

2017198

by Ryan James

Submission date: 18-Jan-2022 11:36AM (UTC+0000)

Submission ID: 169452586

File name: 16861_Ryan_James_2017198_179500_274701514.pdf (1.16M)

Word count: 5688

Character count: 21825

Course: HNC Electrical and Electronic Engineering
Module: SGEE5000 – Electronics II
Assignment 1: Finite State Machines
Name: Ryan James
University number: 2017198
Date submitted: 18/01/2022
Word count: 2660

- ① RESEARCH WOULD BENEFIT FROM DIAGRAMS.
- ② DESIGN METHOD WAS EXPLAINED IN DETAIL.
- ③ TESTING WAS CARRIED OUT BUT LOGIC ANALYSER WASN'T ALWAYS SET UP CORRECTLY.
- ④ EXCESSIVE DIAGRAMS/CIRCUIT IN THE TEXT, SOME SHOULD BE IN THE APPENDIX.
- ⑤ MODIFICATION WAS SUCCESSFULLY IMPLEMENTED
- ⑥ VALID CONCLUSIONS WITH SUGGESTIONS FOR SUSTAINABLE IMPLEMENTATION.

Contents

1. Executive Summary.....	3
2. Research into advanced digital design techniques.....	4
3. Design method, choices, and calculations.....	7
4. Testing, results, and analysis.....	36
5. Conclusions and recommendations.....	37
6. Works Cited.....	38
Appendix	39

1. Executive Summary *PAST TENSE.*

The following report will contain research that will improve understanding of the tasks required to complete, but also aid the explanations in the analysis and the final conclusion. Research should contain references to UK pedestrian crossings, technologies used in traffic light control systems and finally electronic devices used to promote sustainable environments with accurate references throughout.

A Pelican crossing should be designed and created with use of a finite state machine (FSM), in order to control the flow of traffic at a set of lights. The pedestrian should use a wait button in order to cross and use components such as internal clocks with timers. Once complete multisim should be used to test the system as specified to do so in the brief. Upon completion of this, the design should be adapted to create a Puffin crossing where it will detect when a pedestrian wants to cross the road via a sensor. It should be tested using PLD mode on multisim.

From the gathered results, appropriate analysis and conclusions are to be drawn with evidence of points accurately referenced. Testing can be shown in the report but where possible it should be in the appendix.

2. Research into advanced digital design techniques

Throughout the UK, there are various types of pedestrian crossings ranging from zebra crossings to wig wag signals at railway crossings. The most common types that UK pedestrians and road users will use on a day-to-day basis will usually come in the form of Pelican crossings and Puffin crossings.

A Pelican crossing is a type of crossing controlled by traffic lights which pedestrians request to cross the road by pressing a button and waiting for a green man figure to appear when it is safe to cross the road. However, the vehicles will have a red light appear to indicate pedestrians are crossing and therefore must stop. Upon the latter end of this the green man will begin to flash for pedestrians and also an amber light for vehicle users meaning that the drivers can only “go” when there are no more pedestrians crossing and to also indicate to pedestrians there is a limited amount of time to finish crossing before the drivers can continue their journey (TheoryTest.org.uk, 2021).

A Puffin crossing differs from a pelican crossing in a minor way; where the pelican crossing requires a button to be pushed in order for pedestrians to cross the road, the puffin crossing works by an electronic sensor detecting when a pedestrian is waiting to cross in order to change the lights. This type of crossing also detects when the pedestrian is crossing the road and it will hold the red light until they have crossed. However, the pedestrian crosses the road before the start sequence it will automatically cancel the sequence in order to prevent vehicles from stopping at a red light (Crown Copyright 2005, 2005).

There are a variety of components used in traffic light systems such as buttons, clocks, and LEDs, however there are also components in the form FPGA (Field Programmable GATE Array) and ASIC (Application Specific Integrated Circuit).

An FPGA can be described as a set of digital sub- circuits that can be programmed to specific applications ranging from traffic light systems to military applications. The main difference between an FPGA and an ASIC chip is that where an FPGA can be re-programmed to a different application, the ASIC can't as it is custom made for specific tasks, thus making it more expensive (SDR NEWS, 2021).

A further advantage that FPGAs have is that they are programmed at logic level, which means it is faster at processing signals as opposed to ASICs. However, as previously stated ASICs are more expensive this is only short term. When the quantity of FPGA chips increase so does the cost therefore make it a more expensive option on large orders (RF Wireless World 2012, 2012).

ASICs are normally used where a mass production of components are required for example it mobile phones. This is because they have the ability to be job specific which means the overall design can be contained in 1 integrated circuit, meaning less components required (electronics-notes.com, 2022).

There are modern day technologies that are used to control traffic circuits like solar power and wind powered road signs. Solar powered road signs work by use of photovoltaic technology which converts solar energy from the sun into electricity, which can be used straight away to power equipment or used to recharge batteries. Major advantages of using this type of energy is that it doesn't contribute to greenhouse gasses as it is a renewable energy source but also it can work independently meaning it doesn't need to be connected to a power grid as it is deemed to always have power through the sun, however this isn't always the case as sometime trees or high buildings can block the sun from the solar panels thus leading to no power (John Bird, 2017).

Another type of energy used to power road signs is by the use of a miniature wind turbine. They work by having a wind turn the propeller blades which then in turn spins a miniature generator to produce electricity which powers the road signs. The generator of the wind turbine works when energy in the wind turns the propeller blades around a rotor, the pressure of the wind on a blade increase forcing it to move onto the next blade. This process continues resulting in the rotor spinning. The rotor is connected to the main shaft, which spins a generator to create electricity. Although the wind turbine is a form of renewable energy as it doesn't produce any greenhouse gasses it does have problem, as when there is no wind it wouldn't work thus it would need to be used in areas of a constant supply of wind (7 energy ltd 2022, 2022).

It can be seen that an example of modern day electrical components can be described as an LED (Light Emitting Diode). This type of diode is used in a variety of electronic devices like digital clocks, street lighting, television screens car lights and more commonly seen in traffic light systems and signs. A key reason why these smaller components are being chosen over incandescent lightbulbs is because they don't have filaments that could burn out which as a result increases life expectancy (HowStuffWorks, 2022).

These are becoming more common in road signs as well as traffic lights due to the brightness, speed of response, and long lasting life expectancy in comparison to the older incandescent light bulbs, where it was expected only to last 1000 hours to a much improved 80,000 hours (Sinowatcher technology Co, 2021).

Although the advantages outweigh the disadvantages for LEDs there are some drawbacks with one being the cost as they are initially more expensive, but the life expectancy is longer.

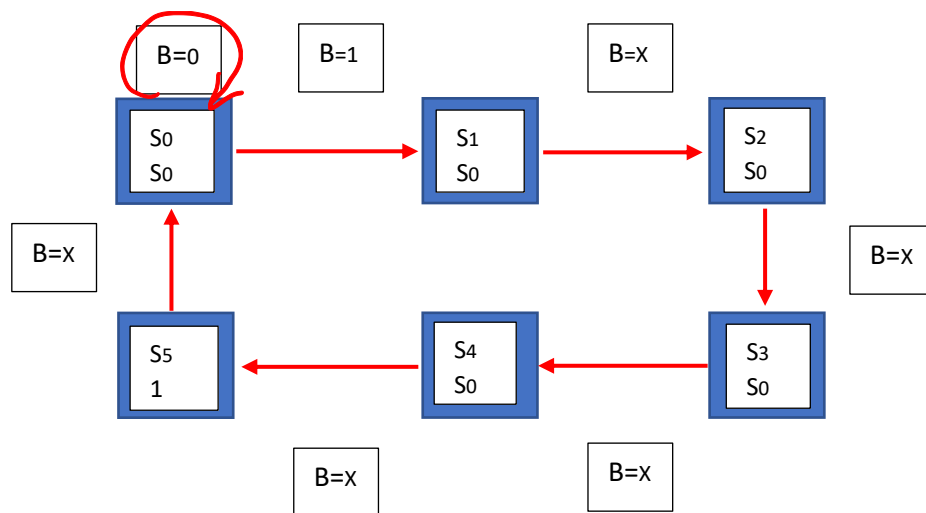
Another drawback is the sensitivity to temperature as they rely heavily on an optimal ambient temperature (Greenie World Poland, 2022).

WOULD BENEFIT FROM DIAGRAMS AND
HEADINGS.

3. Design method, choices, and calculations

For the traffic light system to be designed and created to control a pelican crossing, it had to combine circuits which included the following: 0-5 counter, 0-15 counter and finally a sequence RAG controller. In order to complete the circuits for the pelican crossing a design method had to be followed in the way of truth tables, Karnaugh maps, Boolean expressions and also state diagrams.

The specification of the 0-5 counter states that it needed to count from 0-5 but had to be non-repetitive, had to start counting when B=1 and also when the count reaches 5 the wait begins hence w=1. This can be explained visually by use of the state diagram.



KEY

- B= 0 it stays where the count is.
- B=1 it starts counting.
- B=x (don't care state) this means you will not have to keep your finger on the button and will count regardless of what state it is in.

Table 1: State table for 0-5 counter.

State Table 0-5 Counter												
Current State	Next State		Current			Next						Output (W)
	B=0	B=1				B=0			B=1			
			QA	QB	QC	QA+	QB+	QC+	QA+	QB+	QC+	
S0	S0	S1	0	0	0	0	0	0	0	0	1	0
S1	S2	S2	0	0	1	0	1	0	0	1	0	0
S2	S3	S3	0	1	0	0	1	1	0	1	1	0
S3	S4	S4	0	1	1	1	0	0	1	0	0	0
S4	S5	S5	1	0	0	1	0	1	1	0	1	0
S5	S0	S0	1	0	1	0	0	0	0	0	0	1
S6	X	X	1	1	0	X	X	X	X	X	X	X
S7	X	X	1	1	1	X	X	X	X	X	X	X

Table 2: Karnaugh Map QA+.

QB QC \ QA	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

$$QA^+ = QA \text{ not } QC + QBQC$$

Table 3: Karnaugh Map QB+.

QB QC \ QA	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

$$QB^+ = \text{not } QA \text{ not } QB \text{ } QC + QB \text{ not } QC$$

THE COLUMNS HERE ARE JUST A TEACHING AID.

Table 4: Karnaugh Map Q_C^+ .

QB QC \ QA	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

$$Q_C^+ = Q_B \text{ not } Q_C + Q_A \text{ not } Q_C + B \text{ not } Q_C$$

Table 5: Karnaugh Map W .

QC QD \ QA	00	01	11	10
0	0	0	0	0
1	0	1	X ¹	X

$$W = Q_A Q_C$$

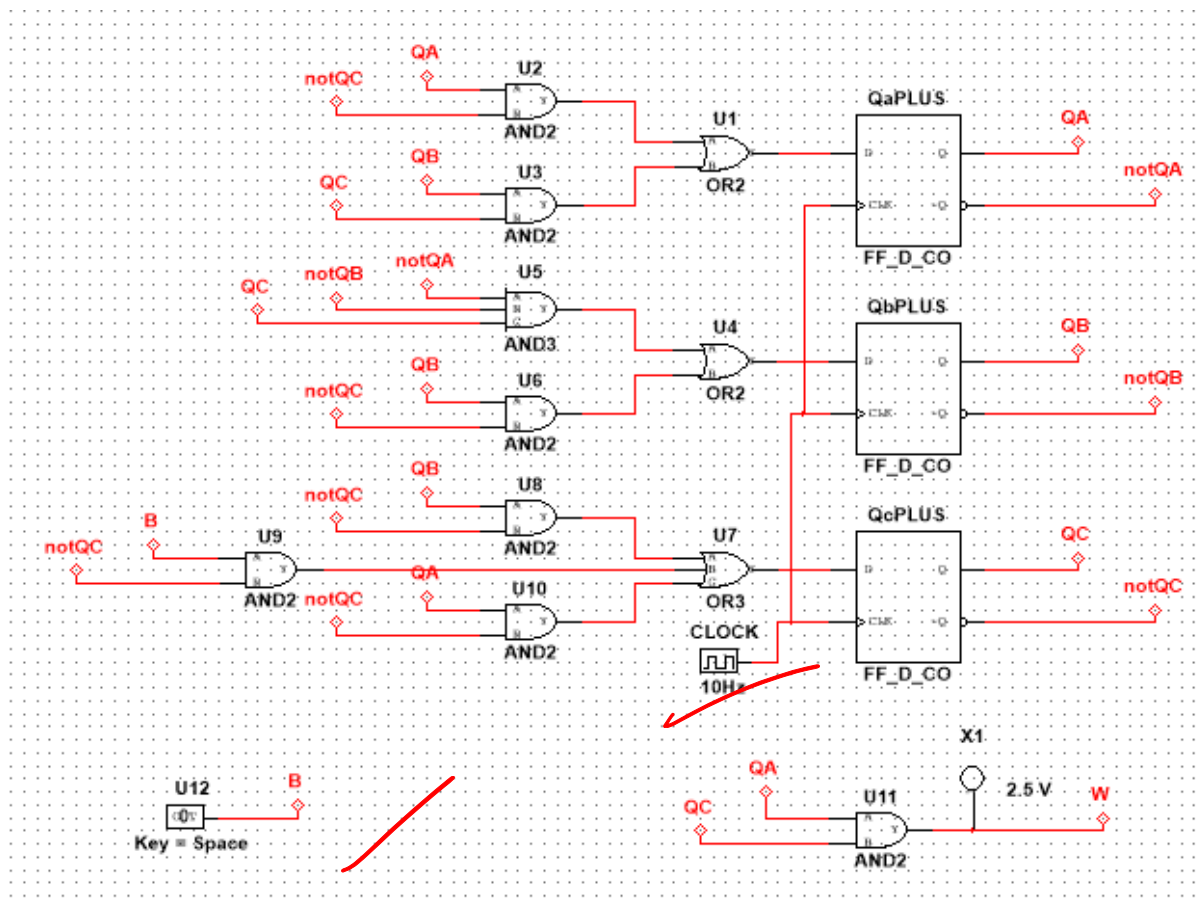


Figure 1: 0-5 Count design on multisim.

Each Karnaugh map in the above tables 2-5 have their own person Boolean expressions next to them; these were used to help aid in the design of the 0-5 count on multisim as shown in figure 1. As the specification states that the design must count from 0-5 non- repetitively and can only count when B=1 and then stop when W=1, it therefore had to be tested by use of a logic analyser as shown in figure 2. The outputs from the design were connected to the logic analyser, that being QA, QB, QC, and W.

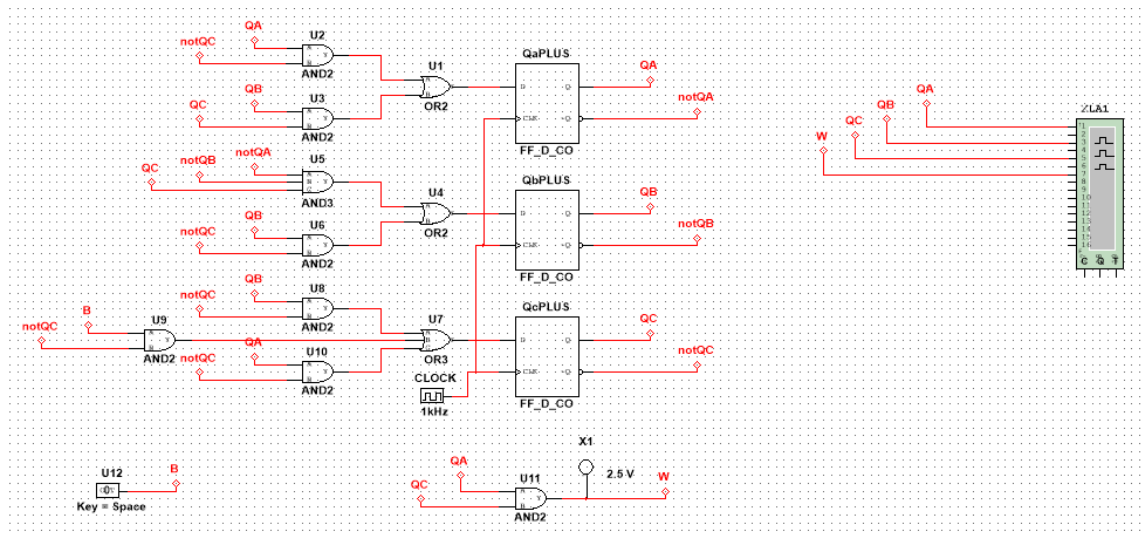


Figure 2: 0-5 count, connected to a logic analyser.

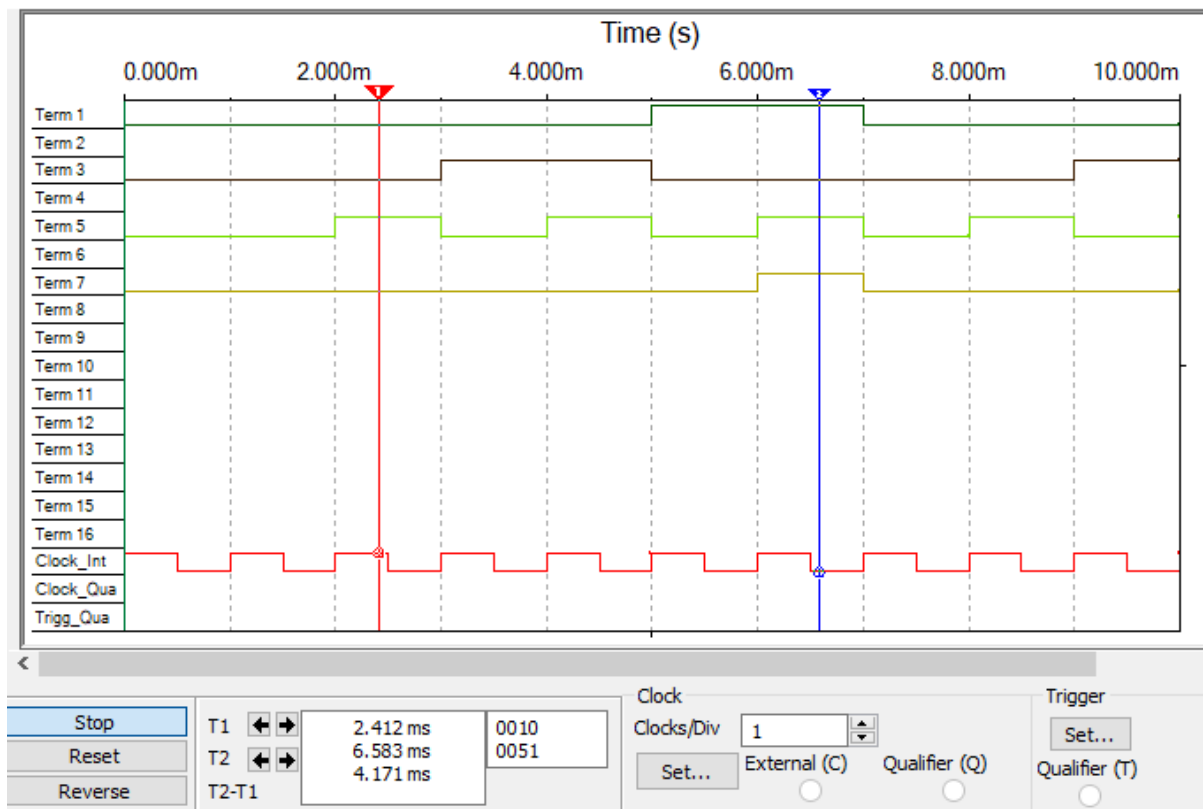


Figure 3: 0-5 Count displayed on logic analyser.

From looking at the graph developed by the logic analyser in figure 3 it can be seen that the design of the 0-5 counter works. The red cursor shows that it is currently in state 1 which is when the button is pressed and the count from 0-5 begins hence the output being notQA notQB QC notW. Furthermore, the blue cursor shows the count being in current state 5 which is when the count stops, and the wait begins hence the output being QA notQB QC W.

The next step of building the traffic light system is to adapt the circuit design as a sub- circuit by removing all components such as probes and clocks and connect it using the PLD mode in order to fully test the system as shown in figure 4.

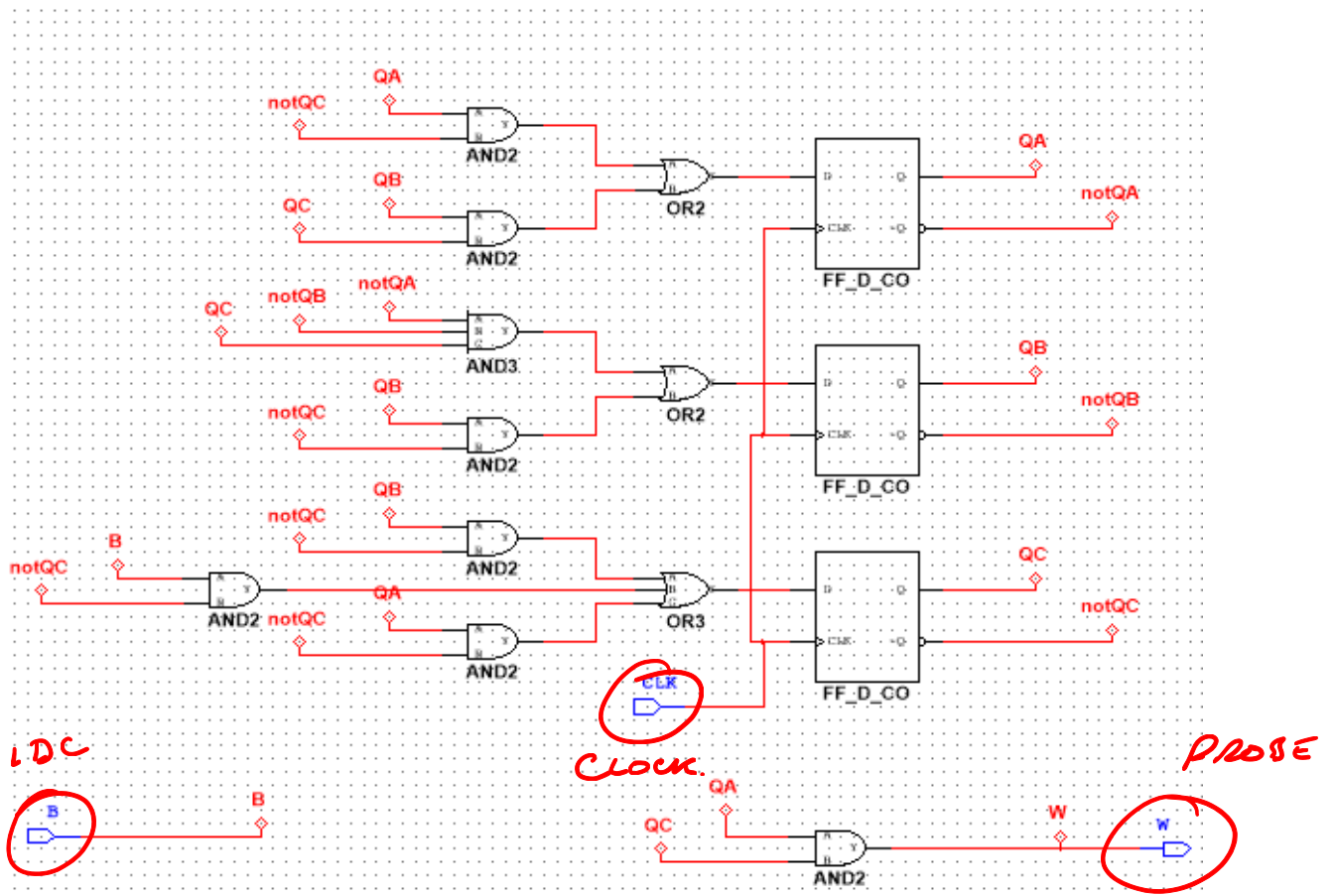


Figure 4: Image of sub-circuit for 0-5 count.

A 0-15 counter was also needed in order to create the traffic light system. The 0-15 counter specified that it had to count from 0-15 non-repetitively. The count starts when the input is read and stops when $T=1$.

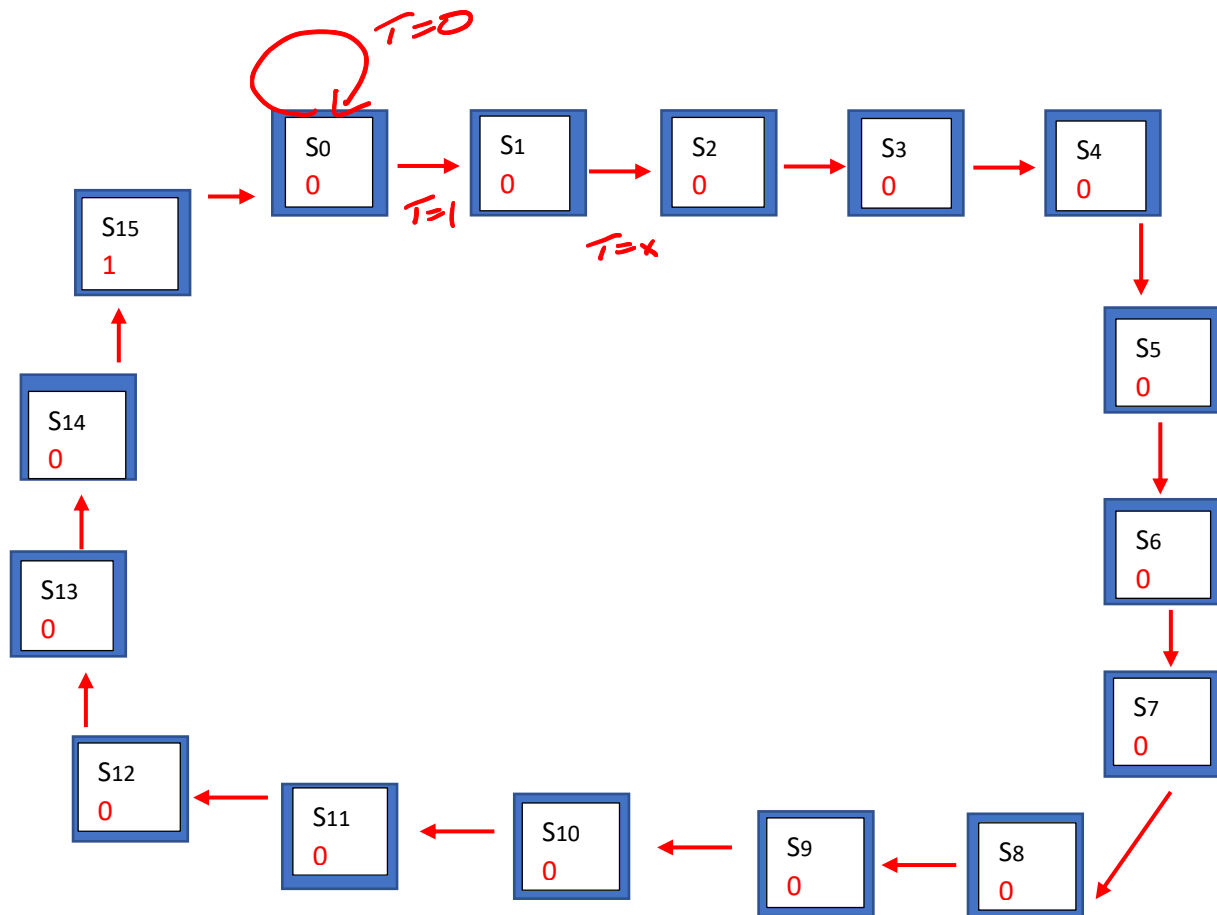


Table 6: State table for 0-15 counter.

0-15 COUNTER															
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+	
S ₀	S ₀	S ₁	0	0	0	0	0	0	0	0	0	0	0	0	0
S ₁	S ₂	S ₂	0	0	0	1	0	0	1	0	0	0	0	1	0
S ₂	S ₃	S ₃	0	0	1	0	0	0	1	1	0	0	1	1	0
S ₃	S ₄	S ₄	0	0	1	1	0	1	0	0	0	1	0	0	0
S ₄	S ₅	S ₅	0	1	0	0	0	1	0	1	0	1	0	1	0
S ₅	S ₆	S ₆	0	1	0	1	0	1	1	0	0	1	1	0	0
S ₆	S ₇	S ₇	0	1	1	0	0	1	1	1	0	1	1	1	0
S ₇	S ₈	S ₈	0	1	1	1	1	0	0	0	1	0	0	0	0
S ₈	S ₉	S ₉	1	0	0	0	1	0	0	1	1	0	0	1	0
S ₉	S ₁₀	S ₁₀	1	0	0	1	1	0	1	0	1	0	1	0	0
S ₁₀	S ₁₁	S ₁₁	1	0	1	0	1	0	1	1	1	0	1	1	0
S ₁₁	S ₁₂	S ₁₂	1	0	1	1	1	1	0	0	1	1	0	0	0
S ₁₂	S ₁₃	S ₁₃	1	1	0	0	1	1	0	1	1	1	0	1	0
S ₁₃	S ₁₄	S ₁₄	1	1	0	1	1	1	1	0	1	1	1	0	0
S ₁₄	S ₁₅	S ₁₅	1	1	1	0	1	1	1	1	1	1	1	1	0
S ₁₅	S ₀	S ₀	1	1	1	1	0	0	0	0	0	0	0	0	1

Table 7: Karnaugh map for QA+, R=0

QC+ QD+ \ QA+ QB+	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	1	0	1
10	1	1	1	1

R=0

$QA^+ = QA \text{ not } QB + QA \text{ not } QD + \text{not } QA$
 $QB \text{ } QC \text{ } QD$

Table 8: Karnaugh map for QA+, R=1

QC+ QD+ \ QA+ QB+	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	1	0	1
10	1	1	1	1

R=1

$QA^+ = QA \text{ not } QB + QA \text{ not } QD + \text{not } QA$
 $QB \text{ } QC \text{ } QD$

Table 9: Karnaugh map for QB+, R=0

QC+ QD+ \ QA+ QB+	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	1	1	0	1
10	0	0	1	0

R=0

$QB^+ = QB \text{ not } QC + \text{not } QB \text{ } QC \text{ } QD + B$
 $\text{not } QD$

Table 10: Karnaugh map for QB^+ , $R=1$

$QC^+ QD^+$ $QA^+ QB^+$	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	1	1	0	1
10	0	0	1	0

$R=1$

$$QB^+ = QB \text{ not} QC + \text{not} QB \text{ } QC \text{ } QD + B \text{ not} QD$$

Table 11: Karnaugh map for QC^+ , $R=0$

$QC^+ QD^+$ $QA^+ QB^+$	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$R=0$

$$QC^+ = \text{not} QC \text{ } QD + QC \text{ not} QD$$

Table 12: Karnaugh map for QC^+ , $R=1$

$QC^+ QD^+$ $QA^+ QB^+$	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$R=1$

$$QC^+ = \text{not} QC \text{ } QD + QC \text{ not} QD$$

Table 13: Karnaugh map for QD^+ , $R=0$

$QC^+ QD^+$ $QA^+ QB^+$	00	01	11	10
00	0	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

$R=0$

$$QD^+ = QC \text{ not} QD + \text{not} QD \text{ } R + B \text{ not} QD + QA \text{ not} QD$$

Table 14: Karnaugh map for QD^+ , $R=1$

Q_C+Q_D+ Q_A+Q_B+	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

$R=1$
 $Q_D^+ = Q_C \text{ not } Q_D + \text{not } Q_D R + B \text{ not } Q_D + Q_A \text{ not } Q_D$

Table 15: Karnaugh map for T

Q_C+Q_D+ Q_A+Q_B+	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	1	0
10	0	0	0	0

$T = A B C D$

Again, like the 0-5 counter the 0-15 counter has its own Karnaugh maps and Boolean expressions ranging from tables 7-15. By use of these tables and expressions it was possible to build the 0-15 counter on multisim as shown in the below figure 5. By the aid of a logic analyser, it could be tested to prove it works before being converted into a sub- circuit via use of the PLD mode as shown in figure 6 and 7.

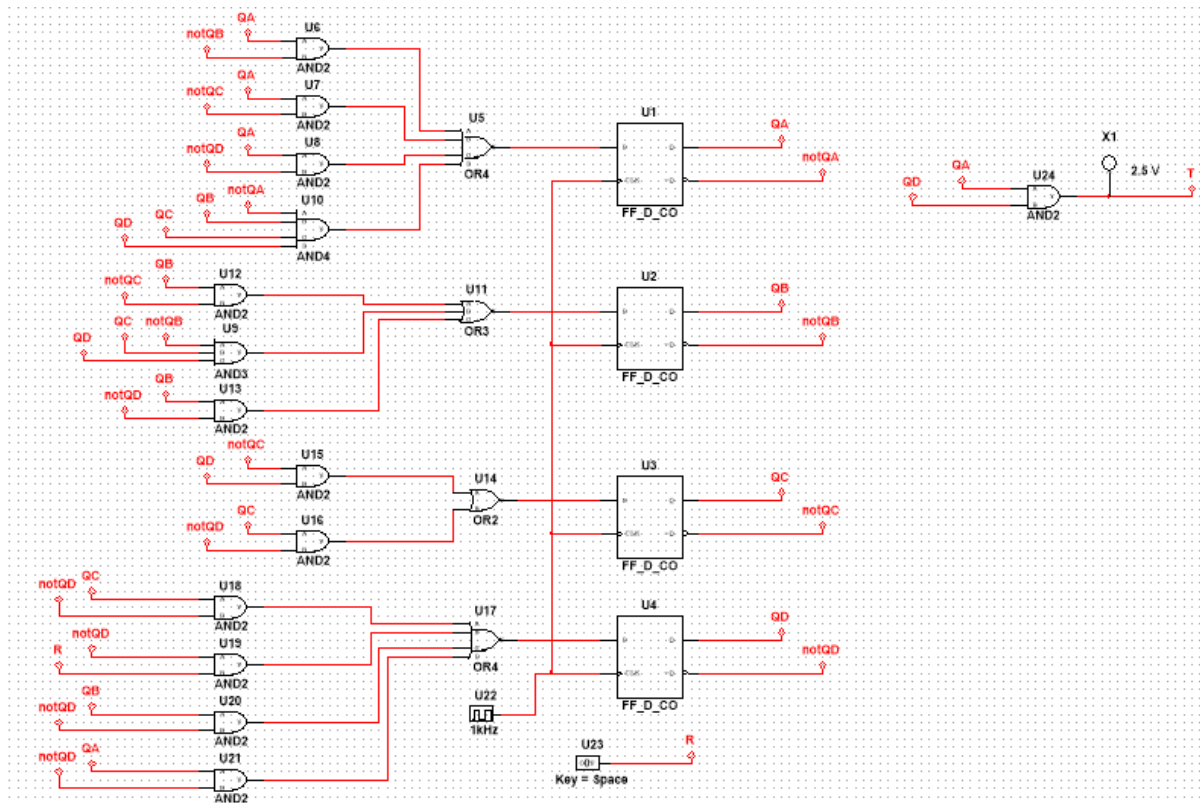


Figure 5: 0-15 count design on multisim.

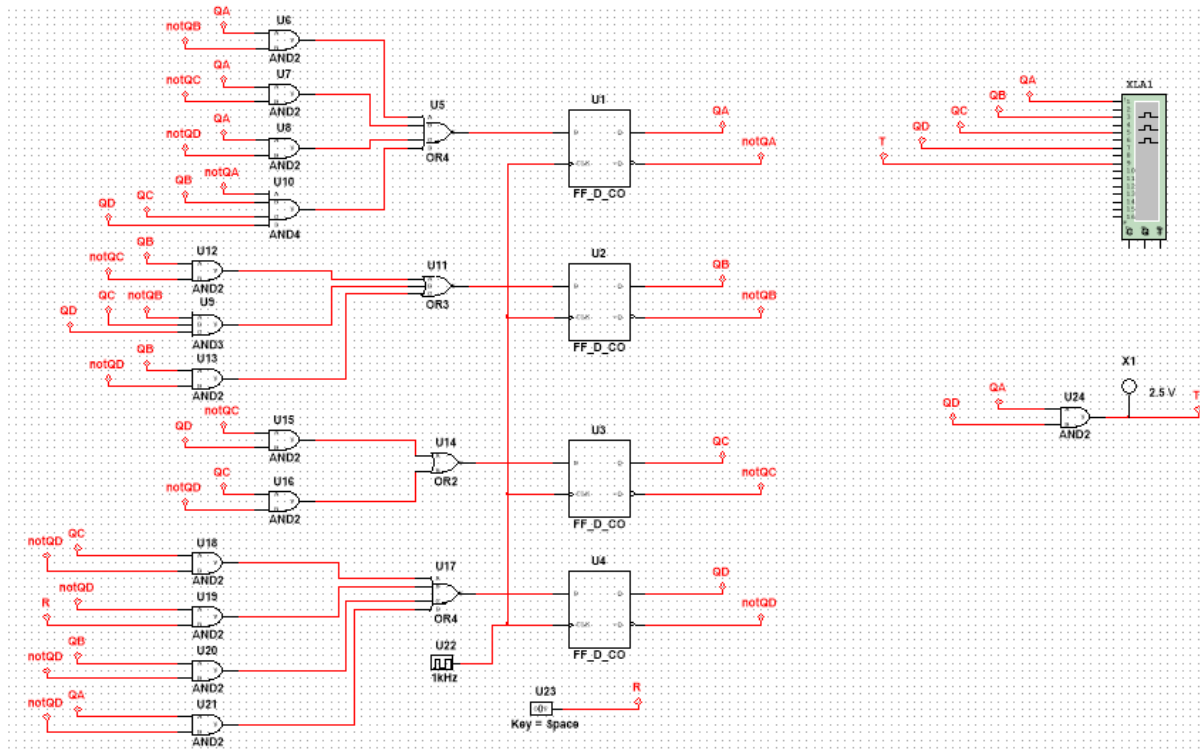


Figure 6: 0-15 counter connected to logic analyser.



Figure 7: 0-15 count being tested.

As shown in figure 7 it can be seen that at the red cursor the 0-15 count has begun since its current state is 1, notQA notQB notQC QD notT. This means that the traffic has stopped to allow pedestrians to cross. Further on in graph the blue cursor is indicating that the current state is at 15 since all the inputs are on. This means that the 0-15 count is finished which as a result indicates the next state being 0 which allows the traffic to flow again.

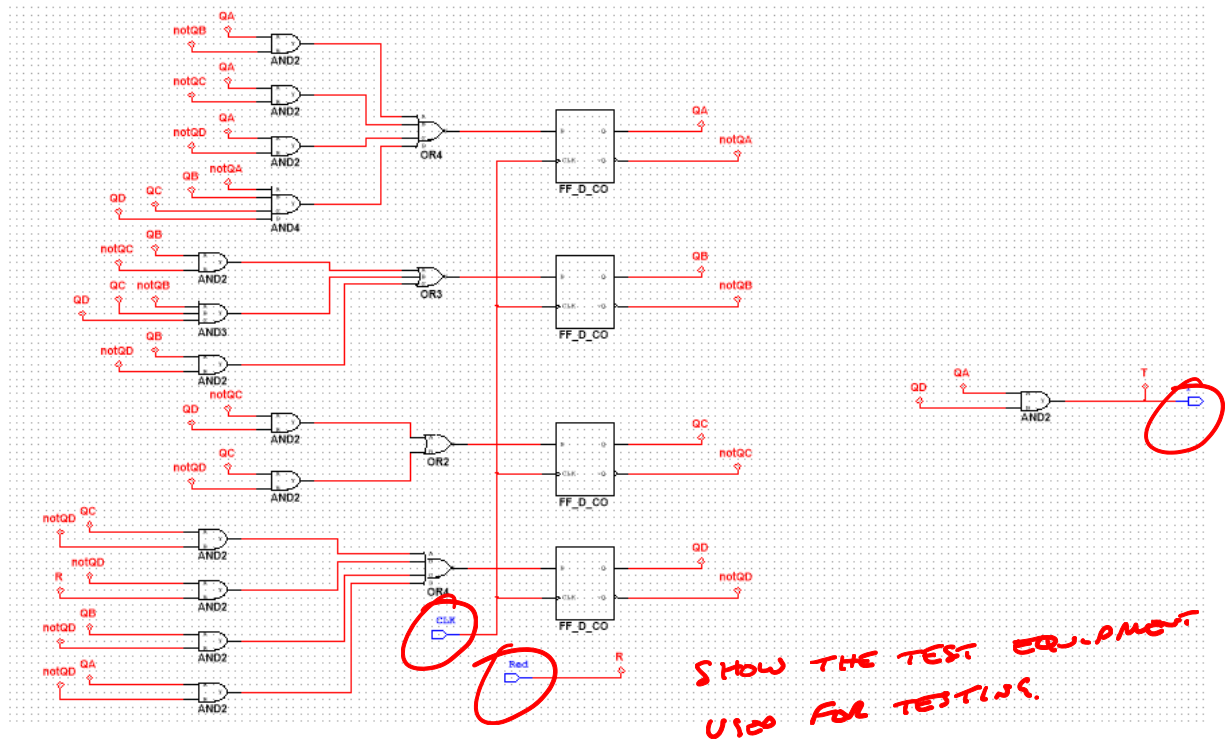


Figure 8: Sub circuit created for 0-15 counter using PLD mode.

The final stage of designing the pelican crossing was to design and create a RAG controller (Red/ Amber/ Green). The controller specified that it had to follow the set sequence below:

S0 = Green

S1 = Amber

S2 = Amber

S3 = Red

S4 = Red/ Amber

S5 = Red/ Amber

S0 = Green

KEY

- W= 0 the pedestrian stays where they are.
- W=1 it goes to amber.
- T=x (don't care state)

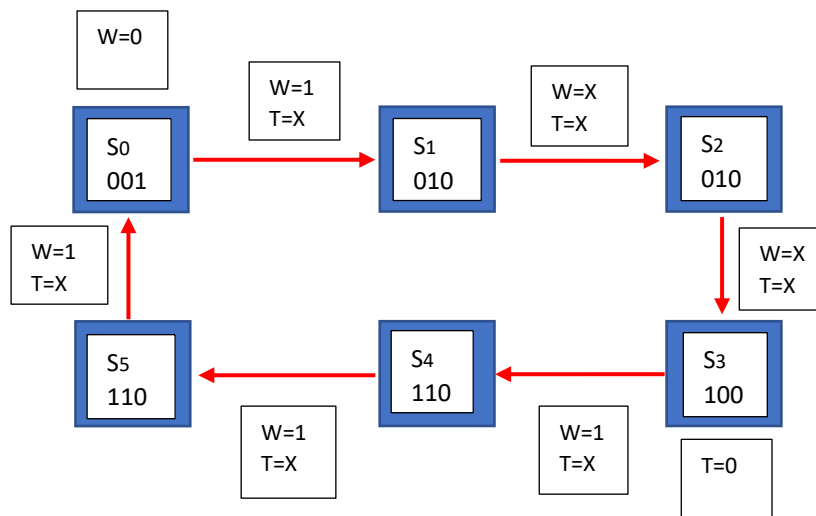


Table 16: State table for RAG controller.

Present State	RAG sequence controller state table																Output		
					Next State														
	Next State				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+			
S ₀	S ₀	S ₀	S ₁	S ₁	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
S ₁	S ₂	S ₂	S ₂	S ₂	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
S ₂	S ₃	S ₃	S ₃	S ₃	0	1	1	0	1	1	0	0	1	1	0	1	1	0	1
S ₃	S ₃	S ₄	S ₃	S ₄	0	1	1	1	0	0	0	0	1	1	0	0	1	0	0
S ₄	S ₅	S ₅	S ₅	S ₅	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0
S ₅	S ₀	S ₀	S ₀	S ₀	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
S ₆	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
S ₇	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 17: Karnaugh map for QA+, W=0

QB+ QC+	00	01	11	10
T QA+				
00	0	0	0	0
01	1	0	X	X ¹
11	1	0	X ¹	X ¹
10	0	0	1	0

W=0

$$QA^+ = QA \text{ not } QC + T \text{ } QB \text{ } QC$$

Table 18: Karnaugh map for QA+, W=1

QB+ QC+ \ T QA+	00	01	11	10
00	0	0	0	0
01	1	0	X	X ¹
11	1	0	X ¹	X ¹
10	0	0	1	0

W=1

$$QA^+ = QA \text{ not } QC + T \text{ } QB \text{ } QC$$

Table 19: Karnaugh map for QB+, W=0

QB+ QC+ \ T QA+	00	01	11	10
00	0	1	1	1
01	0	0	X ¹	X ¹
11	0	0	X	X ¹
10	0	1	0	1

W=0

$$QB^+ = QB \text{ not } QC + \text{not } T \text{ } QB + \text{not } QA \text{ not } QB \text{ } QC$$

Table 20: Karnaugh map for QB+, W=1

QC+ QD+ \ QA+ QB+	00	01	11	10
00	0	1	1	1
01	0	0	X ¹	X ¹
11	0	0	X	X ¹
10	0	1	0	1

W=1

$$QB^+ = QB \text{ not } QC + \text{not } T \text{ } QB + \text{not } QA \text{ not } QB \text{ } QC$$

Table 21: Karnaugh map for QC+, W=0

QC+ QD+ \ QA+ QB+	00	01	11	10
00	0	0	1	1
01	1	0	X ¹	X ¹
11	1	0	X	X ¹
10	0	0	0	1

W=0

$$QC^+ = QA \text{ not } QC + \text{not } T \text{ } QB + QB \text{ not } QC + W \text{ not } QC$$

Table 22: Karnaugh map for QC^+ , $W=1$

$QC^+ QD^+$ $QA^+ QB^+$	00	01	11	10
00	1	0	1	1
01	1	0	X^1	X^1
11	1	0	X	X^1
10	1	0	0	1

$W=1$
 $QC^+ = QA \text{ not } QC + \text{not } T \text{ } QB + QB \text{ not } QC + W \text{ not } QC$

Table 23: Karnaugh map for R .

$QB^+ QC^+$ QA^+	00	01	11	10
0	0	0	1	0
1	1	1	X^1	X

$R = QA + QB \text{ } QC$

Table 24: Karnaugh map for A .

$QB^+ QC^+$ QA^+	00	01	11	10
0	0	1	0	1
1	1	1	X	X^1

$A = QA + \text{not } QB \text{ } QC + QB \text{ not } QC$

Table 25: Karnaugh map for G .

$QB^+ QC^+$ QA^+	00	01	11	10
0	1	0	0	0
1	0	1	X	X

$G = \text{not } QA \text{ not } QB \text{ not } QC$

Like with the 0-5 and 0-15 counters, the RAG controller has had Karnaugh maps and truth tables (tables 17-25) created which can be used to aid the creation on multisim. Furthering the design process, it can again be tested by use of a logic analyser.

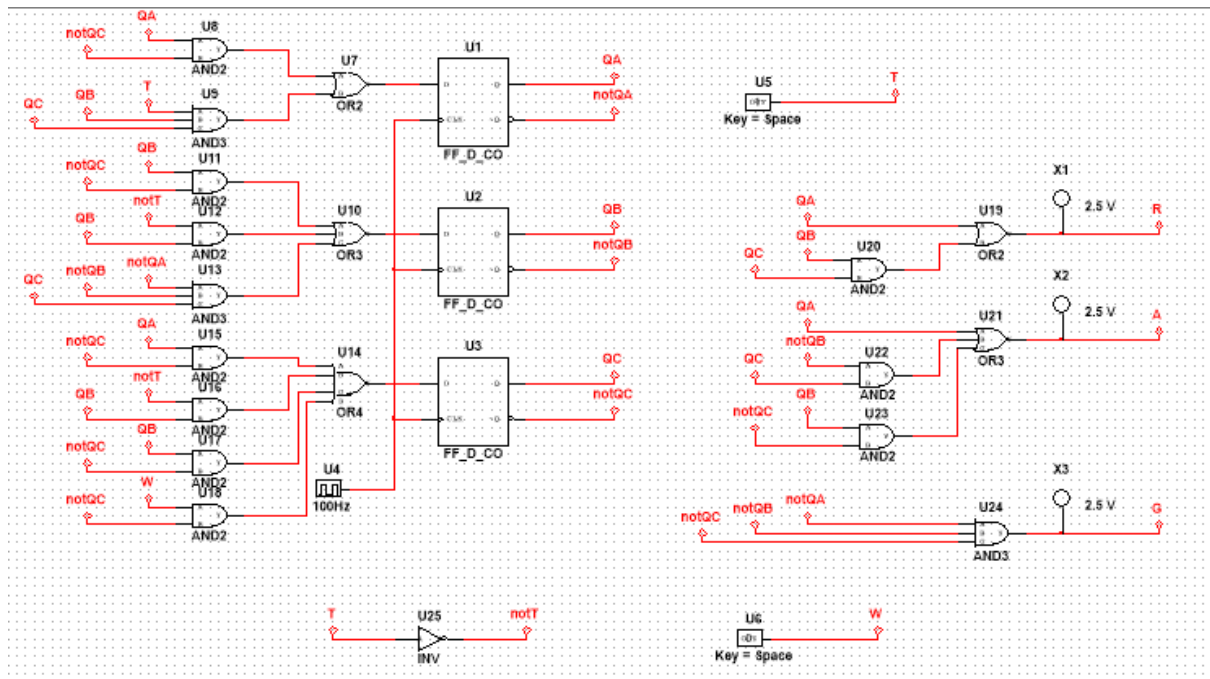


Figure 9: Image of the RAG Controller.

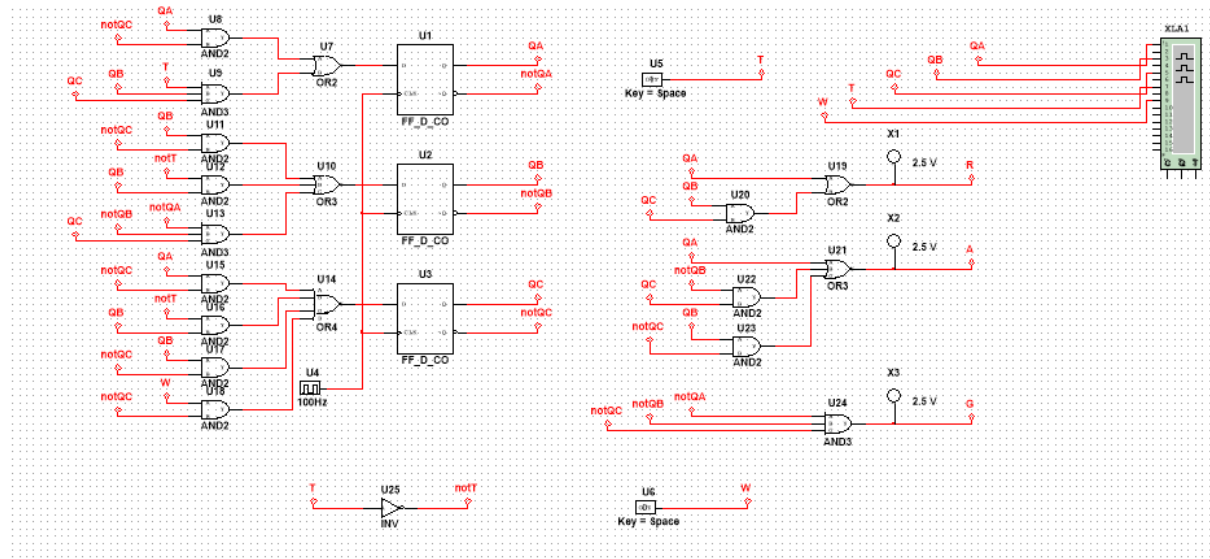


Figure 10: Image of the RAG Controller connected to the Logic analyser.



Figure 11: Logic analyser testing green traffic light (state 0).

Figure 11 shows the traffic light status being green as when at state 1 the inputs are notQA notQB notQC (red cursor) and also notQA notQB QC (blue cursor).

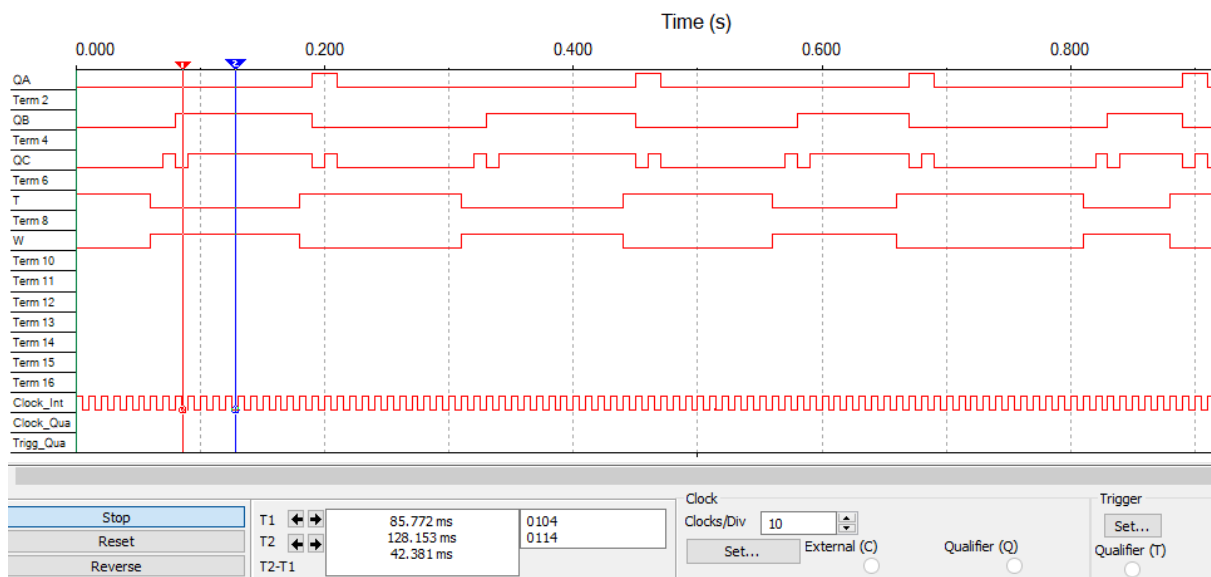


Figure 12: Logic analyser testing amber traffic light (state 1 and 2).

Figure 12 shows the traffic light status being amber when at state 1 (red cursor) since the inputs are notQA QB not QC and also when at state 2 (blue cursor) notQA QB QC.

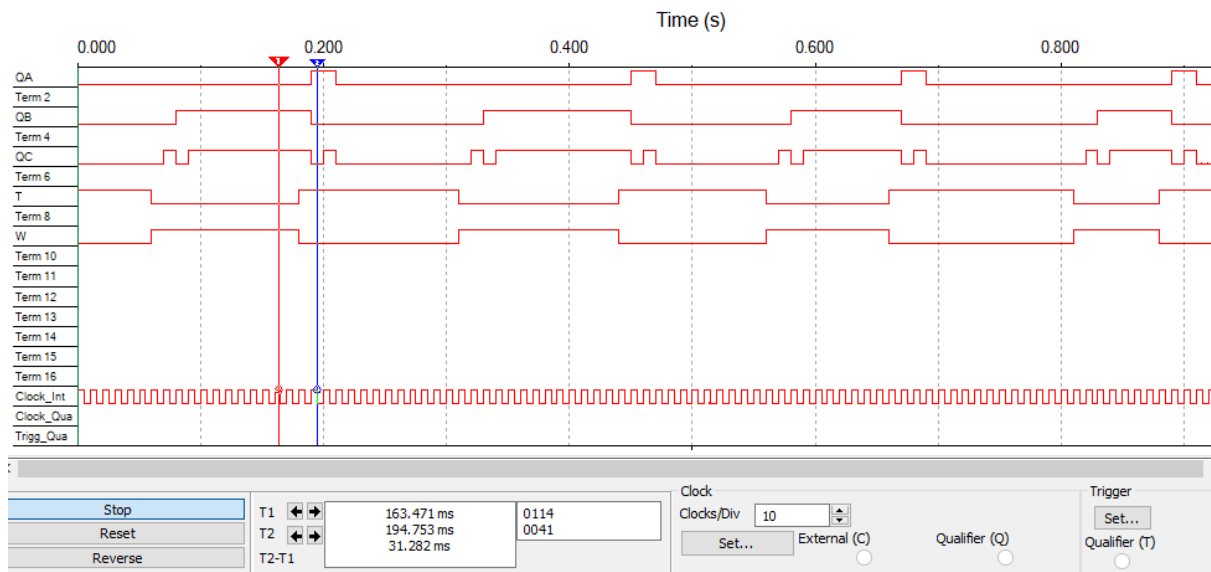


Figure 13: Logic analyser testing red traffic light (state 3).

Figure 13 is showing state 3 being red as the red cursor indicates notQA QB QC and the blue cursor indicates QA notQB not QC.

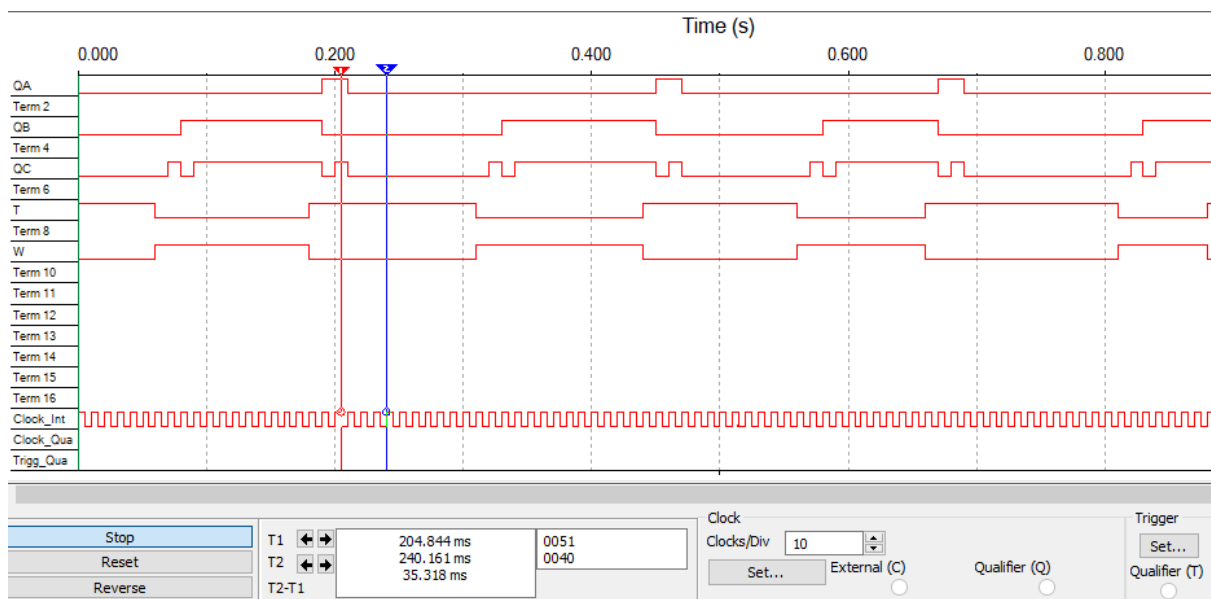


Figure 14: Logic analyser testing red/ amber traffic light (state 4 and 5).

Figure 14 is showing state 4 being red and amber as the red cursor indicates QA notQB QC and the blue cursor indicates state 5 as the inputs are notQA notQB notQC.

Finally, the RAG controller can be removed of components like the digital clock and probes and adapted using the PLD mode to a sub circuit where all 3 circuits are combined to fully test the pelican crossing.

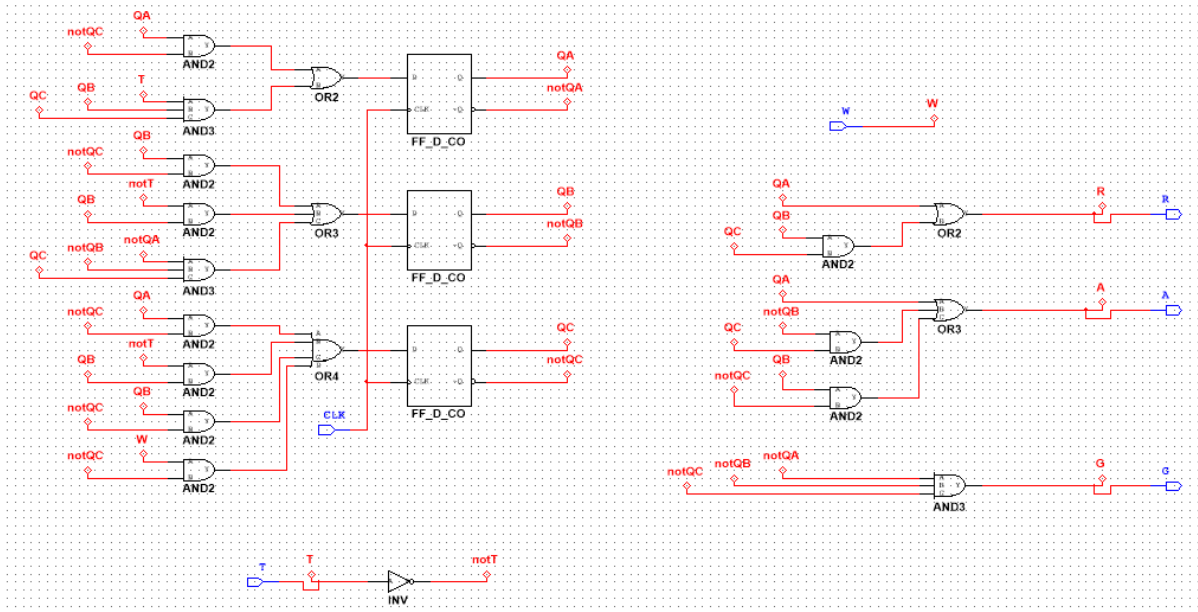


Figure 15: Sub-circuit created for RAG controller using PLD mode.

As a result of all the sub-circuits being made and combined together using the PLD mode it is possible to test them all together and witness a working pelican crossing with use of a finite state machine (FSM).

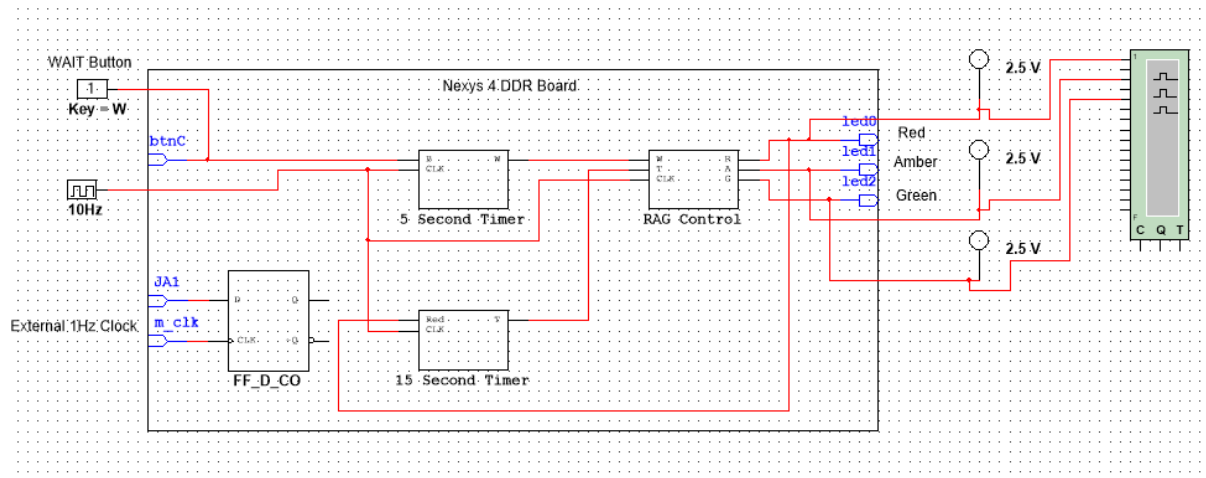


Figure 16: Image of the complete design.

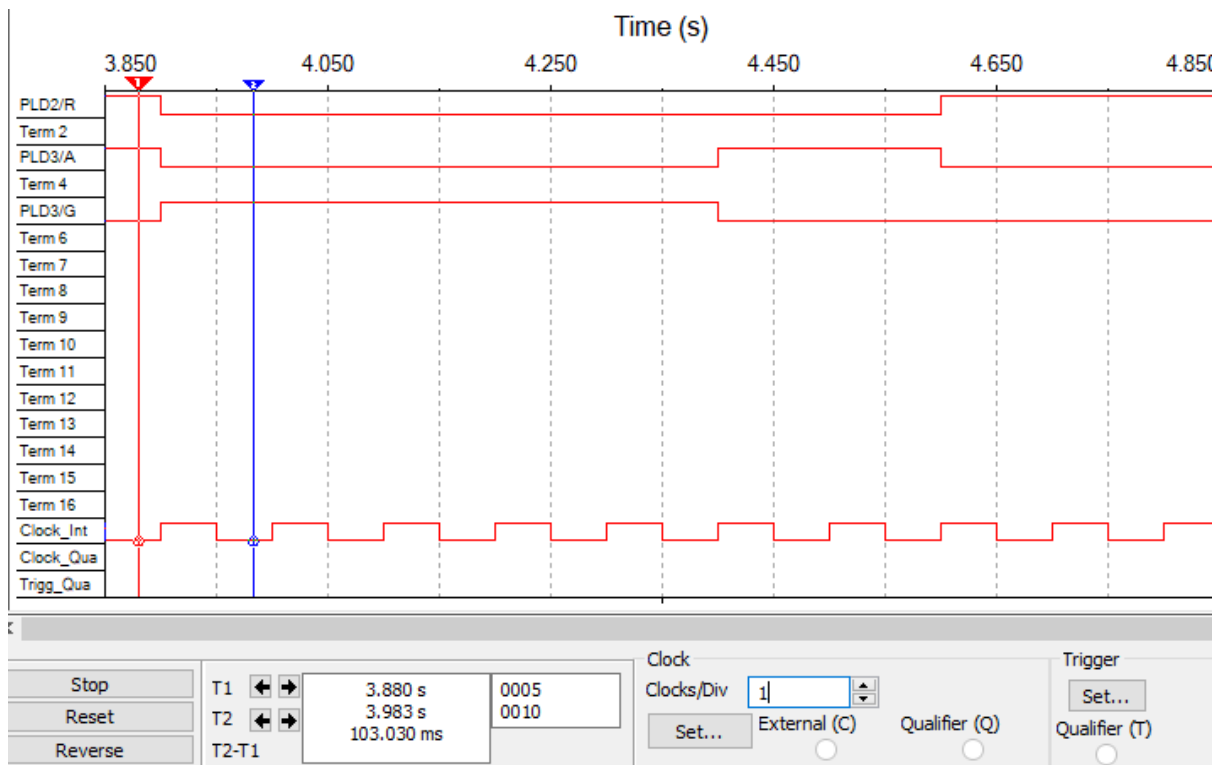


Figure 17: Testing final circuit with use of logic analyser.

In figure 17 the red cursor indicates the lights changing from red and amber to green as shown by the blue cursor.

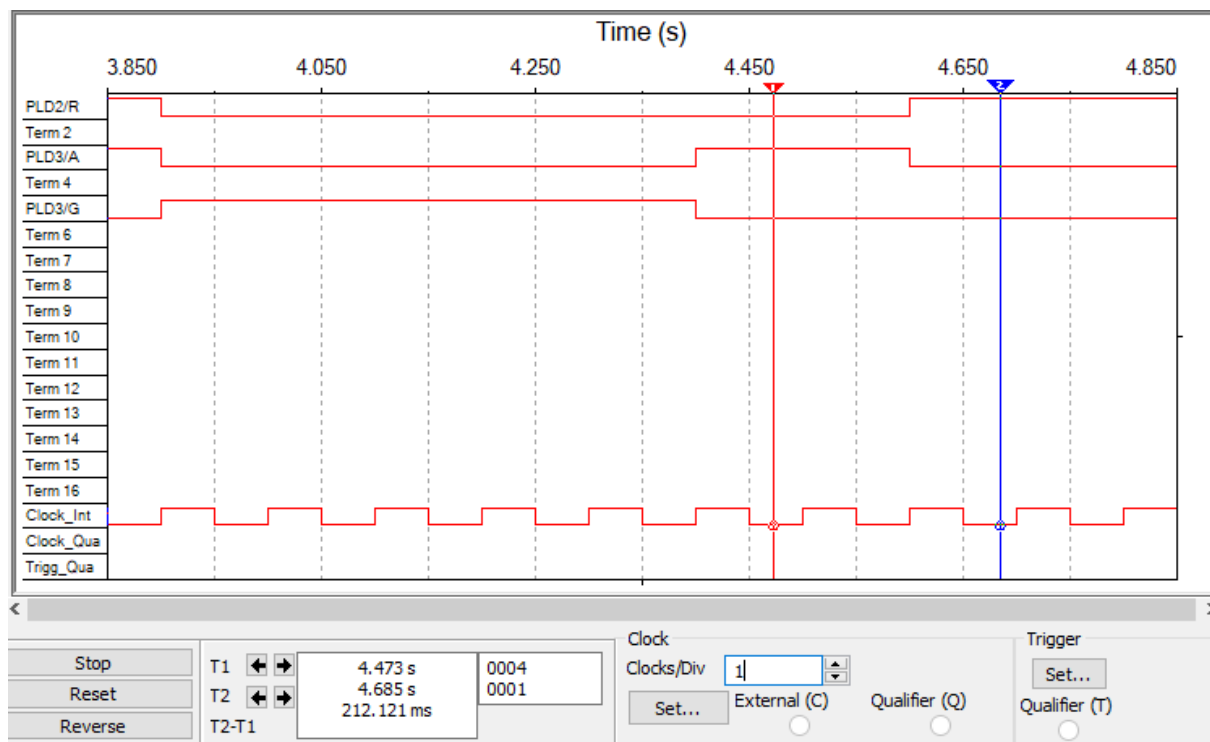


Figure 18: Testing circuit with use of logic analyser.

In figure 18, the red cursor shows the lights changing from green to amber and the to red as indicated by the blue cursor.

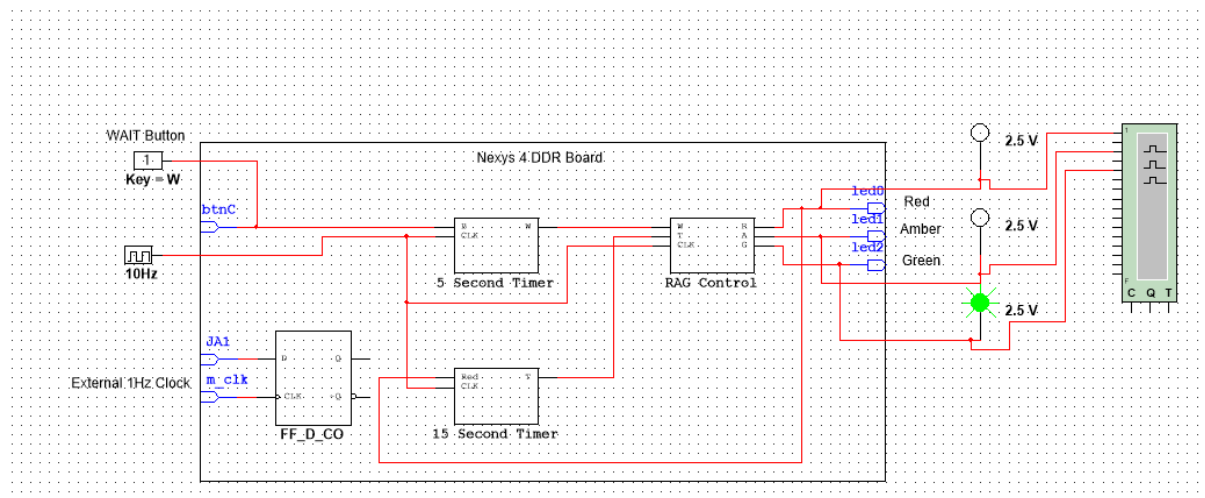


Figure 19: Green light for pelican crossing.

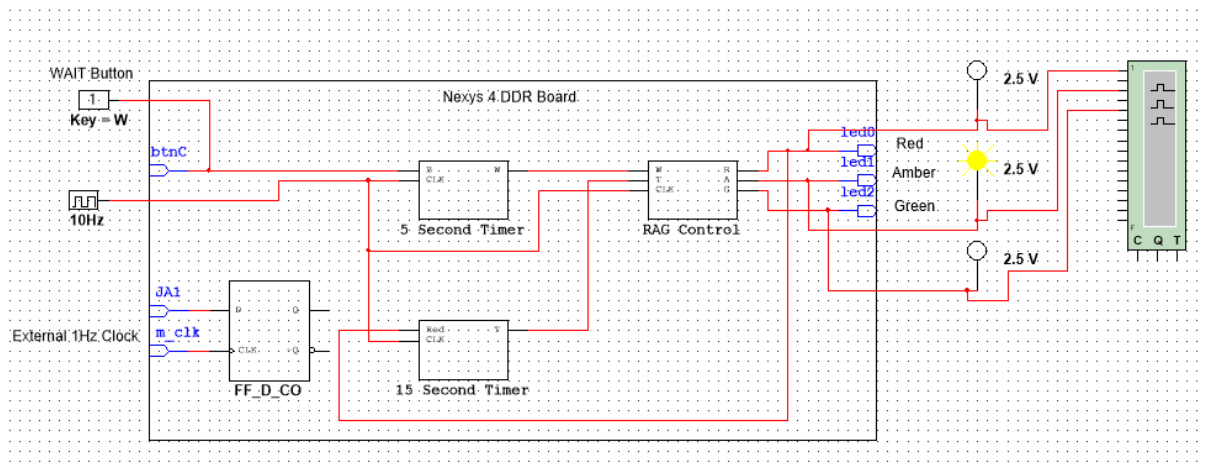


Figure 20: Amber light for pelican crossing.

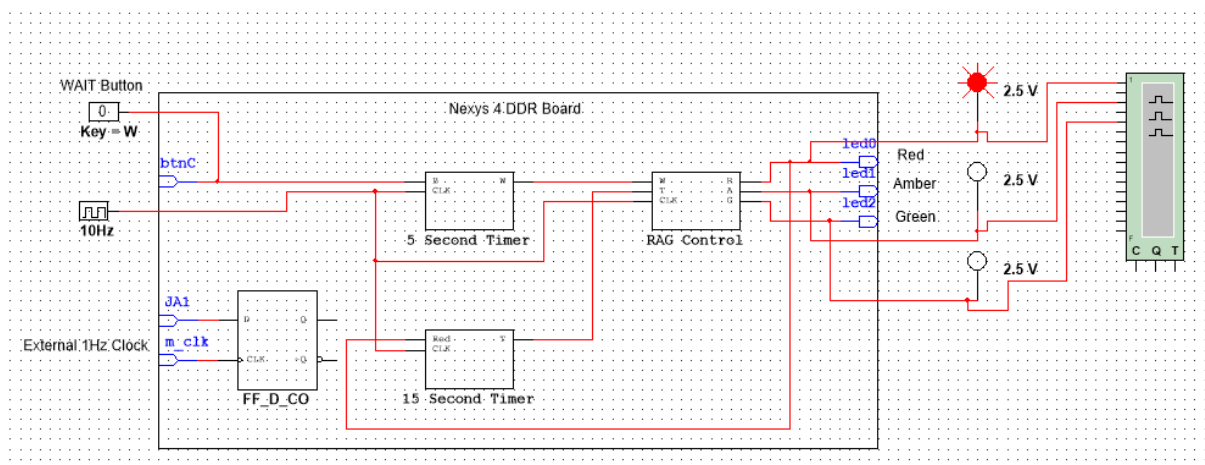


Figure 21: Red light for pelican crossing.

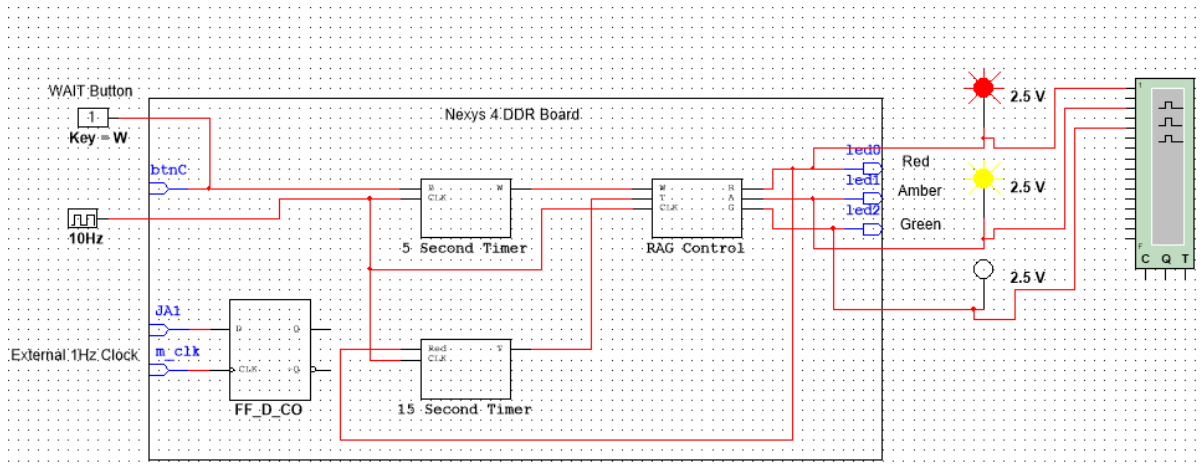


Figure 22: Red and Amber light for pelican crossing.

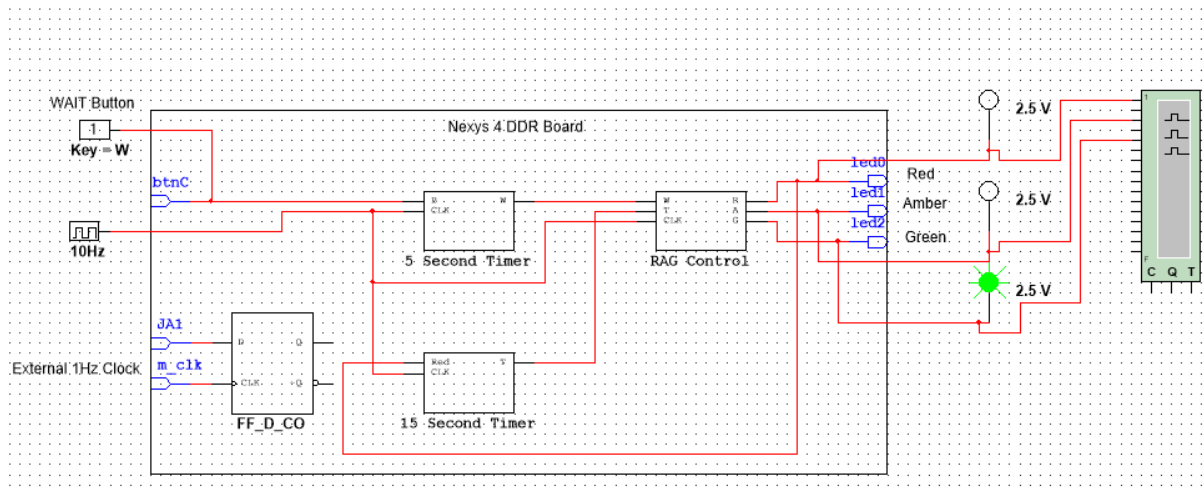
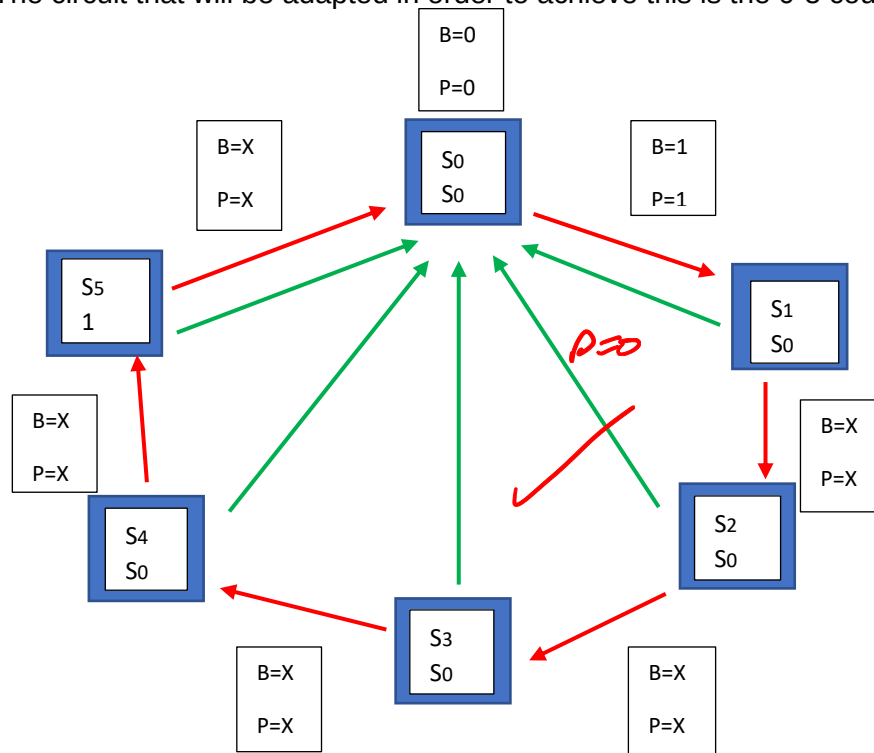


Figure 23: Green light for pelican crossing.

APPENDIX FOR THESE.

As it is known a puffin crossing works when an electronic sensor detects when someone wants to cross the road, it is possible to adapt the design of the pelican crossing in order to achieve this. The circuit that will be adapted in order to achieve this is the 0-5 counter circuit.



The state diagram shows in green that if the sensor no longer detects a pedestrian waiting to cross it ends the count and returns to state 0, however if a button is pressed the count continues like the 0-5 counter.

Table 26: Puffin crossing state table.

Current State	Puffin crossing state table																			Output
	Next State				Current State				Next State											
									P=0						P=1					
	B=0		B=1		B=0			B=1			B=0			B=1						
	P=0	P=1	P=0	P=1	QA	QB	QC	QA+	QB+	QC+	QA+	QB+	QC+	QA+	QB+	QC+	QA+	QB+	QC+	
S ₀	S ₀	S ₀	S ₀	S ₁	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
S ₁	S ₂	S ₂	S ₀	S ₂	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0
S ₂	S ₃	S ₃	S ₀	S ₃	0	1	0	0	0	0	0	1	1	0	0	0	0	1	1	0
S ₃	S ₃	S ₄	S ₀	S ₄	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0
S ₄	S ₅	S ₅	S ₀	S ₅	1	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0
S ₅	S ₀	S ₀	S ₀	S ₀	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
S ₆	X	X	X	X	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X
S ₇	X	X	X	X	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 27: Karnaugh map for QA^+ , $P=0$

$QB+QC+$ $B QA^+$	00	01	11	10
00	0	0	0	0
01	0	0	X	X
11	0	0	X	X
10	0	0	0	0

$P=0$

Table 28: Karnaugh map for QA^+ , $P=1$

$QB+QC+$ $B QA^+$	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

$P=1$

$$QA^+ = P QA \text{ not } QC + P QB QC$$

Table 29: Karnaugh map for QB^+ , $P=0$

$QB+QC+$ $B QA^+$	00	01	11	10
00	0	0	0	0
01	0	0	X	X
11	0	0	X	X
10	0	0	0	0

$P=0$

Table 30: Karnaugh map for QB^+ , $P=1$

$QC^+ QD^+$ $QA^+ QB^+$	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

$P=1$

$$QB^+ = P \text{ not}QA \text{ not}QB \text{ } QC + P \text{ } QB \text{ not}QC$$

Table 31: Karnaugh map for QC^+ , $P=0$

$QB^+ QC^+$ $B \text{ } QA^+$	00	01	11	10
00	0	0	0	0
01	0	0	X	X
11	0	0	X	X
10	0	0	0	0

$P=0$

Table 32: Karnaugh map for QC^+ , $P=1$

$QB^+ QC^+$ $B \text{ } QA^+$	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

$P=1$

$$QC^+ = P \text{ } QA \text{ not}QC + P \text{ } QB \text{ not}QC + P \text{ } B \text{ not}QC$$

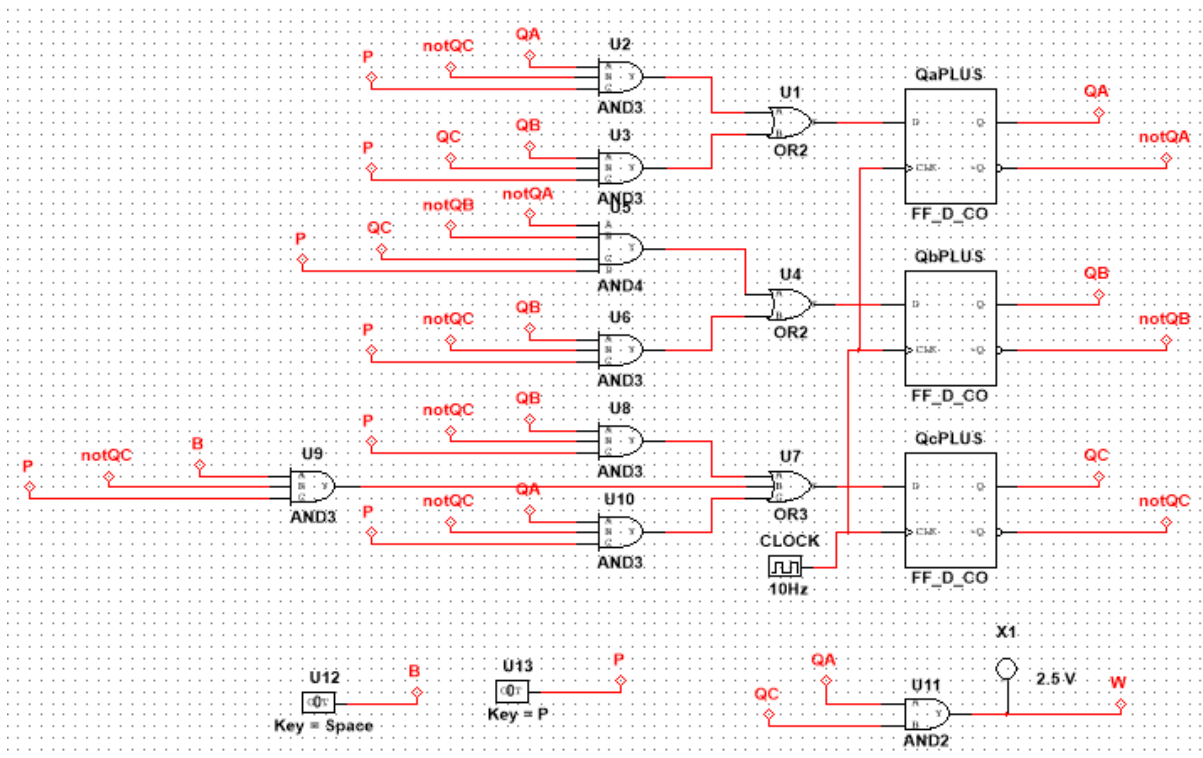


Figure 24: Puffin crossing built on multisim.

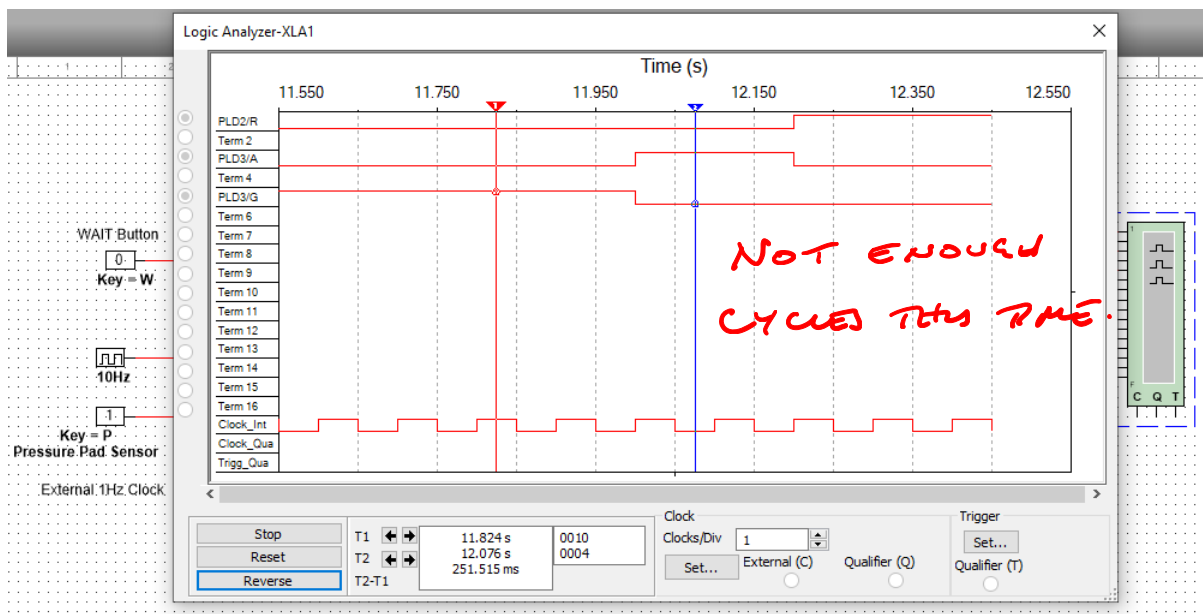


Figure 25: Testing puffin crossing when pressure pad input is 1.

It can be seen that when the pressure sensor had detected a pedestrian the lights change from green to amber to red, as shown by the red and blue cursors in figure 25; but also change back after the pressure has been released and the 15 second count is up as shown in figure 26 by the red and blue cursors.

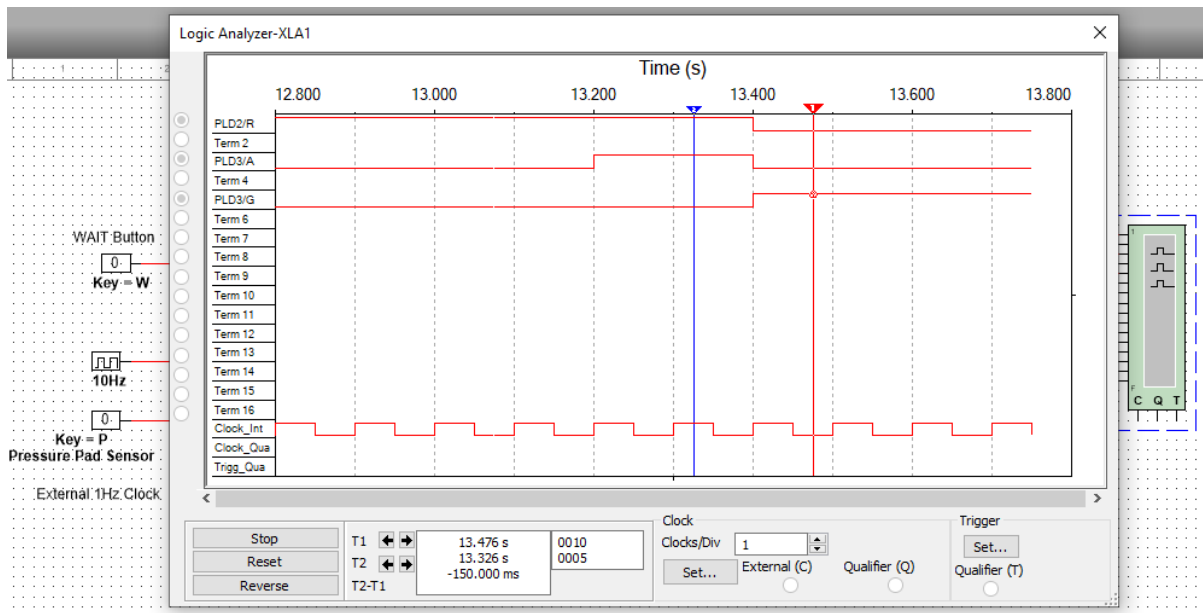


Figure 26: Testing puffin circuit when pressure pad is 0.

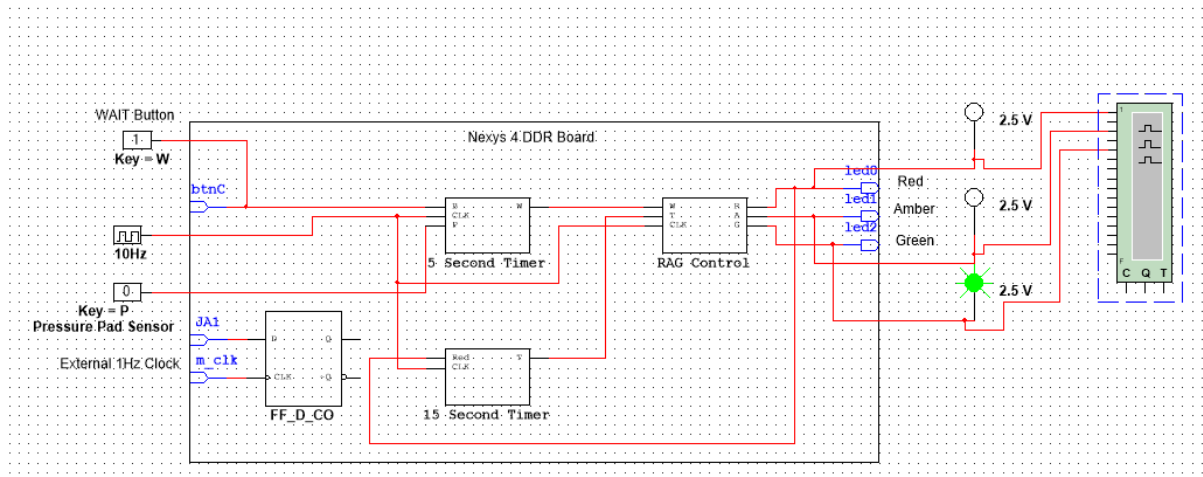


Figure 27: Puffin crossing green light when pressure pad is 0.

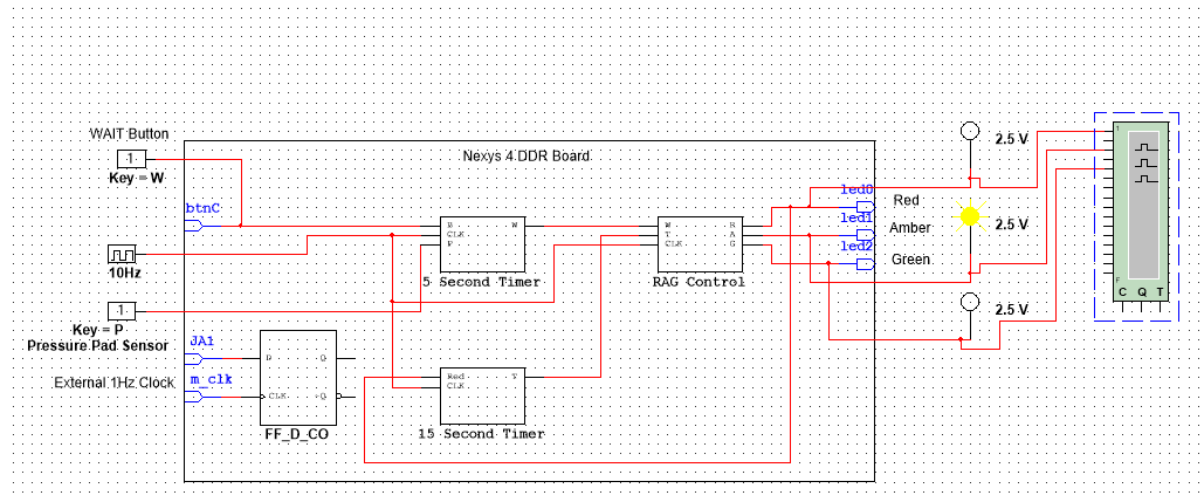


Figure 28: Puffin crossing amber light when pressure pad is 1.

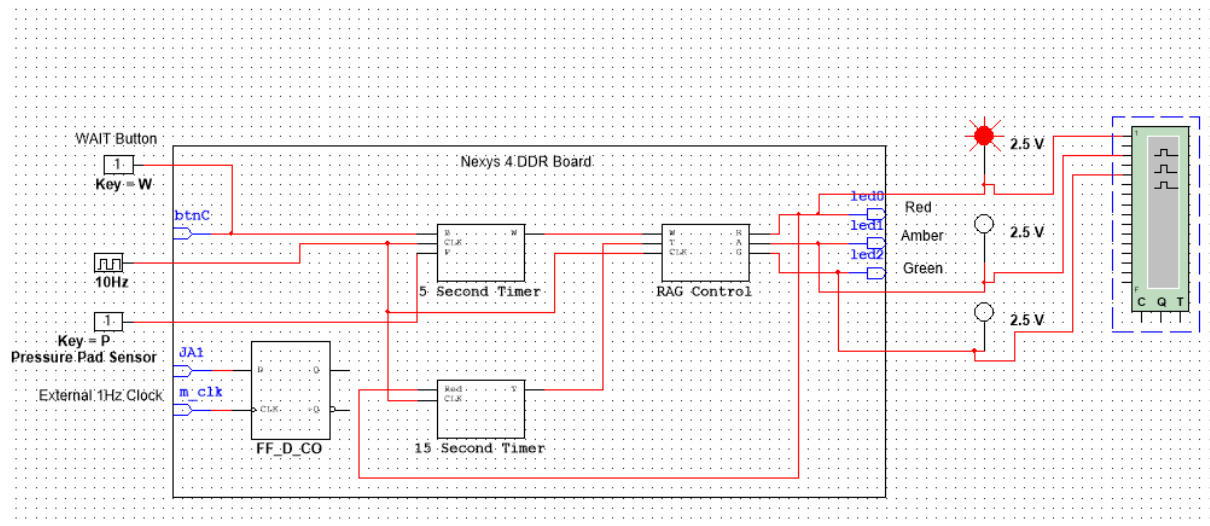


Figure 29: Puffin crossing red light when pressure pad is 1.

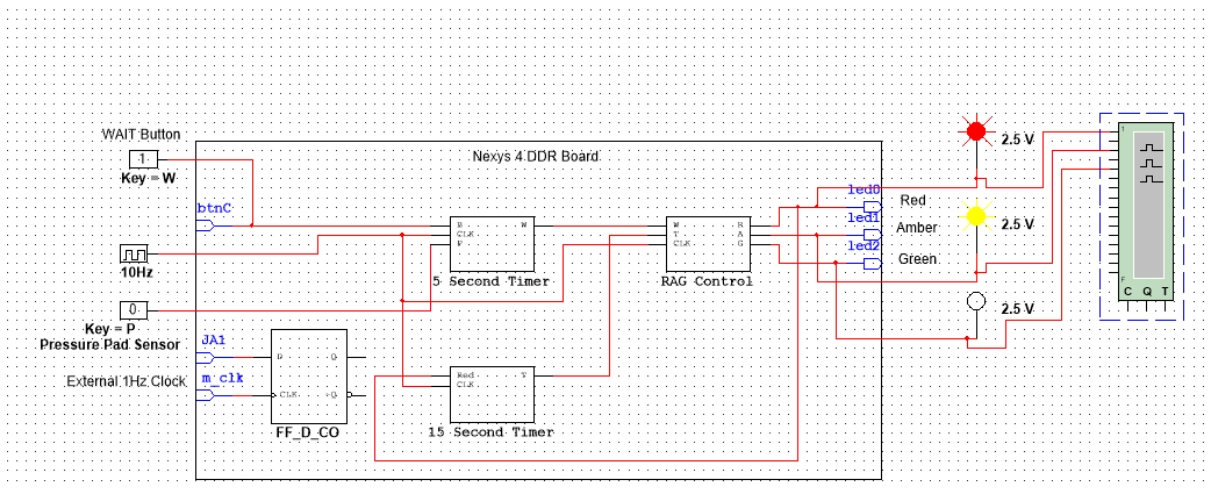


Figure 30: Puffin crossing red and amber light when pressure pad is 0.

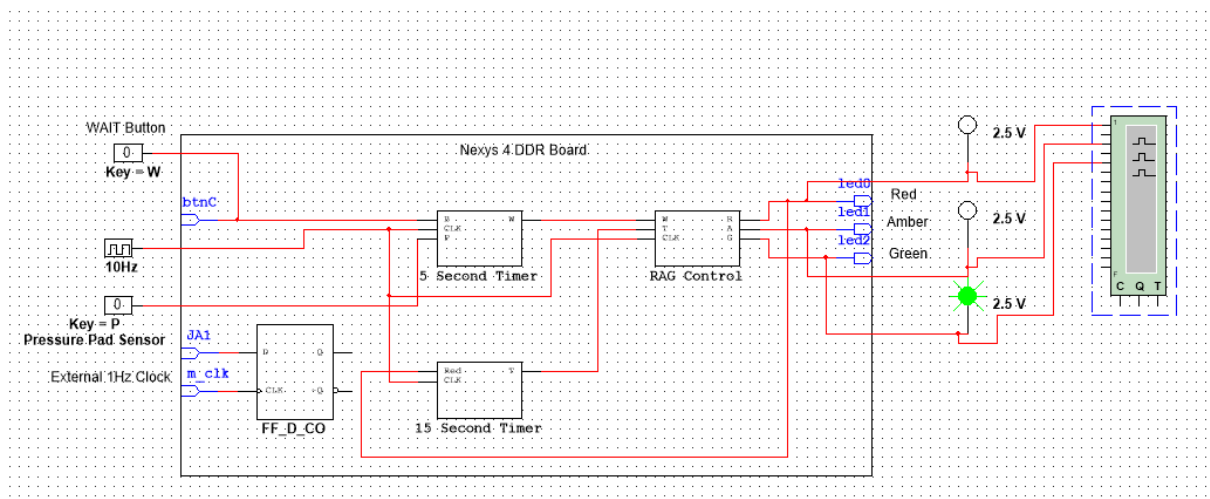


Figure 31: Puffin crossing green light when pressure pad is 0.

4. Testing, results, and analysis

As it is known from research that the pelican crossing works by pedestrians pressing a button in order to cross the road, this can be seen by the individual circuits and also when the subcircuits were combined to show the full evaluation of the circuit.

The gathered results show that the 0-5 counter worked as it was specified to do so. It was mainly asked to count from 0-5 non- repetitively and this was proven to do so looking at figure 3. Furthermore, the same can be said for the 0-15 counter when looking at figure 7 as this was also non- repetitively specified. Figures 19-23 show the sequence of lights that were specified for the RAG controller and, can further be proven by looking at figures 11-18 where the logic analyser is connected to the circuit.

During this assignment it required the pelican crossing to be adapted to work as a puffin crossing which meant an additional input was required to act a sensor to detect when a pedestrian wanted to cross as well as pressing a button.

The pelican crossing requires that if a pedestrian has crossed the road before the lights change to red then the sequence of events prior to that return to “state 0” which indicates a green light to traffic as shown in the state diagram.

The circuit that was adapted was the 0-5 counter as this was the most similar in design but also the one that controls the lights with the button. It can be seen that the Karnaugh maps for the 0-5 counter and the pelican crossing are identical other than the additional input of the pressure pad sensor. It was adapted by adding an additional input to all the and gates titled “P” as this would indicate the pressure pad sensor as shown in figure 24.

Within the research it is stated that FPGA microchips can be reprogrammed and are commonly used in traffic light systems. As a result of this the microchip being used for this assignment is an FPGA as the original designed program was for a puffin crossing but then redesigned to a pelican crossing. Also, these chips have been described as a number of digital sub-circuits which is what was used with the aid of PLD mode and a finite state machine.

CONCLUSIONS.

5. Conclusions and recommendations

Overall, it can be said that each circuit made done what it was specified to do. For example, it was proven that the 0-5 counter and 0-15 counter both counted non- repeatedly and also the specified sequence of events for the RAG controller was followed. Furthering this when all the sub circuits were designed and put together using an FSM it worked as a Pelican crossing and also, when adapted to include a pressure sensor it worked as a pelican crossing.

Further conclusion to make would be as stated in the analysis that an FPGA microchip was used due to the ability of it being able to be reprogrammed. If an ASIC microchip were to be used instead it would have to be a mass produced due to the cost, as it is known from research that they are more cost effective when used in that manner as opposed to the FPGA chip.

A recommendation that could be made as a result of the latest conclusion is that if this design was to mass produced then possibly looking at using an ASIC microchip would be a more cost effective alternative, as they would be built for a specific task.

Another sustainable feature that could be added as researched is the possibility of using solar panels to power the traffic lights as no greenhouse gasses would need to be produced. Also, by use of using LEDs over halogen light bulbs it enables a more environmentally friendly alternative.

6. Works Cited

7 energy ltd 2022, 2022. 7 energy. [Online]
Available at: <http://7energy.co.uk/wind/>
[Accessed 08 01 2022].

Crown Copyright 2005, 2005. the official DSA guide to DRIVING the essential skills. In: DSA, ed. *the official DSA guide to DRIVING the essential skills*. Nottingham: ISBN-9780115526412, p. 129 and 131.

electronics-notes.com, 2022. *electronicsnotes*. [Online]
Available at: <https://www.electronics-notes.com/articles/electronic components/programmable-logic/what-is-an-asic-application-specific-integrated-circuit.php>
[Accessed 16 01 2022].

Greenie World Poland, 2022. *Greenie polska*. [Online]
Available at: <https://greenie-world.com/en/article/the-advantages-and-disadvantages-of-led-lighting/>
[Accessed 08 01 2022].

HowStuffWorks, 2022. *howstuffworks*. [Online]
Available at: electronics.howstuffworks.com
[Accessed 08 01 2022].

John Bird, 2017. Electrical circuit theory and technology. In: Routledge, ed. *Electrical circuit theory and technology*. Abingdon: ISBN-9781138673496, pp. 86-87.

RF Wireless World 2012, 2012. *RF Wireless World*. [Online]
Available at: <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-FPGA.html>
[Accessed 16 01 2022].

SDR NEWS, 2021. *SDR NEWS*. [Online]
Available at: <https://sdr.news/latest/what-is-fpga-and-what-is-it-used-for/>
[Accessed 16 01 2022].

Sinowatcher technology Co, 2021. *Sinowatcher*. [Online]
Available at: <https://www.trafficsolution.cn/the-advantages-of-led-traffic-lights.html>
[Accessed 08 01 2022].

TheoryTest.org.uk, 2021. *Pedestrian Crossings*. [Online]
Available at: <https://theorytest.org.uk/pedestrian-crossings/>
[Accessed 19 October 2021].

Appendix

FINAL GRADE

GENERAL COMMENTS

Instructor

75/100

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

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

GRADING FORM: CSG-RUBIC

RESEARCH-INFORMED LITERATURE

 There was very good evidence of further reading and research from credible and reliable sources. Your interpretation and understanding of the additional research was good and demonstrated a very good underlying appreciation of the subject. Your references were appropriate to the task and applied with few errors or inconsistencies. Your reference list was of an appropriate length for this assignment.

6

KNOWLEDGE AND UNDERSTANDING

 The knowledge and understanding of the subject demonstrated by this assignment was generally very good. There were few gaps in your knowledge of the basic principles and your understanding of the higher level concepts was demonstrated to a high degree. You have demonstrated a competent critical understanding of the main concepts and expanded your knowledge beyond that contained in the course notes.

24

ANALYSIS



Your analysis of the task was excellent, both in class and as shown in this assignment. You have demonstrated highly competent analytical skills and your approach to the task was highly logical. You carried out a detailed evaluation of the results and included an excellent conclusion relating your practical results to accepted theory. You have demonstrated a capacity for independent thought and your interpretation of the quantitative and qualitative data was excellent with sound judgements based on wholly valid arguments.

25

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.

15

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.

5