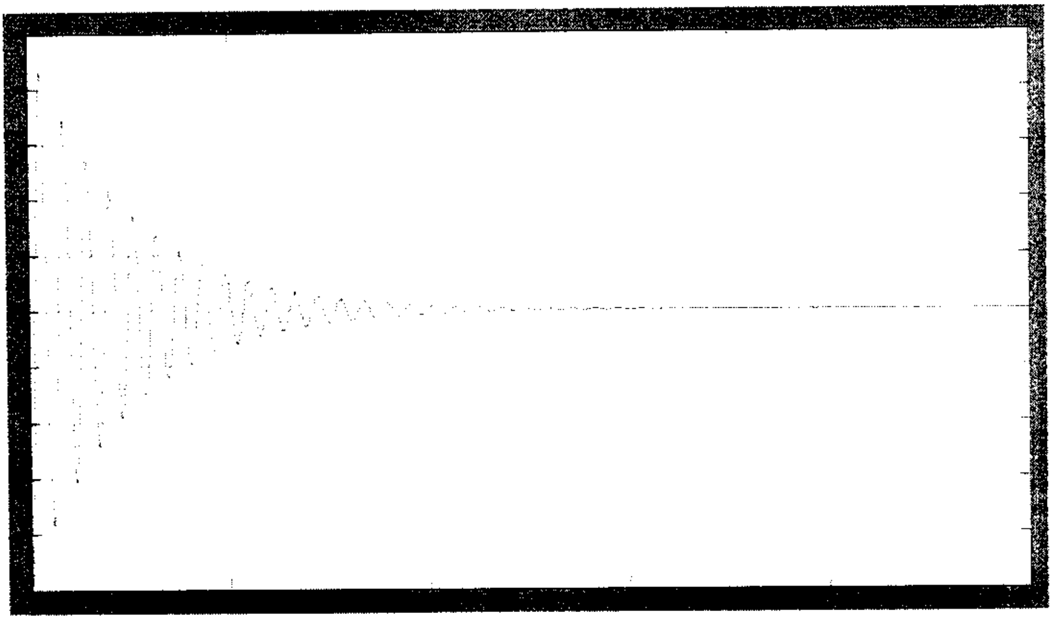


FIG 3.3 RESPONSE OF CONVENTIONAL PID FOR STEP CHANGE

TUNING

Tuning a control loop is the adjustment of its control parameters (proportional gain, integral gain and, derivative gain) to the optimum values for the desired control response. There are three types to tune the PID namely Manual tuning, Ziegler–Nichols method and PID tuning software. In manual tuning method 'I' and 'D' values are set to zero and then Increase the P until the output of the loop oscillates. Then increase 'D' until any offset is correct in sufficient time for the process. Too much 'D' will cause instability. Finally value of 'I' is increased until the loop is acceptably quick to reach its reference speed. Too much 'I' will cause excessive response and overshoot. A fast PID loop tuning usually overshoots slightly to reach the set point more quickly but in some systems over shoot is not accepted.

3.3 ZIEGLER–NICHOLS TUNING

Ziegler-Nichols tuning rule is used to tune the conventional PID in Matlab. It is different from normal Ziegler-Nichols tuning. Gain margin, phase Margin and gain cross over frequency were obtained from the transfer function by using the formula margin (system transfer function).By using those values we can calculate K_u and P_u by using the formula $k_u = G_m$ and $p_u = 2*p_m/W_{cg}$. Then controller gain is calculated for each controller as follows

For P-controller

$k_c = k_u/2$

For PI-controller

$k_c = k_u/2.2$

$t_i = p_u/1.2$

For PID-controller

$k_c = k_u/1.7$

$t_i = p_u/2$

$t_d = p_u/8$

3.4 Results

- Conventional PID has too many oscillations and very high peak overshoot.
- PID cannot be used in system having longer dead time or system with oscillatory modes.
- By using fuzzy with PI and fuzzy with PID we can reduce the overshoot and we can reduce the oscillations as well.

CHAPTER 4

CHAPTER 4
FUZZY LOGIC CONTROLLER

4.1 INTRODUCTION

Fuzzy" the term indicates indistinct or haziness. The Fuzzy logic was invented by professor L.A.Zadeh of California University at Berkeley in the year 1965. The invention was not recognized globally until it was implemented practically by Dr. E. H. Mamdani a professor at London University. Fuzzy controller is an automatic controller that can be used in control application to control the object based on its desired behavior. Unlike the traditional 'yes' or 'no' decisions the fuzzy logic allows a graduation from 'yes' to 'no'. The fuzzy logic works on set of rules and these will be created by the engineers in the design phase and thus fuzzy controller is clearly a branch of intelligent control.

For example if you take a fuzzy controlled air conditioner, the rules behind them are less precise. For instance if the room temperature is warm then the controller will automatically increase the cooling. Fuzzy logic systems are not only used for smaller application but it's also used for complex applications. Some complex problems can be solved only using fuzzy logic due its rules and sets and the one more thing is that it is faster compared to the conventional controllers. In this project fuzzy control is implemented with PI control for position control of the DC motor.

4.2 IMPORTANT TERMS

Some important terms related to fuzzy logic are briefly summarized below.

Fuzzy set: The values or the parameters used in fuzzy control are put together to form fuzzy set and they are usually described in qualitative terms like large, medium, small, Zero, positive and negative.

Membership Value: The degree a control variable or parameters belongs to a fuzzy set is denoted by a membership value between 0 and 1.

Membership Function: A membership function associated with the fuzzy set maps a control variable or parameter to its appropriate membership value.

Fuzzy Rule: The fuzzy rule is usually in the form

If X is A *and* Y is B *then* Z is C where x and y are inputs and z is output. The *if then* statements are combined by using either *and* or *or* connection.

Defuzzification: The output of fuzzy should be converted in to a single value so that it can be used as a control signal to control the plant and Defuzzification block do this process.

4.3 FUZZY CONTROLLER THEORY

The block diagram of fuzzy control is shown in fig 4.1.

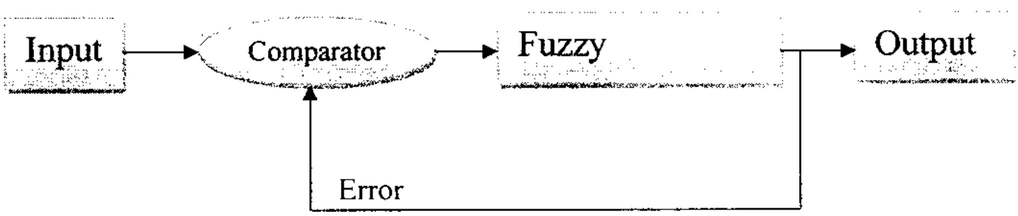


FIG 4.1 FUZZY CONTROLLER

The input block gives the input to the comparator. The comparator compares the input and output values and gives the error value to the fuzzy controller. Depending on the error value the fuzzy controller adjust the output value.

Fuzzy controller components

The fuzzy controller is made up of Fuzzification block, Rule base block, Interference engine block and the Defuzzification block. Fuzzification block converts each and every input into a look up table in the membership functions to derive the membership grades.

Rule base block consists of set of rules that links the input and the output. The rules are formed using *if then* statements and are connected using either *and* or *or* connection. Interference engine block is the one which links the Fuzzification and Defuzzification block. Defuzzification block converts the output to a single value so that it can be used as a control signal to the plant.

4.4 FUZZY WITH PI CONTROL

An intelligent PI controller i.e. Fuzzy controlled PI is used for position control of DC motor since the system has time varying parameter and nonlinearities. The rules which are represented in terms of fuzzy logic for better tuning are used to adjust the PI parameters in the system. The output results of PID and the intelligent controller can be verified using the simulation results.

The input to the fuzzy logic controller is present error and change in error. The fuzzy logic controller has the Fuzzification, rule base and Defuzzification blocks. Based on the fuzzified input to the FLC the Corresponding rule base is selected and then it is defuzzified. The defuzzified output from FLC is applied to PI control and then response is obtained.

The block diagram of the fuzzy with PI control is shown in fig 4.2.

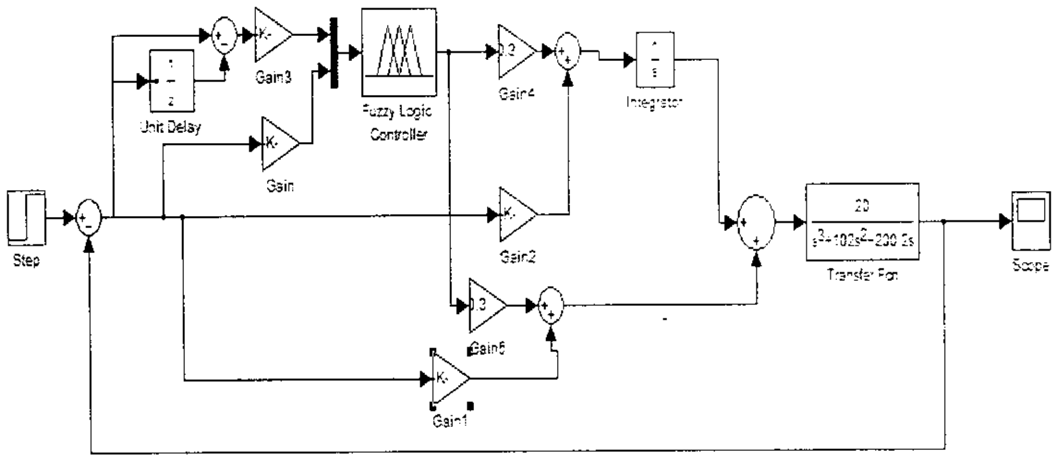


FIG 4.2 FUZZY WITH PI CONTROL

By simulating above block, the response of the above system obtained is shown in fig 4.3.

RESPONSE

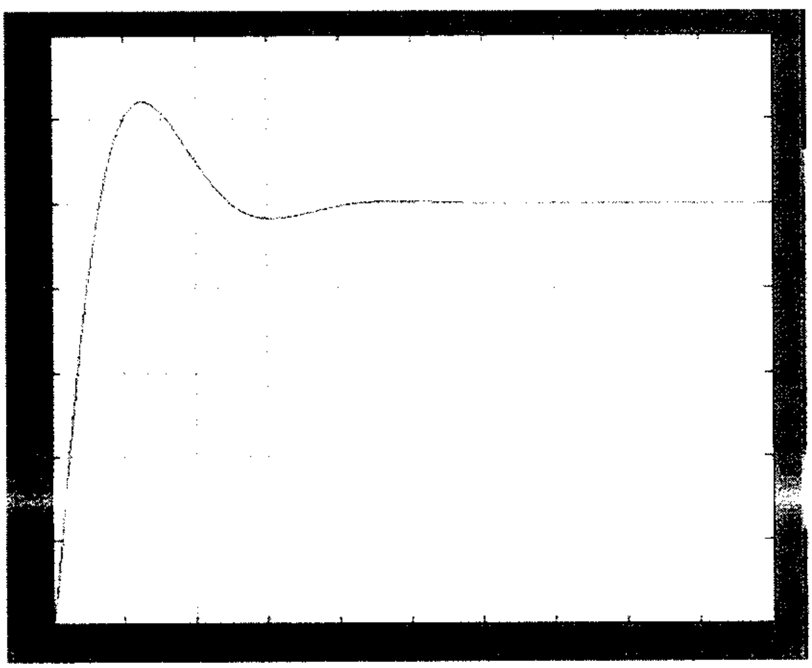


FIG 4.3 RESPONSE OF FUZZY WITH PI CONTROL

4.5 FUZZY WITH PID CONTROL

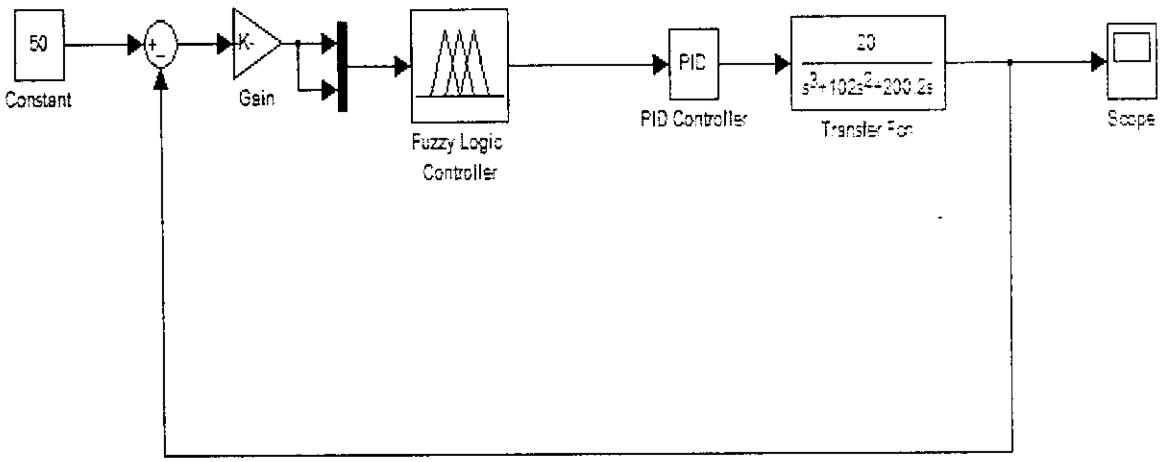


FIG 4.4 FUZZY WITH PID CONTROL

In fuzzy with PID control the defuzzified output is given to PID controller and then the response is obtained. In both fuzzy with PI and PID control same rule base is used. The triangular membership function and centroid Defuzzification method is used in both case.

Tuning is similar to that of tuning of conventional controller. Ziegler-Nichols tuning rule is used to tune the fuzzy with PI and PID in Matlab. It is different from normal Ziegler-Nichols tuning. Gain Margin, phase Margin and gain cross over frequency were obtained from the transfer function. From those values K_u and P_u is calculated. Calculate controller gain K_c, K_i, K_d by using K_u and P_u values.

4.6 RULE BASE

ERROR \ RATE	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NB	NB	NB
NM	NB	NB	NB	NB	NM	NM	NM
NS	NB	NB	NB	NB	NM	NS	NS
Z	NB	NM	NS	Z	PS	PM	PB
PS	Z	Z	Z	PS	PS	PM	PB
PM	Z	PS	PM	PM	PM	PM	PB
PB	PB	PB	PB	PB	PB	PB	PB

TABLE 4.1 RULE BASE

The rules and rule base is obtained with reference to the IEEE paper titled" Development of Fuzzy-Logic-Based Self Tuning PI Controller for Servomotor". Fuzzy logic controller's output depends on the rule base and control action takes place according to the rules in the rule base.

The rule base is designed in such a way that it reduces the oscillation and overshoots in the conventional controller. In the above rule base "NB" stands for negative big, "NM" stands for negative medium, "NS" stands for negative small, "Z" stands for zero, "PS" stands for positive small, "PM" stands for positive medium, and "PB" stands for positive big. By simulating above block, the response of the above system obtained is shown in fig 4.5.

RESPONSE

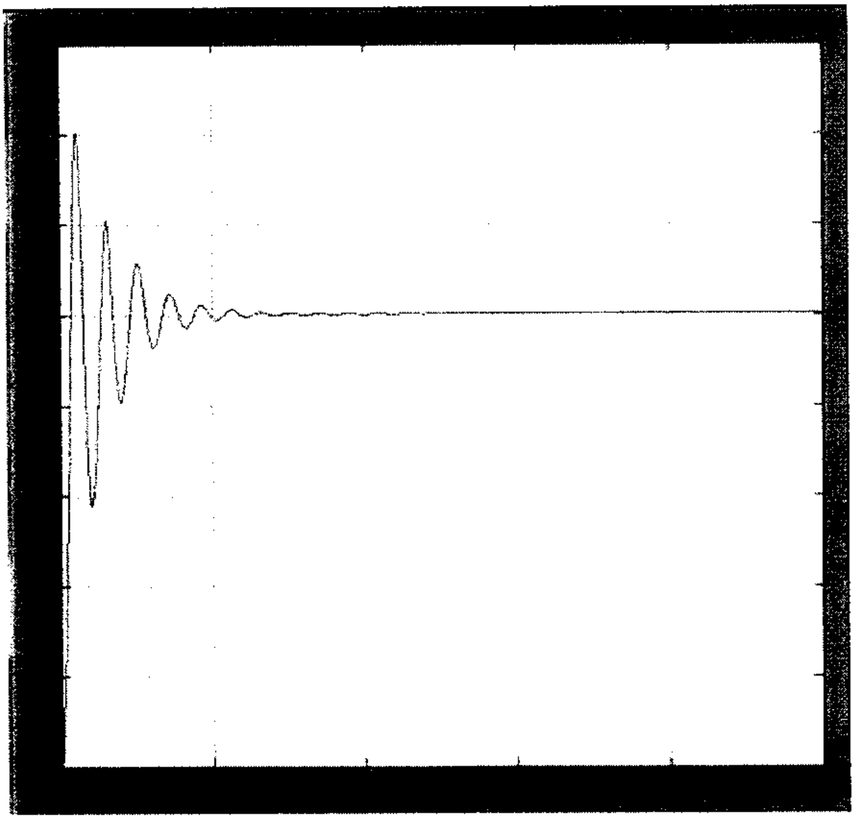


FIG 4.5 RESPONSE OF FUZZY WITH PID CONTROL

4.8 RESULTS

It is clear that fuzzy with PI have better response with less oscillations and zero steady state error .Conventional PID has better rise time and quick settling time but it has too many oscillations. Fuzzy with PID has much better rise time and quick settling time than Conventional PID but still it has steady state error and few oscillations .So Fuzzy With PI control is used in servo position control.

The comparision of various controllers based on different parameters are listed in table 4.2.

4.9 COMPARISION TABLE

Controllers/ parameters	PID	FUZZY WITH PI	FUZZY WITH PID
RISE TIME	0.13	6.7	0.7
SETTLING TIME	16	45	9
STEADY STATE ERROR	0	0	0.3
OVERSHOOT	43	12.5	21
OSCILLATIONS	More	Very less	Less

FIG 4.2 COMPARISION TABLE

CHAPTER 5

CHAPTER 5

FPGA

5.1 INTRODUCTION

The field-programmable gate array (FPGA) is a semiconductor device containing electrically programmable components that can be programmed after manufacturing. Instead of being restricted to any predetermined hardware function, an FPGA allows you to program product features and functions, adapt to new standards, and reconfigure hardware for specific applications even after the product has been installed in the field—hence the name "field-programmable". You can use an FPGA to implement any logical function that an application-specific integrated circuit (ASIC) could perform, but the ability to update the functionality after shipping offers advantages for many applications. It can be programmed to duplicate the functionality of basic logic gates such as AND, OR, NOT or more combinational functions such as decoders or simple math functions.

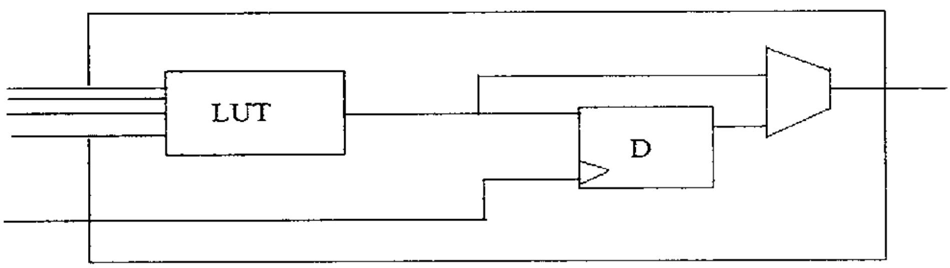
Unlike previous generation FPGAs using I/Os with programmable logic and interconnects, today's FPGAs consist of various mixes of configurable embedded SRAM, high-speed transceivers, high-speed I/O, logic blocks, and routing. Specifically, an FPGA contains programmable logic components called logic elements (LEs) and a hierarchy of reconfigurable interconnects that allow the LEs to be physically connected. You can configure LEs to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, the logic blocks also include memory elements, which may be simple flip flops or more complete blocks of memory. FPGA is similar to a PLD but it supports 1000's of gates whereas PLD support only 100's of gates.

5.2 ARCHITECTURE

The typical architecture consists of an array of configurable logic blocks (CLB's) and routing channels. Multiple I/O pads may fit into the height of one row or the width of one column in the array. Generally all the routing channels have the same width (no of wires).The basic components of FPGA are

- Configurable Logic Block(CLB's)
- Interconnect
- Input Output Block(IOBs)
- Memory
- Complete Clock Management

An application circuit must be mapped into an FPGA with adequate resources. The typical FPGA logic block consists of a 4-input lookup table(LUT),and a flip-flops, as shown



Logic Cell

FIG 5.1 LOGICBLOCK

There is only one output, which can be either the registered or the unregistered LUT output. The logic block has four inputs for the LUT and a clock input. Since clock signals (and often other high- fanout signals) are normally routed via special purpose dedicated routing networks in commercial FPGAs they and other signals are separately managed.

Each input is accessible from one side of the logic block, while the output pin can connect to routing wires in both the channel below the logic block. Each logic block output pin can connect to any of the wiring segments in the channels adjacent to it.

5.3 FPGA MANUFACTURERS

Field programmable devices are manufactured by several companies and a few of them are listed

- Altera corporation
- Xilinx Inc
- Atmel corporation
- Quick Logic Corporation
- Lattice semiconductor corporation

Xilinx first formed in 1984,are the major manufacturers of CPLDs and SRAM based FPGAs. Xilinx leads industry by its products.

- Virtex series
- Spartan series

5.4 FPGA DESIGN FLOW:

The standard design flow for the Spartan generation FPGA's include following three major steps

- Design entry synthesis
- Design Implementation
- Design Verification

Design Description:

Designer's integrated design environment allows you to design, implement and debug a microprocessor-based digital design in an FPGA. The design is captured as a schematic, or using a mixture of schematic and HDL (VHDL or Verilog). The embedded software is written in a coding-aware editor, ready for compilation and download onto the processor in the design.

Synthesis:

Once the hardware design is complete it is synthesised, a process that transforms it from the capture form into a low-level gate form. After design synthesis a place and route is performed, a process where device-aware software implements the design in the target FPGA. The Vendor-specific place and route software required to synthesize for the target architecture is operated by the Designer environment, which automatically manages all project and file handling aspects required to generate an FPGA program file. To test and debug the design the system includes a Nano Board, an implementation platform that includes an FPGA, as well as an array of general purpose peripheral components.

Design Implementation:

Implementation includes partition, place and route. Place and Route translates the logic design into physical design, maps the components used in the into specific elements, places them and routes the interconnection between them, place and route also helps to do timing analysis. After all cells are placed and routed, the output of the place and route tools consist of data files that can be used to implement the chip. The output of the design implementation phase is bit-stream file. These files describe all the connections needed to make the FPGA macro cells implement the functionality required.

Design Verification:

Once the design has been implemented on the NanoBoard it can be debugged, using virtual instruments and boundary scan pin status technology to debug the hardware, and the integrated debugger for the embedded software. Since debugging is performed live from within the same environment as the design is captured in, design iterations can be carried out quickly and software/hardware solutions rapidly explored. µP High-level system specification& partitioning Embedded Software Development FPGA Design Capture(Schematic/HDL)PCB Design Capture PCB Place& Route PCB CAM/Fabrication FPGA Place& Route(vendor tools)Compile Embedded Code Debug Embedded Code Conversion pin assignments Download program file Debug code HDL Simulation FPGA Synthesis.

5.5 FLOW DIAGRAM OF THE FPGA DESIGN PROCESS:

Create FPGA project file setup project options Create embedded project file setup project options FPGA component libraries Define constraints and set up configuration Link embedded project Select project/configuration combination in

Devices view Write code Capture design(HDL)Synthesize and correct errors Compile and correct errors Build (place and route)Compile and correct errors Synthesize and correct errors Program device

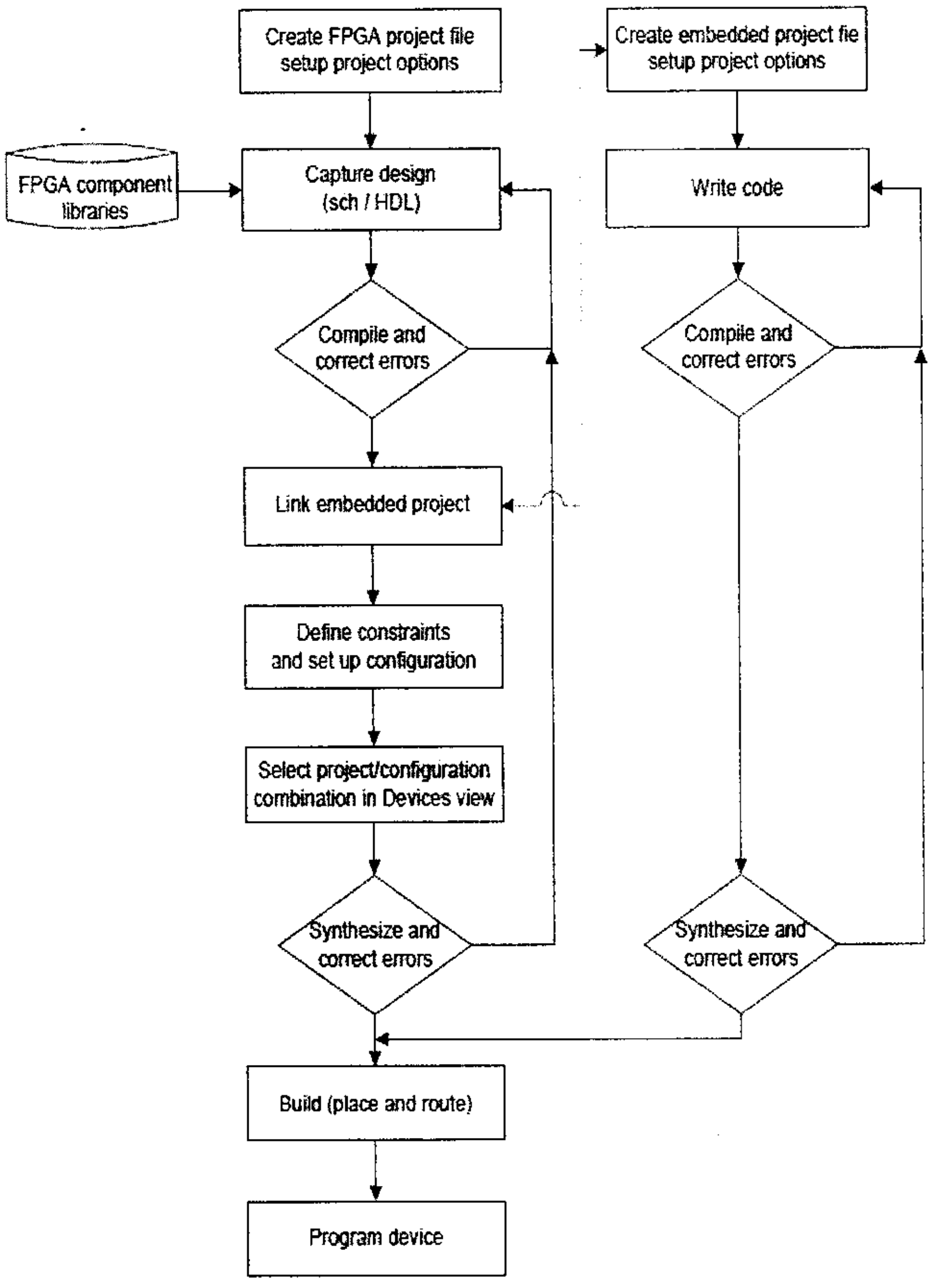


FIG 5.2 FLOW DIAGRAM OF THE DESIGN PROCESS

5.6 XILINX

Fpga Pins

FPGA pins fall into 2 categories: "dedicated pins" and "user pins". About 20% to 30% of the pins of an FPGA are "dedicated pins", which means that they are hard-coded to a specific function. The dedicated pins fall into the 3 following sub-categories.

- Power pins: ground and core/IO power pins.
- Configuration pins: used to "download" the FPGA.
- Dedicated inputs, or clock pins: these are able to drive large nets inside the FPGA, suitable for clocks or signals with large fan-outs.

The rest are user pins (called "IOs", or "I/Os", or "user I/Os", or "user IOs", or "IO pins"). IO stands for "input-output".

- You usually have total control over user IOs. They can be programmed to be inputs, outputs, or bi-directional (tri-stable buffers).
- Each "IO pin" is connected to an "IO cell" inside the FPGA. The "IO cells" are powered by the VCCIO pins (IO power pins).

An FPGA has many VCCIO pins, usually all connected to the same voltage. But new generations of FPGAs have a concept of "user IO banks". The IOs are split into groups, each having its own VCCIO voltage. That allows using the FPGA as a voltage translator device, useful for example if one part of your board works with 3.3V logic, and another with 2.5V.

Fpga Power

FPGAs usually require two voltages to operate: a "core voltage" and an "IO voltage". Each voltage is provided through separate power pins.

- The internal core voltage (called VCCINT here) is fixed (set by the model of FPGA that you are using). It is used to power the logic gates and flip flops inside the FPGA. The voltage was 5V for older FPGA generations, and is coming down as new generations come (3.3V, 2.5V, 1.8V, 1.5V, 1.2V and even lower for the latest devices).
- The IO voltage (called VCCIO here) is used to power the I/O blocks (= pins) of the FPGA. That voltage should match what the other devices connected to the FPGA expect.

The internal voltage is named "VCC" for Xilinx and "VCCINT" for Altera. The IO voltage is named "VCCO" for Xilinx and "VCCIO" for Altera.

5.7ARCHITECTURE OF XC3S100E:

The spartan –III family of FPGAs is implemented with a regular, flexible, programmable architecture of Configurable Logic Blocks (CLBs) surrounded by a perimeter of programmable INPUT / Output Blocks(IOBs), interconnected by a powerful hierarchy of versatile routing resources. The architecture also provides advanced functions such as Block RAM and Clock blocks as shown below

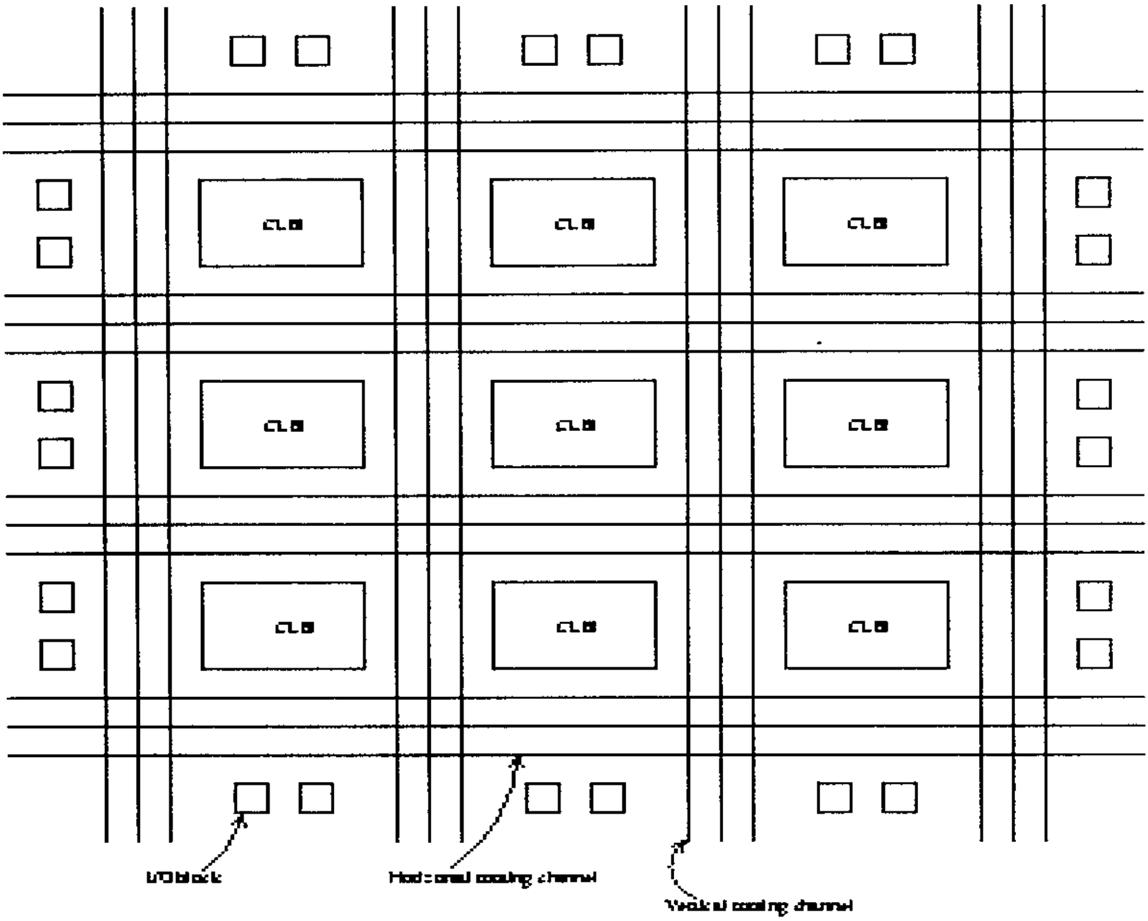


FIG 5.3 ARCHITECTURE OF XC3S100E

Features of XC3S100E are

- System Gates-1,00,000
- Logic Cells-2,160
- CLBs-960
- Block RAM(bits)-72k
- I/O-108

Input / Output Block

The Spartan-iii IOB features inputs and outputs that support 16 I/O signaling standards including LVCMOS, HSTL, SSTL, GTL. The high-speed inputs and outputs are capable of supporting various state of the art memory and bus interfaces. The three IOB register function either as edge triggered D-type flip-flops or as a level sensitive latches.

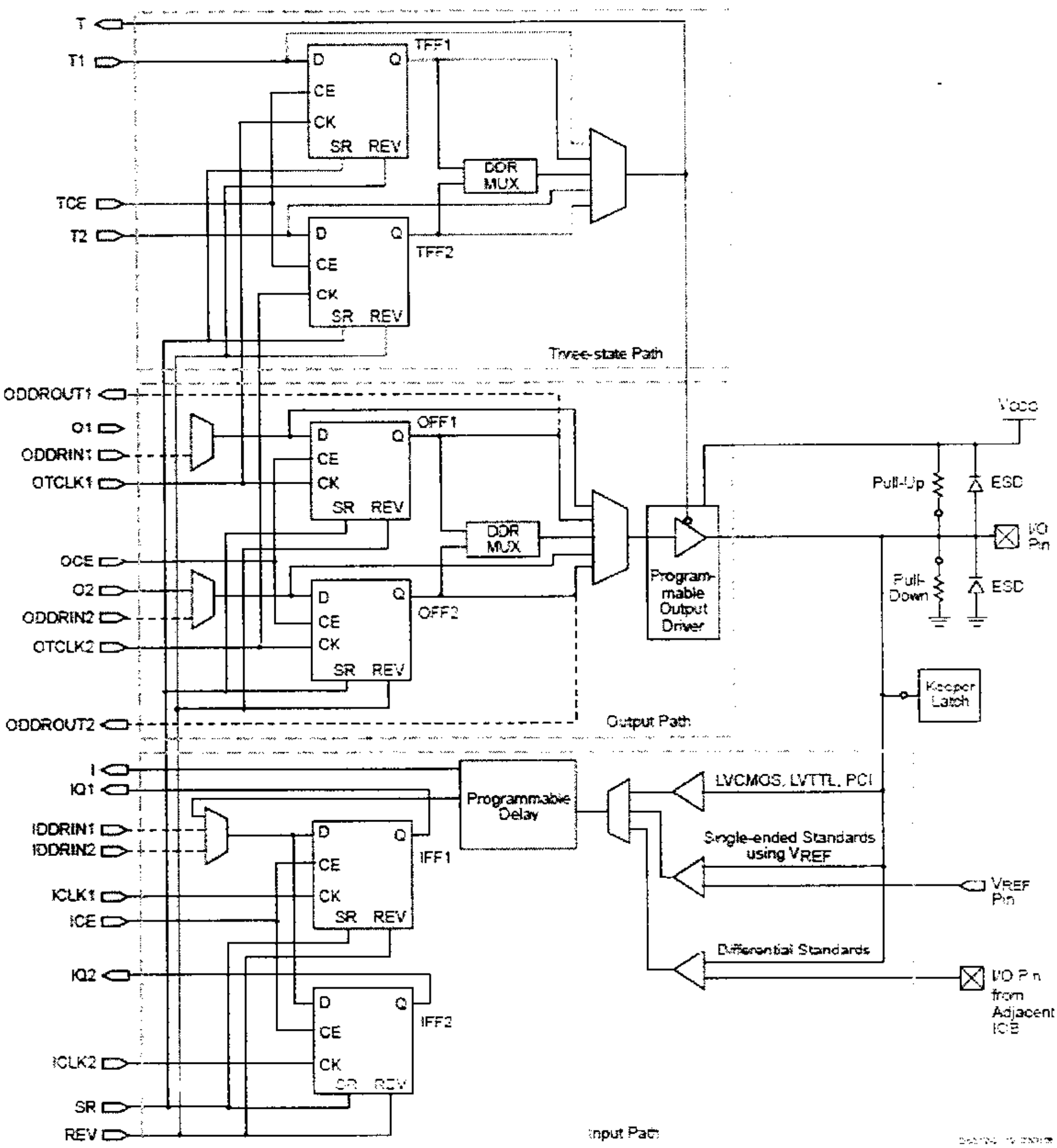


Fig 5.4 Input / Output Block

Logic Cells:

The basic building block of spartan-iii CLB is the logic cell. The combination of a LUT and a storage element is known as a "Logic Cell". The additional features in a slice, such as the wide multiplexers, carry logic, and arithmetic gates, add to the capacity of a slice, implementing logic that would otherwise require additional LUTs. Benchmarks have shown that the overall slice is equivalent to 2.25 simple logic cells. The LUTs located in the top and bottom portions of the slice are referred to as "G" and "F", respectively, or the "G-LUT" and the "F-LUT". The storage elements in the top and bottom
portions of the slice are called FFY and FFX, respectively. Each slice has two multiplexers with F5MUX in the bottom portion.

Block RAM

Spartan-3E devices incorporate 4 to 36 dedicated block RAMs, which are organized as dual-port configurable18 Kbit blocks. Functionally, the block RAM is identical to the Spartan-3 architecture block RAM. Block RAM synchronously stores large amounts of data while distributed RAM, previously described, is better suited for buffering small amounts of data anywhere along signal paths. This section describes basic block RAM functions. Each block RAM is configurable by setting the content's initial values, default signal value of the output registers, port aspect ratios, and write modes. Block RAM can be used in single-port or dual-port modes. The XC3S100E has

- Total no RAM blocks: 4
- Total no of addressable locations(bits) :73,728
- Total columns: 1

The Internal Structure of the Block RAM:

The block RAM has a dual port structure. The two identical data ports called A and B permit independent access to the common block RAM, which has a maximum capacity of 18,432 bits, or 16,384 bits with no parity bits . Each port has its own dedicated set of data, control, and clock lines for synchronous read and write operations. There are four basic data paths, as shown:

1. Write to and read from Port A
2. Write to and read from Port B
3. Data transfer from Port A to Port B
4. Data transfer from Port B to Port A

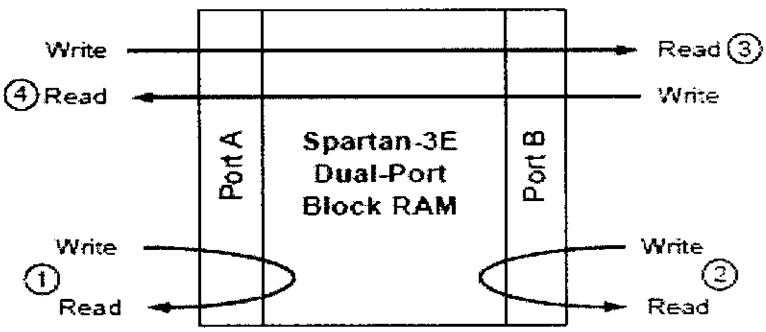


FIG 5.5 STRUCTURE OF RAM

5.8 FEAUTURES OF XC3S100E:

The XC3S200 features up to 200,000 gates14 Ram blocks and up to 284 I/O pins in three low cost, high-volume packages. It includes

- Distributed and Block Memory
- Four Digital Delay Locked Loops per Device
- Versatile I/O Interface Technology
- Full PCI Compliance

5.9 APPLICATIONS OF FPGA

Application of FPGA s include DSP, software-defined radio, aerospace and defense systems, asic prototyping, medical imaging, computer vision, speech recognition, cryptography, bioinformatics, computer hardware emulation and a growing range of other areas. FPGAs originally began as competitors to CPLDs and competed in similar spaces, that of glue logics of PCBs. As their size, capabilities and speed increased they began to take over larger and larger functions to the state where some are now marketed as full systems on chips (SOC).due to their programmable nature, FPGAs are an ideal fit for many different markets such as

- Aerospace &Defence
- Automotive
- Broadcast
- Consumer
- Industrial/science/medical
- Storage &Server

CHAPTER 6

CHAPTER 6
HARDWARE DESCRIPTION

6.1 INTRODUCTION

The Spartan-3E family of Field-Programmable Gate Arrays (FPGAs) is specifically designed to meet the needs of high volume, cost-sensitive consumer electronic applications. The five-member family offers densities ranging from 100,000 to 1.6 million system gates. The Spartan-3E family builds on the success of the earlier Spartan-3 family by increasing the amount of logic per I/O, significantly reducing the cost per logic cell. New features improve system performance and reduce the cost. These Spartan-3E enhancements, combined with advanced 90 nm process technology, deliver more functionality and bandwidth per dollar than was previously possible, setting new standards in the programmable logic industry. Because of their exceptionally low cost, Spartan-3E FPGAs are ideally suited to a wide range of consumer electronics applications.

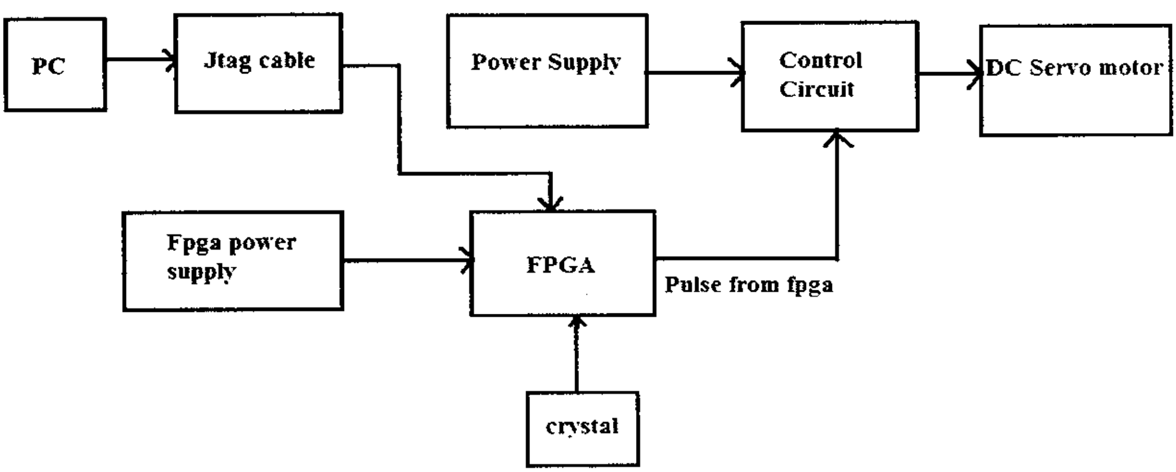


FIG 6.1 SYSTEM BLOCK DIAGRAM

The FPGA device is programmed using Verilog code to generate the gating pulses. The fpga produces the control pulse that is used to control the driver circuit. Servo motor rotates according the driver circuit output and its output is determined by the duration of the pulse or pulse width.

6.2 DRIVER CIRCUIT

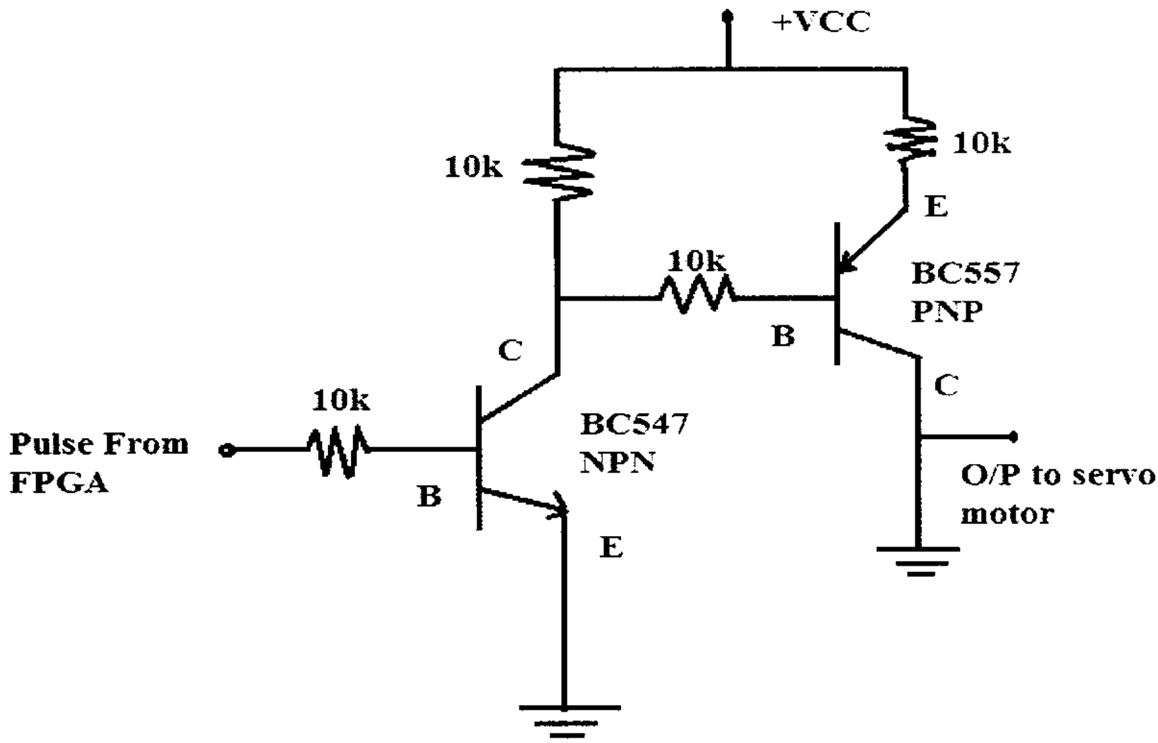


FIG 6.2 CONTROL CIRCUIT

When it receives high pulse from FPGA, NPN transistor starts conducting and the ground is applied to the base of the PNP transistor. PNP transistor starts conducting when it receives low pulse, so its starts conducting and thus output is produced. Based on pulse width, the control output is produced. The output of control circuit is then given to dc servo motor and the position corresponding to the control output is obtained.

The control pulse is used to communicate the angle. The angle is determined by the duration of a pulse that is applied to the control wire. This is called Pulse Coded Modulation .The length of the pulse will determine how far the motor turns.

A 1.5 millisecond pulse, for example, will make the motor turn to the 90 degree position (often called the neutral position). If the pulse is shorter than 1.5ms, then the motor will turn the shaft to closer to 0 degrees. If the pulse is longer 1.5ms, the shaft turns closer to 180 degrees.

6.3 TIMING DIAGRAM

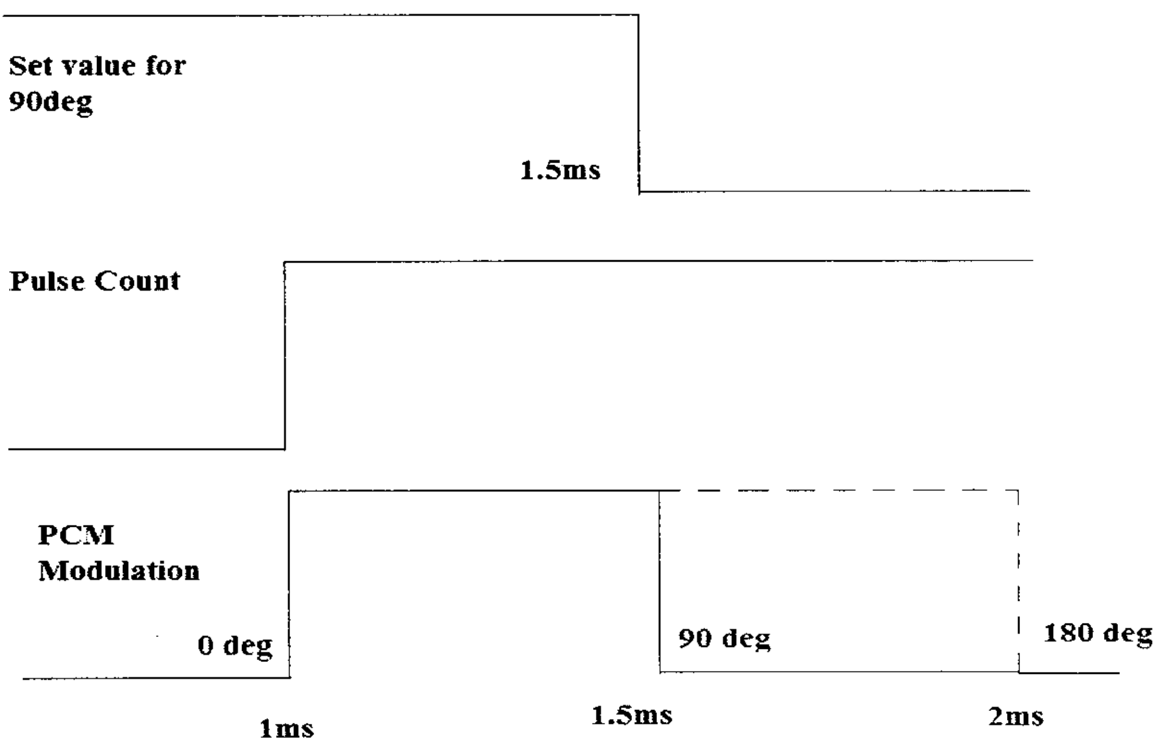


FIG 6.3 TIMING DIAGRAM

6.4 POWER SUPPLY

The present chapter introduces the operation of power supply circuits built using filters, rectifiers, and then voltage regulators. Starting with an ac voltage, a steady dc voltage is obtained by rectifying the ac voltage, then filtering to a dc level, and finally, regulating to obtain a desired fixed dc voltage. The regulation is usually obtained from an IC voltage regulator unit, which takes a dc voltage and provides a somewhat lower dc voltage, which remains the same even if the input dc voltage varies, or the output load connected to the dc voltage changes.

Block Diagram

The ac voltage, typically 220V rms, is connected to a transformer, which steps that ac voltage down to the level of the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation. A regulator circuit removes the ripples and also remains the same dc value even if the input dc voltage varies, or the load connected to the output dc voltage changes. This voltage regulation is usually obtained using one of the popular voltage regulator IC units.

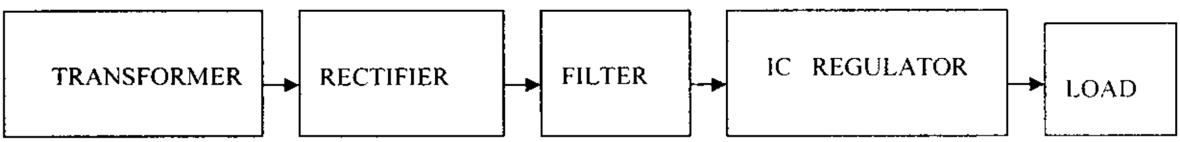


FIG 6.4 BLOCK DIAGRAM (POWER SUPPLY)

6.5 WORKING PRINCIPLE

Transformer

The potential transformer will step down the power supply voltage (0-230V) to (0-6V) level. Then the secondary of the potential transformer will be connected to the precision rectifier, which is constructed with the help of op–amp. The advantages of using precision rectifier are it will give peak voltage output as DC, rest of the circuits will give only RMS output.

Bridge rectifier

When four diodes are connected as shown in figure, the circuit is called as bridge rectifier. The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners.

Let us assume that the transformer is working properly and there is a positive potential, at point A and a negative potential at point B. the positive potential at point A will forward bias D3 and reverse bias D4.

The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow.

The path for current flow is from point B through D1, up through RL, through D3, through the secondary of the transformer back to point B. this path is indicated by the solid arrows.

One-half cycle later the polarity across the secondary of the transformer reverse, forward biasing D2 and D4 and reverse biasing D1 and D3. Current flow will now be from point A through D4, up through RL, through D2, through the secondary of T1, and back to point A. This path is indicated by the broken arrows. The current flow through RL is always in the same direction. In flowing through RL this current develops a voltage corresponding to that . Since current flows through the load during both half cycles of the applied voltage, this bridge rectifier is a full-wave rectifier.

One advantage of a bridge rectifier over a conventional full-wave rectifier is that with a given transformer the bridge rectifier produces a voltage output that is nearly twice that of the conventional full-wave circuit. This may be shown by assigning values to some of the components shown in views A and B. assume that the same transformer is used in both circuits.

The peak voltage developed between points X and y is 1000 volts in both circuits. In the conventional full-wave circuit, the peak voltage from the center tap to either X or Y is 500 volts. Since only one diode can conduct at any instant, the maximum voltage that can be rectified at any instant is 500 volts.

The maximum voltage that appears across the load resistor is nearly-but never exceeds-500 vOlts, as result of the small voltage drop across the diode. In the bridge rectifier shown in view B, the maximum voltage that can be rectified is the full secondary voltage, which is 1000 volts.

Therefore, the peak output voltage across the load resistor is nearly 1000 volts. With both circuits using the same transformer, the bridge rectifier circuit produces a higher output voltage than the conventional full-wave rectifier circuit.

6.6 IC VOLTAGE REGULATORS

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC.

A power supply can be built using a transformer connected to the ac supply line to step the ac voltage to desired amplitude, then rectifying that ac voltage, filtering with a capacitor and RC filter, if desired, and finally regulating the dc voltage using an IC regulator.

The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from mill watts to tens of watts.

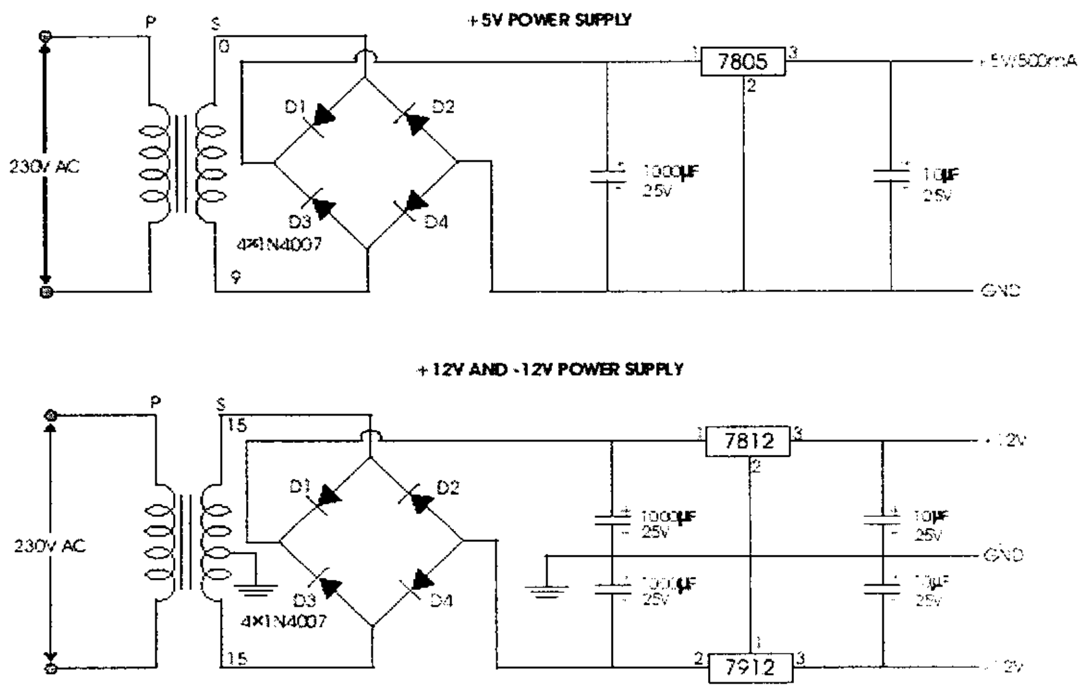


FIG 6.5 CIRCUIT DIAGRAM (POWER SUPPLY)

A fixed three-terminal voltage regulator has an unregulated dc input voltage Vi, applied to one input terminal, a regulated dc output voltage, Vo, from a second terminal, with the third terminal connected to ground.

The series 78 regulators provide fixed positive regulated voltages from 5 to 24 volts. Similarly, the series 79 regulators provide fixed negative regulated voltages from 5 to 24 volts.

- For ICs -5 volts
- For driver circuits, relay circuits – 12 volts

Three-terminal voltage regulators

Fig shows the basic connection of a three-terminal voltage regulator IC to a load. The fixed voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated output dc voltage, Vo, from a second terminal, with the third terminal connected to ground. For a selected regulator, IC device specifications list a voltage range over which the input voltage can vary to maintain a regulated output voltage over a range of load current. The specifications also list the amount of output voltage change resulting from a change in load current (load regulation) or in input voltage (line regulation).

Fixed Positive Voltage Regulators

The series 78 regulators provide fixed regulated voltages from 5 to 24 V. IC 7812 is connected to provide voltage regulation with output from this unit of +12V dc. An unregulated input voltage Vi is filtered by capacitor C1 and connected to the IC's IN terminal. The IC's OUT terminal provides a regulated + 12V which is filtered by capacitor C2 (mostly for any high-frequency noise).

The third IC terminal is connected to ground (GND). While the input voltage may vary over some permissible voltage range, and the output load may vary over some acceptable range, the output voltage remains constant within specified voltage variation limits. These limitations are spelled out in the manufacturer's specification sheets. A table of positive voltage regulated ICs is provided.

IC Part	**Output Voltage (V)**	**Minimum Vi (V)**
7805	+5	7.3
7806	+6	8.3
7808	+8	10.5
7810	+10	12.5
7812	+12	14.6

TABLE 6.1 POSITIVE VOLTAGE REGULATORS IN 7800 SERIES

6.9 ALGORITHM FOR FPGA IMPLEMENTATION OF FUZZY WITH PI CONTROL

STEP1: START

STEP2: ASSIGN INPUT OUTPUT PORTS FOR THE CLK, RST, PCM SIGNALS

STEP3: INITIALIZE DATA REGISTER

STEP4: CALCULATE COUNT VALUE

STEP5: IMPLEMENT FUZZY RULES IN DATA REGISTER

STEP6: SETPOINT IS GIVEN AS BINARY CODES

STEP7: INCREMENT COUNT VALUE AND COMPARE THE DATA REGISTER WITH COUNT VALUE

STEP8: PCM IS PRODUCED BASED ON THE GIVEN SET POINT OUTPUT THE PULSE

STEP9: STOP

CHAPTER 7

CONCLUSION

The position control of dc servo motor was simulated for various controllers such as conventional PID, fuzzy with PI and fuzzy with PID. The performance of all the above controllers were analyzed and fuzzy with PI controller provides the better result.

In this project, the design and implementation of position control of dc servo motor using FPGA is carried out. The Verilog code was implemented and verified on XILINX SPARTAN-3E FPGA device. The control pulses from the output of FPGA device was given to the Servomotor through Control circuit. The position of servo motor controlled by the duration of pulse generated from FPGA is verified using hardware.

REFERENCES

1. "*Development of Fuzzy-Logic-Based Self Tuning PIController for Servomotor*", 2008 10th Intl. Conf. on Control, Automation, Robotics and Vision Hanoi, Vietnam, 17–20 December 2008, IEEE.

2. "*Adaptive fuzzy-neural-network controls for a DSP-based permanent magnet linear synchronous motor servo drive*", Faa-Jeng Lin; Po-Hung Shen; Fuzzy Systems, IEEE.

3. "*Position control of DC motors via variable structure systems control: a chattering alleviation approach*", Fu-Juay Chang; Hsiang-Ju Liao; Shyand Chang; Industrial Electronics, IEEE.

4. "*Automated calibration of a fuzzy logic controller using a cell state space algorithm*", Petersen, D.P.; Smith, S.M.; Comer, D.J.; Control Systems Magazine, IEEE.

5. "*Basics of VHDL programming*", A.Baskar.

6. "*Fuzzy logic controller*", Timothy J Ross.

APPENDIX XLIX

FEATURES OF XC3S100E

- Very low cost, high-performance logic solution for high-volume, consumer-oriented applications
- Proven advanced 90-nanometer process technology
- Multi-voltage, multi-standard Select IO interface pins
- Up to 376 I/O pins or 156 differential signal pairs
- Enhanced Double Data Rate (DDR) support
- Abundant, flexible logic resources
- Densities up to 33,192 logic cells, including optional shift register or distributed RAM support
- Efficient wide multiplexers, wide logic
- Fast look-ahead carry logic
- Enhanced 18 x 18 multipliers with optional pipeline
- IEEE 1149.1/1532 JTAG programming/debug port
- Hierarchical Select RAM memory architecture
- Up to eight Digital Clock Managers (DCMs)
- Eight global clocks and eight clocks for each half of device, plus abundant low-skew routing
- Complete Xilinx ISE, WebPACK development system support
- MicroBlaze, PicoBlaze embedded processor cores
- Fully compliant 32-/64-bit 33/66 MHz PCI support

SUMMARY ABOUT SPARTAN-3E FAMILY

Device	System Gates	Equivalent Logic Cells	CLB Array (One CLB = Four Slices)				Distributed RAM bits[1]	Block RAM bits[1]	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs	Total Slices						
XC3S100E	100K	2,160	22	16	240	960	15K	72K	4	2	108	40
XC3S250E	250K	5,508	34	26	612	2,448	38K	216K	12	4	172	68
XC3S500E	500K	10,476	46	34	1.164	4,656	73K	360K	20	4	232	92
XC3S1200E	1200K	19,512	60	46	2,168	8,672	136K	504K	28	8	304	124
XC3S1600E	1600K	33,192	76	58	3,688	14,752	231K	648K	36	8	376	156

Table a.1 Spartan-3E family