Design and Simulation of Digital PWM Generator Using Verilog HDL

Gurram Sravanthi[1], J Nithish Reddy[2], S Sukanth[3], N Yogendra Sai[4], P Sravanthi[5]

[1]Assisstant Professor Dept. of ECE, CMR Technical Campus, Telangana, India
[2 3 4 5]Student, Dept. of ECE, CMR Technical Campus, Telangana, India
Email: gurramsravanthi26@gmail.com, 247r5a0411@cmrtc.ac.in, 237r1a04b8@cmrtc.ac.in, 237r1a04a9@cmrtc.ac.in, 237r3a04b6@cmrtc.ac.in

---

Abstract - This paper presents the design and simulation of a digital Pulse Width Modulation (PWM) generator using Verilog HDL. PWM is widely used in applications like motor control, LED dimming, power converters and embedded systems, where accurate switching and power control is important. The proposed design supports both edge-aligned and center-aligned PWM modes using counter and comparator logic. A synchronized duty cycle update mechanism is also implemented to reduce unwanted glitches during output switching. The design was developed and simulated using Vivado software, and different duty cycle values were applied to verify the output waveform. The simulation results shows that the generator produces PWM signals with proper pulse width control and stable switching behaviour. The design also provides flexibility in selecting the operating mode, making it useful for different digital and power electronic applications. Overall, the proposed system provides a simple and flexible approach for digital PWM generation, and it can be further improved for practical FPGA based hardware implementation.

Key Words: Pulse Width Modulation, Verilog HDL, FPGA, Vivado, Duty Cycle Control, VLSI Design.

1. INTRODUCTION

Pulse Width Modulation (PWM) is an important technique used in modern electronic and embedded systems for controlling power and switching operations. It is commonly used in applications such as motor speed control, DC-DC converters, LED dimming, inverter control and renewable energy systems. As technology is developing, the demand for compact, high speed and low power digital systems is increasing. Traditional analog PWM circuits have some limitations like less flexibility, sensitivity to noise and difficulties in modifying the design according to different requirements. Because of these limitations, digital PWM generators using Hardware Description Languages (HDL) are becoming more useful in VLSI and digital system design.

This project focuses on the design and simulation of a digital PWM generator using Verilog HDL. The proposed design supports both edge-aligned and center-aligned PWM generation using counter and comparator logic. A synchronized duty cycle updating method is also used to avoid unwanted glitches and maintain stable output waveforms. The design is verified using Vivado software by testing different duty cycle values and operating modes. The simulation results shows that the proposed generator can produce PWM signals with proper pulse width control and reliable switching behaviour. This design provides a flexible solution for digital applications and can be further developed for FPGA implementation and practical hardware use.

2. LITERATURE SURVEY

Before finalizing the system architecture, it is useful to examine prior work addressing similar design challenges. The studies summarized below were reviewed to identify effective design strategies, recurring limitations in existing approaches, and opportunities that could inform the present design.

Shankar A et al. [1] (2021): implemented PWM and PPM generators for microcontroller applications using Verilog, demonstrating how digital modulation techniques can produce analog-equivalent control signals for motor drives, LED dimming, and communication systems. Their counter-and-comparator-based architecture was validated through FPGA implementation and simulation, achieving high operating speed and low power consumption suitable for embedded use. The design, however, was restricted to 8-bit resolution and did not explore further optimization.

Nimma Vandana et al. [2] (2026): developed an FPGA-based Space Vector PWM (SVPWM) system for three-phase inverter control, implemented in Verilog HDL. Their design produced three-phase PWM signals with reduced harmonic distortion and improved DC-link utilization, and FPGA implementation offered faster execution and lower latency than conventional DSP- or microcontroller-based approaches. While simulation and experimental results confirmed stable duty cycles and reliable switching, the study did not extend its analysis to industrial-scale deployment.

Rohan Srivastava et al. [3] (2025): presented an FPGA-based PWM generation technique for power-electronics applications, aiming to overcome the flexibility and complexity limitations of analog PWM systems. Their counter-based architecture generated stable, controllable-duty-cycle PWM signals, with RTL verification confirming accurate waveform generation suitable for inverter applications. The design, however, was limited to fixed-frequency operation and lacked adaptive frequency control or glitch-free duty-cycle updating — features that the present work specifically incorporates.

Anil Kumar C. S. et al. [4] (2024): proposed an FPGA-based digital PWM architecture for DC-DC converters in Software-Defined Radio (SDR) applications, employing Gray counters and one-hot encoders to improve switching efficiency and reduce power loss. Their results showed improved response time, efficiency, and voltage regulation at high operating frequencies, along with the low-latency and flexibility benefits typical of FPGA implementations. However, the added design complexity and SDR-specific focus limited the applicability of their approach to general-purpose systems.

E. Han et al. [5] (2024): combined PWM generation with Direct Digital Synthesis (DDS) for railway stopping-control applications, using FPGA-based digital techniques to produce highly accurate low-frequency signals in place of conventional analog methods. This combination improved precision, stability, and disturbance rejection while enabling high-speed operation and easier reconfiguration. The integration of DDS and PWM, however, introduced additional hardware complexity that required careful resource management.

F. R. Shahroury [6] (2021): presented a fully digital PWM demodulation method for low-power implantable sensor applications, combining digital calibration with ring oscillators, digitizers, and data detectors to improve robustness against process, voltage, and temperature variation. This fully digital approach offered better scalability and reliability than analog alternatives, and simulations confirmed low power consumption with accurate high-frequency signal recovery.

V. V. S. Vijaya Krishna [7] (2019): implemented FPGA-based PWM generation using digital sine generators, frequency triggers, and finite state machines to produce signals with variable duty cycles. The approach offered high speed, flexibility, and programmability compared to traditional hardware methods, and simulations confirmed stable, precise PWM output. Because multiple functional modules were required, however, FPGA resource utilization increased, and the study emphasized functional correctness over power efficiency.

C. Sinduja et al. [8] (2023): integrated an FPGA-based PID controller with a PWM module for intelligent control applications, using a Particle Swarm Optimization (PSO) algorithm to tune PID parameters and improve settling time, rise time, and steady-state accuracy. The PWM module generated precise control pulses derived from the PID output, and both simulation and hardware verification demonstrated stable PWM generation with improved control performance suitable for industrial automation and real-time embedded systems. Collectively, this body of work underscores the value of combining intelligent optimization with digital PWM architectures to maintain adaptability under varying operating conditions: in Sinduja et al.'s implementation [8], PSO-based tuning allowed the controller to adjust its parameters automatically in response to changing load conditions and external disturbances, without manual recalibration.

3. PROPOSED SYSTEM

The proposed system is a digital PWM generator designed using Verilog HDL by following the basic VLSI design methodology. The system consists of five main blocks, which are clock generation, counter module, duty cycle register, comparator block and PWM output stage. All these blocks works together to generate PWM signals with adjustable duty cycle for different digital and power electronic applications. The design supports both edge-aligned and center-aligned PWM modes, which helps to improve the waveform quality and flexibility of the system. A synchronized duty cycle updating method is also used to reduce unwanted glitches and maintain stable switching operation. Overall, the proposed design provides good waveform accuracy, reliable switching and flexibility for future improvements.

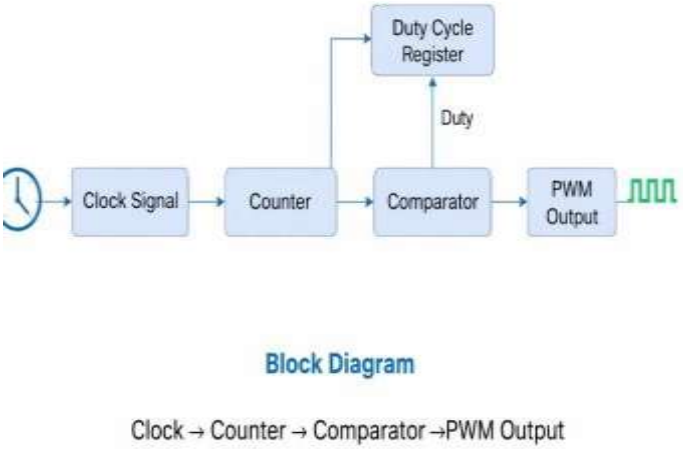


Fig 1: Block Diagram of the Proposed PWM Generator

Fig. 1 shows the overall block diagram of the proposed PWM generator. The main parts of the system are clock source, counter, duty cycle register, comparator and PWM output block. The clock signal acts as the main timing reference for the entire system and continuously drives the counter. The counter generates digital values in sequence, which are used to decide the timing and pulse width of the PWM output.

Clock Signal

The clock signal is used to synchronize all the digital operations in the system. Its frequency directly affects the switching frequency of the PWM output. Therefore, a stable clock signal is necessary to maintain correct timing and generate a reliable output waveform.

Counter Module

The counter is one of the main parts of the PWM generator, which counts in increasing or decreasing order depending on the selected operating mode. In edge-aligned mode, it works as an up-counter, while in center-aligned mode it operates as an up-down counter. The counter values determines the timing period of the PWM signal and helps in generating the required waveform.

Duty Cycle Register

The duty cycle register stores the digital value which controls the ON-time of the PWM output. To avoid unwanted glitches and sudden changes in the output signal, the duty cycle value is updated in a synchronized manner. This makes sure that the new value takes effect at the correct point in the switching cycle, maintaining a stable output.

Comparator Unit

The comparator compares the counter value with the stored duty cycle value continuously. When the counter value is less than the duty cycle value, the PWM output remains HIGH, and when it becomes equal or greater than the duty cycle value, the output changes to LOW. This comparison is repeated during every clock cycle to generate PWM pulses with different widths according to the applied duty cycle.

PWM Output Generation

The PWM output stage generates the final waveform based on the comparator output. By changing the duty cycle value, the width of each pulse can be increased or decreased while keeping the switching frequency constant. This allows the average power supplied to a load to be controlled in an efficient way.

Edge-Aligned and Center-Aligned Operation

The proposed design supports two types of PWM operation, namely edge-aligned and center-aligned modes. In edge-aligned mode, the pulse starts at a fixed edge of the switching period. In center-aligned mode, the pulse is arranged symmetrically around the center of the period. This symmetry helps to reduce harmonic distortion and improves the waveform quality, which is useful for applications where switching noise is a concern.

Output Stage

The generated PWM signal can be used in various applications such as motor control, DC-DC converters, LED dimming, renewable energy systems, embedded controllers and VLSI based platforms. The design was simulated using Vivado software to check its functionality and output waveforms. The simulation results confirms that the system generates PWM signals with proper duty cycle control and stable switching behaviour.

4. ALGORITHMS

The working of the PWM generator is based on a counter and comparator algorithm along with a shadow register for updating the duty cycle. The design supports both edge-aligned and center-aligned PWM operation. It is implemented using Verilog HDL, which helps to generate PWM signals with high speed and adjustable duty cycle. The proposed method provides glitch-free output waveforms and can be used in FPGA and VLSI applications. By updating the duty cycle according to the PWM period, unwanted changes in the output and timing errors can be reduced.

Counter-Based PWM Generation Algorithm

The PWM generation process starts with a digital counter that increases its value at every clock cycle. This counter acts as a timing reference for the complete system. During each clock cycle, the counter value is compared with the stored duty cycle value. If the counter value is less than the duty cycle, the output remains HIGH, and when it reaches or exceeds the duty cycle value, the output becomes LOW. This process repeats continuously and generates a periodic PWM signal with adjustable pulse width.

Duty Cycle Control Algorithm

The duty cycle represents the percentage of time the PWM signal remains ON during one complete period. It determines the pulse width of the output signal and is calculated by dividing the ON-time (TON) by the total period (T), then multiplying by 100.

$$Duty\ Cycle\ (\%) = (TON / T) \times 100$$

When the duty cycle is increased, the average power delivered to the load also increases. Similarly, reducing the duty cycle decreases the average power. This method allows the output power to be controlled without changing the switching frequency.

Comparator Operation Algorithm

The comparator continuously checks the counter value with the stored duty cycle value in every clock cycle. The output is generated based on the following condition:

$$PWM\ Output = 1,\ if\ Counter < Duty\ Cycle$$

$$PWM\ Output = 0,\ if\ Counter \geq Duty\ Cycle$$

This simple comparison process helps the system to generate PWM pulses with the required timing and pulse width. The output changes according to the counter and duty cycle values.

Edge-Aligned PWM Algorithm

In edge-aligned mode, the counter starts from zero and increases up to its maximum value before resetting again. Each PWM pulse begins at the same clock edge, while its width changes according to the selected duty cycle. Due to its simple operation and easy implementation, this method is suitable for applications such as LED brightness control and DC motor speed regulation.

Center-Aligned PWM Algorithm

In center-aligned mode, the counter first counts upward to its maximum value and then starts counting downward until it reaches zero. This up-down counting operation produces PWM pulses that are arranged symmetrically around the center of the switching period. It helps to reduce harmonic distortion and switching noise, making this mode useful for motor drives and power electronic applications where waveform quality is important.

Glitch-Free Duty Update Algorithm

To avoid sudden changes in the PWM output, the new duty cycle values are first stored in a shadow register. The value from this register is transferred to the active duty cycle register only when the counter reaches zero. This ensures that the duty cycle changes at a proper point in the switching cycle instead of changing during pulse generation. By using this method, unwanted glitches can be reduced and the output waveform remains more stable during duty cycle changes.

5. RESULTS & DISCUSSIONS

To check the working of the proposed design, the PWM generator was synthesized and simulated using Vivado software for the Xilinx Zynq-7000 (xc7z020clg484-1) FPGA. The verification was carried out in three stages, which includes RTL schematic analysis, device implementation and behavioural simulation. These stages helps to verify the working of the design, resource utilization and PWM signal generation. The results shows that the proposed generator produces PWM signals with proper duty cycle control in both edge-aligned and center-aligned modes.

RTL Schematic – Top-Level View

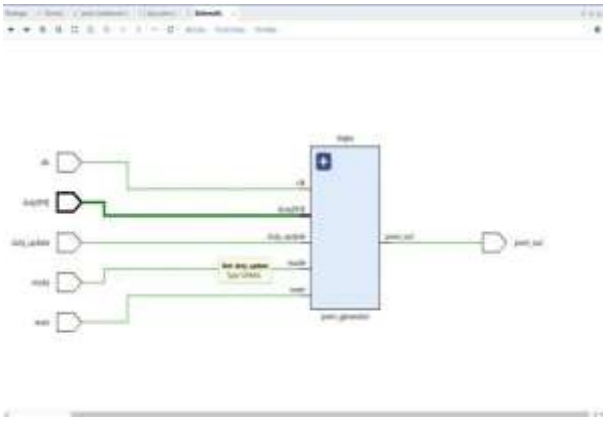


Fig 2: RTL Schematic of the PWM Generator (Top-Level View)

Fig. 2 shows the top-level RTL schematic of the PWM generator after synthesis. The pwm_generator module contains the main input and output signals such as clk, duty[9:0], duty_update, mode, reset and pwm_out. The generated netlist consists of 48 cells, 15 I/O ports and 114 nets. This indicates that the design is properly elaborated and the main interface signals are connected correctly.

RTL Schematic – Internal Logic View

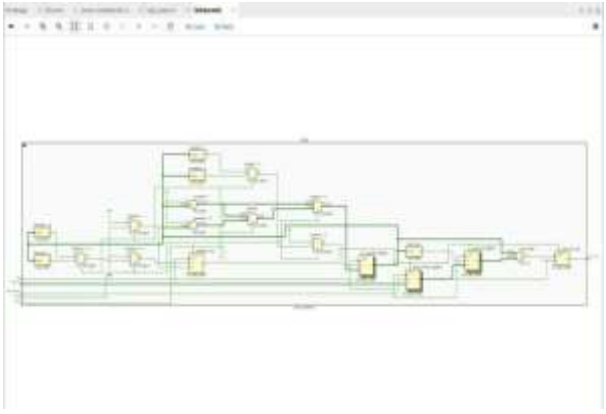


Fig 3: RTL Schematic of the PWM Generator (Internal Logic View)

Fig. 3 shows the internal RTL schematic of the proposed design. It contains the main internal components, including counter logic (counter_i, counter_reg), up/down direction control registers, duty register (duty_reg_i), shadow duty register (duty_shadow_reg) and magnitude comparator (RTL_LT) connected to pwm_out_reg. The schematic consists of 48 cells and 99 nets. It also shows the shadow register used for duty cycle updating and the up-down counter required for both edge-aligned and center-aligned operation. This confirms that the internal structure follows the proposed design.

Elaborated Design (Device Floorplan View)

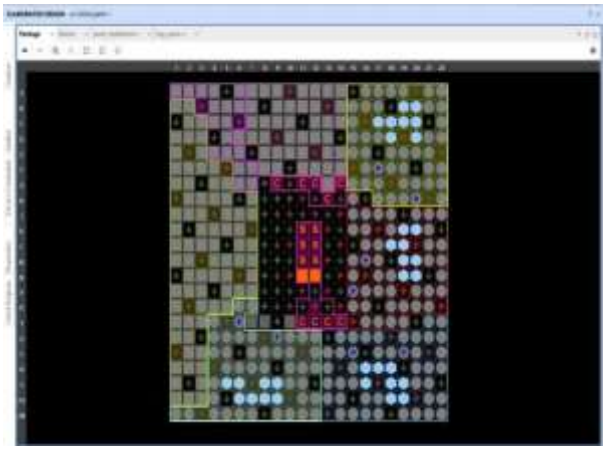


Fig 4: Elaborated Design Showing Device Floorplan on xc7z020clg484-1

Fig. 4 shows the elaborated design mapped onto the device floorplan of the Xilinx Zynq-7000 FPGA (xc7z020clg484-1). It illustrates the distribution of the design logic across different device resources such as configurable logic blocks (CLBs), block RAMs, DSP slices and I/O columns. The highlighted clock regions and resource areas shows the selected placement boundary for the PWM generator. This indicates that the design has been mapped to the available FPGA resources before implementation.

Implemented Design (Device Layout View)

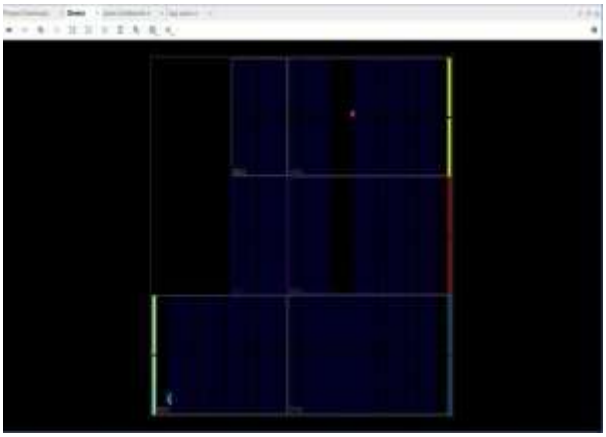


Fig 5: Implemented Design Showing Placement and Routing on the FPGA Device

Fig. 5 shows the implemented design after placement and routing in the FPGA clock regions X0Y0 to X1Y2. The synthesized logic occupies a relatively small area of the device, which indicates good slice utilization and less routing congestion. This result shows that the proposed design has been successfully implemented and is ready for bitstream generation and further hardware testing. It also demonstrates that the design is suitable for FPGA-based implementation.

Simulation Waveform Results

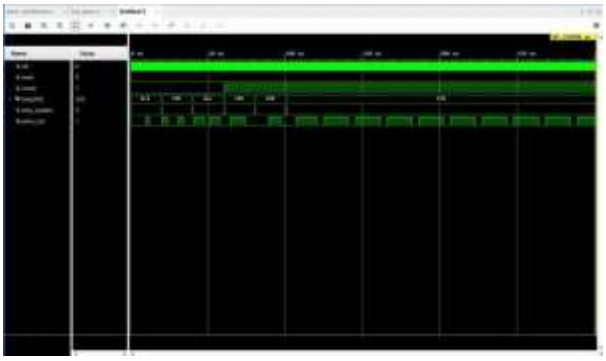


Fig 6: Simulation Waveform Showing PWM Output for Varying Duty-Cycle Values

Fig. 6 shows the simulation waveform of the PWM generator with the clock signal active and reset de-asserted. The operating mode was changed from edge-aligned (0) to center-aligned (1) during simulation. Different values were applied to the duty[9:0] input, including 0x0c8, 0x190, 0x2bc, 0x100, 0x200 and 0x320. The duty_update signal was applied to transfer each new duty cycle value into the shadow register. The pwm_out waveform changes its pulse width according to the applied duty cycle values. The transition between edge-aligned and center-aligned modes also occurs without unwanted glitches, which confirms the correct working of the duty cycle update mechanism.

6. CONCLUSIONS

This work presents the design and simulation of a digital PWM generator using Verilog HDL, which is developed using basic VLSI design techniques. By combining a counter-comparator architecture with synchronized duty cycle updating, the proposed system generates stable and glitch free waveforms with better switching accuracy. It supports both edge-aligned and center-aligned modes, which provides more flexibility in operation and improves waveform symmetry and control precision. The design uses efficient and scalable digital logic, keeping the overall complexity low while maintaining good performance at high speed. The simulation and waveform verification results shows that the design works correctly and reliably. This PWM generator can be used in various applications such as motor control, power converters, embedded systems and digital power management systems.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to G. Sravanthi of the Department of Electronics and Communication Engineering, CMR Technical Campus, Hyderabad, for her invaluable guidance and unwavering support throughout this research. We are also thankful to our peers and instructors for their constructive feedback, which contributed meaningfully to this study. Finally, we acknowledge the institution's infrastructure and resources, which were essential to the design, simulation, and testing of the digital PWM generator presented in this work.

REFERENCES

[1] Shankar A, Lokeshwari A, Manohar B, Pranavi S, "Implementation of Simple PWM/PPM Generator for Microcontroller Using Verilog," International Journal of Advanced Research in Engineering and Technology (IJARET), Vol. 12, Issue 3, pp. 97–102, Mar. 2021.

[2] Nimma Vandana, K. Bhaskar, "Generation of SVPWM on FPGA to Control 3 Phase Inverter," International Journal for Research in Applied Science & Engineering Technology (IJRASET), Vol. 14, Issue 1, Jan. 2026.

[3] Rohan Srivastava, Yogesh K. Chauhan, Bhavnesh Kumar, "Generation of PWM Using Verilog in FPGA," Proceedings of International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pp. 4593–4597, 2016.

[4] Anil Kumar C. S., M. Devanathan, "Optimizing Digital PWM Techniques for DC DC Converters in Software-Defined Radios (SDRs): A Robust Architectural Approach," International Journal of Intelligent Systems and Applications in Engineering (IJISAE), Vol. 12, No. 1, pp. 154–164, 2024.

[5] E. Han et al., "Research on Implementation of a PWM Generation Algorithm for Train Stationary Stopping Frequency," Electronics, 2024.

[6] F. R. Shahroury, "The Design Methodology of Fully Digital Pulse Width Modulation," Journal of Low Power Electronics and Applications, 2021.

[7] V. V. S. Vijaya Krishna, "FPGA Realization of PWM using HDL," International Journal of Computer Sciences and Engineering, 2019.

[8] C. Sinduja, P. Thenmozhi, "Implementation of PID Controller PWM Module on FPGA," International Journal of Trend in Scientific Research and Development, 2023.