



Implementation of PWM on the Xilinx Spartan 6 FPGA with Programmable Period and Duty Cycle

Archishman Guha⁽¹⁾, Subhasri Chavakula⁽¹⁾, Harsha Avinash Tanti⁽¹⁾ and Abhirup Datta⁽¹⁾

(1) Indian Institute of Technology, Indore, India, e-mail: ms2104121005@iiti.ac.in, mt2102121005@iiti.ac.in, phd1901121009@iiti.ac.in, abhirup.datta@iiti.ac.in.

Pulse Width Modulation (PWM) is a simple but effective way to control analog circuitry, servo motors, ADCs, sampling circuits and much more [1]. A PWM signal is generated by keeping the signal “ON” for a specific time span and keeping it “OFF” for a specific time span. The ON-OFF times govern a very important parameter, the Duty Cycle. With a wide variety of applications of PWM, introduction of Re-programmability and Reconfigurability to it would certainly allow us to leverage more out of it. Hence, we choose an FPGA for the implementation of the same. An FPGA implementation comes with benefits of fast prototyping and reconfigurability of the design. The FPGA generated PWM signal can be used to control antenna dishes at various angular speeds and with varying precision. This is achieved through the user programmable nature of the PWM signal. The user has the liberty to change the duty cycle and the period in real-time.

The proposed implementation of the PWM signal on the Spartan 6 FPGA involves counting integer cycles of the FPGA clock that have elapsed. Higher is the period, more would be the number of duty cycles that we can accommodate. For high-speed applications, we may choose to reduce the period at the cost of precision of analog values that can be emulated. Implementing the design on an FPGA offers quick prototyping, parallel execution and reconfigurability. The proposed design presents the design and implementation of Pulse Width Modulation with its period and duty cycle being user defined. This design was implemented in Verilog. It has a total of 4 I/O ports, a clock input, a switch each for increasing & decreasing the duty cycle, a 4-bit wide vector for setting the period & an output port to output the PWM signal. The user needs to press the corresponding push button to either decrease or increase the duty cycle. The digital input from the push buttons is being sampled by a high-speed clock, it is beyond human ability to press or release a switch at such a high speed. To address this, a slower enable signal operating at 4Hz, obtained by scaling the actual FPGA clock, was used to sample the input from the push buttons. Lastly, a counter runs in coordination with a comparator that continuously compares the counter value to that of the duty cycle set by the user. Figure 1 shows a time period of 160ns, that amounts to a frequency of 6.25MHz. This result is obtained when we assign 15(Binary 1111) as the value of the period. The clock, running at 100MHz, takes 16 cycles to count a period of 160ns.

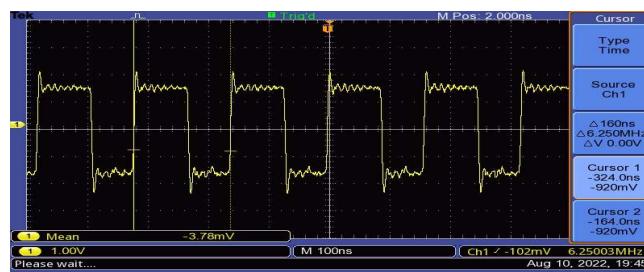


Figure 1. Oscilloscope output for a time period of 160ns running with a time period of 6.25MHz.

1. A. Al-Safi, A. Al-Khayyat, A. M. Manati and L. Alhafadhi, "Advances in FPGA Based PWM Generation for Power Electronics Applications: Literature Review," 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2020, pp. 0252-0259, doi:10.1109/IEMCON51383.2020.9284821.