VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

EEE4019 – Advanced Digital System Design using FPGAs

Project Report

Servo Motor Control using FPGA

Team Members

Ruturaj A. Nanoti – 18BEE0134

Amitvikram S. Pujar – 18BEE0135

Advait R. Marathe – 18BEE0164

Under the Guidance

Dr. Balamurugan S.

Associate Professor

School of Electrical Engineering

VIT, Vellore

Winter Semester 2020-21

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF ELECTRICAL ENGINEERING

CERTIFICATE

This is to certify that the project work entitled "Servo Motor Control using FPGA" by Ruturaj A. Nanoti (18BEE0134), Amitvikram S. Pujar (18BEE0135), Advait Ravi Marathe (18BEE0164) submitted to Vellore Institute of Technology University, Vellore, in partial fulfilment of the requirement for J component of the course titled **"Advanced Digital System Design using FPGAs" (EEE4019)** is a work carried out by us under my supervision.

The project fulfils the requirement for J component as per the regulations of this Institute and in my opinion meets the necessary standards for submission.

Sr. No.	Name	Registration Number
1.	Ruturaj A. Nanoti	18BEE0134
2.	Amitvikram S. Pujar	18BEE0135
3.	Advait Marathe	18BEE0164

Dr. Balamurugan S

Subject Faculty

Asso.Prof / SELECT

VIT

Table of Contents

I. Introduction

The introduction of AC drives has been a great revolution in the industry. With less bulky components and same power rating as the DC motor, in lesser cost, are some of the factors which drove the revolution. But the grave disadvantage in AC Drives is complex control of the motor.

II. Problem statement and Background

Now, the main problem associated with all the hardware associated with control of the AC Motor, is that they are quite rigid. For changing the duty ratio of a certain motor, the circuit has to be modified, which is a very tedious task, especially when the complex pulses are generated, as the pulses become complex, the circuitry follows the same trend.

Thus, here we are presenting **FPGA based Servo Motor Control.**

Now, the main issue of generating PWM pulses can be carried out by using DSPs and Microcontrollers as well, but modification of both of these is quite complex. Since, the programming in FPGA is quite simple, the modification can be done quite easily. Plus, for circuit verification, lesser volume production and prototyping purposes, FPGA can be implemented at much lower cost.

Now, why servo motor? This is because servo motor has been a huge success since its invention. The discrete control of motor makes its great for mainly robotic applications, i.e., precise joint control in robotic arm. And since such control can be achieved, it will be of great use in Industry 4.0, where now most of the tasks will be automated and will be controlled remotely.

III. Literature Survey

SI.No.	Name of the Paper	Authors	Year
1.	Design and Implementation of SVPWM servo control System based on FPGA	Pang Haiyan, Xie Yun, Xiao Shanshan, Chen Bingcheng	2010
2.	Embedded FPGA Controller for robot arm in material handling using reconfigurable nicompact RIO	B.Hemalatha V.R.Ravi, S.Divya, M.Uma	2014
3.	FPGA-based stepper motor controller: An FPGA-based design project to improve level of the complexity in project-based learning	Ruzali Rustam, Mariam Imbrahim Al Khoory	2018
4.	FPGA-Based Intelligent-Complementary Sliding-Mode Control for PMLSM Servo-Drive System	Faa Jeng Lin, Jonq-Chin Hwang, Po-Huan Chou, Ying-Chih Hung	2010
5.	FPGA Based Functional Link Radial Basis Function Network Control for PMLSM Servo Drive System	Faa Jeng Lin, Po-Huan Chou	2010

IV. Design Analysis

1. Design and Implementation of SVPWM servo control System based on FPGA

SVPWM (Space Vector Pulse Width Modulation) has the advantage of low harmonic distortion, high utilization of voltage, and easy digital realization, over conventional SPWM technique. All the current processes use complex software algorithms in microcontrollers (DSP). To realize SVPWM in hardware FPGA is used. The hardware implementation is achieved by calculating the 2-axis stator voltage component and transforming it into 3-axis, and the output waveform is converted into the best PWM pulse through the adjusted SVPWM converter.

2. Embedded FPGA Controller for robot arm in material handling using reconfigurable nicompact RIO

PID controller is designed in embedded FPGA architecture of National Instruments CompactRIO (NI cRIO-9073) and interfaced with servo motors through NI 9505 DC Servo drive modules. A path generation technique is used to compute the speed profile of the motor. The servo control system is implemented using an encoder loop, PID control loop and PWM generation loop. The PID algorithms developed in LabView are embedded in FPGA of CompactRIO module to achieve precision.

3. FPGA-based stepper motor controller: An FPGA-based design project to improve level of the complexity in project-based learning

An FPGA-based stepper motor controller for open loop control system has been applied. The open loop system for the controller has been developed by utilizing the Xilinx's FPGA prototyping board along with VHDL, stepper motor, power driver circuit and tachometer.

V. Detailed Methodology

1. Studying and implementation of previously developed techniques to control the servo motor.

2. The control methodology involves switch to angle and angle to constant vale decoder, a counter and a comparator.

3. The counter takes in the system clock and clear. Every clock cycle it increments a count value until it hits a certain constant value. This value is what is needed to get the appropriate frequency. Once it reaches that value it resets. So, it counts to the constant and then goes back to zero. This controls the frequency.

4. The switch to angle decoder takes in a switch value and maps it to an angle value.

5. The angle decoder takes in an angle value and maps it to a constant value that will get the appropriate duty cycle.

6. The comparator then takes in the count value and constant value from the angle decoder. If the count value is less than the constant than the output is 1, otherwise the output is 0. This creates the PWM signal.

VI. Algorithm Development

The algorithm is explained by step-by-step as follows:

1. Takes the position of a servo motor as an 8-bit input.

2. The position of the servo motor is upscaled from 0 to 255 to a new range of the 50,000 to 1,00,000. It is essential, as we have a fast clock signal as compared to the desired PWM waveform. The upscaling is done by adding 165 to the position and shifting it left by 8-bits. For example, the 0 + 165 = 165 and 165 * 2^8 is equal to 42,240.

3. The upscaled position is compared with the desired position. If the position is greater than what expected a pulse is generated according to the requirement.

4. The clock signal of 50 MHz is used for the purpose, which means the width of each pulse is approximately 1ms to 2ms, considering the upscaling of 50,000 to 1,00,000.

5. The varying pulses are generated by the code. Since the operation includes serial input and serial output operation, it takes approximately 10 cycles for bit transmission. Since, each bit manipulation takes maximum of 2ms or 1,00,000 clock cycles, each PWM waveform is of period of 20ms approximately.

VII. Standards Adopted

- In order to prevent potentially unsafe attributes of HDL code from leading to unsafe design issues, the use of HDL coding standards is required by various safety-critical industries such as DO-254.

- The VHDL/Verilog standards included cover essential areas in HDL coding such as coding style, readability, simulation, clock/reset management, design reuse, coding for safe synthesis and implementation, clock domain crossings (CDC) and design for test (DFT).

- Additionally, Clock Domain Crossing issues like non-synchronized CDC transfers, convergence/divergence, combinational logic on CDC paths and incorrect synchronizer structures detected during linting process can be easily investigate by using dedicated windows like CDC Viewer, CDC Schematic and RTL Schematic.

VIII. Trade Off Identified

- The main Trade Off is accuracy for speed. For getting quick response, the servo motor may not exactly match to the exact desired position due to high frequency of CLK. This implies that the value has to be upscaled to introduce sufficient delay required for the application.

- Secondly, cost for resolution. For higher resolution of the PWM pulse-width, servo motor with higher no. of poles has to be selected which compromises on the cost bared for the project implementation.

IX. Algorithm Implementation

The algorithm was implemented using the Verilog code. This Verilog code was executed by the means of Xilinx Vivado Version 2020.2. The desirable motor position was kept constant while the position of servo motor was varied in the testbench. Here, are the codes for the algorithm -

- **Top Module Code –**

```
/*
  Servo Controller
  Takes an 8-bit position as an input
  Output a single pwm signal with period of ~20ms
  Pulse width = 1ms -> 2ms full scale. 1.5ms is
center position
*/

module RC_Servo (input clk, rst, input [7:0]
position, output servo);
  reg pwm_q, pwm_d;
  reg [19:0] ctr_q = 74880, ctr_d;
  assign servo = pwm_q;

  // Position (0-255) maps to 50,000 - 100,000 clock
cycles (which corresponds to 1ms - 2ms @ 50MHz)
  // This is approximately (position + 165) << 8
  // The servo output is set by comparing the
position input with the value of the counter ctr_q
  // The ctr_q can be considered desired position of
servo motor
  // To get to desired position wave is given to the
motor.
```

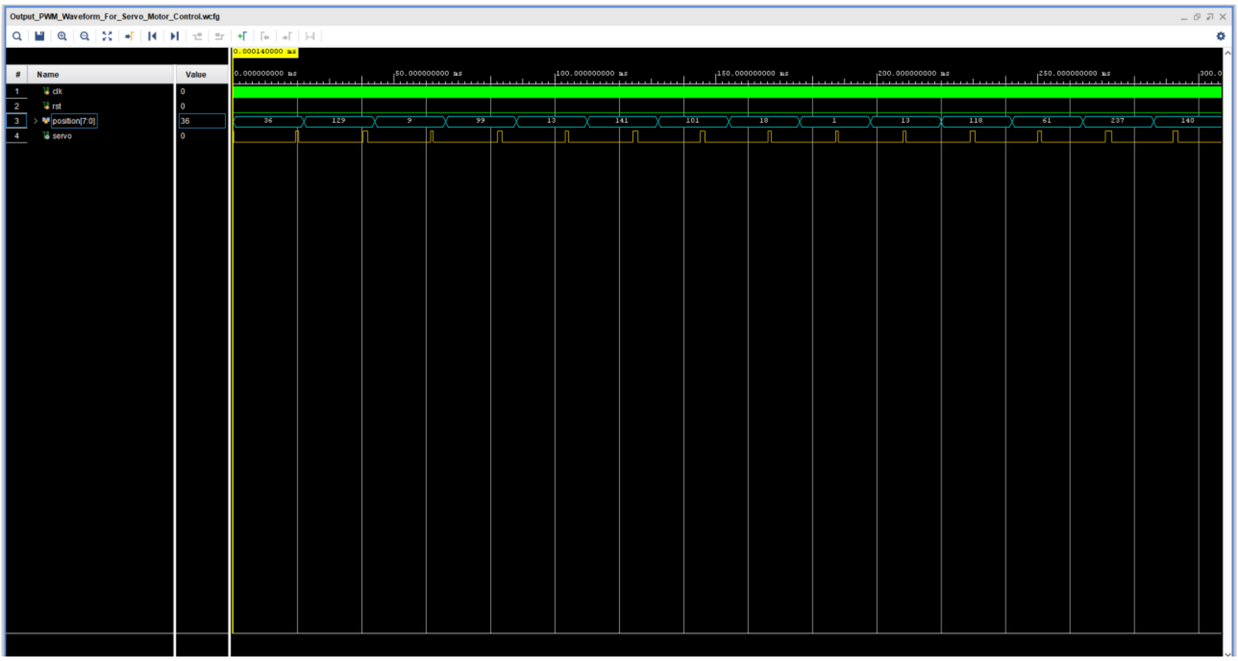
```
  always @(*)
  begin
    ctr_d = ctr_q + 1'b1;
    if (position + 9'd165 > ctr_q[19:8])
    begin
      pwm_d = 1'b1;
    end
    else
    begin
      pwm_d = 1'b0;
    end
  end
  always @(posedge clk) begin
    if (rst)
    begin
      ctr_q <= 1'b0;
    end
    else
    begin
      ctr_q <= ctr_d;
    end
    pwm_q <= pwm_d;
  end
endmodule
```

- **Testbench Code –**

```
`timescale 1ns / 1ps
module RC_Servo_tb;
reg clk = 1'b0;
reg rst = 1'b0;
reg [7:0] position;
wire servo;
RC_Servo UUT (clk, rst, position, servo);
initial repeat(100)
begin
position = $random;
#22000000;
// The pulse width is approx 21.5ms
end
always #10 clk = ~clk;
// As 50 MHz CLK signal
// The time period is 20 ns and clock complements at
every 10ns.
endmodule
```

X. Results & Discussion

The above code was executed for 2.2 seconds. Here, is the output we obtained from the code.



Output_PWM_Waveform_For_Servo_Motor_Control.wcfg

#	Name	Value
1	clk	0
2	rst	0
3	position[7:0]	36
4	servo	0

The position of servo motor is changed after every 22 ms. It can be clearly seen that for various position, the pulses of varying width are generated. These pulses can be fed to the servo motor. As long as pulse output is 1, the motor moves and it settles once the pulse output becomes zero.

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 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 in the counter clockwise direction toward the 0° position, and any longer than 1.5ms will turn the servo in a clockwise direction toward the 180° position.

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.

XI. Challenges

One of the major challenges include achieving high speed performance while not compromising on accuracy. As the response gets faster, the accuracy of the system tends to decrease, hence, a middle ground needs to be adopted. The next challenge is the appropriate scaling of the maximum and minimum step size of the position of servo motor. The upscaling is essential to obtain the waveform of appropriate time period of ~ 20ms.

XII. Conclusion

Servos are used in radio-controlled airplanes to position control surfaces like elevators, rudders, walking a robot, or operating grippers. Servo motors are small, have built-in control circuitry and have good power for their size.

In food services and pharmaceuticals, the tools are designed to be used in harsher environments, where the potential for corrosion is high due to being washed at high pressures and temperatures repeatedly to maintain strict hygiene standards. Servos are also used in in-line manufacturing, where high repetition yet precise work is necessary. With so many diverse applications, the FPGA control of servo motor can help to ease out the challenges and achieve a better control and help us to automate most of the mundane task.

XIII. Individual Contribution

The project included various steps and they were executed accordingly –

- First of all, in the start of project all the relevant references were gathered. About 3 to 4 references were looked after which were simultaneously documented.
  (Done by – Amitvikram S. Pujar and Ruturaj A. Nanoti)

- Then the next step was to write an appropriate code which satisfied the project requirements. Most, of the online references contained VHDL code, the detailed code helped us to get a better idea of the algorithm. Now, the main task was to implement the same algorithm via the means of the Verilog code. Since, the model is solely software based, the module code along with the relevant testbench code was to be written.
  (Done by – Ruturaj A. Nanoti and Advait R. Marathe)

- Since, the Verilog code was developed, its prone to human errors. The code may have syntax errors or the logical errors. The logical errors were resolved by looking after some more references. The syntax errors were resolved by looking the various standards in the Verilog language.
  (Done by – Advait R. Marathe and Amitvikram S. Pujar)

- The project and its pitch are not successful unless its documented properly. The document should contain all the expected content in proper format and all the results should be clearly perceivable.
  (Done by – Ruturaj A. Nanoti, Amitvikram S. Pujar, Advait R. Marathe)

References

[1] B. Hemalatha, V. R. Ravi, S. Divya and M. Uma, "Embedded FPGA controller for robot arm in material handling using reconfigurable nicompact RIO," Second International Conference on Current Trends In Engineering and Technology - ICCTET 2014, Coimbatore, India, 2014, pp. 125-131, doi: 10.1109/ICCTET.2014.6966274.

[2] Pang Haiyan, Xie Yun, Xiao Shanshan and Chen Bingcheng, "Design and implementation of SVPWM servo control system based on FPGA," 2009 Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA), Wuhan, China, 2009, pp. 333-336, doi: 10.1109/PACIIA.2009.5406593.

[3] R. Rustam and M. I. Al Khoory, "FPGA-based stepper motor controller: An FPGA-based design project to improve level of the complexity in project-based learning," 2018 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, Sharjah, Abu Dhabi, United Arab Emirates, 2018, pp. 1-5, doi: 10.1109/ICASET.2018.8376928.

[4] Lin, Faa-Jeng, et al. "FPGA-based intelligent-complementary sliding-mode control for PMLSM servo-drive system." IEEE transactions on power electronics 25.10 (2010): 2573-2587.

[5] Lin, Faa-Jeng, and Po-Huan Chou. "FPGA based functional link radial basis function network control for PMLSM servo drive system." The 2010 International Power Electronics Conference-ECCE ASIA-. IEEE, 2010.