COM3525 MICROPROCESSORS AND EMBEDDED SYSTEMS

YAHYA BATURAY SARAÇOĞLU – 18290121

1. **The Problem:**

The main problem in this project is complexity and disordering dominate as a result of simultaneous execution of multiple processes. Traffic light simulation emerges as the real-life adaptation of this explanation.

2. **What you learned in brief:**

Real life systems that require more than one processing capacity can be simulated through modeling systems and vhdl language.

Systems embeded with the simulation method can save us many costs and save time.

As the number of code blocks we write starts to increases, it also starts to get complicated and get out of control. The way to solve this situation is to always use divide and conquer. In this way, we can solve the code by dividing it into smaller subprograms, we can make the code look simpler and work more efficiently.

I learned that a process with a sensitivity list is equivalent to a process with a wait-on statement at the end.

I learned to use functions and procedures as subprograms instead of reusing the same formula or the same piece of code in many places.

Some of the problems I encountered while writing the project and the alternative solutions I applied to them:

1. **When I try compile both of my files (code file and test bench file) together,**

    I got an error like -> “Unknown expanded name”.

    Later I realized I got this error because I didn’t compile the two files sequentially in a simulator.

    In order for test bench to use the compiled source file, it must not have been compiled before from the source code file.

    I solved this problem by compile the files in corect order in simulation.

2. **When I saw that my variable that I wanted to update its value in the process could not react the value I wanted.**

    I learned that the process must be paused so that the variable in the process can be updated continuously. Otherwise, only the last assignment operator in the process is executed.

3. **During the traffic light simulation while I am creating a finite-state machine I saw that my output is delayed by one clock cycle causing stuff to happen in the wrong state.**

    Then I realized that the output signals and the state signal changed simultaneously.

    As a solution for this problem, I delayed the state changes in time with using counter signal for controlling how long a state machine stays in each state.

4. **As another problem, the simulation I created took place in a very small time frame and I found a solution by creating a timer to sync simulation time to real life time.**

3. **Any ideas to improve:**

    If we manage to use subprograms such as functions and procedures more efficiently, we can finish many functions in a shorter way.

    For larger real-life simulations we can use auto functions for less manual identification.

Drive link for the project demonstration video:

https://drive.google.com/file/d/1YiPsdXHY_7V84GdplxdezREPndw_gRXl/view?usp=sharing