PWM Generator with variable duty cycle

Jaladi Deekshitha
Department of Electronics and
Communication Engineering
Amrita School of Engineering,
Coimbatore, Amrita Vishwa
Vidyapeetham,
India
cb.en.u4ece22024@cb.students.amrita.edu

Kethireddy Pujitha Reddy
Department of Electronics and
Communication Engineering
Amrita School of Engineering,
Coimbatore, Amrita Vishwa
Vidyapeetham,
India
cb.en.u4ece22030@cb.students.amrita.edu

Challa Poojitha
Department of Electronics and
Communication Engineering
Amrita School of Engineering,
Coimbatore, Amrita Vishwa
Vidyapeetham,
India
cb.en.u4ece22064@cb.students.amrita.edu

***Abstract*— The developed system is oriented for the generation of a PWM signal of frequency and duty cycle that can be changed by using the system clock. What's more, using two controls namely DUTY_INCREASE and DUTY_DECREASE, this system allows adjusting the duty cycle in increments of 10% from the minimum of 0% to the maximum of 100%. With the usage of asynchronous resets and period counters, it is possible to achieve correct control over every PWM cycle. The design objectives include changing the duty cycle of the PWM output, preventing bouncing when changing parameters, and maintaining FPGA compatibility. This simple and efficient PWM generator can be used in real life scenarios such as in driving motors' speed control and power management in VHDL for a more reliable and robust modulation approach to variations in pulse width.**

Keywords — Counter, Duty cycle control ,Pmod Keypad , Basys3 , Pulse width modulation , LED control

I. Introduction

Pulse Width Modulation, or PWM, is a really handy technique that lets us control analog devices using digital signals. It's commonly used in things like controlling how fast a motor spins, adjusting the brightness of LEDs, or even regulating power in embedded systems. The basic idea is simple: by turning a signal on and off very quickly, we can control how much power gets delivered to a device — and that's all done by adjusting the duty cycle of the signal.

In this project, we built a PWM generator using VHDL, with a special feature — real-time control using a Pmod Keypad. Instead of setting parameters in code or through a computer interface, the user can just press buttons on the keypad to increase or decrease the duty cycle in 10% steps, from 0% up to 100%. This makes the system more interactive and easier to use, especially for learning or prototyping.

The whole thing runs on a Basys 3 FPGA board, using its built-in 100 MHz clock. We've added a few smart design features, like asynchronous resets to avoid glitches, and counter logic that keeps the timing accurate. There's also logic to handle key debouncing, so one press equals one change — no accidental multiple jumps in duty cycle.

In short, this PWM generator is simple, reliable, and very practical. It's a great example of how hardware design and user-friendly controls can come together, and it's especially useful in areas like motor control, power systems, or any embedded project where adjustable output is needed.

Pulse Width Modulation, or PWM, is a really handy technique that lets us control analog devices using digital signals. It's commonly used in things like controlling how fast a motor spins, adjusting the brightness of LEDs, or even regulating power in embedded systems. The basic idea is simple: by turning a signal on and off very quickly, we can control how much power gets delivered to a device — and that's all done by adjusting the duty cycle of the signal.

In this project, we built a PWM generator using VHDL, with a special feature — real-time control using a Pmod Keypad. Instead of setting parameters in code or through a computer interface, the user can just press buttons on the keypad to increase or decrease the duty cycle in 10% steps, from 0% up to 100%. This makes the system more interactive and easier to use, especially for learning or prototyping.

II. Objectives

The primary objectives of this project are as follows:

- To design a Pulse Width Modulation (PWM) generator using VHDL.
- To implement variable duty cycle control for the PWM output, allowing adjustments in 10% increments from 0% to 100%.
- To integrate a debouncing mechanism for reliable and stable duty cycle changes through keypad input.
- To ensure the design is compatible with FPGA-based implementation, specifically on the Basys 3 Artix-7 development board.
- To demonstrate the versatility and practical effectiveness of the PWM generator in real-world applications such as motor speed control and power regulation.

III. Hardware Setup

This project uses two main hardware components: the Basys 3 Artix-7 FPGA development board and the Digilent Pmod KYPD 16-button keypad. Together, they form a simple and effective setup for building and testing the PWM generator with real-time user input.

A. Basys 3 Artix-7 FPGA Board The

Basys 3 is a beginner-friendly FPGA development board from Digilent, featuring the Xilinx Artix-7 (XC7A35T) chip. It includes a 100 MHz onboard clock, numerous input/output interfaces, and supports programming through the Xilinx Vivado Design Suite.

Some of the onboard features include:

1. 16 user switches
2. 5 push buttons
3. 16 LEDs
4. A 4-digit seven-segment display

Several

Pmod connectors for adding peripherals

In

this project, the Basys 3 board is used to implement the PWM logic, scan keypad inputs, apply debouncing, and manage duty cycle changes. Its flexible I/O and reliable clock make it ideal for generating and adjusting PWM signals in real-time.

Fig1. Basys3 Board

B. Digilent Pmod KYPD (Keypad Module)

The Pmod KYPD is a compact 4x4 matrix keypad, providing 16 push buttons arranged in a grid. It connects easily to the Basys 3 through a standard Pmod port.

Each key corresponds to a unique value.

The keypad uses a scanning method to detect button presses. In this setup, specific buttons are used to increase or decrease the PWM duty cycle in 10% steps.

To avoid errors from rapid or noisy button presses (common with mechanical switches), debouncing logic is added in VHDL to ensure smooth operation.

Fig 2. Keypad

C. Putting It All Together

The Pmod keypad is plugged into the Basys 3 board using one of its Pmod ports. When a user presses a button, the system detects the input, debounces the signal, and adjusts the PWM duty cycle accordingly. The generated PWM output can be monitored through the onboard LEDs or routed to other circuits for testing with motors, lights, or other components.

This simple and interactive hardware setup allows users to manually control the PWM signal in real time, making it great for both learning and practical embedded applications.

IV. Results and Discussions

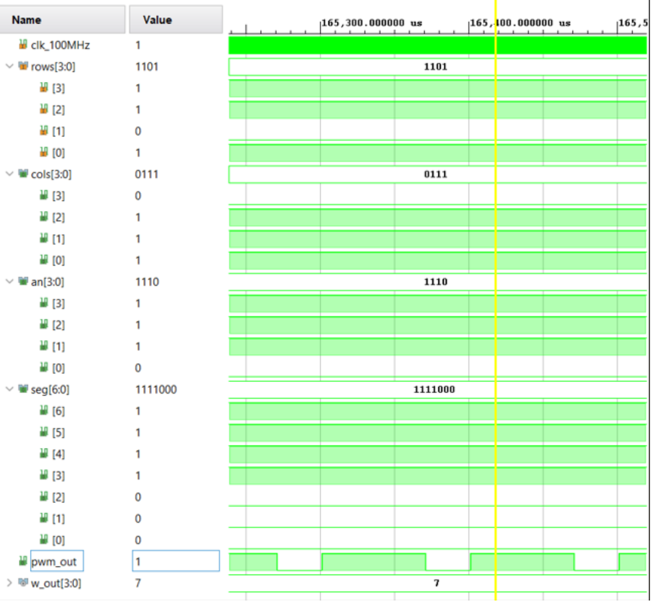
The simulation of the PWM generator was carried out in Xilinx Vivado, and the waveform results confirm its correct functionality. As shown in Fig. 1, key signals such as the clock (clk_100MHz), keypad inputs (rows[3:0], cols[3:0]), seven-segment display (seg[6:0]), and PWM output (pwm_out) were analyzed.

When a button is pressed on the Pmod keypad, the corresponding duty cycle value is updated and displayed on the seven-segment display, while the PWM output (pwm_out) adjusts accordingly. The waveform clearly shows stable pulse width modulation, with the output toggling as expected.

Key observations:

- Keypad inputs successfully update the PWM duty cycle.
- Seven-segment display dynamically reflects changes.
- PWM signal remains stable and accurate, confirming correct operation.

This validates that the VHDL-based PWM generator is working as intended, providing real-time user control over duty cycle adjustments, making it reliable for FPGA-based applications like motor speed control and power regulation



Name	Value
clk_100MHz	1
rows[3:0]	1101
[3]	1
[2]	1
[1]	0
[0]	1
cols[3:0]	0111
[3]	0
[2]	1
[1]	1
[0]	1
an[3:0]	1110
[3]	1
[2]	1
[1]	1
[0]	0
seg[6:0]	1111000
[6]	1
[5]	1
[4]	1
[3]	1
[2]	0
[1]	0
[0]	0
pwm_out	1
w_out[3:0]	7

Fig 3 . Simulation Result

The schematic diagram of the PWM generator system shows how different components are connected within the Basys 3 Artix-7 FPGA. The system is designed to generate a PWM signal with a variable duty cycle, which can be adjusted using a Pmod keypad. The selected duty cycle is also displayed on a seven-segment display for easy monitoring.

The system starts with the 100 MHz clock, which acts as the main timing source. A clock buffer ensures the clock signal remains stable. The Pmod keypad serves as the input device, allowing users to increase or decrease the PWM duty cycle in steps. These keypad inputs pass through input buffers before reaching the PWM generator module.

The PWM generator module processes the keypad input and generates a PWM signal (pwm_out) with the corresponding duty cycle. This output is then sent through an output buffer to ensure smooth operation. At the same time, the seven-segment display controller updates the display to show the selected duty cycle percentage. The display signals are also passed through output buffers for proper communication with the FPGA.

In summary, this schematic confirms that the PWM generator is working correctly, allowing users to easily adjust the duty cycle using a keypad while monitoring the changes on a display. This setup makes the system useful for applications like motor speed control, LED dimming, and power regulation.

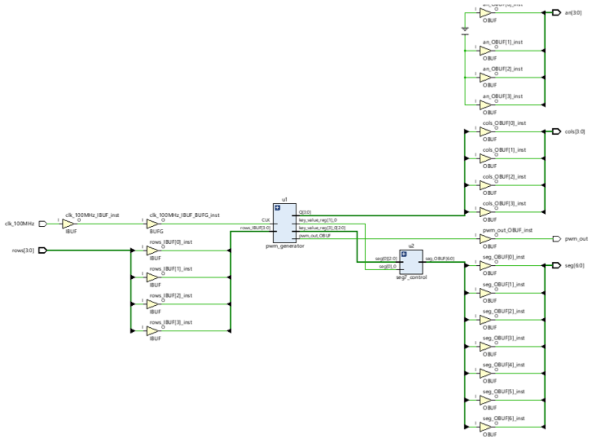
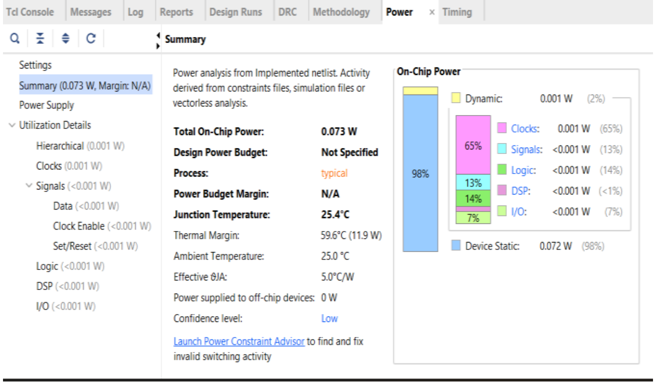


Fig4. Schematic

The power analysis for the PWM generator system implemented on the Basys 3 Artix-7 FPGA reveals a total on-chip power consumption of 0.073 W. This power consumption is primarily composed of static power (0.072 W), which accounts for 98% of the total power, while dynamic power contributes only 0.001 W (2%).

The clock network is the most significant contributor to dynamic power, consuming 65% of the total dynamic power. Other elements such as signals (13%), logic (14%), DSP (<1%), and I/O (7%) also contribute minimally to power usage.

The system operates at a junction temperature of 25.4°C, with an ambient temperature of 25.0°C. The thermal margin is 59.6°C, indicating that the system runs efficiently within thermal limits. The effective thermal resistance (θJA) is 5.0°C/W, ensuring proper heat dissipation.



Settings
Summary (0.073 W, Margin: N/A)
Power Supply
Utilization Details
Hierarchical (0.001 W)
Clocks (0.001 W)
Signals (<0.001 W)
Data (<0.001 W)
Clock Enable (<0.001 W)
Set/Reset (<0.001 W)
Logic (<0.001 W)
DSP (<0.001 W)
I/O (<0.001 W)

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**0.073 W**
Design Power Budget:	**Not Specified**
Process:	typical
Power Budget Margin:	**N/A**
Junction Temperature:	**25.4°C**
Thermal Margin:	59.6°C (11.9 W)
Ambient Temperature:	25.0 °C
Effective θJA:	5.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

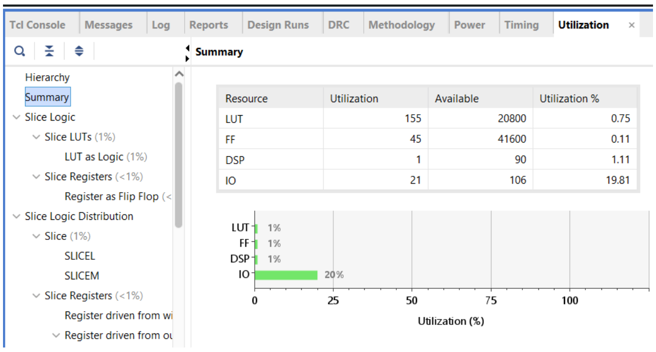
Dynamic:	0.001 W	(2%)
Clocks:	0.001 W	(65%)
Signals:	<0.001 W	(13%)
Logic:	<0.001 W	(14%)
DSP:	<0.001 W	(<1%)
I/O:	<0.001 W	(7%)
Device Static:	0.072 W	(98%)

Fig 5. Power Report

Overall, the PWM generator design is power-efficient, making it suitable for FPGA-based applications where low power consumption is crucial. The analysis confirms that the system operates well within the FPGA's power and thermal constraints.

The FPGA resource utilization for the PWM generator implementation on the Basys 3 Artix-7 FPGA is minimal, ensuring efficient hardware usage.

- LUTs: 155 used out of 20,800 (0.75%)
- Flip-Flops (FFs): 45 used out of 41,600 (0.11%)
- DSP Blocks: 1 used out of 90 (1.11%)
- I/O Pins: 21 used out of 106 (19.81%)



Hierarchy
Summary
Slice Logic
Slice LUTs (1%)
LUT as Logic (1%)
Slice Registers (<1%)
Register as Flip Flop (<
Slice Logic Distribution
Slice (1%)
SLICEL
SLICEM
Slice Registers (<1%)
Register driven from wi
Register driven from ou

Resource	Utilization	Available	Utilization %
LUT	155	20800	0.75
FF	45	41600	0.11
DSP	1	90	1.11
IO	21	106	19.81

Fig6.Utilization Report

The design uses a small fraction of available resources, leaving ample space for further enhancements or additional features. The I/O utilization is relatively higher, reflecting external interfacing needs. This lightweight implementation makes the design scalable and well-suited for FPGA deployment

V. CONCLUSION

This project demonstrates the successful implementation of a Pulse Width Modulation (PWM) generator with real-time duty cycle control using a Pmod Keypad on the Basys 3 FPGA board. Written in VHDL, the design enables the user to adjust the duty cycle in 10% steps—from 0% to 100%—through simple keypad inputs, making the system interactive and easy to operate.
The system is both functionally effective and hardware-efficient:
It uses only a small fraction of available logic (LUTs) and sequential elements (Flip-Flops),
Requires minimal DSP resources,

And makes purposeful use of I/O pins to connect with external components like the keypad and PWM output.
The inclusion of key debouncing ensures reliable user input, preventing multiple increments or decrements from a single keypress. The design is lightweight, scalable, and well-suited for applications such as LED dimming, fan speed control, or other real-time embedded control systems.

In summary, this PWM generator achieves its intended goals with minimal hardware overhead and provides a practical solution for digital-to-analog-style control using simple digital inputs.

solution for digital-to-analog-style control using simple digital inputs.

REFERENCES

[1] S. K. Shukla, R. N. Patel, and P. Agrawal, "Generation of Variable Duty Cycle PWM using FPGA," *IOSR Journal of VLSI and Signal Processing*, vol. 4, no. 6, pp. 1–3, 2014. [Online]. Available: https://www.iosrjournals.org/iosr-jvlsi/papers/vol4-issue6/Version-2/A04620103.pdf.

[2] P. Agarwal and A. Kumar, "Generation of PWM using Verilog in FPGA," *International Journal of Engineering Research & Technology (IJERT)*, vol. 5, no. 12, pp. 1–5, Dec. 2016. [Online]. Available: https://www.researchgate.net/publication/311253291_Generation_of_PWM_using_verilog_In_FPGA

[3] FPGA4Student, "PWM Generator in VHDL with Variable Duty Cycle," 2017. [Online]. Available: https://www.fpga4student.com/2017/06/pwm-generator-in-vhdl.html.

[4] H. S. Shah, J. H. Bhatt, and R. M. Patel, "Generating PWM Signals with Variable Duty from 0% to 100% Based FPGA SPARTAN3AN," in *Proceedings of the International Conference on Innovations in Electronics and Communication Engineering (ICIECE)*, Hyderabad, India, Jul. 2015. [Online]. Available: https://www.researchgate.net/publication/280563174_Generating_PWM_Signals_With_Variable_Duty_From_0_to_100_Based_FPGA_SPARTAN3AN.

[5] S. Kumar and S. Verma, "FPGA Realization of PWM using HDL," *International Journal of Recent Trends in Engineering & Research (IJRTER)*, vol. 5, no. 4, pp. 24–28, Apr. 2019. [Online]. Available: https://www.researchgate.net/publication/332115816_FPGA_Realization_of_PWM_using_HDL.