Homework 1: Traffic Light

1. Overview

The purpose of this lab is to get you familiar with the Verilog development environment. Please implement the functionality of a traffic light using a finite state machine written in **Verilog**.

2. General rules for deliverables

You need to complete this homework **INDIVIDUALLY**. You can discuss the homework with other

students, but you need to do the homework by yourself. You should **NOT copy** anything from

someone else, and you should **NOT distribute** your homework to someone else. If you violate any of

under these rules, you will get **NEGATIVE scores**, or even fail this course directly.

When submitting your homework, compress all files into a **single zip file**, and upload the

compressed file to Moodle.

Please follow the file hierarchy shown in the following figure.

A screenshot of a computer

Description automatically generated with medium confidence

1. F740XXXXX (folder) (學號，預設是PXXXXXXX)
2. src (folder)
3. report.docx (Briefly explain you design)

3. Design specification

Diagram

Description automatically generated

Diagram

Description automatically generated

作業規則如下：

1. 綠燈維持1024 個cycles。 （起始狀態）
2. 沒有任何燈號維持128個cycles。
3. 綠燈維持128個cycles。
4. 沒有任何燈號維持128個cycles。
5. 綠燈維持128個cycles。
6. 切換成黃燈維持512個cycles。
7. 再切換成紅燈維持1024個cycles

輸入訊號：(電路為clock正緣觸發

1. pass：1bit訊號，當pass為1時，若當前狀態非起始狀態之綠燈，強制切換成起始狀態之綠燈第1個cycle，若原本為起始狀態之綠燈則不改變燈號和cycle。當pass為0則沒有任何動作。
2. rst：1bit訊號，非同步正緣時觸發，將燈號狀態設成綠燈第1個cycle。
3. clk：1bit clock訊號。

輸出訊號：

R：1bit訊號，代表紅燈的輸出訊號。

G：1bit訊號，代表綠燈的輸出訊號。

Y：1bit訊號，代表黃燈的輸出訊號。

4. Project structure

In this course, our work falls in the category of **Cell-based IC design** which consists of the following steps:

1. Verilog coding
   1. Write your Verilog code to meet the design specification.
2. RTL simulation
   1. Verify you design by running **test bench**
   2. The simulation process is done purely with software. No hardware has been inferred yet.
3. Logic synthesis
   1. Synthesize your Verilog design using EDA tools that convert your Verilog code to **actual hardware.**
   2. EDA tools will synthesize your Verilog code into **gate-level netlist**, which is a file specifying the logic gates and the wire connection between them.
4. Pre-layout simulation
   1. In this step we will verify your design after logic synthesis. Make sure your design meets the specification.
   2. Take the **gate-level netlist**,and verify your design again using the Verilog **test bench.**
5. Automatic Placement & Routing (APR)
   1. APR stands for **Automatic Placement & Routing.** You don’t have to do this part in this course.

In this lab, you have to verify your design till **pre-layout simulation in our computing environment.**

To help you design your traffic light module in Verilog, we prepared a Makefile-based project that take case of the following process for you:

1. RTL simulation => **make rtl**
2. Logic synthesis => **make syn**
3. Pre-layout simulation => **make pre**

First download the project for this lab from this [repo](https://github.com/WeiCheng14159/DLIC_2021_pub/releases). Unzip the file and go to the directory labeled “**hw1\_traffic\_light**”. There are several folders in this project.

|  |  |
| --- | --- |
| conf | This directory stores the configuration files for different waveform viewing software (i.e. simvision, nWave) |
| doc | Contains documentation for this project. |
| script | Contains **synopsys\_dc.setup, synthesize.tcl, traffic\_light.sdc.**   1. **synopsys\_dc.setup** file specifies the cell library related information used in this project. We set the configuration to match the setup of the U18 process. 2. **synthesize.tcl** script tells the **Design Compiler** software how to synthesize your design into hardware. 3. **traffic\_light.sdc** is the **design constraint file** specifying the performance requirements of your design. You are allowed to change the cycle time of your design by changing the line **“set cycle xx.x”** in this file. |
| sim | This directory contains the **Verilog test bench** and **related text files** used for testing your design. Also, there is a soft link **fsa0m\_a\_generic\_core\_21.lib.src** file that points to a file in the cell library. This file is necessary for pre-layout simulation please don’t delete it. |
| src | This is where you write your **Verilog code.** There should be an empty Verilog file named “**traffic\_light.v**” where you can design your module. You are allowed to add multiple Verilog files here if you know what you are doing.  **Make sure you submit ALL Verilog files in this directory to Moodle so we can grade your work.** |
| **Makefile** | This Makefile defines multiple targets that help you test your design in a very efficient way. This Makefile contains the following targets:   1. **make rtl: Run RTL simulation** 2. **make nw: Run nWave program** 3. **make syn: Synthesize your design using Design Compiler** 4. **make pre: Run pre-layout simulation** |

5. Start coding

The design flow consists of the following steps:

1. Write your Verilog code in **src/** directory using any text editor you like.

Rectangle

Description automatically generated with medium confidence

1. Compile and run RTL simulation by typing **“make rtl”** in the project directory. You will see the simulation process running and telling you which part of your design does not meet the specification. Type “**make nw**” to launch the **nWave** software that shows the waveform of your design. Then, observe the waveform and make modifications to your code.

A screenshot of a computer

Description automatically generated

1. Make sure you pass the RTL simulation before you proceed to the next step.

Text

Description automatically generated

1. Synthesize your design by typing “**make syn**” in the project directory and the **Design Compiler** software will start running. At the end of the process, the **Design Compiler** software will stop and wait for your inputs.

Text

Description automatically generated

1. You can type **“report\_timing”** to look at your timing information or type “**report\_area”** to look at your area information. Or type “**exit**” to leave **Design Compiler.** Make sure the timing of your design is **MET** before you proceed to the next step.

Text

Description automatically generatedText

Description automatically generated

1. You can now find **two files: traffic\_light\_syn.sdf, traffic\_light\_syn.v** in the **syn/** directory. These two files will be used in pre-sim later.
2. Type “**make pre**” to run pre-sim. If you see “**Congratulations !! Simulation PASS !!**” then you have successfully completed this lab. If not, you should modify your design until everything is correct.

Text

Description automatically generated

6. Homework requirement and Grading

* **50%**
  + **Pass RTL simulation to get 50%**
  + **Pass Pre-layout simulation to get the remaining 50%**
* **50%**
  + **Report**