Laboration 1
SystemC TrafficLight

Björn Hvass
Cyril Barrelet

March 10, 2019

Course:	TDTS07
Liu Ids:	Hvass bjohv276
	Barrelet cyrba593

1 Sensor module

The purpose of the sensor module is to keep track of how many cars are waiting for a traffic light. To do this it has one increment method and one decrement thread. Furthermore, it has two input signals. The first is from the generator that is used by the increment method. The second is from the controller and is used by the decrement thread. The module also has an output signal to the controller module.

1.1 Sensor increment method

The purpose of this function is to count the number of cars waitting for the green light.

1.2 Sensor decrement thread

The purpose of this function is to empty the cars queue in a time interval when the light is green. Since we wanted to be on a time interval, we implanted using a thread. The time interval is the time needed by car to leave the junction.

N	:	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
S	:	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
E	:	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0
W	:	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0

N	-	N	-	N	-	N	-	N	-	N	-	N	-	N	-	N	-
-	S	S	-	-	S	S	-	-	S	S	-	-	S	S	-	
-	-	-	E	E	E	E	-	-	-	-	E	E	E	E	-	
-	-	-	-	-	-	-	W	W	W	W	W	W	W	W	-	

2 Generator module

The generator module simulates cars in order to show the different properties of the lights. To prove those properties, two differents modules has been created. The first one generates pseudo-random cars to show the behaviour of the lights during a time period. The other one generates cars by reading a text file file to show all possible cases, one by one.

The generator contains a print method, a generator method and a thread to trigger the generator method.

2.1 Event thread

The event_creator is a thread that triggers the generator method periodically. The generator output will be '1' if there is a car or '0' if there isn't.

2.2 Random generator method

For every events, the module has a 30% chance of generating a car.

The toggle is used to invoke a change in the channel when there is a car. The change is used by the sensor to trigger its increment method.

2.3 Generator method

This module reads a dedicated text to show the different properties.

All cases are presented below:

3 Controller module

The controller module contains all to the logic needed to control a traffic light in a four-way junction with traffic flowing in four directions. However does not support turning traffic, for example, a car can't go from north to west. The module has four input signals, one form each direction. This signal tells the controller if there are any cars waiting for a green light. There are also four boolean output signals to represent a green or red light. The controller has one thread and a print method. The print method is triggered by the tread and prints the current state of the traffic lights. The controller's thread is covered in greater detail in the next section.

3.1 Controller Thread

The controller thread contains a state machine with seven states. The states are "*WEST, EAST, EAST_ AND_ WEST, SOUTH, NORTH, NORTH_ AND_ SOUTH, NONE*". These states each represents one or two traffic lights. There are states that enable green light for two directions at the same time. This is due to the fact that cars from the north and south or east and west doesn't cross each other's driving lanes. Thus there is no problem if both have a green light at the same time. The *NONE* state is for when there is no cars.

To transition between states the machine uses the input signal to check if there is cars waiting for a green light in any one of the four directions north, south, east and west in combination with some logic in the states. This logic for the states is described below.

A transition from one of the one-directional states can happen in one of four ways. First of if there are no more cars watching for green light the state machine will switch state to *NONE*. Secondly, if there are one or more cars in the opposite direction the machine will switch state to the corresponding dual light state. This will keep the current green light green and switch the light for the opposite lane to green as well. The final two ways is if there is one or more cars waiting in either of the crossing directions. Then the machine will wait for six seconds, this enables at least three cars to cross before swishing the signals.

The procedure for transitioning from a bidirectional state is largely the same. With one addition if there are no cars left waiting for a green light in any of the directions that the state represents the machine will switch to the correct one directional state. The criteria for switching to the *NONE* state is also for there to be no cars in either of the directions of the state.

The state machine is implemented using a systemC thread, this is so it can control the time that the output signals are high or low. This enables the state

machine to stay in one state for a specific period of time. This is required for good traffic flow through the junction.

4 System

The traffic light system has four generator modules, four sensors modules and one controller module. There's one generator and sensor per direction. Each generator have a out channel to a sensor. Each sensor have a out channel to the controller. The controller have four out channels, one for each sensor. See figure 1 for a visual representation.

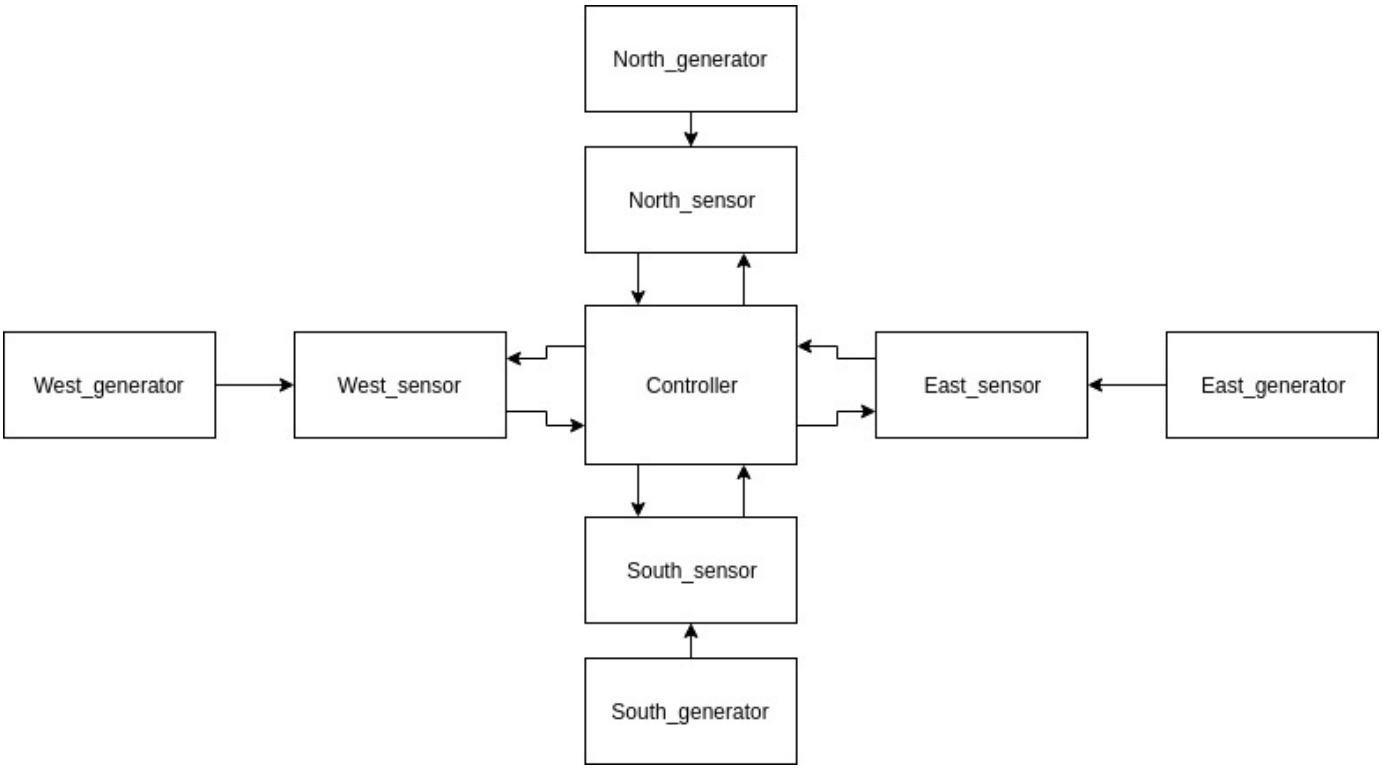

Figure 1: Traffic light system.

5 How to run

To run the system with the test files first make the project with < make -f MakeFile > and then <main.x fil1 file2 file3 file4 fileOut>. The files are input files for the directions with 0 representing no car and 1 represents a car. The last file is an output file. To compile the project to generate pseudo-random cars, use <make -f MakeFile$_{R}andom > andrun < main.x > fs$

6 Independence and Safety

The proof that the lights work independently and safety is demonstrated by the test bench. This test bench, given the test files, represents all sixteen possible cases.

For example, if the light NS is green, the light SN is red if there are no cars coming in the direction SN. Furthermore, the North or South lights can't be green at the same time as the East or West lights. And even more, if a vehicle arrives at the crossing, it will eventually be granted the green light, according to the safety property.

This result is possible thanks to a state machine or controller" who prevents any problem by reading the sensors and set the lights properly.

7 Conclusion

After all those properties have been proved, the random car generation can ensure that the system works correctly in the time.