PWM Generator

Special Assignment and Lab Project Report

Course: 2EC202CC23 - FPGA based System Design

Submitted by:

Vansh Muchhala

(23BEC204)

Department of Electronics and Communication Engineering,

Institute of Technology,

Nirma University,

Ahmedabad 382 481

[March- 2025]

Abstract

Pulse Width Modulation (PWM) is a circuit which produces a periodic digital signal with different duty cycles. Duty cycle determines the percentage of the time signal which remains high in a single period. It is widely used in controlling power delivery in various applications, such as motor control, LED brightness adjustment, and communication systems. This project aims to design and simulate a PWM generator using Verilog, which takes an 8-bit input to control the duty cycle of the output signal. While we provide the duty cycle which is in the form of 8 bit input value, this variable duty cycle provides flexibility and makes it more controllable

Contents

1. Introduction

At its core, PWM works by varying the duty cycle of a square wave signal, The Duty Cycle refers to the proportion of the time the signal is high (ON) versus low (OFF) for the square wave period. Adjusting the duty cycle allows for the changing of average power voltage, thereby controlling the power without significant heat loss, So, PWM is most suitable for applications where energy efficiency is important.

A digital circuit or PWM generator can be implemented in Verilog HDL. Verilog is favorable when precise management of timing and signal is needed, such as in FPGA. In Verilog, a PWM generator can be modelled by counters, comparators, and clock divisions. A clock oscillating at high frequency is divided into a square wave with variable duty cycle.

This paper explains the design and implementation of PWM generator in Verilog. The project will outline the steps involved in creating a configurable PWM generator that can be used for various applications, such as motor speed control, power regulation, and LED dimming. Additionally, it will explain the hardware description and simulation process to ensure the system performs efficiently and as expected. Through this project, we aim to provide insight into the power of Verilog in creating high-performance, real-time systems for embedded applications.

2. Literature survey/State of the art technology available

PWM is generated by varying the pulse width in a fixed time period. This can be done in various ways, including using Analog Circuits, Digital Controllers, and more sophisticated Microcontroller-based methods. The key elements involved in the design and operation of PWM generators are frequency control, duty cycle, and signal integrity:

1. **Microcontrollers (MCUs)**: Arduino, STM32, PIC, and ATmega microcontrollers feature PWM output capabilities.
2. **Field-Programmable Gate Arrays (FPGAs)**: FPGA-based PWM implementations offer high-speed and highly customizable PWM generation.
3. **Dedicated PWM ICs**: ICs like the TL494 and SG3525 are specialized for PWM signal generation.
4. **Software-Based PWM**: Software implementations on general-purpose processors can generate PWM but are often limited by CPU speed and real-time constraints.

PWM generation technologies have evolved significantly, moving from simple Analog circuits to highly sophisticated digital and hybrid systems. These systems have enabled precision control over power, enhanced efficiency in power conversion, and reduced the impact of noise. Recently, the use of machine learning (ML) and artificial intelligence (AI) has been explored to optimize PWM generation, particularly in power electronics.

3. Limitations of currently available technology

Pulse Width Modulation (PWM) technology has seen significant advancements, there are still several limitations and challenges that affect the performance and application of PWM generators in modern systems. These limitations primarily arise from the complexity of control, noise, thermal management and efficiency

Electromagnetic Interference (EMI) and Noise:

PWM signals have sharp transitions between high and low states, which can generate electromagnetic interference (EMI) and noise. These high-frequency components are a significant concern in sensitive electronic systems such as communications, medical devices, and audio equipment

Thermal Management and Power Loss

High-frequency PWM operation requires switching devices to turn on and off rapidly. The rapid switching leads to power losses in the switching elements, particularly in the form of heat. In high-power applications like motor control or power supplies, managing the thermal load becomes a major issue. Also, Excessive heat can lead to device failure or reduced system lifespan.

Duty Cycle Resolution: Many PWM generators, especially those in microcontrollers or simpler digital circuits, have limited resolution in the control of the duty cycle. The resolution of the duty cycle determines how finely it can be adjusted.

Current PWM generator technology offers great flexibility and precision in a variety of applications, it still faces several limitations related to EMI, thermal management, resolution, and power losses. The challenge lies in overcoming these limitations to improve efficiency, reduce noise, and maintain performance in high-power, high-frequency, and multi-phase systems.

4. Proposed Solution/Methodology

To address the limitations of current PWM generator technology, several methodologies and innovative solutions can be proposed to enhance performance, reduce noise, improve efficiency, and manage power dissipation in various applications.

Reducing Electromagnetic Interference (EMI) and Noise

Spread Spectrum PWM: One of the most effective ways to reduce EMI is by using spread-spectrum PWM. This technique spreads the PWM signal's frequency across a range instead of a single sharp frequency. This reduces the peak EMI by lowering the harmonic content and making the interference less concentrated.
Filters and Shielding: Incorporating low-pass filters on the output and shielding of the PCB can significantly reduce high-frequency noise. Filters can attenuate unwanted high-frequency components that arise from PWM switching.

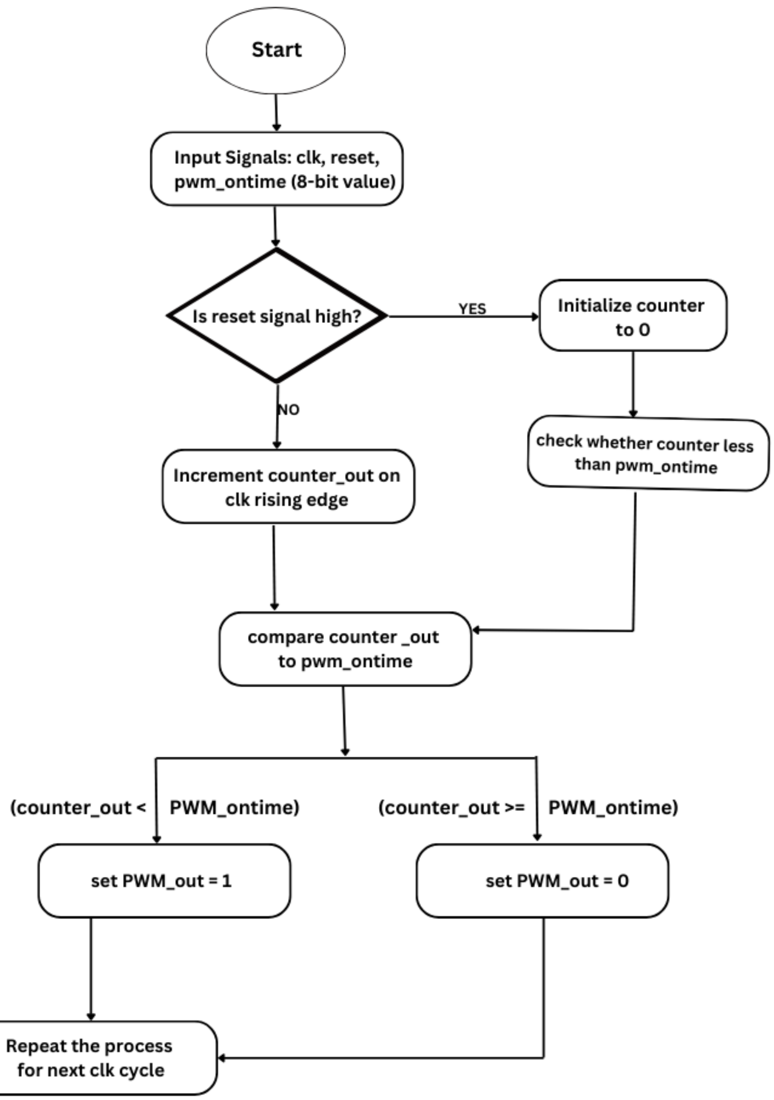
Improving Thermal Management and Reducing Power Losses

Use of High-Efficiency Switches (Wide-Bandgap Semiconductors): Replacing conventional silicon-based switching devices with wide-bandgap semiconductors like SiC (Silicon Carbide) or GaN (Gallium Nitride) can significantly reduce thermal losses.
Thermal Management Solutions: Active cooling systems, like heat sinks or even liquid cooling in high-power applications, can be used to manage the thermal load effectively. Advanced thermal design techniques, such as optimizing PCB layout

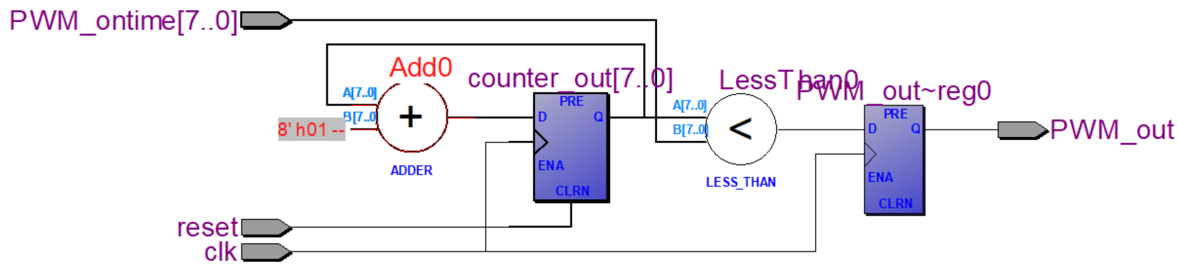
Improving Duty Cycle and Frequency Resolution

High-Resolution Digital PWM Generators: Modern microcontrollers, Digital Signal Processors (DSPs), and FPGAs can be programmed to generate PWM signals with higher resolution in both frequency and duty cycle.

5. Flowchart of the code

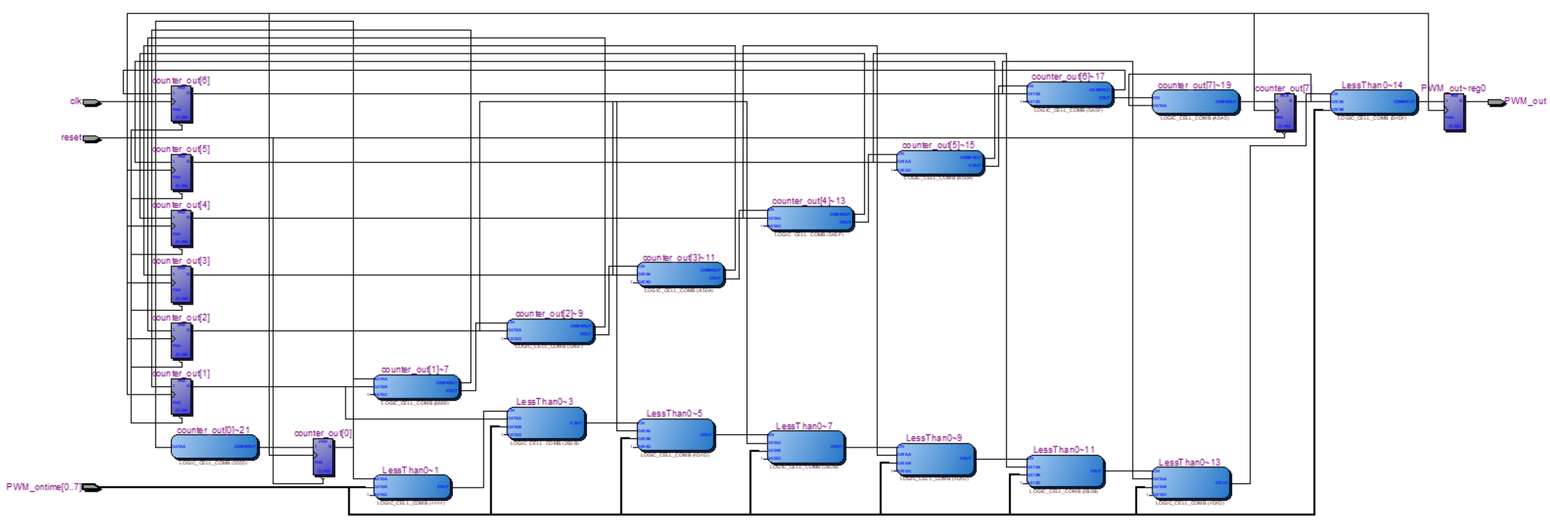


6. RTL



PWM_ontime[7..0]
Add0
counter_out[7..0]
LessThan0
PWM_out~reg0
A[7..0]
B[7..0]
8' h01 --
+
PRE
D
Q
ENA
CLRN
ADDER
A[7..0]
B[7..0]
<
LESS_THAN
PRE
D
Q
ENA
CLRN
PWM_out
reset
clk

7. TTL



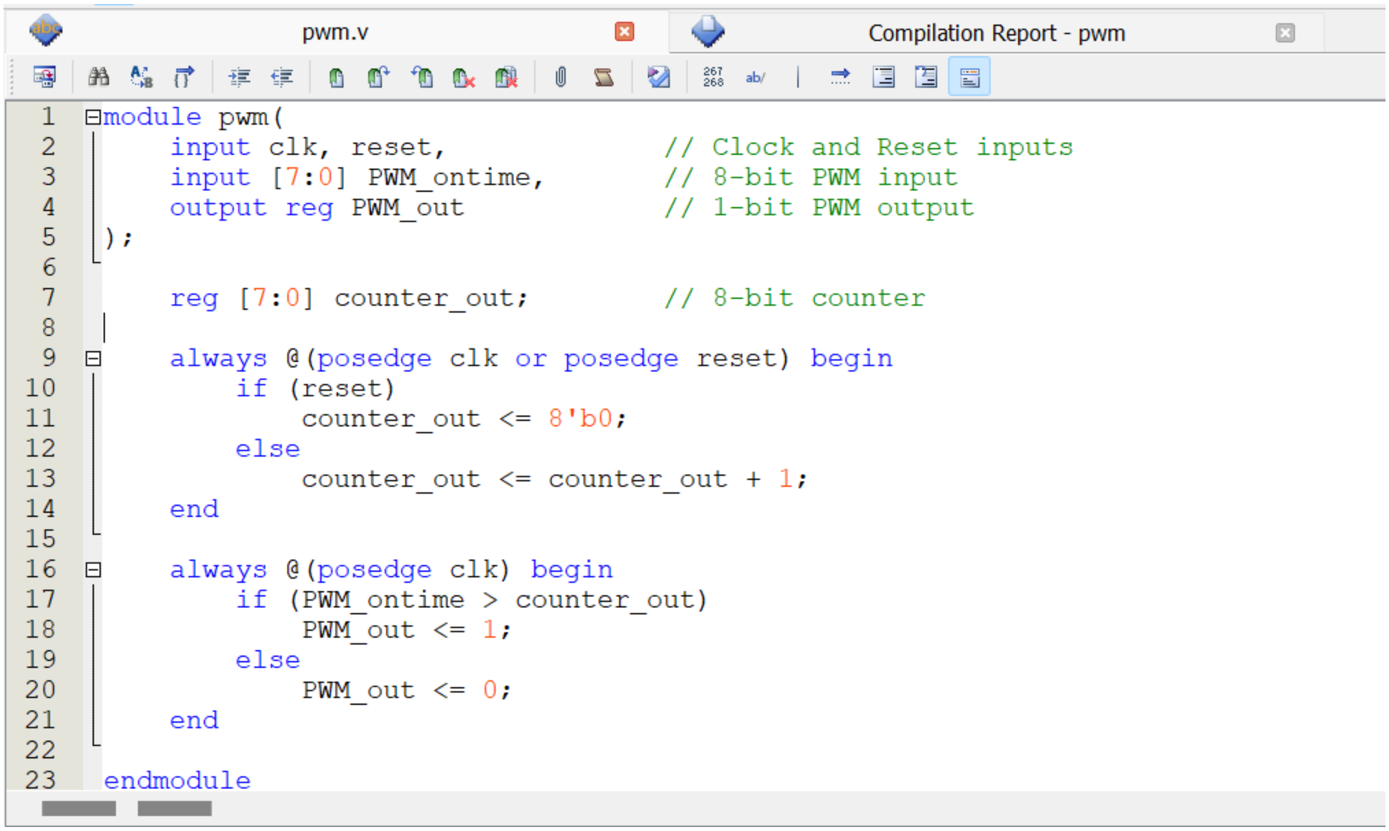
8. Conclusion

In this project, we explored the various aspects of Pulse Width Modulation (PWM) generators, focusing on the current limitations and proposing solutions to address the challenges that arise in modern applications. PWM technology plays a crucial role in efficiently controlling power delivery in diverse fields such as motor control, power conversion, audio amplification, and communication systems. However, despite its wide usage, PWM generators face limitations related to electromagnetic interference, thermal management, precision in frequency and duty cycle control, and efficiency in high-power applications. As the demand for high-efficiency, low-noise, and high-precision systems continues to grow, especially in areas like electric vehicles, renewable energy systems, and consumer electronics, the continued development and optimization of PWM generators will be essential for meeting these challenges

9. References

1. Brown, S. & Vranesic, Z. (2009). *Fundamentals of Digital Logic with Verilog Design*. McGraw-Hill.
2. Xilinx. (2021). *Vivado Design Suite User Guide*.
3. Altera (Intel). (2021). *Quartus Prime Handbook*.
4. IEEE Papers on FPGA-based PWM design and motor control.

10. Code




```
module pwm(
    input clk, reset,              // Clock and Reset inputs
    input [7:0] PWM_ontime,        // 8-bit PWM input
    output reg PWM_out             // 1-bit PWM output
);

    reg [7:0] counter_out;         // 8-bit counter

    always @(posedge clk or posedge reset) begin
        if (reset)
            counter_out <= 8'b0;
        else
            counter_out <= counter_out + 1;
    end

    always @(posedge clk) begin
        if (PWM_ontime > counter_out)
            PWM_out <= 1;
        else
            PWM_out <= 0;
    end

endmodule
```

Text:

```
module pwm(
   input clk, reset,           // Clock and Reset inputs
   input [7:0] PWM_ontime,       // 8-bit PWM input
   output reg PWM_out          // 1-bit PWM output
);

   reg [7:0] counter_out;       // 8-bit counter

   always @(posedge clk or posedge reset) begin
      if (reset)
         counter_out <= 8'b0;
      else
         counter_out <= counter_out + 1;
   end

   always @(posedge clk) begin
      if (PWM_ontime > counter_out)
         PWM_out <= 1;
      else
         PWM_out <= 0;
```

```
    end
endmodule
```