Technical Sciences
Academy of Romania
www.jesi.astr.ro

Journal of Engineering Sciences and Innovation
Volume 10, Issue 2 / 2025, p. 233-244

F. Electrical, Electronics Engineering, Computer Sciences and Engineering

Received *9 December 2024* ***Accepted*** *10 June 2025*
Received in revised form *23 Apryl 2025*

Versatile PWM module used for cross-teaching

AUREL GONTEAN, ALBERT BARBU*, SAMUEL MATEA, ELISEI ILIES, MAGDALENA MARINCA

Faculty of Electronics, Telecommunications and Information Technology, Politehnica University Timişoara, România

Abstract. Pulse Width Modulation (PWM) technique has the advantage of providing precise control combined with high efficiency and flexibility, which makes it widely used in modern electronic systems. Therefore, our paper's main goal is to present the results obtained by cross linking a similar design using different implementation techniques for students' better understanding. In this regard, the PWM technique is presented through three different courses, accompanied with custom physical implementations, allowing students to simulate and test a PWM modulator composed of digital integrated circuits, generate such a signal using a microcontroller, and implement a PWM modulator in an FPGA using the Verilog/SystemVerilog hardware description language.

Keywords: PWM, digital design, education, FPGA, embedded systems.

1. Introduction

Pulse Width Modulation (PWM) is a widely known technique, being the foundation of control in many areas, such as: automation, power electronics, lighting, data transmission and so on. A PWM module can be built using analog parts (a sawtooth waveform generator followed by an analog comparator) or using pure digital circuits as counters and digital comparators [1].

The PWM modulator is encountered in the first years of bachelor at the Electronics, Telecommunications and Information Technologies in our university at the following courses:

A. Digital Integrated Circuits, as an application for sequential circuits
B. Microcontrollers, when studying the CCP modules for PIC microcontrollers [1]
C. FPGA systems, where such a modulator is described in HDL, tested and implemented.

*Correspondence address: albert.barbu@student.upt.ro

The aim of this paper is to detail our teaching experience in this field and to emphasize how these approaches can be combined for a better understanding and learning curve for the student. Having the same circuit built in different technologies and used in several ways proves more than useful for the student if also the design and simulation is followed by real experiments; and provides an anchor point for the student, easing the learning process for newer systems by implementing an already well-known technology through them.

2. The PWM Working Principle

A digital PWM (Fig. 1) uses an oscillator, a counter and two digital comparators. The oscillator provides a higher frequency clock, fixed or with the use of a prescaler and a multiplexer having some higher period alternatives. The counter starts counting at the beginning of the cycle (PWM period), when the PWM output becomes High (Fig. 2). The first comparator detects the moment when the PWM output is Low (after the Duty Cycle or On time), while the second timer detects the end of the period, when both the PWM signal and the counter are cleared.

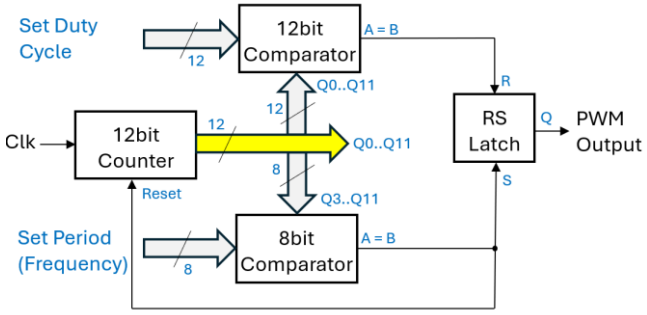

Fig. 1. The block diagram of the PWM modulator

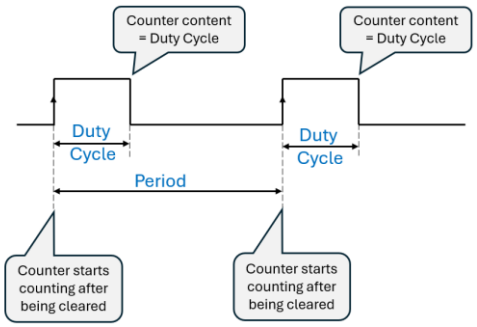

Fig. 2. The working principle of the PWM modulator

3. Simulation

The simulation schematic (Fig. 3) closely follows the block diagram from Fig. 1: U1 – U3 composes a 12bit synchronous counter, U7 is the 8bit period comparator and U4 – U6 implements the 12bit counter for the On (Duty Cycle) time. The U7

"$\overline{P = Q}$" output clears the 12bit counter and the U9:A Q output (the PWM signal), while the U6 $QA = B$ output sets the U9 flip-flop (used as a latch), thus making the PWM output signal high.
The PWM frequency is set via K7..K0:

$$T_{PWM} = T_{Clk} \cdot [16(K7{:}0) + 1] \quad (1)$$

For a 200kHz Clock signal and $K7{:}0 = 0x05$, $f_{PWM} = 2.469\text{kHz}$, as the simulation also shows (the 200kHz frequency is linked to simulator's limited capabilities – real time simulation is not possible above, using the current software and PCs).
The Duty Cycle is set via A11:A0:

$$DC = T_{Clk} \cdot [(A11{:}0) + 1] \quad (2)$$

For a 200kHz Clock signal and $A11{:}0 = 0x3E$, $DC = 315\mu s$.

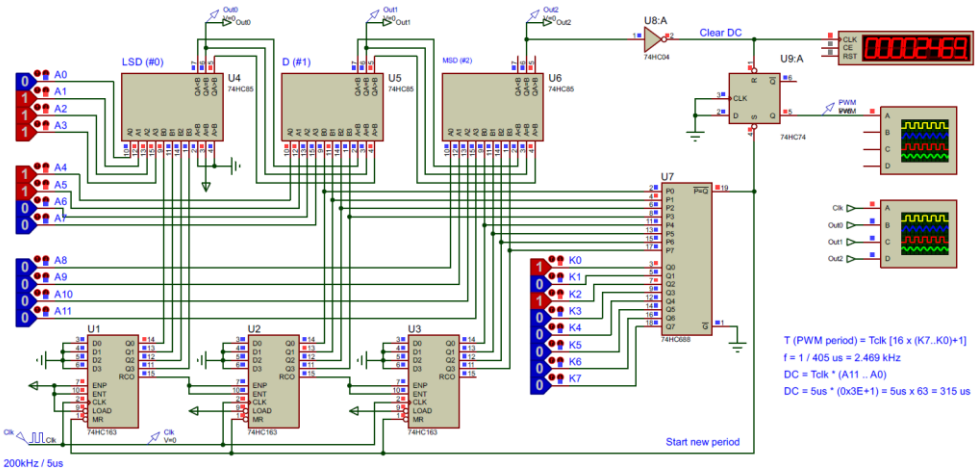

Fig. 3 Proteus simulation schematic

The 12bit digital comparator outputs are introduced in Fig.4; as the 4 upper bits of A11:0 is 0, $OUT1 \equiv OUT2$.

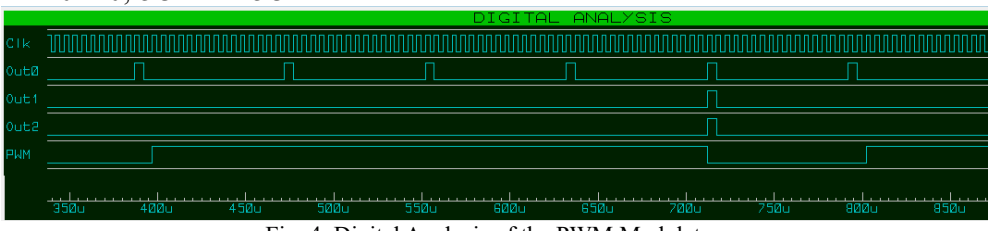

Fig. 4. Digital Analysis of the PWM Modulator

The waveforms from Fig 5 show in real time the same behavior of the 12bit comparator outputs and the actual PWM signal.

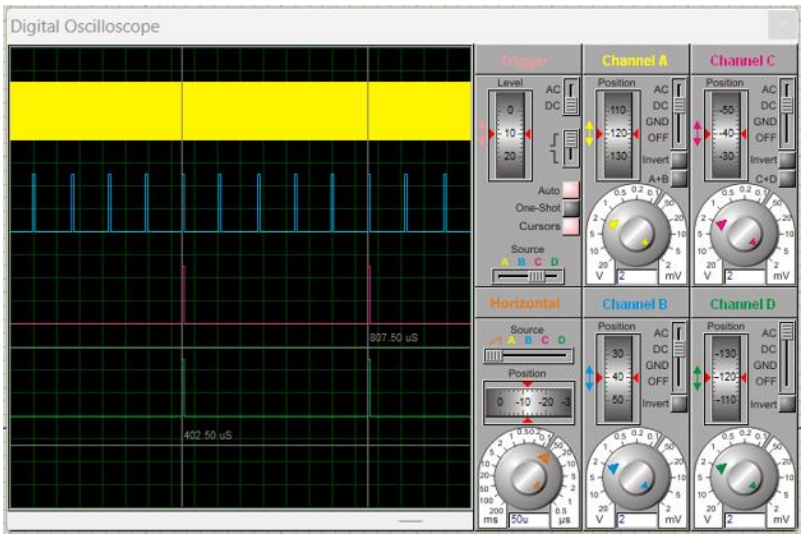

Fig. 5.a. 12bit comparator outputs

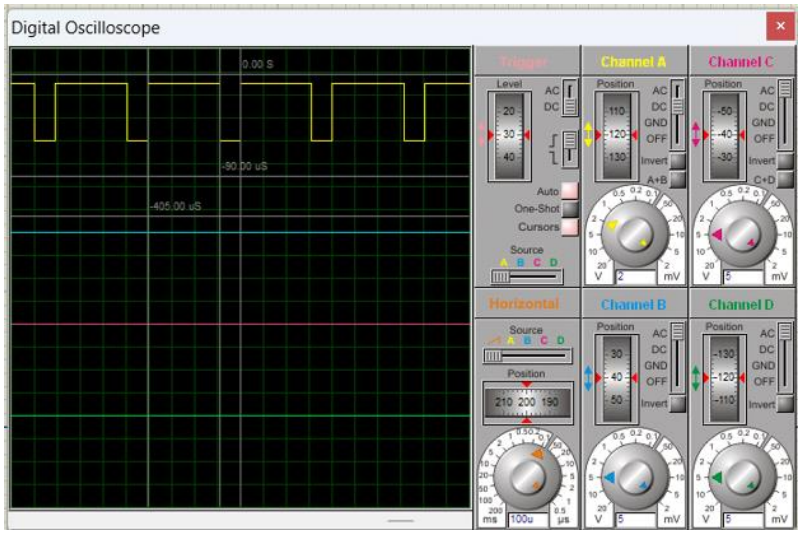

Fig. 5.b. PWM Output signal

The student must face some challenges here:

a. Understanding that to obtain a high frequency PWM signal, the system clock must be considerably higher.
b. Not all Period – duty cycle combinations are valid
c. There is a limit in PWM resolution

All these shortcomings are addressed in A – C laboratories:

For the A course (Digital Integrated Circuits), as hardware obvious limitations (the maximum clock frequency is below 20 MHz), the PWM maximum frequency for a 10bit PWM resolution is:

$$f_{PWM,Max} = \frac{f_{Clk}}{10^{PWM\_res \cdot log2}} = 19.53\text{kHz} \quad (3)$$

It is worth mentioning that there is no versatility in this case, any hardware changes leading to the board redesign and implementation. For the B Course, the limitations come from the microcontroller's manufacturer performance, using the same (3) equation, but the clock frequency can be much higher (150 MHz for Teensy 4.1, where the CPU clock is 600 MHz). For FPGAs course (C), there is no internal CPU to peripheral clock division, so the full oscillator frequency can be used; moreover, the modulator design can accommodate different updates, and the student can add any feature he wants, the tradeoff being the higher price when compared with the microcontroller counterpart (the total cost being however lower versus the first hardware approach (Fig. 3).

4. Real world schematic

The actual schematic is based on fig. 3, while adding 2 oscillators, preset jumpers, an LDO voltage regulator, and most important, the possibility to be also configured via a microcontroller board, such as PIC Curiosity or even an Arduino.

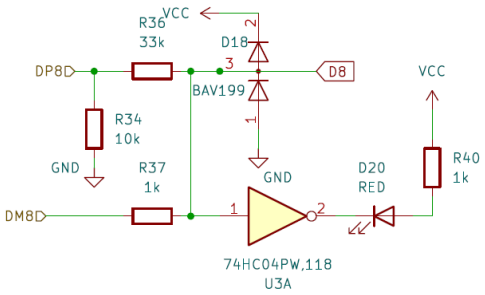

Fig. 6. The input circuit

Each of the 12 and 8 comparator inputs uses the schematic depicted in Fig. 6, where DP8 connects to a jumper wired to Vcc and DM8 is controlled by a microcontroller board (Arduino, PIC Curiosity etc.). D8 is fed to the comparator's A8 equivalent input. The resistor network (R36, R34 and R37 together with the clamping diodes D18 ensure proper operation for both logic levels and overvoltage protection against wrong manipulation by the user when connecting the controller board (the PWM modulator is 3V3 supplied and some microcontroller boards still operate at 5V).

5. PCB Design, PCB Assembly

The PCB design (Fig. 7) was performed in KiCad [2] and uses a 4 layer approach, taking advantage of the small size of the SMD components used and the bottom

placement of the interface circuits. During the design phase, the 3D rendering of the PCB model proved very useful (Fig. 8).

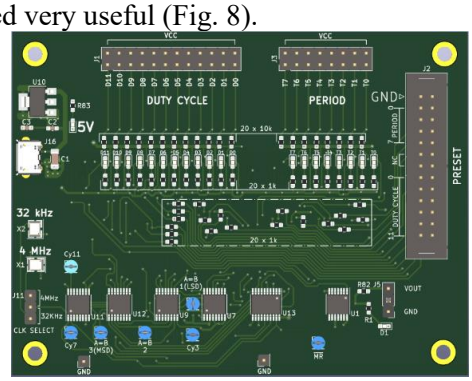

Fig. 7.a. PCB Layout - Top

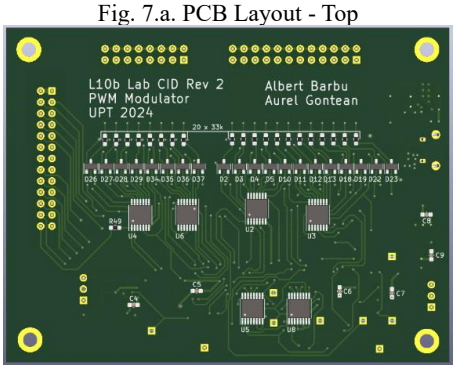

Fig. 7.a. PCB Layout – Bottom

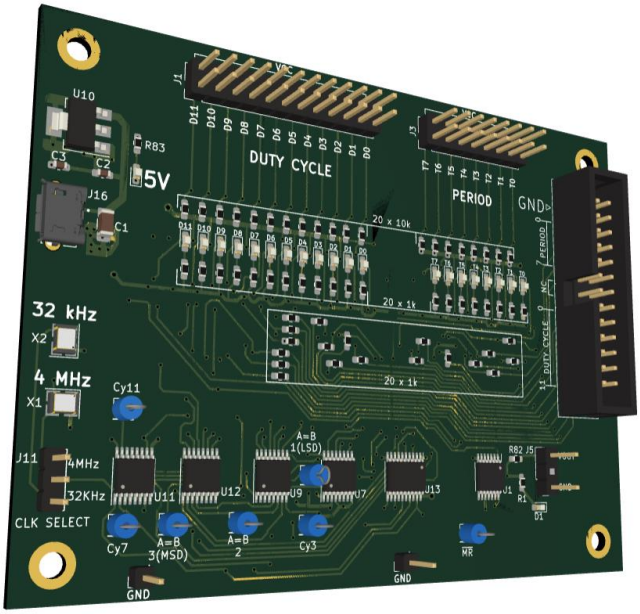

Fig. 8. 3D PCB Rendering

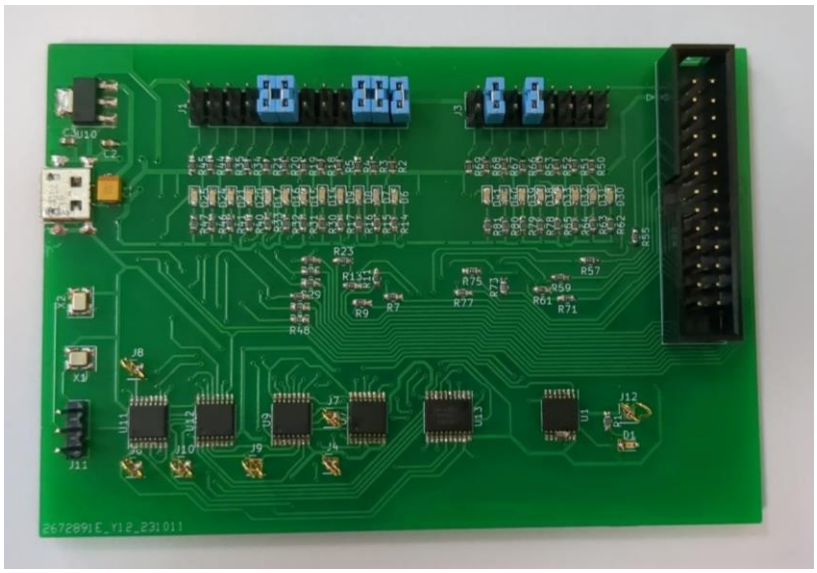

Fig. 9. Prototype PCB

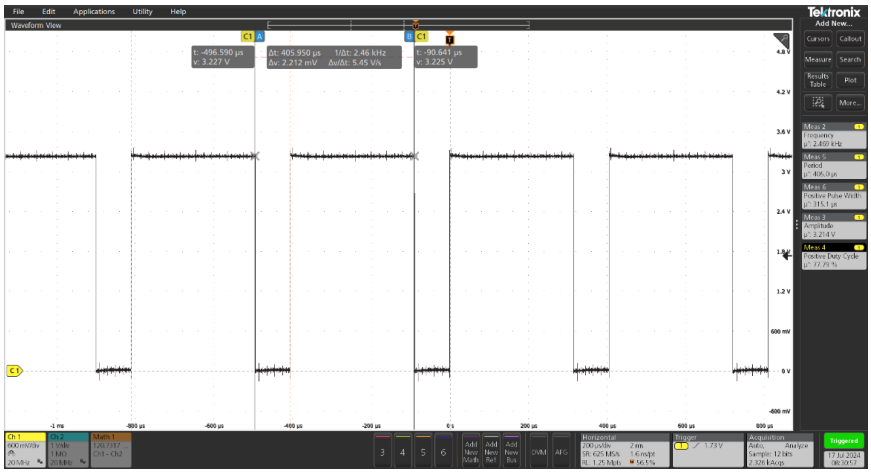

Fig. 10. Oscilloscope readout of prototype PCB

6. EDAPlayground Verilog Simulation

Due to space constraints, the code is not listed here; however, it can be inspected and run using the EDAplayground website: https://edaplayground.com/x/Kvcq

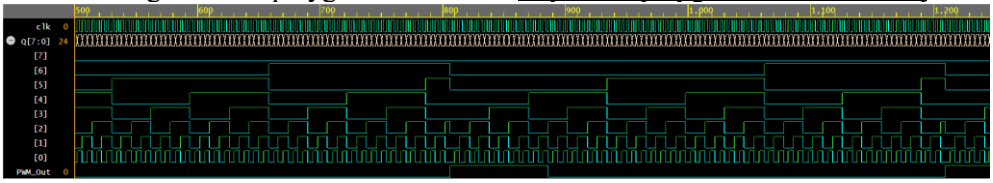

Fig. 11. Verilog Testbench Functional Simulation results (DC = 20%, period = 100 Clock cycles)

7. FPGA Design

The PWM generator was developed in SystemVerilog using Vivado 2019.2 and Nexys4DDR (Nexys A7) FGPA board. The output of the PWM module will be given by the result of a comparator which evaluates the counter, incremented on every clock cycle, and the duty cycle set by the user. The period and the duty cycle of the PWM can be set using the 8 switches available on the board, maximum frequency being 100MHz and minimum is 392.156kHz=400kHz(100MHz/255). The period and duty cycle are displayed on the eight 7-seg displays available on the board as can be seen in Fig. 15.

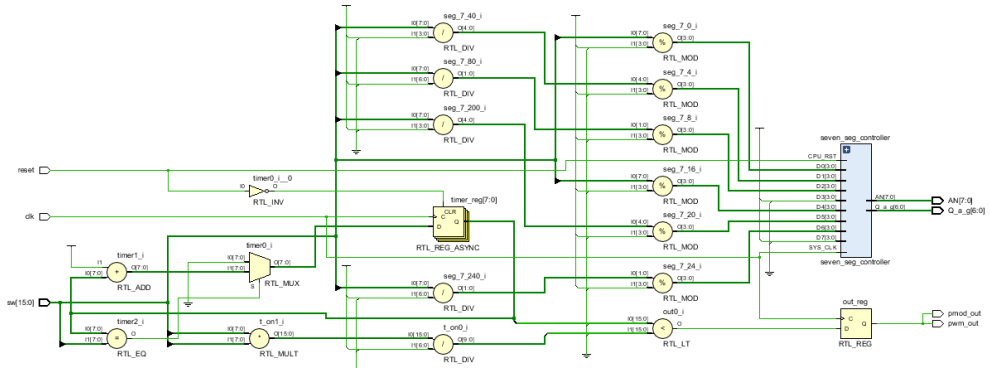

Fig.12. Vivado. PWM Modulator Implementation

Tcl Console | Messages | Log | Reports | Design Runs | Package Pins | I/O Ports

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed
synth_1	constrs_1	synth_design Complete!								236	29	0.0	0	0	7/15/24, 2:56 PM	00:00:42
impl_1	constrs_1	route_design Complete!	6.957	0.000	0.206	0.000	0.000	0.115	0	230	30	0.0	0	0	7/15/24, 2:56 PM	00:01:01

Fig.13. Report utilization

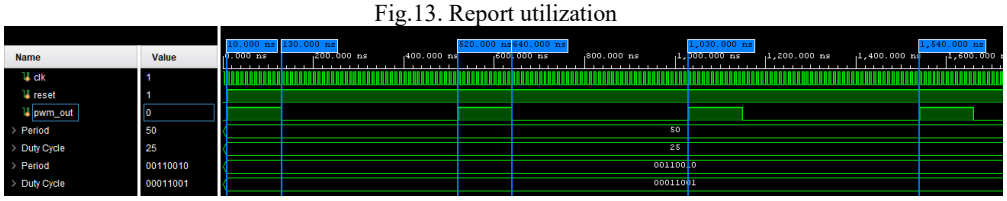

Fig.14.a. Simulation (DC = 25%, T_{PWM} = 50 Clock cycles)

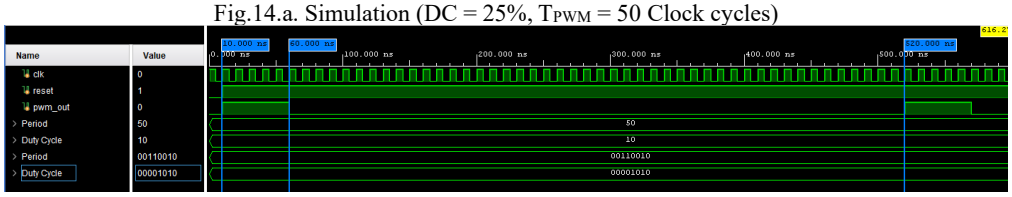

Fig.14.b. Simulation (DC = 10%, T_{PWM} = 50 Clock cycles)

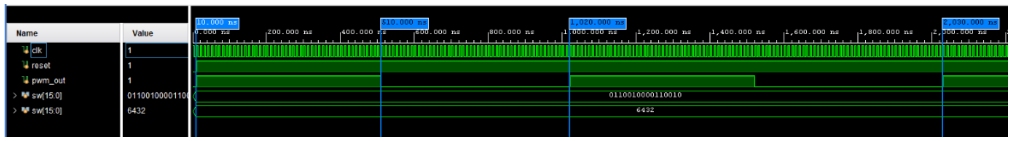

Fig.14.c. Simulation (DC = 50%, T_{PWM} = 100 Clock cycles)

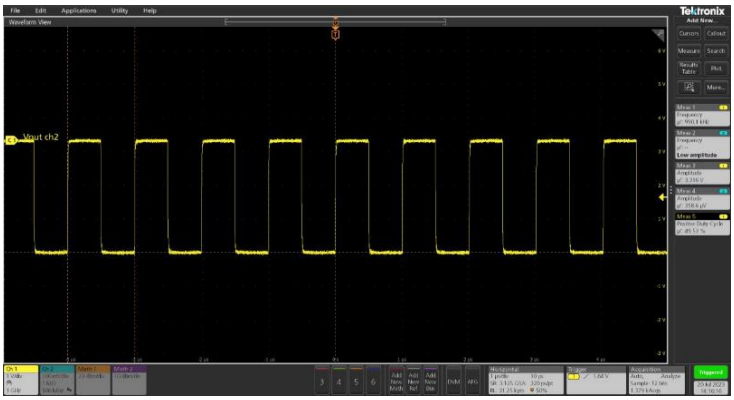

Fig.15.a. 50%DC, T_{PWM} = 100 Clock cycles

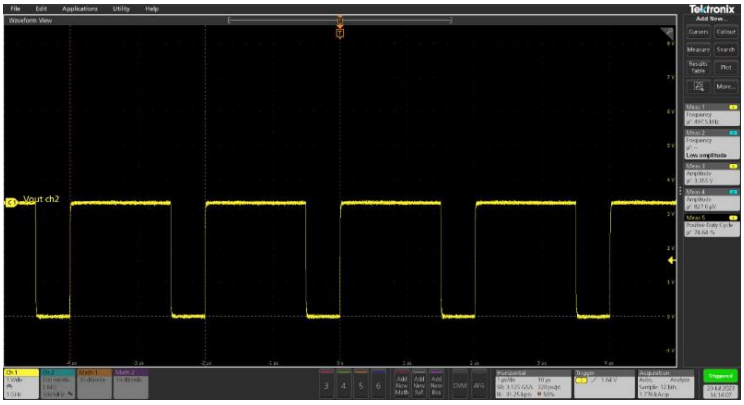

Fig.15.b. 75%DC, T_{PWM} = 200 Clock cycles

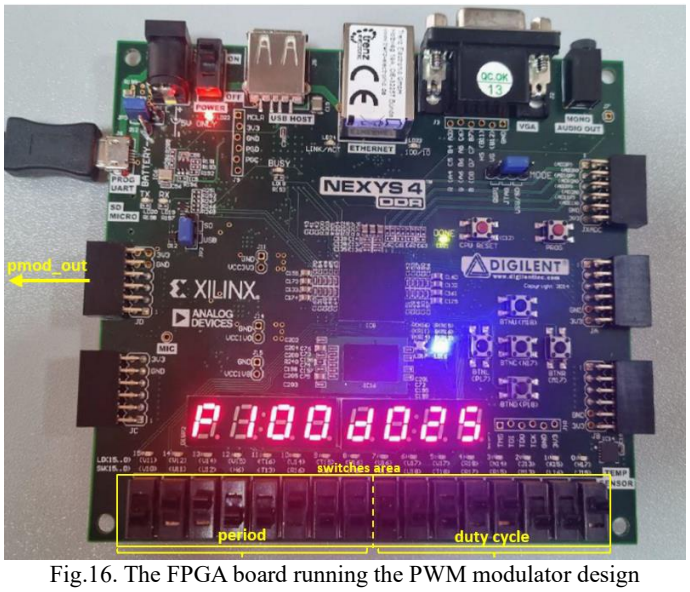

Fig.16. The FPGA board running the PWM modulator design

8. ASIC Design

Tiny Tapeout is an open-source educational initiative aimed at making chip design more accessible and cost-effective. It enables users, even those with minimal digital design experience, to swiftly and easily create custom chips. The project utilizes the Skywater 130 nm process design kit (PDK) and QFN packaging. Development is carried out using Openlane2, which can be accessed directly on GitHub or on personal Linux systems. [14]

Parameter	Value
Maximum output frequency	33 MHz
Maximum input frequency	66 MHz
Drive strength (source/sink)	4 mA
IO supply voltage *	1.71V - 5.5V
1 Tile area	160x100um
Costs for: 1 tile + ASIC + demo board	$300 + shipping
* The demo board provides 3.3V IO supply voltage. The inputs pins are not 5V tolerant.	

Fig. 17. Specifications for Tiny Tapeout ASIC[15]

The design description: it contains two 12-bit registers: one for the duty cycle (duty_reg) and one for the period (period_reg); when the sel signal is set to "0," the duty_reg is selected, and when sel is "1," the period_reg is selected. If values for the duty cycle or period are provided at the input, they are written to the registers only when wr_en is set to "1." For the duty cycle, only 7 bits (from 0 to 6) are used, with the remaining bits hardcoded to 0. The value for period_reg can range from 2 to 4095 (using 12 bits). The pwm_out signal will be available only when out_en is set to "1." The source code was developed in Verilog, input clock is 50MHz, output maximum frequency is 25 MHz (25%) and minim frequency is 12.21 kHz. Link to my project: https://github.com/MateaSamuel/tt08-pwm-generator

The design is tested with cocotb (a python library) which creates the stimuli from a python script and verifies that gate level simulation works properly. The .vcd file obtained after Gate Level testing, can be visualized with GTKwave opensource tool. In Fig. can be seen capture from GTKwave from Gate Level test.

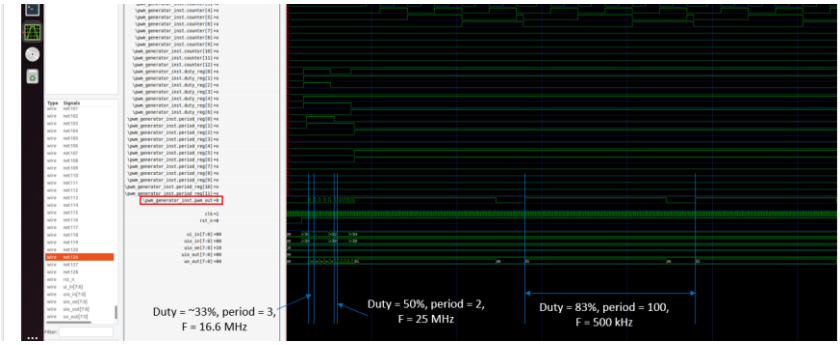

Fig. 18. Gate Level testing (GTKwave window)

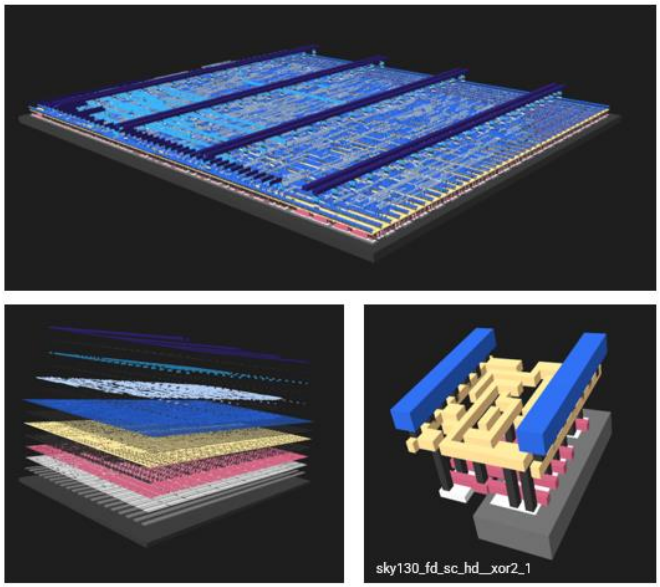

Fig. 19 Layout from 3D viewer

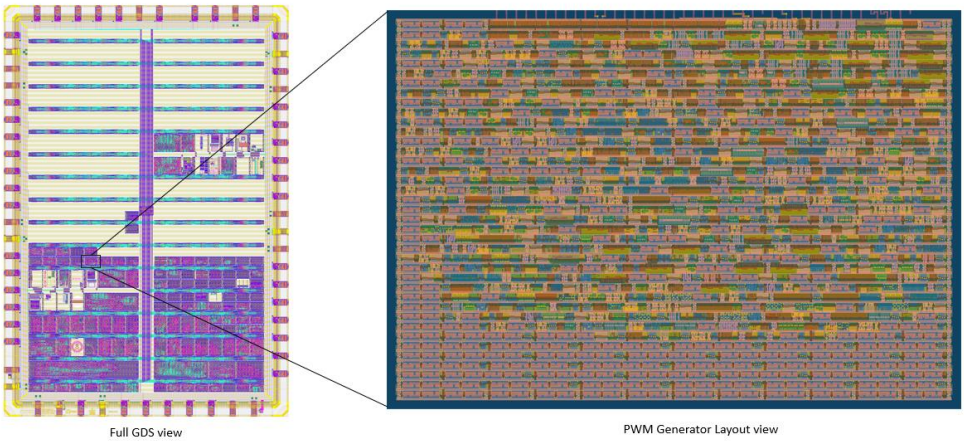

Fig. 20.

Conclusions

All the hardware and FPGA implementations and their related simulations presented in this paper are original. PCB soldering has been performed with students and proved to be rewarding when the results were the same as expected.
Addressing the same topic from different perspectives proved a valuable tool for consolidating students' knowledge. Besides the obvious preference of most students for software, this integrated and combined approach allowed us to make them visualize the key points of each technique. Furthermore, even the pure

hardware implementation was interesting when combined with an Arduino interface to program the PWM parameters.
Another positive outcome is the perfect match between simulations and hardware implementations. The price comparison was also emphasized and offered a broader perspective and offers clear advantages for future decisions.

Acknowledgement
This work has been partially supported by Continental Automotive Romania via the IPCEI partnership with Politehnica University Timisoara.

References

[1] Buccella C., Cecati C., Latafat H., *Digital Control of Power Converters—A Survey*, IEEE Transactions on Industrial Informatics, **8**, 3, 2012, p. 437-447.
[2] Hua G., Lee F. C., *Soft-switching techniques in PWM converters*, IEEE Transactions on Industrial Electronics, **42**, 6, 1995, p. 595-603.
[3] He S., Zhou D., Wang X., Zhao Z., Blaabjerg F., *A Review of Multisampling Techniques in Power Electronics Applications*, IEEE Transactions on Power Electronics, **37**, 9, 2022, p. 10514-10533.
[4] Lee M.-C., Chen Y. -C., *Implementation of a ramp generator with Schmitt trigger circuit for PWM modulator applications*, 2017 Progress in Electromagnetics Research Symposium - Fall (PIERS - FALL), Singapore, 2017, p. 2176-2182.
[5] Nandhakumar R., Jeevananthan S., Dananjayan P., *Design and implementation of an FPGA-Based high performance ASIC for open loop PWM inverter*, 2006 India International Conference on Power Electronics, Chennai, India, 2006, p. 349-354.
[6] Sathyan A., Milivojevic N., Lee Y. -J., Krishnamurthy M., Emadi A*., An FPGA-Based Novel Digital PWM Control Scheme for BLDC Motor Drives*, IEEE Transactions on Industrial Electronics, **56**, 8, 2009, p. 3040-3049.
[7] Microchip, PIC18F2420/2520/4420/4520 Data Sheet, 2008, last accessed on July, 15th, 2024.
[8] Teensy® USB Development Board, https://www.pjrc.com/teensy/, last accessed on 25th July, 2024.
[9] KiCad EDA, https://www.kicad.org/, last accessed on 25th July, 2024.
[10] EDAplayGround, https://edaplayground.com/, last accessed on 25th July, 2024.
[11] Shieh C. -S., *Quick implementation of Pulse Wise Modulation(PWM), Pulse Frequency Modulation(PFM) and mixed PWM/PFM on FPGA chip*, 2020 International Symposium on Computer, Consumer and Control (IS3C), Taichung City, Taiwan, 2020, p. 444-447.
[12] Bor J.-C., Wu C.-Y., *Realization of the CMOS pulse width-modulation (PWM) neural network with on-chip learning,* IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, **45**, 1, 1998, p. 96-107.
[13] Chindris G., *Digital PWM control implementation for System on Chip devices*, 2007 30th International Spring Seminar on Electronics Technology (ISSE), Cluj-Napoca, Romania, 2007, p. 521-526.
[14] Matthew David Venn. Tiny Tapeout: A Shared Silicon Tapeout Platform Accessible To Everyone. TechRxiv. July 09, 2024. **DOI: 10.36227/techrxiv.172055642.27780676/v1**
[15] TinyTapeout, https://tinytapeout.com/, last accessed on 25th September, 2024