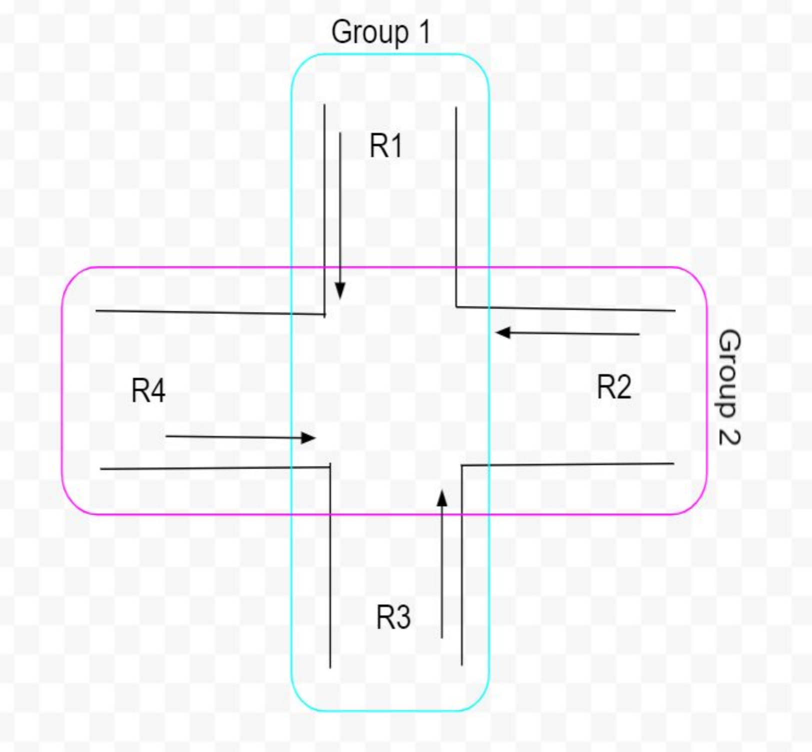
4 Way Traffic Light Controller

Cali Malone, Bianca Muller, Azra Jakupovic,
Marielly Perez

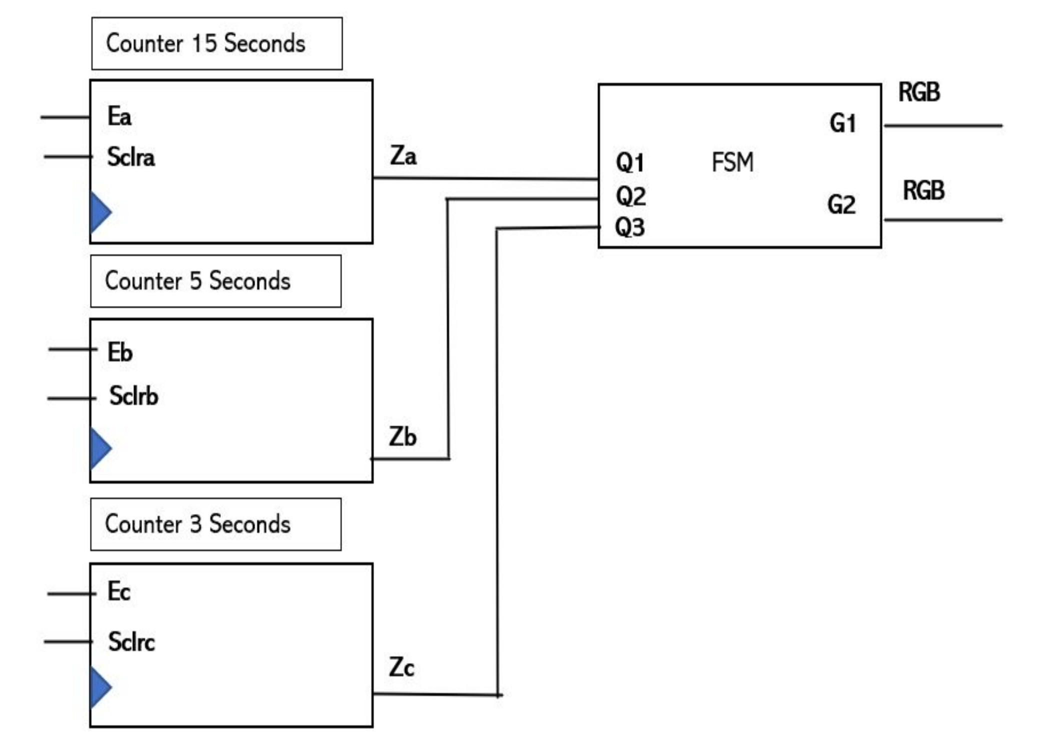
Methodology

- A 4 way traffic light controller was created in order to simulate a real life scenario of how traffic light timing works in an intersection.
- The traffic lights are divided into two perpendicular groups. One group goes north and south and the other goes from east and west.
- When group 1 is green, group 2 will be red, and vise versa.
- Each group will have different timing for the yellow and then green lights so that the cars may slow down before they transition to red.
- There will be a period where both groups will be red, acting as the reset of the cycle.

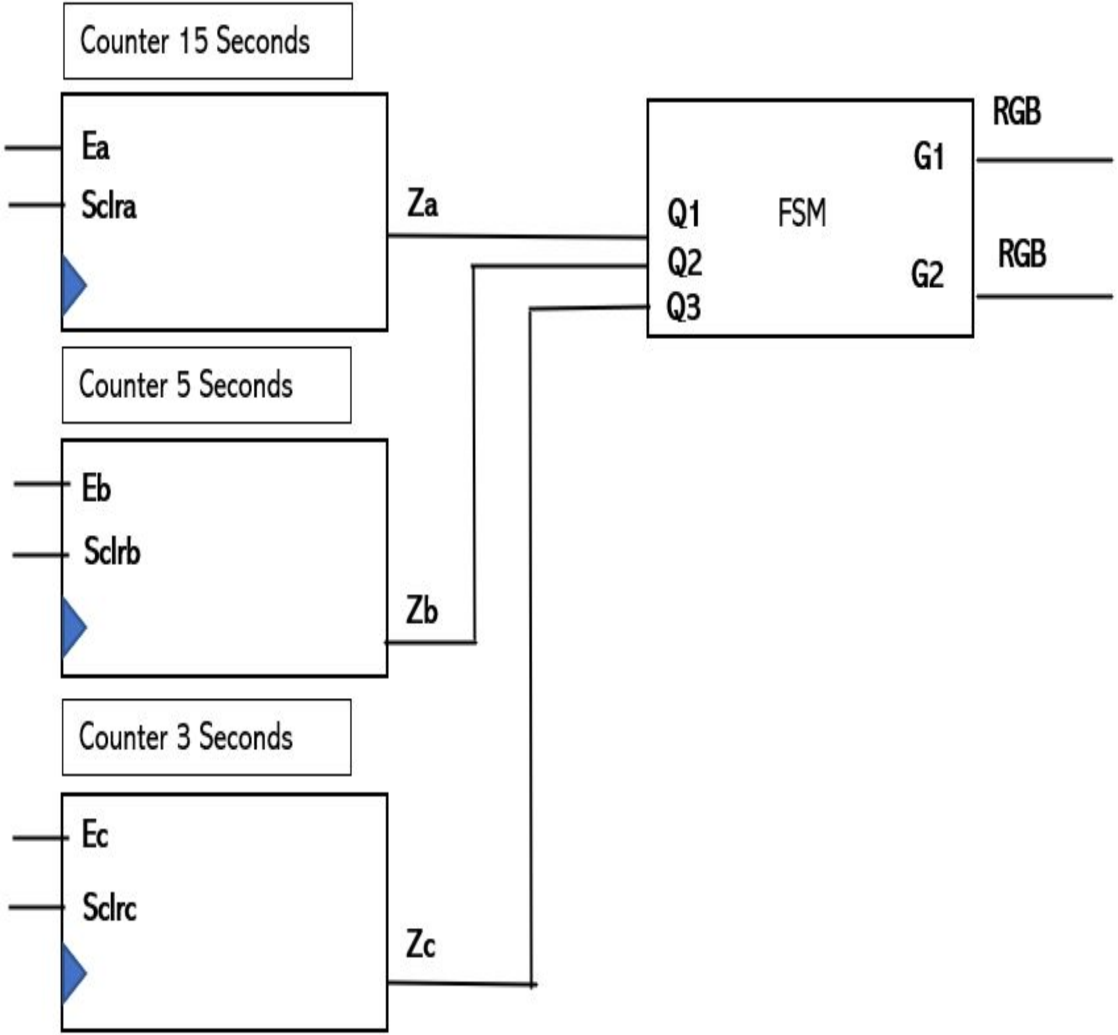
- **Hardware: Artix-7 Nexys A7 DDR Board**
- **Software: Xilinx Vivado using VHDL**



Circuit for Traffic Light



- We will be utilizing 3 counters and an FSM machine.
- Each counter will be responsible for a specific period of time.
- We have 6 inputs in total from the counters
- **Ea, and Sclra** - inputs for the first counter.
- **Eb, and Sclrb** - inputs for the second counter.
- **Ec and Sclrc** - inputs for the third counter.
- **Q1** - Input from **Za**
- **Q2**- Input from **Zb**
- **Q3**- Input from **Zc**
- The output from the FSM will be
- **G1**- Group 1 for **RGB**
- **G2**- Group 2 for **RGB**



Counter :

This counter will be used to sequence pulses. This digital device is a form of a flip-flop.

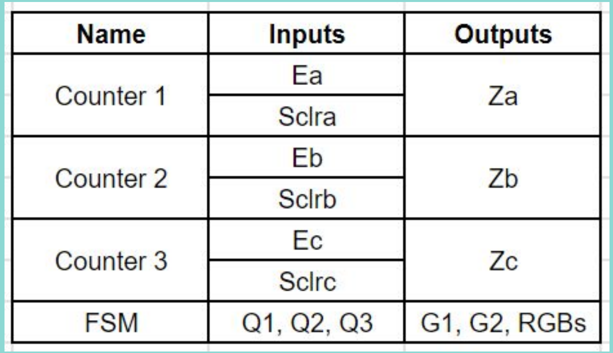
-Counter A will pulse the clock for 15 seconds.
-Counter B will pulse the clock for 5 seconds.
-Counter C will pulse the clock for 3 seconds.

FSM :

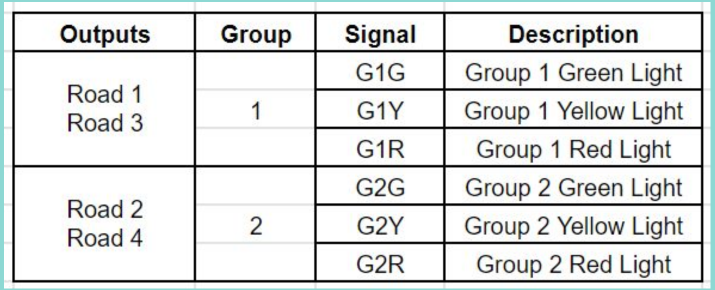
FSMs are used to represent system with a finite number of states. In our case it will have states S1-S5, which will control the RBG lights.

States S1-S5 will be controlled by
-Inputs **Q1, Q2, Q3**
-Outputs **G1, G2**

Inputs/ Outputs

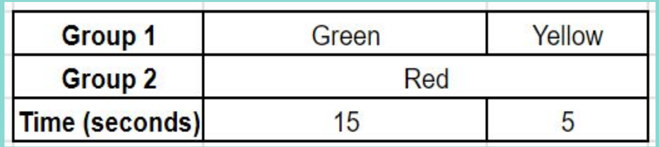


Name	Inputs	Outputs
Counter 1	Ea	Za
	Sclra	
Counter 2	Eb	Zb
	Sclrb	
Counter 3	Ec	Zc
	Sclrc	
FSM	Q1, Q2, Q3	G1, G2, RGBs

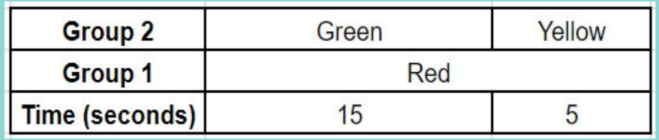


Outputs	Group	Signal	Description
Road 1 Road 3	1	G1G	Group 1 Green Light
		G1Y	Group 1 Yellow Light
		G1R	Group 1 Red Light
Road 2 Road 4	2	G2G	Group 2 Green Light
		G2Y	Group 2 Yellow Light
		G2R	Group 2 Red Light

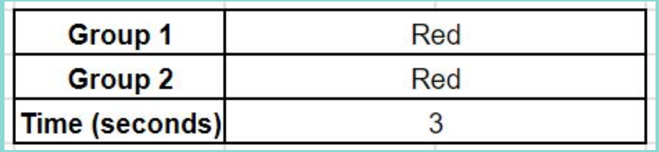
- Road 1 and 3, which are relatively N and S are in Group 1, represented with **(G1G)** for green, **(G1Y)** for yellow, **(G1R)** for red.
- Road 2 and 4, which are relatively E and W will be Group 2, represented with **(G2G)** for green, **(G2Y)** for yellow, **(G2R)** for red.



Group 1	Green	Yellow
Group 2	Red	
Time (seconds)	15	5



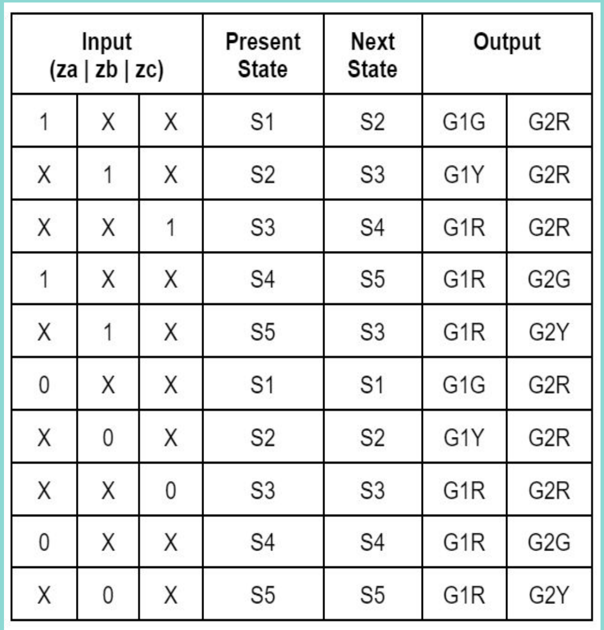
Group 2	Green	Yellow
Group 1	Red	
Time (seconds)	15	5



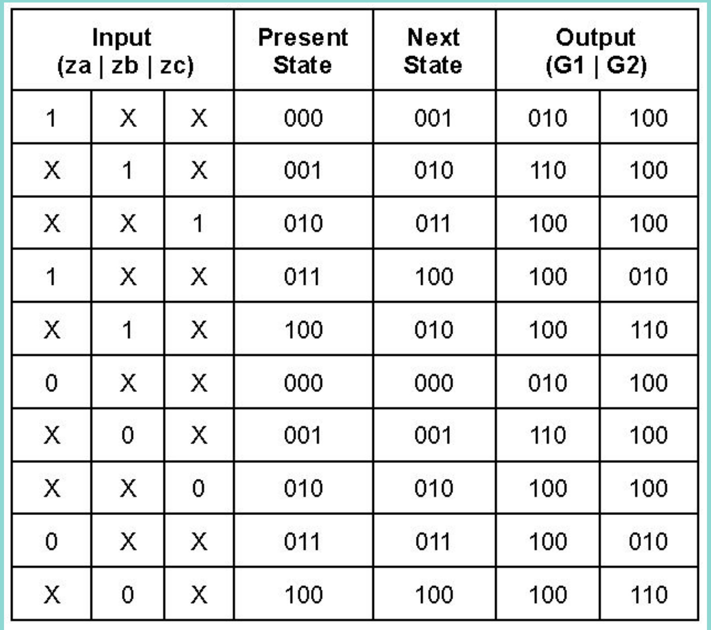
Group 1	Red
Group 2	Red
Time (seconds)	3

- Group 1 will be green for 15 seconds, and yellow sequentially following for 5 seconds. Simultaneously, group 2 will be red for the entirety of the 20 seconds.
- Afterwards, both groups will be red for 3 seconds serving as a reset. This will allow all the cars to come to a halt before the process cycles again.
- The process is the same vice versa for Group 2.

State Table/Excitation Table

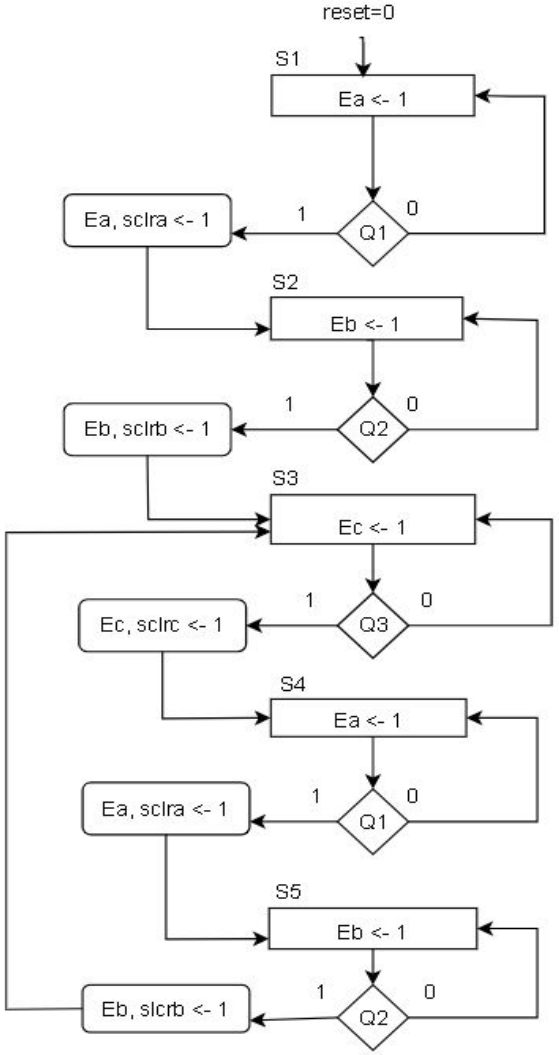


Input (za | zb | zc)			Present State	Next State	Output	
1	X	X	S1	S2	G1G	G2R
X	1	X	S2	S3	G1Y	G2R
X	X	1	S3	S4	G1R	G2R
1	X	X	S4	S5	G1R	G2G
X	1	X	S5	S3	G1R	G2Y
0	X	X	S1	S1	G1G	G2R
X	0	X	S2	S2	G1Y	G2R
X	X	0	S3	S3	G1R	G2R
0	X	X	S4	S4	G1R	G2G
X	0	X	S5	S5	G1R	G2Y



Input (za | zb | zc)			Present State	Next State	Output (G1 | G2)	
1	X	X	000	001	010	100
X	1	X	001	010	110	100
X	X	1	010	011	100	100
1	X	X	011	100	100	010
X	1	X	100	010	100	110
0	X	X	000	000	010	100
X	0	X	001	001	110	100
X	X	0	010	010	100	100
0	X	X	011	011	100	010
X	0	X	100	100	100	110

- Inputs za, zb, and zc control the states.
- Only one input is need at a time.
- When za, zb, or zc is 1 it will prompt the circuit to move to the next state.
- If za, zb, or zc is 0, the circuit remains in the same state.
- When the input in S3 is 1, it changes to S4.



S1:

- **In S1** Group 1 would be green and Group 2 would be red.
- When **Q1 = 0** stays in **S1**.
- When **Q1 = 1** the state changes **S2** the counter runs for 15 seconds. **sclra** will also be **1.** While **Ea** will always be **1.**

S2:

- In S2 Group 1 would be yellow and Group 2 would stay red.
- **Q2 = 0** stays in **S2**.
- **Q2 = 1** the state changes to **S3** and the counter runs for 5 seconds. sclrb will also be 1. While **Eb** will always be 1.

S3:

- In S3 Group 1 would be red and Group 2 would be red.
- **Q3 = 0** stays in S3.
- **Q3 = 1** the state changes to **S4** and the counter runs for 3 seconds. Sclrc will also be 1. While **Ec** will always be 1.
- **S3** is the rest state where both lights will be red.

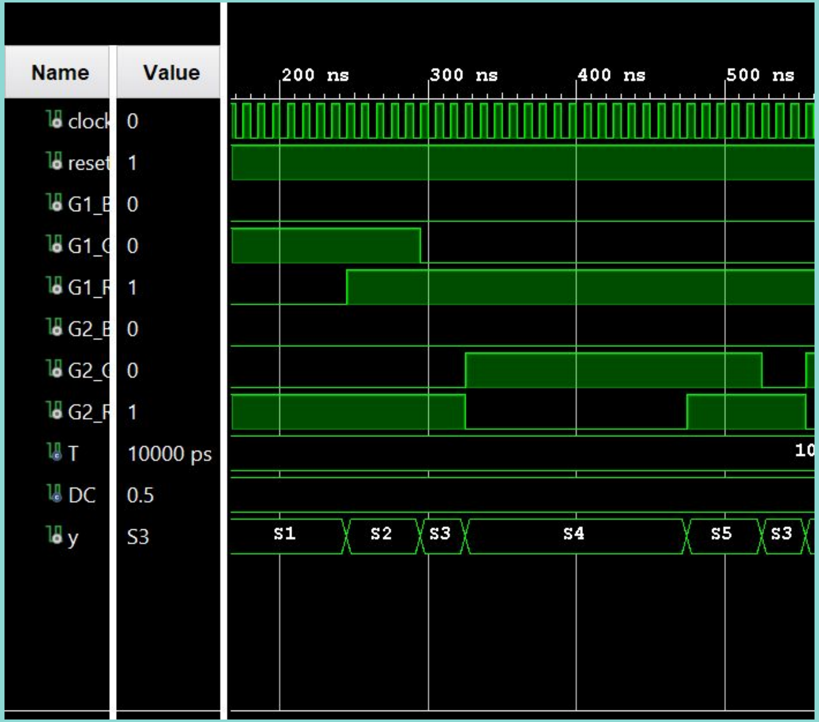
S4:

- In S4 Group 1 would stay red and Group 2 would be green.
- **Q1 = 0** stays in **S4.**
- **Q1 = 1** the state changes to **S5**, and the counter runs for 15 seconds. sclra will also be 1. While **Ea** will always be 1.

S5:

- In S5 Group 1 would stay red and Group 2 would be yellow..
- **Q2 = 0** goes to **S5**.
- **Q2 = 1** changes the state to **S3** and the counter runs for 5 seconds.. sclrb will also be 1. While **Eb** will always be 1.

Simulation



Name	Value
clock	0
reset	1
G1_B	0
G1_C	0
G1_R	1
G2_B	0
G2_C	0
G2_R	1
T	10000 ps
DC	0.5
y	S3

References

[1] Llamocca. “RECRlab.” *VHDL Coding for Fpgas*, http://www.secs.oakland.edu/~llamocca/VHDLforFPGAs.html.

[2] “Finite State Machines: Sequential Circuits: Electronics Textbook.” *All About Circuits*, https://www.allaboutcircuits.com/textbook/digital/chpt-11/finite-state-machines/.

[3] “Digital Circuits - Finite State Machines.” *Digital Circuits - Finite State Machines*, https://www.tutorialspoint.com/digital_circuits/digital_circuits_finite_state_machines.htm.

[4] *Vivado Tutorial - Xilinx.* https://www.xilinx.com/support/documentation/university/Vivado-Teaching/HDL-Design/2013x/Nexys4/Verilog/docs-pdf/Vivado_tutorial.pdf.