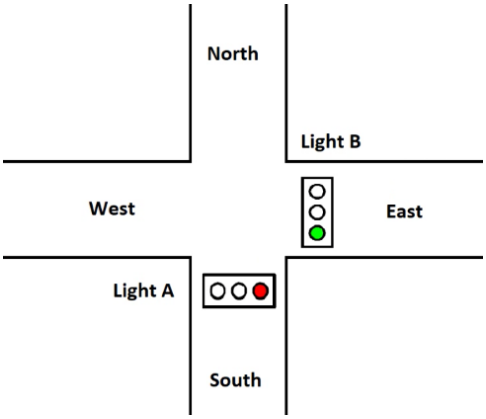
Digital Logic Design

Final Project

In this project, you are going to controll a traffic light system. For that matter, we define several simple rules.

Consider there are two traffic lighs you want to controll:



1. Ligh A: Controls the traffic flow between North and South.
2. Ligh B: Controls the traffic flow between West and East.

Rules:

1. Each light has the ability to show three different colors (Red, Yellow, Green).
2. Both lights can not be green simultaneously.
3. Both lights can not be yellow simultaneously.
4. The green color has to last for 6 seconds to let cars pass.
5. The yellow color has to last for 1 second to alert drivers not to pass

To implement these rules into a fully functional system, you need to design a simple state machine. So here is a list of possible states:

State	West-East	North-South	Delay (seconds)
S0	Green	Red	6
S1	Yellow	Red	1
S2	Red	Red	1
S3	Red	Green	6
S4	Red	Yellow	1
S5	Red	Red	1

In the First state, West-East is green and the North-South is red, and the green light lasts for 6 seconds. Then in the state S1, West-East turns into yellow for 1 seconds, but The North-South light is still red to avoid any accidents. And so on…

Why do we need 6 states? Because after State S5, We can jump to state S0 and repeat everything.

Good Luck!

Mosayebi
Do not hesitate to ask your question
Via eeXBee@gmail.com

Digital Logic Design

Final Project

Part 1. Design and draw the state machine.

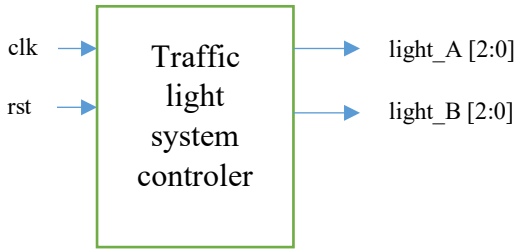
Part 2. Code a verilog module to cotroll these traffic lights. You can use as many internal modules as you need. Here are some details that might help:

We have two lighs and each light has three colors. So we will have six outputs:

```
output [2:0] light_A; output [2:0] light_B;
```

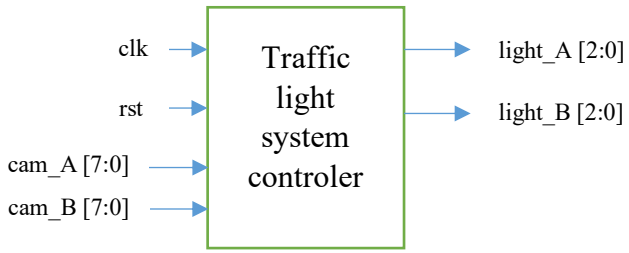
As you can guess, each wire/bit can control a single color in a single light. For example `light_A = 6'b100` turns on the green light of A, and also turns off yellow and red.

We also have two inputs: reset and clock. So the black box looks like this:



Another important point is that you can not use delay assignments. As you know already, delay assignments are meant to be used in testbenches and can not be synthesised. Use another mothod to keep track of passed seconds. Set clock frequency to 1 KHz if it helps.

Part 3. Imagine there are two traffic camera setups, each of which can count all of the cars stopped behind thier red lights (These cameras are using DNN and Image processig to do so and you will learn to implement them using HDL in an other course). Each camera can output a number between 0 to 255, indicating number of the cars behind their red lights and You receive this number in your module. Based on these two numbers, adjust red and green delay to gain an optimum traffic flow.



Good Luck!

Mosayebi
Do not hesitate to ask your question
Via eeXBee@gmail.com

Digital Logic Design

Final Project

Note 1: You should at least provide two Verilog files: one for the module, and one for the testbench. There should be enough test cases in your testbenches to test modules for different input and output values.

Note 2: This project must be done individually; thus, in case of any similarities between the codes provided by the students, all of those will receive a "-50".

Note 3: Upload your codes as one zip file.

Note 4: Please name your files as below:

{Your_Last_Name}.{Your_First_Name}.{Student_Number}.{Module_Name}.v

{Your_Last_Name}.{Your_First_Name}.{Student_Number}.{Module_Name}.Testbench.v

Example: Cruise.Tom.98777777.Main.Testbench.v

Note 5: Provide a report in pdf format alongside with project files. Report should be at least 3 pages, proving that your design actually works. It should contain schematics, state machines, several screenshots from simulation waves and your descriptions.

Good Luck!

Mosayebi
Do not hesitate to ask your question
Via eeXBee@gmail.com