

# 1302736

*by* Jason Booth

---

**Submission date:** 18-Jan-2022 09:52PM (UTC+0000)

**Submission ID:** 169528332

**File name:** 19735\_Jason\_Booth\_1302736\_179500\_1783551175.docx (1.59M)

**Word count:** 3240

**Character count:** 16388

60-65

Course: HNC Electrical and Electronic Engineering

Module: SGEE5000 – Electronics II

Assignment Title: Finite State Machines

Name: Jason Booth

University number: 1302736

Date Submitted: 18/01/2022

Word Count: 2552 words

- ① RESEARCH WAS OK, BUT IN-TEXT CITATIONS ARE NOT HARVARD.
- ② DESIGN METHOD WAS GOOD, WITH WELL PRESENTED MAPS.
- ③ TESTING WAS CARRIED OUT USING THE LOGIC ANALYSER. SOME SIGNALS WERE NOT MONITORED.
- ④ ANALYSIS WAS INCONSISTENT ACROSS THE 3 CIRCUITS.
- ⑤ MODIFICATION NOT ATTEMPTED.
- ⑥ PRESENTATION WAS OK, SMALLER PICTURES WOULDN'T HURT.

## Contents

|                                                                  |    |
|------------------------------------------------------------------|----|
| 1. Executive Summary.....                                        | 4  |
| 2. Research into advanced digital design techniques (1000) ..... | 5  |
| 3. Design method, choices and calculations (500).....            | 9  |
| 4. Testing, results and analysis (500) .....                     | 18 |
| 5. Conclusion and recommendations (500) .....                    | 20 |
| 6. Reference list and bibliography .....                         | 21 |

|                                            |    |
|--------------------------------------------|----|
| Figure 1-Pelican Crossing .....            | 5  |
| Figure 2-Puffin Crossing .....             | 6  |
| Figure 3-FPGA .....                        | 7  |
| Figure 4-ASIC .....                        | 7  |
| Figure 5-5 second state diagram.....       | 9  |
| Figure 6-State Table .....                 | 10 |
| Figure 7-5 seconds Karnaugh Maps .....     | 11 |
| Figure 8- 5 seconds MultiSim Design .....  | 11 |
| Figure 9-15 seconds state diagram .....    | 12 |
| Figure 10-15 seconds state table .....     | 12 |
| Figure 11-15 seconds Karnaugh maps .....   | 13 |
| Figure 12-Karnaugh Map for output T .....  | 14 |
| Figure 13-15 seconds MultiSim design ..... | 14 |
| Figure 14-RAG State diagram.....           | 15 |
| Figure 15-RAG State table.....             | 15 |
| Figure 16-RAG Karnaugh Maps .....          | 16 |
| Figure 17-RAG MultiSim Design .....        | 17 |
| Figure 18-Final Design.....                | 17 |
| Figure 19-5 second counter results .....   | 18 |
| Figure 20-15 second counter results .....  | 18 |
| Figure 21-RAG results .....                | 19 |

## 1. Executive Summary

Research was carried out on various pedestrians crossing, mainly a pelican and puffin crossing as they use digital logic circuits and components to operate, unlike a zebra crossing. Characteristics and features were distinguished between the two. The point in this is to gain an understanding of what will be needed to design and develop a working traffic light system through MultiSim. Knowledge and understanding of technologies such as FPGA and ASIC were obtained to establish suitability for developing a working traffic light system. FPGAs are ideally suited for prototype developing as they can be re-programmed so can accommodate changes to the circuit, where as an ASIC cannot be altered. To use an FPGA in mass production of traffic light systems would be deemed overkill as they have a lot more connections for inputs and outputs which are not needed. The ASIC is suited for the project if it was to go into production, as these are more efficient, smaller and cost less to produce when purchasing in mass quantities. This project can be made very sustainable by utilizing modern technologies such as LEDs in place of ordinary bulbs as the wattage is far lower, so power consumption will ultimately be lower. In result of a low power consumption system, it can be reliable to use methods of electricity generators such as a mini-wind turbine and solar panels to provide a fully self-sustained system which reduces the total carbon footprint. Results of the MultiSim circuit convinces that they work correctly and to specification. Recommendations can be made to the circuits to incorporate sensors to detect pedestrians which can increase the allowed length of time for pedestrians to cross and behave more like a puffin crossing.

*WHOLE*  
YOU NEED TO INCLUDE THE <sup>WHOLE</sup> REPORT - SO A BIT  
MORE ON THE DESIGN, BIT LESS ON THE CONCLUSIONS.

## 2. Research into advanced digital design techniques (1000)

### Pelican Crossing



Figure 1-Pelican Crossing [Road crossings - do you know your pelicans from your toucans? | RAC Drive](#)

According to (RAC, 2020), “A pelican crossing is a set of traffic lights with sets of zig-zag lines before and after on either side of the road. They are also known as pedestrian light-controlled crossings and were first introduced in UK in 1969 and were the first light-controlled crossing that could be activated by pedestrians and controlled by traffic lights.”

Pedestrians operate the traffic sequence by pressing a button located at the side of the road and wait for a signal to cross which is represented by a standing red man (do not cross) to a walking green man (safe to cross). Pelican crossings does not account for slower moving pedestrians and instead there is a flashing green man/amber light which allows pedestrians to finish crossing or vehicles to start moving if the crossing is clear. Pelican crossings does not implement sensory devices to allow extra time or to cancel the button being pressed accidentally. (Traffic Choices, 2014)

### Puffin Crossing

Puffin Crossing looks and works similar to a pelican crossing but are much more intelligent through the use of sensory devices. Where a pelican crossing will have a red and green man indications mounted high up next to the vehicle traffic light, instead a puffin crossing will have a red and green man pedestrian crossing symbols displayed on the yellow and black box with the button to operate the sequence. (Shale-Hester, 2020) This allows pedestrians to monitor passing traffic while waiting to cross as well as assists visually impaired people as the lights are closer.

REDUCE THE SIZE OF  
PICTOS.



Figure 2-Puffin Crossing [Road crossings - do you know your pelicans from your toucans? | RAC Drive](#)

A puffin crossing has two sensors on top of the traffic lights (PCD – pedestrian crossing detector, and PKD – pedestrian kerb detector). These sensors detect if pedestrians are crossing slowly and can extend the amount of time traffic is held to allow pedestrians to cross. A PKD detects if pedestrians are at the kerb, this is useful because if the button is pressed and the pedestrian walks off, the PKD can detect that no one is present at the kerb and cancel the request for traffic to stop. (Council, 2009)

#### Technologies that could be used to implement the control circuitry of a traffic light system

There are different technologies that can be used to implement the control circuitry of a traffic light system. To develop a traffic light system a good idea would be to use an FPGA for the development stages as they can compute logic blocks and can be customized because of the re-programmability advantage with a FPGA.

#### FPGA

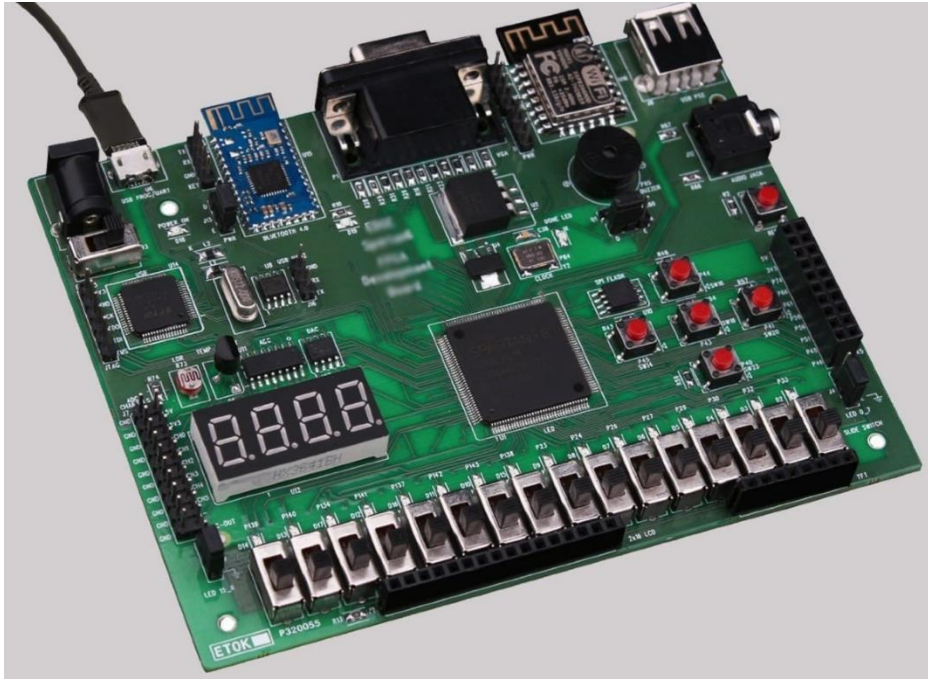


Figure 3-FPGA [How-To-Choose-An-FPGA-Development-Board-1.jpg \(1920×1080\) \(innov8tiv.com\)](#)

According to (Arrow, 2018), “The field-programmable gate array (FPGA) is an integrated circuit that consists of internal hardware blocks known as Configurable Logic Blocks (CLBs) with user-programmable interconnects to customize operation for a specific application. The interconnects can readily be reprogrammed, allowing an FPGA to accommodate changes to a design or even support a new application during the lifetime of the part.”

Because of the ability to re-program an FPGA and configure the hardware, it can be used to optimize a particular set of functions which makes an FPGA a desirable option in many applications. They are often used in developing prototypes. (2018)

## ASIC

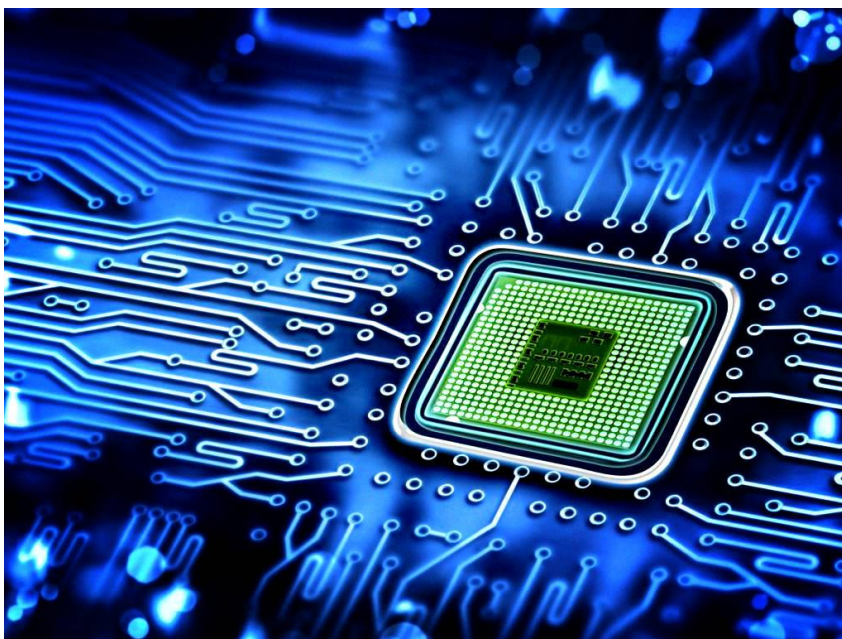


Figure 4-ASIC [chip.jpg \(1024×768\) \(bp.blogspot.com\)](#)

According to (Smith, 2016), “An ASIC, or application-specific integrated circuit, is a microchip designed for a special application, such as a kind of transmission protocol or a hand-held computer. You might contrast an ASIC with general integrated circuits, such as the microprocessor or random-access memory chips in your PC.”

There are two primary design methods for an ASIC, these are gate array and full custom design. The gate array designs the engineering costs are much lower due to little design work that is needed to make a working chip. Production is also much shorter because the process of metallization is fast in comparison to full-custom design. A full custom design is more complex than a gate array but allows the chip to do far more than the gate-array. (2016)

The benefit of using an ASIC is its far smaller than an FPGA, reducing the overall power consumption, the ASIC contains only the circuitry needed for the application, which in result makes the chip much more efficient due to the miniature size and power requirements. (2016)

### Sustainability

In an ever-increasing world population, the demand for electricity also increases, so it is important to think of the sustainability of new electrical systems by implementing ways to reduce total energy expenditure and work towards a net zero. The main power demand will be the lights as they remain illuminated all the time. Many traffic lights today use 67-watt bulbs to produce light which can be improved by using LEDs (light-emitting diodes). LEDs use much less energy than incandescent bulbs because diode light is much more efficient, power-wise than filament light. (Arcadia, 2020) They give off a good amount of light whilst outputting less power.

More and more are being done to reach carbon net zero targets set out by governments and this requires power to be generated by cleaner methods than burning fossil fuels. Solar panels can be installed locally or at a distance, however, locally will minimize cable distance resulting in lower costs and groundwork requirements. Another method is to install mini wind turbines which can be placed on top of the traffic light pole. It is recommended to adapt both technologies to generate power for the traffic lights, for the extra cost of installing them will in-time cover the cost.

### 3. Design method, choices and calculations (500)

To achieve a system that can be used to control a pedestrian crossing such as a pelican crossing a working digital circuit must be created. This is done by a finite state machine that will control the sequencing of the pelican crossing including the manually operated button that can be pressed by pedestrians. To achieve this, 3 circuits are built using MultiSim and incorporated together in one final circuit.

#### 5 Second Counter

The specification of this circuit is to:

- Count 0-5 (non-repetitive)
- Start when B=1
- Output W=1 when count =5

#### State diagram

A state diagram is designed by forming the initial state (the starting point) also known as a pseudo-state. The state is represented by the large circle containing the state number and its binary digit as seen in the below figure. Accompanying the states are transitions, these transitions are a change from one state to another and are represented by the arrows joining the states together. The action that causes the transition is written next to the arrows.

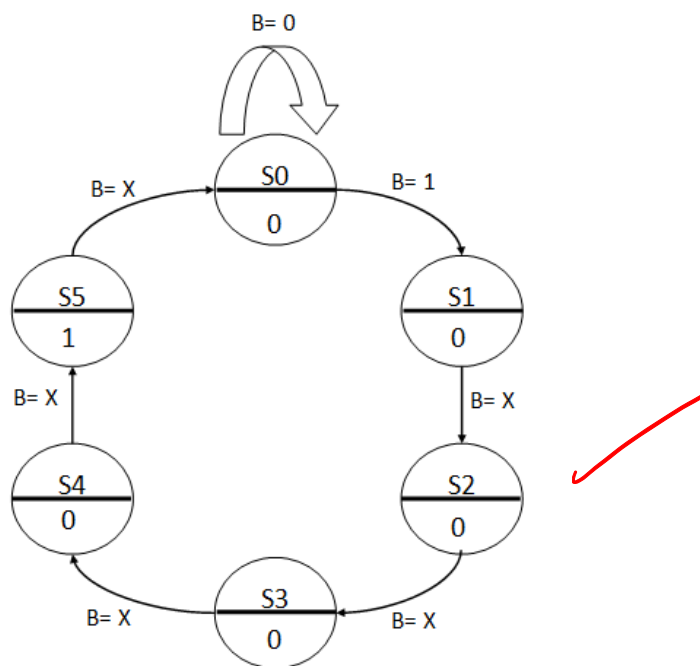


Figure 5-5 second state diagram

Input B is the representation of a physical button to initiate the traffic light sequence so that the light will turn red indicating road users to come to a stop and allow pedestrians to cross. As can be seen in the above diagram when at state 0 and the button is pressed making B = 1 it will transition to the next state, until it reaches state 5 which outputs to W (wait command) to the RAG (red, amber & green) circuit to start the lighting sequence.

## State Table

A state table was made to produce the binary code that will be used to gather the Boolean equations by using Karnaugh maps.

| Current State | Next State |     | Current |    |    | Next |     |     |     |     |     | Output |   |
|---------------|------------|-----|---------|----|----|------|-----|-----|-----|-----|-----|--------|---|
|               |            |     |         |    |    | B=0  |     |     | B=1 |     |     |        |   |
|               | B=0        | B=1 | QA      | QB | QC | QA+  | QB+ | QC+ | QA+ | QB+ | QC+ |        | W |
| S0            | S0         | S1  | 0       | 0  | 0  | 0    | 0   | 0   | 0   | 0   | 0   | 1      | 0 |
| S1            | S2         | S2  | 0       | 0  | 1  | 0    | 1   | 0   | 0   | 0   | 1   | 0      | 0 |
| S2            | S3         | S3  | 0       | 1  | 0  | 0    | 1   | 1   | 0   | 1   | 1   | 1      | 0 |
| S3            | S4         | S4  | 0       | 1  | 1  | 1    | 0   | 0   | 1   | 0   | 0   | 0      | 0 |
| S4            | S5         | S5  | 1       | 0  | 0  | 1    | 0   | 1   | 1   | 0   | 0   | 1      | 0 |
| S5            | S0         | S0  | 1       | 0  | 1  | 0    | 0   | 0   | 0   | 0   | 0   | 0      | 1 |
| S6            | X          | X   | 1       | 1  | 0  | X    | X   | X   | X   | X   | X   | X      | X |
| S7            | X          | X   | 1       | 1  | 1  | X    | X   | X   | X   | X   | X   | X      | X |

Figure 6-State Table

## Karnaugh Maps & Boolean Equations

Karnaugh maps were used to simplify the digital inputs laid out in the state table by forming groups with the binary code as seen in the figures below and forming Boolean equations from the groups. The purpose of this is to create smallest circuits possible for more efficiency and less complexity. If you were to derive the equations from the main state table, they would be far bigger and requiring lots of extra gates in the circuit.

| BQA \ QBQC |    |    |      |      |
|------------|----|----|------|------|
|            | 00 | 01 | 11   | 10   |
| 00         | 0  | 0  | 1    | 0    |
| 01         | 1  | 0  | X(1) | X(1) |
| 11         | 1  | 0  | X(1) | X(1) |
| 10         | 0  | 0  | 1    | 0    |

$$QA+ = QBQC + QANOTQC$$

| BQA \ QBQC |    |    |    |      |
|------------|----|----|----|------|
|            | 00 | 01 | 11 | 10   |
| 00         | 0  | 1  | 0  | 1    |
| 01         | 0  | 0  | X  | X(1) |
| 11         | 0  | 0  | X  | X(1) |
| 10         | 0  | 1  | 0  | 1    |

$$QB+ = NOTQANOTQBQC + QBNOTQC$$

| QBQC |   | QBQA |    |      |    |
|------|---|------|----|------|----|
|      |   | 00   | 01 | 11   | 10 |
| 00   | 0 | 0    | 0  | 1    |    |
| 01   | 1 | 0    | X  | X(1) |    |
| 11   | 1 | 0    | X  | X(1) |    |
| 10   | 1 | 0    | 0  | 1    |    |

$QC+ = QBNOTQC + QANOTQC + BNOTQC$

| QBQC |   | QBQA |      |    |    |
|------|---|------|------|----|----|
|      |   | 00   | 01   | 11 | 10 |
| 0    | 0 | 0    | 0    | 0  |    |
| 1    | 0 | 1    | X(1) | X  |    |

$W = QAQC$

Good.

Figure 7-5 seconds Karnaugh Maps

### Multi-sim Design

With the Boolean equations established, a circuit can be made to perform the operation for the 5 second counter. The circuits are built on MultiSim and are shown in the figures below. Logic gates AND & OR gates were used to create the Boolean equations found through the Karnaugh maps. Three D-type flip-flops were used for its ability to latch and store data as a way of creating internal memory and create a delay in the progress of data. A clock (CLK) is connected to the flip-flops to operate at a certain pace.

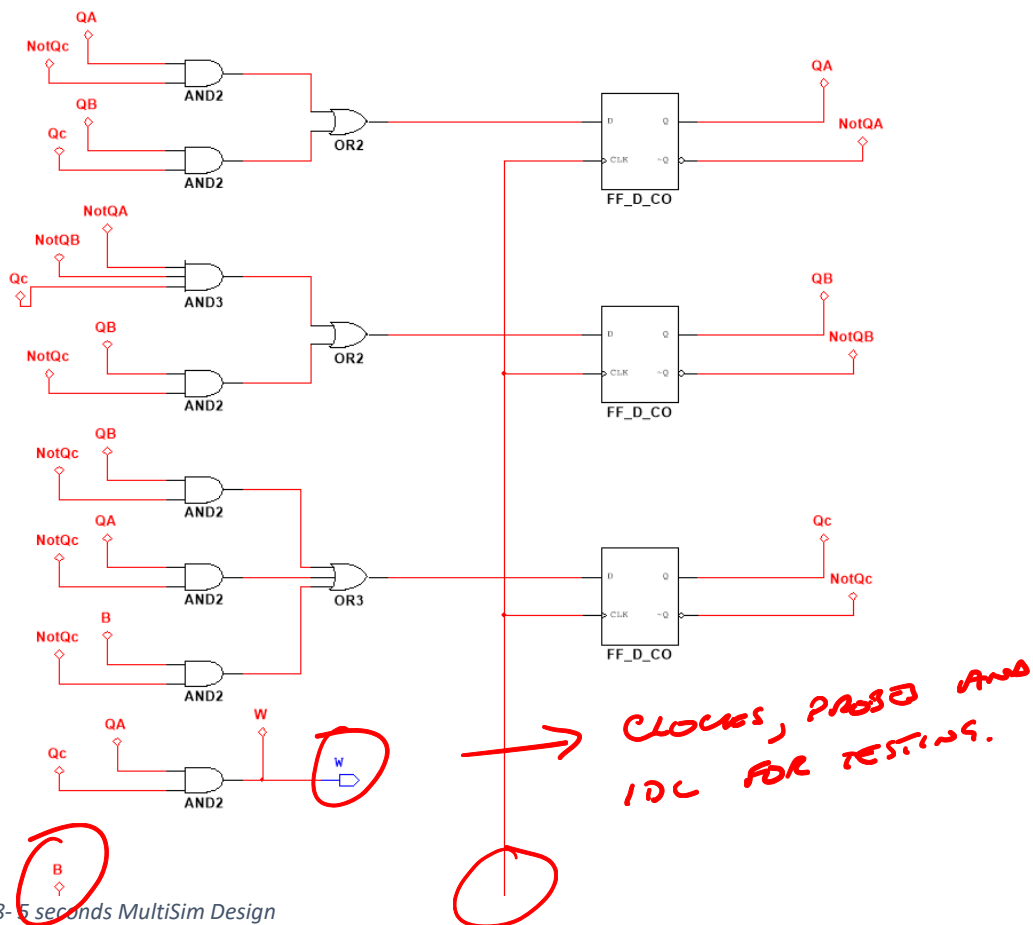


Figure 8-5 seconds MultiSim Design

## 15 Second Counter

### Specification

The specification for this design is to create a counter that will count to 15 seconds and send an output (T) to the RAG circuit to initiate the traffic light sequence so that the traffic lights will green to indicate pedestrians to stop and allowing the road users to continue.

### State Diagram

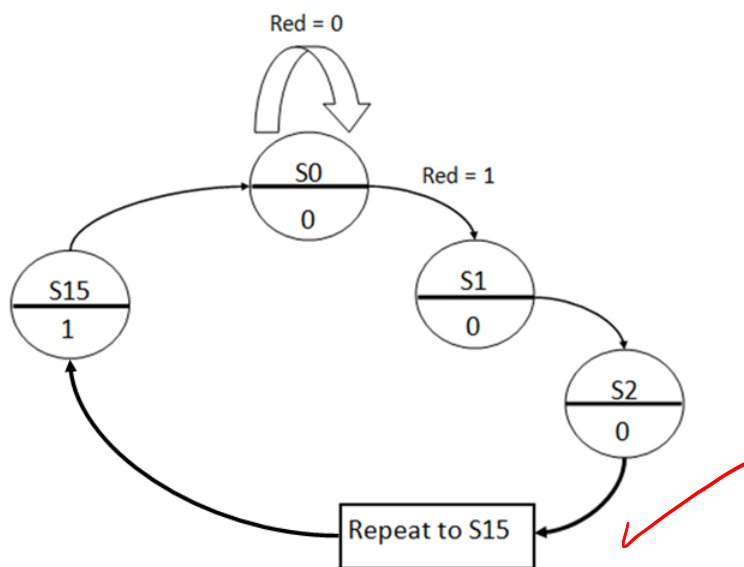


Figure 9-15 seconds state diagram

16 states in total are needed to count from 0 to 15 seconds. At state 0 when red = 0 it will stay where it is, but when red = 1 the sequencing will operate and state 0 will transition to state 1 and continue transitioning to each state and when it reaches state 15 (15 seconds), will output T = 1 to the RAG circuit as shown in the figure above.

### State Table

| Current State | Next State |     | Current |    |    |    | Next |     |     |     |     |     |     |     | Output |   |
|---------------|------------|-----|---------|----|----|----|------|-----|-----|-----|-----|-----|-----|-----|--------|---|
|               |            |     |         |    |    |    | R=0  |     |     |     | R=1 |     |     |     |        |   |
|               | R=0        | R=1 | QA      | QB | QC | QD | QA+  | QB+ | QC+ | QD+ | QA+ | QB+ | QC+ | QD+ |        | T |
| S0            | S0         | S1  | 0       | 0  | 0  | 0  | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1      | 0 |
| S1            | S2         | S2  | 0       | 0  | 0  | 1  | 0    | 0   | 1   | 0   | 0   | 0   | 1   | 0   | 0      | 0 |
| S2            | S3         | S3  | 0       | 0  | 1  | 0  | 0    | 0   | 1   | 1   | 0   | 0   | 1   | 1   | 0      | 0 |
| S3            | S4         | S4  | 0       | 0  | 1  | 1  | 0    | 1   | 0   | 0   | 0   | 1   | 0   | 0   | 0      | 0 |
| S4            | S5         | S5  | 0       | 1  | 0  | 0  | 0    | 1   | 0   | 1   | 0   | 1   | 0   | 1   | 0      | 0 |
| S5            | S6         | S6  | 0       | 1  | 0  | 1  | 0    | 1   | 1   | 0   | 0   | 1   | 1   | 0   | 0      | 0 |
| S6            | S7         | S7  | 0       | 1  | 1  | 0  | 0    | 1   | 1   | 1   | 0   | 1   | 1   | 1   | 0      | 0 |
| S7            | S8         | S8  | 0       | 1  | 1  | 1  | 1    | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0      | 0 |
| S8            | S9         | S9  | 1       | 0  | 0  | 0  | 1    | 0   | 0   | 1   | 1   | 0   | 0   | 1   | 0      | 0 |
| S9            | S10        | S10 | 1       | 0  | 0  | 1  | 1    | 0   | 1   | 0   | 1   | 0   | 1   | 0   | 0      | 0 |
| S10           | S11        | S11 | 1       | 0  | 1  | 0  | 1    | 0   | 1   | 1   | 1   | 0   | 1   | 1   | 0      | 0 |
| S11           | S12        | S12 | 1       | 0  | 1  | 1  | 1    | 1   | 0   | 0   | 1   | 1   | 0   | 0   | 0      | 0 |
| S12           | S13        | S13 | 1       | 1  | 0  | 0  | 1    | 1   | 0   | 1   | 1   | 1   | 0   | 1   | 0      | 0 |
| S13           | S14        | S14 | 1       | 1  | 0  | 1  | 1    | 1   | 1   | 0   | 1   | 1   | 1   | 0   | 0      | 0 |
| S14           | S15        | S15 | 1       | 1  | 1  | 0  | 1    | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 0      | 0 |
| S15           | S0         | S0  | 1       | 1  | 1  | 1  | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0      | 1 |

Figure 10-15 seconds state table

The main difference to this table to the 5 second is that it has 16 states to count from 0-15 seconds. In addition to this there is 5-variable inputs which creates slight complexity when doing the Karnaugh maps which is detailed further in the report.

### Karnaugh Maps & Boolean Equations

| R = 0                                 |  |  |  |  | R = 1                                 |  |  |  |  |
|---------------------------------------|--|--|--|--|---------------------------------------|--|--|--|--|
|                                       |  |  |  |  |                                       |  |  |  |  |
| QAQB \ QBQC                           |  |  |  |  | QAQB \ QBQC                           |  |  |  |  |
| 00                                    |  |  |  |  | 00                                    |  |  |  |  |
| 01                                    |  |  |  |  | 01                                    |  |  |  |  |
| 11                                    |  |  |  |  | 11                                    |  |  |  |  |
| 10                                    |  |  |  |  | 10                                    |  |  |  |  |
| QA+ = ANOTB + ANOTC + ANOTD + NOTABCD |  |  |  |  | QA+ = ANOTB + ANOTC + ANOTD + NOTABCD |  |  |  |  |
| QB+ = BNOTC + BNOTD + NOTBCD          |  |  |  |  | QB+ = BNOTC + BNOTD + NOTBCD          |  |  |  |  |
| QC+ = NOTCD + CNOTD                   |  |  |  |  | QC+ = NOTCD + CNOTD                   |  |  |  |  |
| QD+ = CNOTD + BNOTD + ANOTD           |  |  |  |  | QD+ = NOTD                            |  |  |  |  |
| QD+ = CNOTD + BNOTD + ANOTD + NOTD    |  |  |  |  |                                       |  |  |  |  |

Figure 11-15 seconds Karnaugh maps

The reason there is a 2<sup>nd</sup> set of Karnaugh maps is that there are 5 variable inputs instead of 4 as in the 5 second counter. It is not possible to use a 5-variable input Karnaugh map so a set had to be developed for when R = 0 and when R = 1. The only duplicated Karnaugh map that has changing states is QD+, therefore the expression for QD+ is NOTD and is added to the final equation as can be seen at the bottom of the above figure.

| QAQB \ QBQC | 00 | 01 | 11 | 10 |
|-------------|----|----|----|----|
| 00          | 0  | 0  | 0  | 0  |
| 01          | 0  | 0  | 0  | 0  |
| 11          | 0  | 0  | 1  | 0  |
| 10          | 0  | 0  | 0  | 0  |

$T = QAQBQCQD$

Figure 12-Karnaugh Map for output T

### MultiSim Design

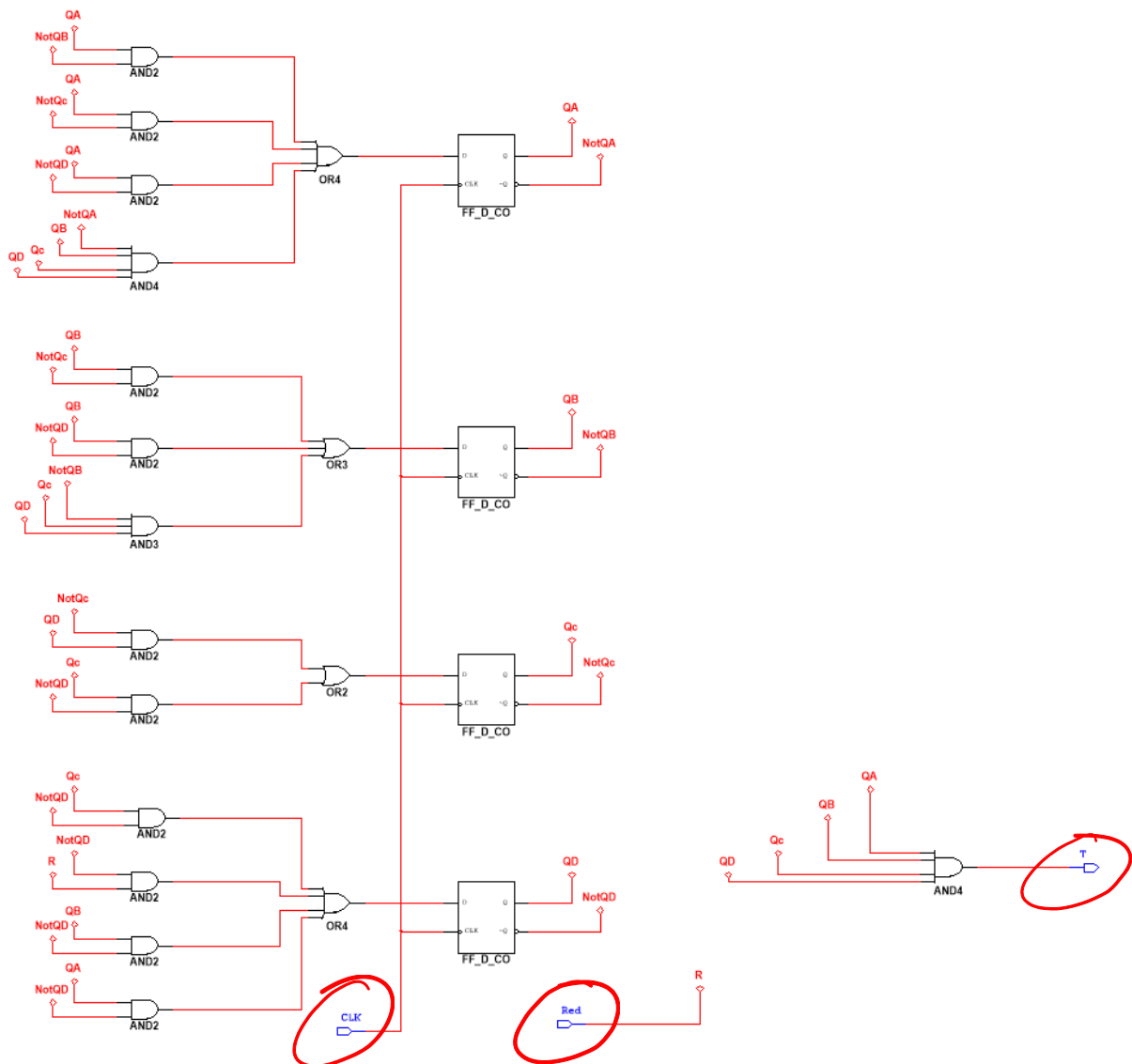


Figure 13-15 seconds MultiSim design

Four flip-flops were used in this circuit as there is more than 8 states.

## RAG Design

### Specification

To illuminate the LED lights (traffic lights) in a determined sequence to stop and control traffic to allow pedestrians to cross. In the real world, green will predominately be illuminated so this will be state 0, the next step is to switch to amber for 2 seconds (this will be state 1 and state 2). Next step is to turn red to stop traffic (state 3) and allow pedestrians to cross. State 4 will bring in amber as well as red for 2 seconds (states 4 and 5) and then back to green (state 0).

### State Diagram

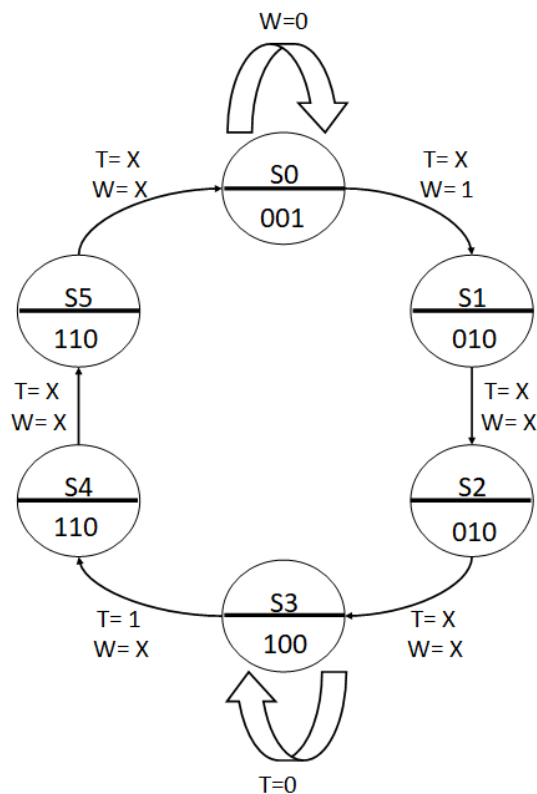


Figure 14-RAG State diagram

### State Table

| Present State | Next State |     |     |     | Next states |     |     |     |     |     |     |     |     |     |     |     | Outputs |   |   |
|---------------|------------|-----|-----|-----|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---------|---|---|
|               |            |     |     |     | W=0         |     |     |     |     |     | W=1 |     |     |     |     |     |         |   |   |
|               | W=0        |     | W=1 |     | T=0         |     |     | T=1 |     |     | T=0 |     |     | T=1 |     |     |         |   |   |
|               | T=0        | T=1 | T=0 | T=1 | QA+         | QB+ | QC+ | QA+ | QB+ | QC+ | QA+ | QB+ | QC+ | QA+ | QB+ | QC+ | R       | A | G |
| (Green) S0    | S0         | S0  | S1  | S1  | 0           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 1   | 0       | 0 | 1 |
| (Amber) S1    | S2         | S2  | S2  | S2  | 0           | 1   | 0   | 0   | 1   | 0   | 0   | 1   | 0   | 0   | 1   | 0   | 0       | 1 | 0 |
| (Amber) S1    | S3         | S3  | S3  | S3  | 0           | 1   | 1   | 0   | 1   | 1   | 0   | 1   | 1   | 0   | 1   | 1   | 0       | 1 | 0 |
| (Red) S3      | S3         | S4  | S3  | S4  | 0           | 1   | 1   | 1   | 0   | 0   | 0   | 1   | 1   | 1   | 0   | 0   | 1       | 0 | 0 |
| (R+A) S4      | S5         | S5  | S5  | S5  | 1           | 0   | 1   | 1   | 0   | 1   | 1   | 0   | 1   | 1   | 0   | 1   | 1       | 1 | 0 |
| (R+A) S5      | S0         | S0  | S0  | S0  | 0           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1       | 1 | 0 |
| S6            | X          | X   | X   | X   | X           | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X       | X | X |
| S7            | X          | X   | X   | X   | X           | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X       | X | X |

Figure 15-RAG State table

## Karnaugh Maps & Boolean Equations

Same again as the 15-second design, two simultaneous Karnaugh maps had to be created because of the 5-variable inputs.

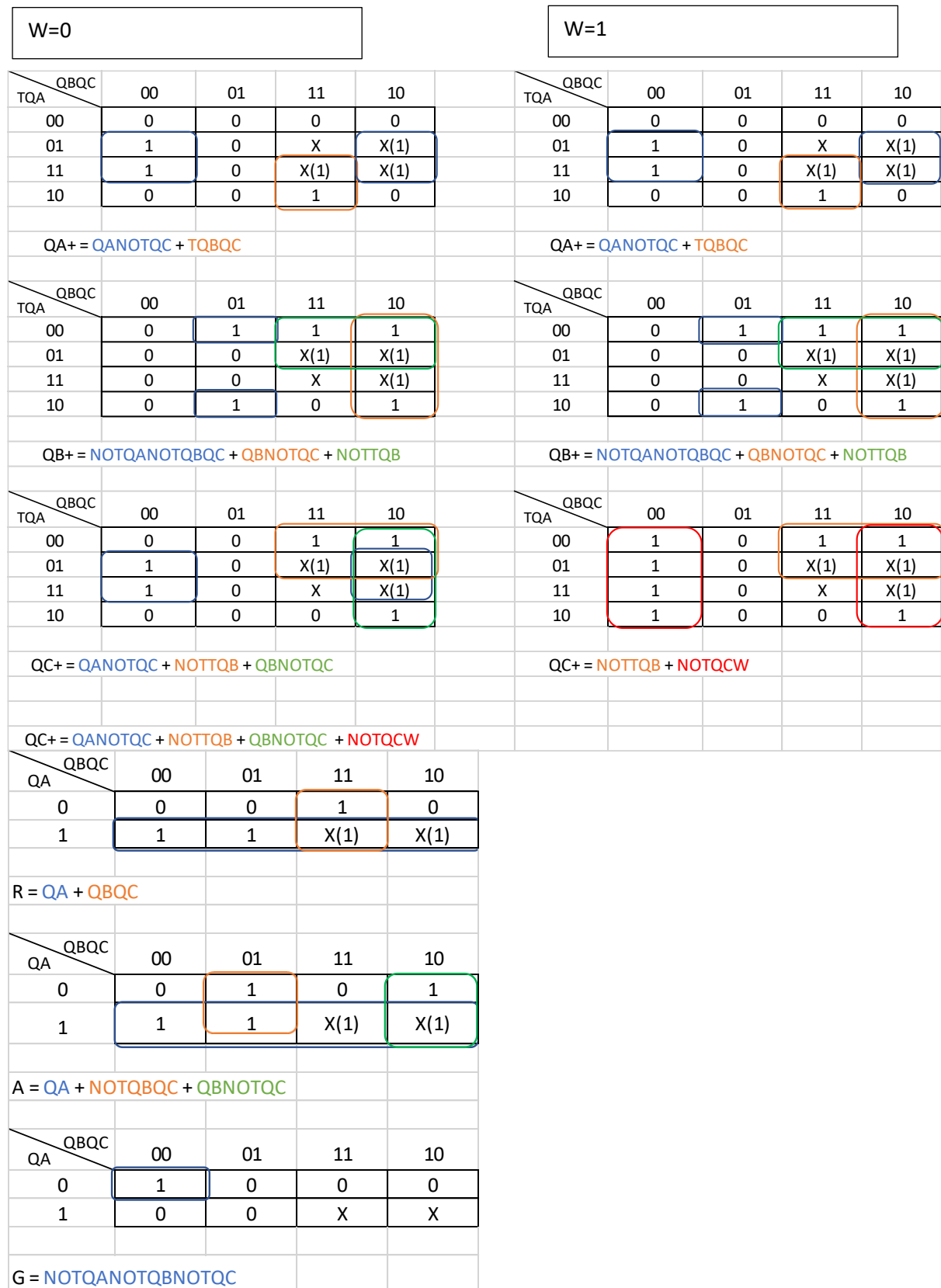


Figure 16-RAG Karnaugh Maps

## MultiSim Design

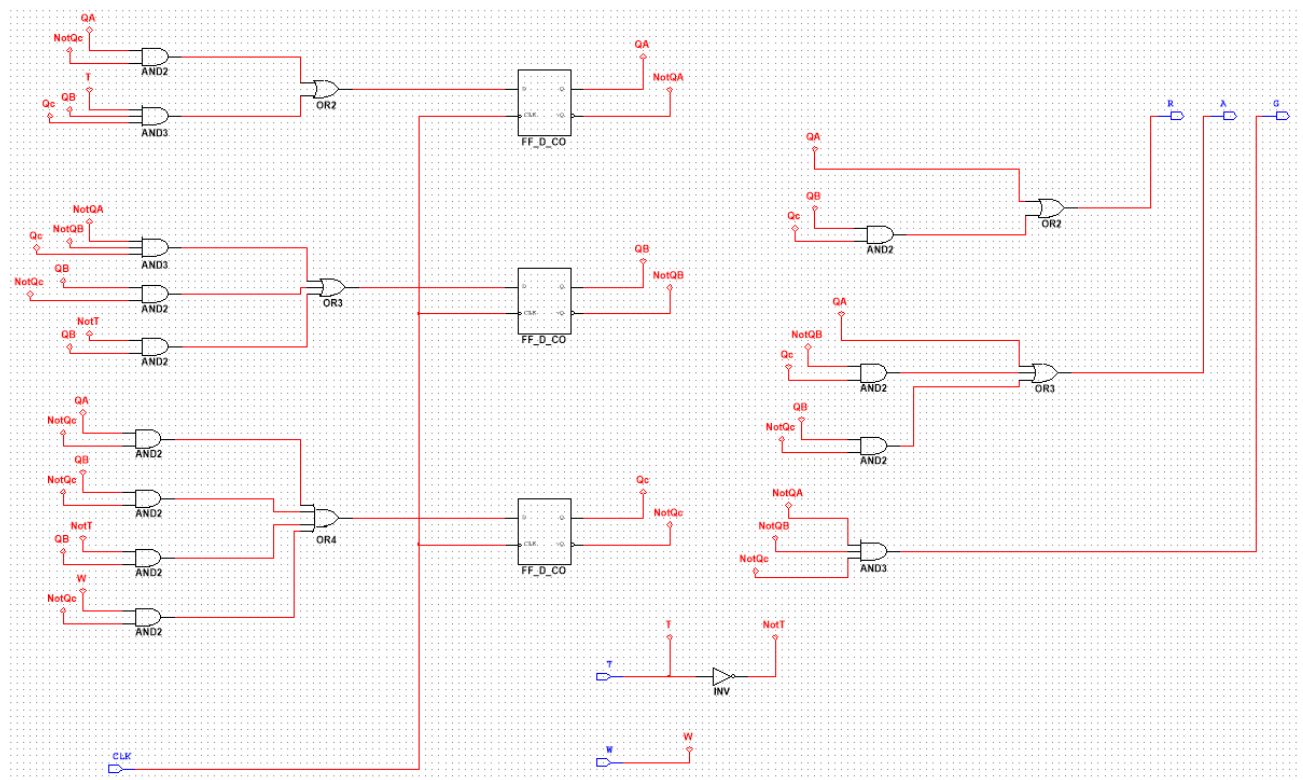


Figure 17-RAG MultiSim Design

The inputs for T and W come from on-page connectors that go high when it reaches the pre-determined condition set out in the previous sub-circuits.

## Final Design

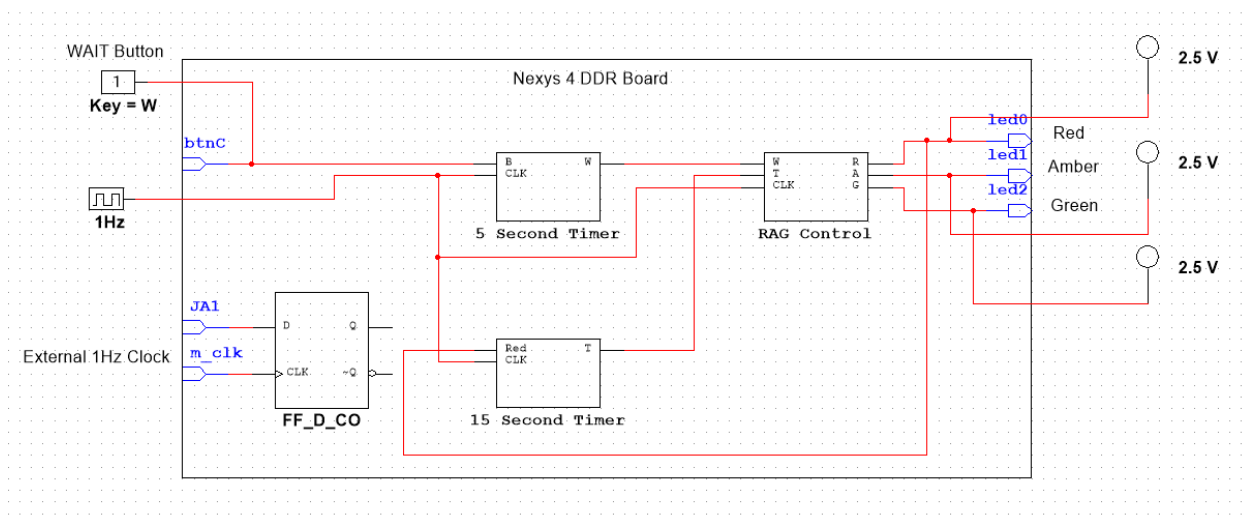


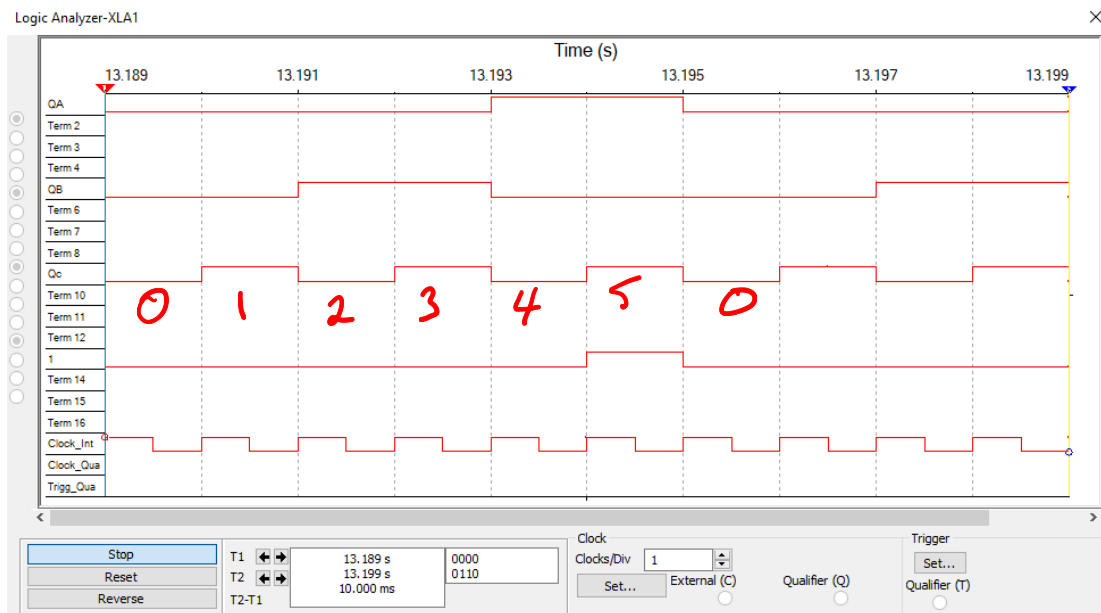
Figure 18-Final Design

The final design incorporates the previous circuits together and works in harmony to produce a working traffic light system. The flip-flop in the figure above is not used and can be deleted in further work.

#### 4. Testing, results and analysis (500)

The testing of the PLD circuits was carried out on MultiSim using logic analysers connected to the outputs.

### 5 second counter Logic analyser results

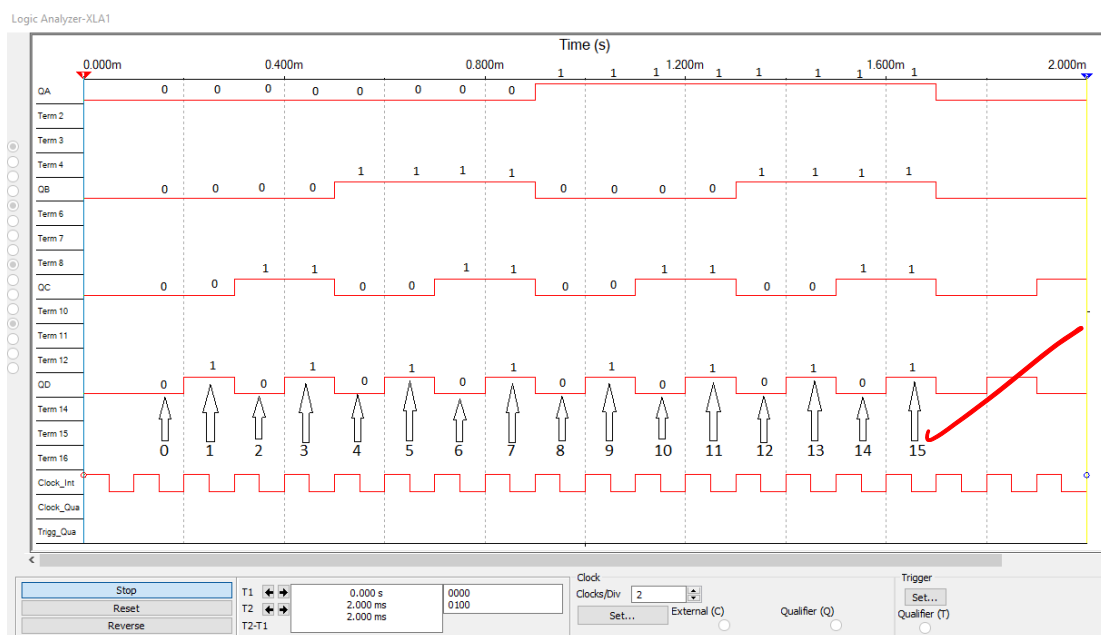


ANNOTATE OR  
USE CURSOR  
TO EXPLAIN  
THE GRAPH.

Figure 19-5 second counter results

The specification for this circuit is to count from 0-5. It can be seen in the figure above that the outputs go high in a sequence. The graphical results above show the circuit counting from 0-5.

15 second counter logic analyser results.



Good.

Figure 20-15 second counter results

As can be seen in the figure above, the outputs go high in the correct order forming a count from 0 – 15 seconds which is annotated.

## RAG Circuit Logic Analyser Results

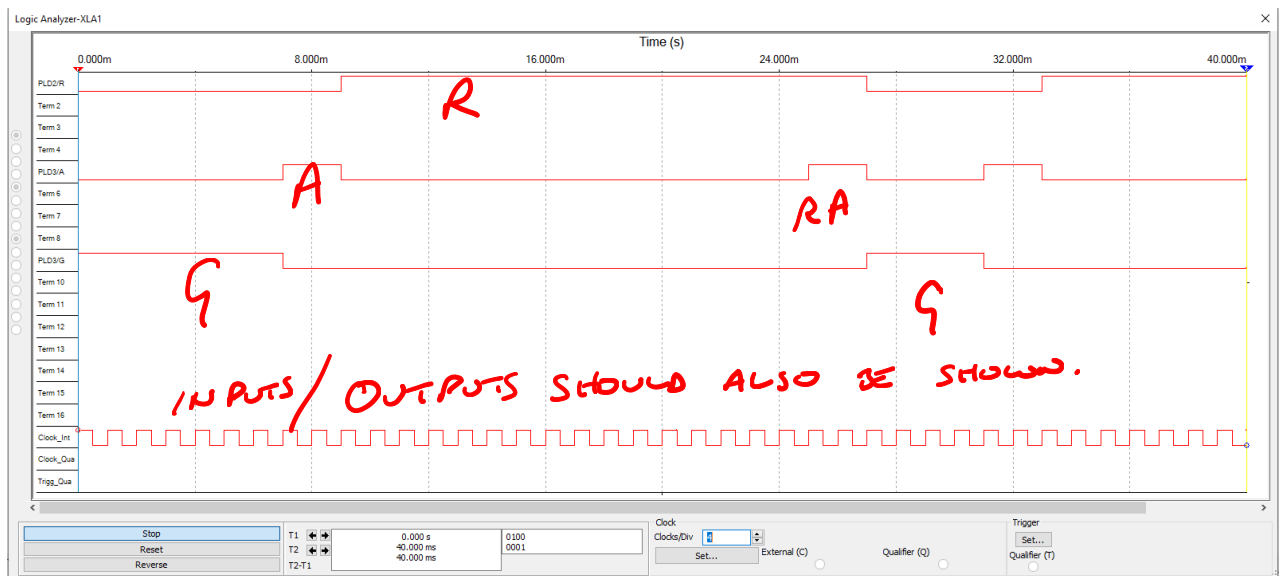


Figure 21-RAG results

The above image shows the Red, Amber & Green LEDs switching on and off corresponding to the order set out in the specification and the state diagram. The timings also correspond correctly to the 5- and 15-seconds counters.

## 5. Conclusion and recommendations (500)

The results from testing the PLD circuits prove the theory and methods such as truth tables, Karnaugh maps, Boolean equations, etc used in the design stages of the circuits correct, and achieves the specifications set out in the assignment brief. Creating and using a finite state machine using bistable gates (D-type flip-flop) to control a set of traffic lights proved successful. The results from the 0-5 counter shows that when a button is pressed a count will start from 0-5 seconds and send an output (W) to the RAG controller to initiate the next step of the sequence. The results can be seen by following the order in when the output go high. For example, when Qc goes high the binary digits are 0/0/1 which equates to 1 second. Next, Qc goes low and QB goes high which equates to 2 seconds, and so on. These results translate over to the 0-15 second counter where it can be seen on the graph the order in which the outputs go high. The RAG circuit behaved exactly as expected and meshes the counter circuits together to control the timing and operation of the lights.

It can be concluded that the circuit can be used and implemented in systems where traffic is controlled and operated by a set of traffic lights to allow pedestrians to cross. The circuit can be burnt onto a chip ready to be installed on a circuit board for a customer. Recommendations can be made to improve the circuit to operate more like a puffin crossing as detailed in the research. It utilizes sensors such as a PCD that can increase the length of time allowed for pedestrians to cross and a PKD which can detect pedestrian presence which can cancel the sequence if the button is falsely pressed.

If it was to be implemented in the real world it is recommended to produce a self-sustaining traffic crossing by using clean renewable energy such as solar and wind power to generate electricity supply. This is done by mounting solar panels and mini wind turbines locally. This provides a reliable source of energy to operate the traffic lights because of advances in technology making equipment a lot more efficient and requiring less power. All this reduces stress placed on the electricity grid and aims to reach carbon net-zero. Initial costs would be slightly higher but in time the free generation of electricity will pay for itself resulting in zero costs for electricity.

It is recommended to use a FPGA in the development stages. Because a FPGA can be re-programmed, it can be used to optimize and tweak the program to operate more efficient or to supplement new changes being incorporated. A FPGA can be used in the final stages of manufacture but tend to be expensive if on a mass scale which is not necessary. Also, because a FPGA can be read and uploaded to, it poses a security risk from hackers stealing intellectual property stored in the memory, so it is not advised to use. Instead, an ASIC should be used where the logic developed on the FPGA is burnt onto a chip where it is stored and cannot be altered. This makes it impossible for hackers to steal or manipulate the circuit. It is also much more economical to use ASICs as they take up less room, uses less power and cost less to manufacture.

## 6. Reference list and bibliography

- Arcadia, 2020. *LED vs. regular lightbulbs: Do they really make a difference?*. [Online]  
Available at: <https://blog.arcadia.com/led-vs-regular-lightbulbs-do-they-really-make-a-difference/>  
[Accessed 16 01 2022].
- Arrow, 2018. *FPGA Basics: Architecture, Applications and Uses*. [Online]  
Available at: <https://www.arrow.com/en/research-and-events/articles/fpga-basics-architecture-applications-and-uses>  
[Accessed 16 01 2022].
- Council, D. B., 2009. *Puffin Crossings Good Practice Guide Release 1*, Darlington: Darlington Borough Council.
- RAC, 2020. *Road crossings - what's the difference between zebra, pelican, puffin and toucan crossings?*. [Online]  
Available at: <https://www.rac.co.uk/drive/advice/learning-to-drive/road-crossings/>  
[Accessed 09 01 2021].
- Shale-Hester, T., 2020. *What is a Puffin crossing?*. [Online]  
Available at: <https://www.autoexpress.co.uk/car-news/105227/what-is-a-puffin-crossing>  
[Accessed 09 01 2022].
- Smith, S., 2016. *What is an ASIC, and why is everyone using them?*. [Online]  
Available at: <https://www.sigenics.com/blog/what-is-an-asic>  
[Accessed 16 01 2022].
- Traffic Choices, 2014. *PELICAN CROSSING*. [Online]  
Available at: <https://www.trafficchoices.co.uk/traffic-schemes/pelican-crossing.shtml>  
[Accessed 09 01 2022].

FINAL GRADE

GENERAL COMMENTS

Instructor

60/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

GRADING FORM: CSG-RUBIC

**JASON BOOTH****60****RESEARCH-INFORMED LITERATURE**

There was some evidence of further reading and research and your interpretation and understanding of the additional research was OK. The sources referenced were insufficient at this level given the amount of potential research areas for this assignment and relied on a small selection of mainly inappropriate and non-authoritative sources. You also need to include more in-text citations using the Harvard reference system.

**3****KNOWLEDGE AND UNDERSTANDING**

The knowledge and understanding of the subject demonstrated by this assignment was reasonably detailed and mainly accurate. There were very few gaps in your knowledge of the basic concepts and your understanding of the higher level concepts was demonstrated to a limited degree - you could

**19**

have explained in more detail how the circuit evolved. Further work – Your explanations need to be more detailed and written with a greater emphasis on critical evaluation rather than description.

## ANALYSIS



Your analysis of the task was generally satisfactory, both in class and as shown in this assignment, however it did tend more towards descriptive rather than analytical. You have demonstrated some analytical skills and your approach to the task was logical. You wrote a short conclusion that demonstrated some skills in analysing quantitative and qualitative data, however this was generally basic in nature. Further work – in your conclusion, you should relate the results to the original specification and to accepted theory and provide a detailed critical analysis of this.

**17**

## PRACTICAL APPLICATION AND DEP.



Your application of practical skills for this task was generally good. You used the resources available appropriately when designing the circuit and you used appropriate options for testing and evaluating the circuit performance. You managed to design an efficient circuit that functioned within the specification and achieved this by only asking for assistance when appropriate. Your results were comprehensive and allowed for a good conclusion to be attempted.

**14**

## SKILLS FOR PROFESSIONAL PRAC.



In this assignment you have demonstrated skills in working individually and also collaboratively. Your written report was generally good and contained most of the required detail, it was also written in a coherent and logical way. The available word count was in the main effectively used.

**7**