**Date:**  February 12nd

**From:** Jaric Sloan

**To:** Dr. Kaputa

**Subject:** PVM module

# Introduction

For the PVM module, the three different sources were used to give me assistance and use their concept to meet the requirements. The three different sources are blink, generic counter, and rising edge synchronizer is used to help reach the requirements as shown below. The period and duty input ports were made into 26-bit vectors to support the period up to one second. Both period and duty inputs were meet the desired frequency for the period to be able to meet the duty cycle of the given percentage of zero percent to 100 percent. The vector sizes were calculated with the provided of 50MHz system clock to support one second of the period such as (log (50000000)/1)/log (2) is approximately 25.6 bits, therefore, using 26 bits is the best idea. By using the generic counter example to use the internal count signal to keep track of the increments.

* To have an enable pulse
* To be able to control both the duty cycle and the period of the PVM pulse train
* Ensure the entity inputs not ‘generics’ for the period and duty inputs

PVM Module will be beneficial to Drone to be able to manifest itself in the motor.

# Analysis

By using the similar code fundamental to the Blink file that was provided, the fundamental is still the same for the lab 2 code fundamental.

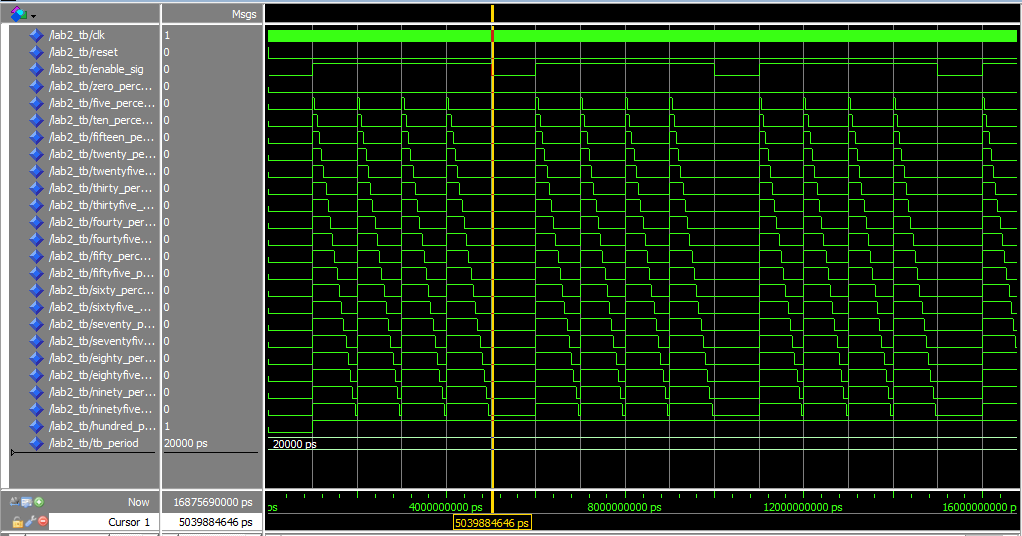


Figure 1:Result of lab2\_tb

As shown in Figure 1, shown the results of the lab2\_tb of increment of five percentage out of 0-100 percent to show a better picture of what it looks like. In the lab2\_tb, a 1 KHz period was used for different cases of duty. The duty remains constant as for other variables remain constant and it would be great if can test the difference of duty changes and something happens during a real-time event. But it does meet all the requirements as shown in Introduction however to control the duty cycle and the period of the PVM pulse train remains remain constant.

# Conclusion

Since the analysis determine that to control the duty cycle and the period of the PVM pulse train was remain constant. As shown in Figure 1 shows all the possibilities result from various output signals of the increments of five for 0-100 percentage of the constant duty cycles. It’s required more testing than forcing them to be constant vectors however it is running smoothly.

# Code Appendix

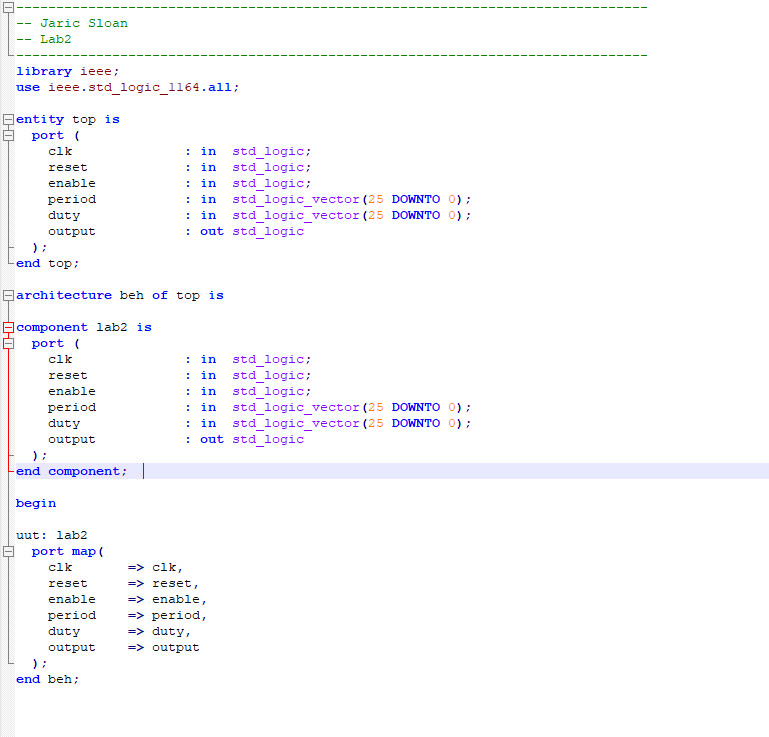


Figure 2: Top.vdhl file shown that lab2 clk are getting clk from the top

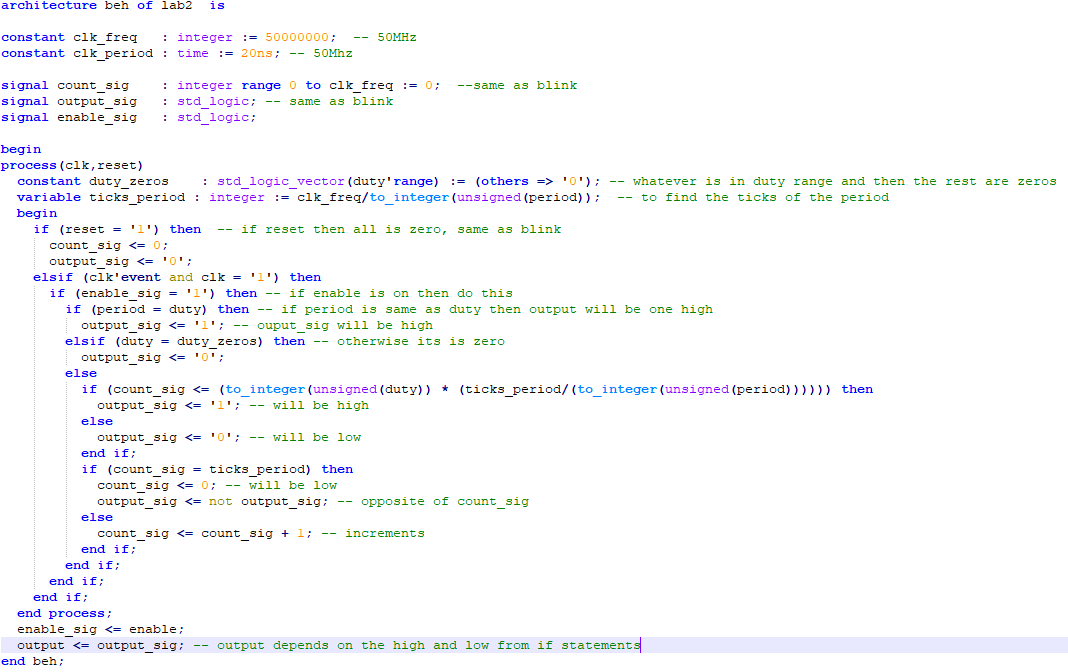


Figure 3 : lab2.vhdl to calculate the count\_sig depends on the duty and ticks of period to show the high and low of the duty percentage

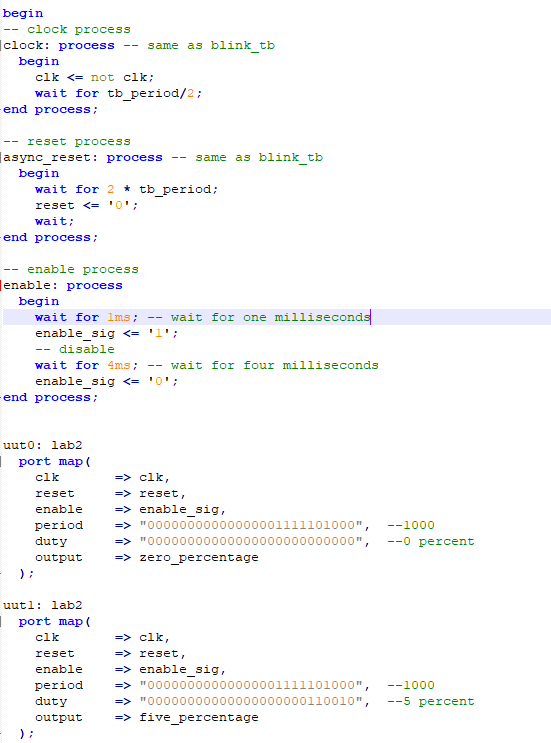


Figure 4: the test bench of lab 2 show the duty percentage of 0 to 100, period of 1000Hz