Name: Ömer Faruk Uysal

Student ID: 19290337

**Microprocessors and Embedded Systems Project**

**Describing the Problem**

The objective of this project is to explain how a Finite-State machine operates using a practical example of traffic lights, and this example is implemented in VHDL hardware description language.

Finite state machines are used to simulate sequential logic. Since, there is a sequential state flow in the traffic lights, a finite state machine is a good way to simulate and control the traffic lights.

In the given traffic lights problem, there is a intersection of 4 roads and in each road we have a traffic light with the colors red, green and yellow. Traffic lights transition from one state to another after a fixed time interval to control the traffic flow of intersection. The roads are heading to each other will act in the same pattern. For example, when the traffic light of the north road is green the south one also should be green. Therefor, these are counted as just one state.

The whole cycle can be represented as 8 states: start-north, north, stop-north, west-next, start-west, west, stop-west and north-next. In each of these 8 states, there are 6 light signals can be changed: north-red, north-yellow, north-green, west-red, west-yellow, west-green.

It is decided that red and green lights take 60 seconds while the yellow lights take 5 seconds. Initially the lights are red in both directions. So all the light signals are set to 0. This is the north-next state. Next state is start-north, and in this state both the yellow and red lights of the north-south direction are on simultaneously for a short period of time. After that state, north-south direction gets the green light while the east-west is red and this is the state north. Then, the lights in a north-south direction will change into the yellow in the stop-north state. In the upcoming state, west-next state, all the lights are red again for a short period of time. The next state is start-west where the lights are red and yellow for a short period of time in the west-east direction. Then in the west state, west-east direction will have the green light while north-south direction is red. And as a last state, stop-west state, the lights will be yellow in the west-east direction for a short period of time. Then, all the states will be repated in a loop. So, with these 8 states, the finite state machine can be constructed to solve the traffic lights problem.

**What Is Learned**

First of all, I learned what VHDL is. VHDL is a hardware description language and it is not a software programming language like C, Python or Java. This language is used to model the behaviour and structure of digital systems.

Along the tutorials, I learned the basics of VHDL language. These can be listed as follows:

* the wait statement suspends the execution of the process while wait for statement delays the process for given time.
* signals are accessible outside of a process but variables are not. So variables are good for develeloping inner process development.
* std\_logic type is a special type which is mostly used type in the VHDL. While an integer type can represent only 2 value ‘1’ or ‘0’, the std\_logic type can represent 9 different values which are possible states that a digital signal can be in.
* a module is a self-contained unit of VHDL code. Modules communicate with the outside world through the entity. Port map is the part of the module instantiation where you declare which local signals the module’s inputs and outputs shall be connected to.
* Clocked logic, commonly referred to as synchronous logic or sequential logic, is used in the great majority of VHDL designs. Only a master clock signal may start a clocked process; other input signals do not initiate it.
* Flip-flops are the fundamental building block of timed logic. There are several variations of it.
* Clocked processes are turned on simultaneously, and inputs are taken all at once, while outputs are taken from the last iteration, which is more effective for deep logic. The algorithm's operations could happen on a clock cycle.

**Ideas To Improve**

It is possible to utilize impure functions and processes since they make code easier to read and managable.