



PWM based servo motor controller project

Project by Mantvydas Klimavičius, Swan Massart, Marek Wilczewski, Roland Policsek

1. Introduction

This project aims to design a digital system to control a servo motor using VHDL and implement it on a Digilent Nexys A7 FPGA board. The core idea is to generate a PWM signal with precise timing based on user input, to adjust the motor's angle from -90° to $+90^\circ$.

Project Description:

In this project, we designed and implemented a servo motor control system using VHDL on a Digilent Nexys A7 FPGA development board. The objective was to control the position of a standard servo motor by generating a precise PWM (Pulse Width Modulation) signal based on user input.

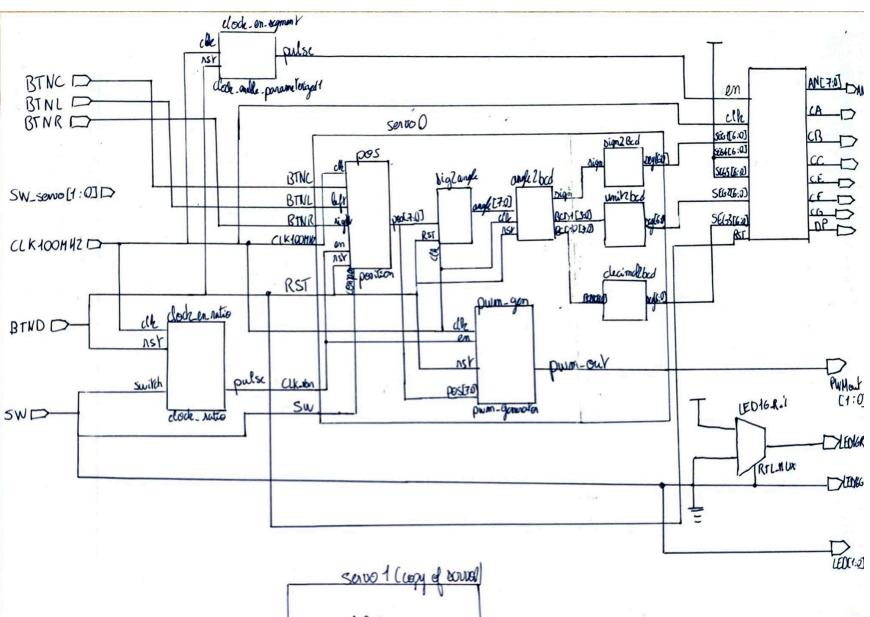
The system allows the user to adjust the servo angle within a defined range (from -90° to $+90^\circ$), with real-time feedback displayed on a 7-segment display. The PWM signal is generated using a custom VHDL module that translates the desired angle into the appropriate pulse width. Timing precision and signal stability were crucial aspects of the design to ensure smooth and accurate movement of the servo.

To control the servo angle, we used input peripherals available on the Nexys board, such as switches or rotary encoders, allowing intuitive adjustment. The project highlights key digital design concepts such as clock division, finite state machines, and pulse generation.

Objectives

The main objective of the project is to achieve accurate angular control of a servo motor by using a VHDL-based system implemented on an FPGA. The user should be able to select an angle in real-time using a rotary encoder. This selected angle is then converted into a corresponding PWM signal that drives the motor. Additionally, the selected angle is displayed on a 7-segment display for direct visual feedback.

VHDL Implementation



The project consists of several key VHDL components. The PWM generator is central to the system and ensures precise pulse widths to control the servo's position. The angle decoder interprets rotary encoder inputs and updates the target angle dynamically. The display driver handles the real-time rendering of the angle on the 7-segment display, while the clock divider provides slower clocks required for PWM generation and display refreshing. The use of finite state machines and counters is essential in managing timing and sequencing for both motor control and visual output.

```
27 begin
28     if (aclr = '1') then
29         q_s <= (others => '0');
30     elsif rising_edge(clk) then
31         q_s <= ('0' & signed(a)) + ('0' & signed(b));
32     end if;
33     -- clk'd
34 end process;
35 end signed adder arch;
```

Challenges and Solutions

Throughout development we ran into several implementation hurdles inside Xilinx Vivado itself. Integrating the design and closing timing took multiple iterations, and we repeatedly hit warnings that forced us to reshuffle our HDL. The most persistent difficulties involved clocking and synchronisation: passing signals between the 100 MHz system clock and the slower enable domains occasionally caused metastability and skew, showing up as sporadic PWM jitter and glitches on the display. We resolved these issues by adding explicit clock-domain-crossing registers, tightening timing constraints, and validating the fixes with Vivado's timing analyser and on-board logic probes.

Applications and Future Work

We also planned to use the accelerometer built into the Nexys board to keep the servo level.

Conclusion

The project successfully demonstrates the potential of FPGA technology and VHDL for real-time motor control. The use of the Nexys A7 board enables a fully hardware-based solution that is flexible, precise, and responsive. The design process helped reinforce key concepts in digital design, including state machines, clock management, and modular architecture. The resulting system is both functional and instructive, showing how even simple hardware components can be controlled effectively using FPGA logic.

