COM3525 – PROJECT

TRAFFIC SIMULATOR

Name, Surname : Elvin Huseynli

Student ID: 20290122

Date: 20/12/2021

Sign: 

1. The problem

In this project we were expected to create a Traffic Simulator using VHDL, a hardware description language used for digital systems. Following some tutorials step by step, at the end we were needed to create a Traffic Simulator model. The problem is mainly focused on how to use VHDL elements, modules and libraries in an efficient way and create an algorithm using FSM, which stands for Finite State Machine. The aim of FSM is to create a sequential logic, which creates a loop of some actions, which are related with each other. While implementing Traffic Simulator, we had to realize the conditions where circumstances could change and we were expected to come up with a correct solution to build up a better code, using different kinds of VHDL components. The code's readability and simplicity are also increased, with complex VHDL elements, to make it easier to understand the main concept of the code and its simulation. Following some tutorials, we learned how to improve code's concept and its implementation step by step. At the end of tutorial, by using what we learned, we tried to implement the topics to create a Traffic Simulator, similar to the sample code.

2. What we learned

While following tutorials, I also made some investigations to have a better understanding of VHDL and comprehend the details of it. To start with, I firstly learned the basic concepts, which are similar and related to the normal programming languages. The loop concepts, wait statements and others work in the same way they do in normal programming languages. Core features of those loops and wait statements are used to control the execution and simulation of code. As it is an example of parallel programming languages, it has some unique and basic features, such as variables, signals, sensitivity lists and if-else statements. Despite the fact that those exist in normal programming languages, theirs features can differ from the parallel one in some contexts. For example, as variables are being declared in processes, signals should be declared inside architectures, before begin part. Also, we learned other data types, which are different than normal ones, such as std_logic, which stands for Boolean values working better in VHDL simulator. By using those, we could work with waves to see and realize the simulation patterns. Then, the next concept was mainly about how to represent integers in VHDL, how to create a module, and basic block designs, such as multiplexer and flip-flop. By learning case-when statements, it became easy to use conditional statements compared to if-else statements. Also, using port map we could include the variables in modules to the main test bench file in an efficient way. Its accessibility made us to use generic and constant maps for a better

visualization and implementation. In addition to it, we learned how to create concurrent statements, which helped us to create a process with certain, clearly defined characteristics. Concurrent statements are barely equal to a process with sensitivity list. In the last part of the tutorial, we were expected to learn and implement the concepts of FSM and real-life problems and solutions to it. Starting by creating a fully functional timekeeping module in VHDL, we learned the difference between simulation time and real-time in VHDL. By creating timers, procedures and FSM, we were able to write a code for Traffic Simulator and implement this in simulator.

Note: I made two separate videos, one with introducing myself, and other one with explanation of video.

Introduction:
https://drive.google.com/file/d/1RIXmGNdG0ttC8Magw2jfJrlG6xAsXDxW/view?usp=sharing

Explanation:
https://drive.google.com/file/d/1hrGK8eSuHMY43MV0DzHgOwOXIqWc5nso/view?usp=sharing