PWM Generator with Variable Duty Cycle Using FPGA

Team A:

Devanshi Chauhan

Jainam Shah

Harsh Maghnani

Siddhpura Hemangi

Overview

Pulse Width Modulation (PWM) is a very popular modulation technique which is mainly used to control the power delivered to electrical devices such as motors. In this task the team will have to generate pulse width signals using Verilog (Hardware Descriptive Language) and realize its hardware using FPGA. A field-programmable gate array (FPGA) is an integrated circuit that can be programmed or reprogrammed to the required functionality or application after manufacturing. Important characteristics of field-programmable gate arrays include lower complexity, higher speed, volume designs and programmable functions

Task 3.1 - Counters and Universal Shift Register

Installation & introduction

The team is expected to download Intel's Quartus 19.1 tool for executing the given task. The team is also advised to go through the concepts of digital electronics before the commencement of the tasks.

Implementing Universal Shift Register and Counter

At this stage the team members are very much familiar with the concepts of counters and universal shift register. Now, for task 2.1 ,the team has to design,

1. Universal shift register
2. Synchronous up counter using J,K flip flop(MOD 14)

using the Verilog HDL,realize their hardware, show its RTL Design and generate its corresponding waveforms .

Note: Individual zip folders for counter and shift register will be considered as submission which would contain the files such as verilog code, compilation report from the quartus tool, waveforms (**prior and after the hardware generation**) and their block diagram and rtl view.

Task 3.2 - Designing main verilog code

Implement the pwm generator block using verilog

Now, the team will start working to create the verilog code for the PWM Generator. The frequency of the pulse should be 2 MHz. Duty cycle variation should be in the range of 5%. Submit the verilog project, Pin assignment, block diagram, RTL layout of the project

Task 3.3 - Testbench File

Testbench for PWM Generator

Your final task is to create a working testbench file using verilog and run the testbench in Modelsim software. You should be able to vary the duty cycles from 0-100% in steps of 5.

Sr No.	Duty Cycle
1.	0%
2.	25%
3.	50%
4.	65%
5.	100%

Table 1. Test cases for testbench