

PSDE5006 SGEE5000, Electronics II Assignment 2018_19

by Ethan Howells

FILE	15325_ETHAN_HOWELLS_PSDE5006__SGEE5000__ELECTRONICS_II_AS SIGNMENT_2018_19_165369_386086337.DOCX (941.01K)		
TIME SUBMITTED	28-JAN-2019 11:16PM (UTC+0000)	WORD COUNT	4482
SUBMISSION ID	99951478	CHARACTER COUNT	20841

HNC ELECTRICAL & ELECTRONIC ENGINEERING

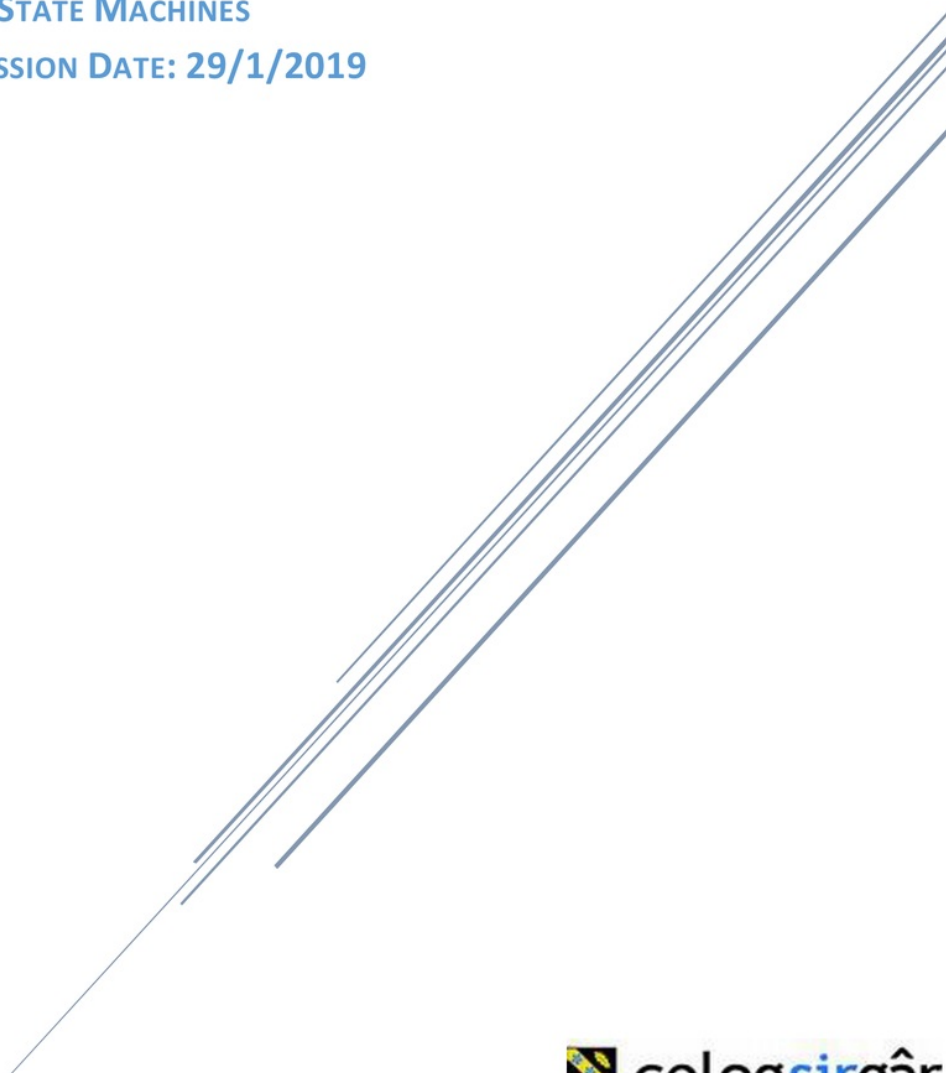
ETHAN HOWELLS

PSDE 5006

ELECTRONICS II

FINITE STATE MACHINES

SUBMISSION DATE: 29/1/2019



Contents

Executive Summary.....	2
Research	3
Design Method, Choices and Calculations	6
Modifying and Improving the Design.....	7
Conclusion.....	2
Reference List.....	3
References	3
Annexes.....	4
Figure 1 - Combinational vs Sequential Logic (Anon., 2019)	3
Figure 2 - Finite State Machine Example (Shead, 2018)	4
Figure 3 - FPGA vs ASIC (Jangra, 2014).....	4
Figure 4 - 0-5 Counter Circuit.....	9
Figure 5 - 0-5 Counter Logic Analyzer	9
Figure 6 - 0-15 Counter Circuit.....	15
Figure 7 - 0-15 Counter Logic Analyzer	16
Figure 8 - Main Controller Circuit	3
Figure 9 - Main Controller Logic Analyzer.....	4
Figure 10 -Main Controller Block Diagram.....	5
Figure 11 - Main Controller Circuit	5
Figure 12 - Logic Analyzer Main Controller	6

Executive Summary

The aim of this task was to design and build a Traffic Light Control System, which consists of a number of sub-circuits, designed to control the system. The Main aim of this task was to successfully develop and test the traffic light sequence, which will be functioning successfully by running through the traffic light sequence of; red, red amber, green and then red.

The first step was to complete research on the differences between combinational and Sequential Logic circuits. The use of a Finite State Machine as a designing tool was also explored as well as the advantages of using Field Programmable Gate Arrays and Application Specific Integrated Circuits. The research provided a good starting point to develop the circuits.

The circuits were all designed, using a finite state machine to create a state table, state diagram and finite state diagram. This enabled the Boolean Expressions to be found, which allowed the circuits to be created. There were three circuits designed the 0-5 Counter, 0-15 Counter and the RAG Controller. The Circuits were all tested individually by using a logic analyzer, which shows the circuit counting. The circuits were then created into sub-circuits and combined to create the Traffic Light Control System. This was tested as well, by using the logic analyzer and by also using coloured LEDs on the outputs to represent the traffic lights.

The final circuit test showed that the circuit was functioning as expected, as the LED lights on the output followed the sequence that was expected and the logic analyzer showed the circuit counting as expected. To further improve the system, the traffic light sequence could be modified into a puffin crossing, that includes a pressure plate. The Pressure Plate will replace the push button, so this will trigger the sequence when someone stands on the pressure plate. The sequence will re-set when someone steps off the pressure plate, preventing people starting the sequence and walking off.

Research

Combinational and Sequential Logic

Both Combinational and Sequential Logic are the building blocks of digital circuits. Combinational logic is an independent circuit, that contains no memory and so therefore acts only on its present input to produce an output. Sequential however contains a memory element, which means the output depends upon past as well as present inputs. Due to it containing no memory, combination circuits are a lot faster than sequential circuits. Due to this the designing process of sequential logic is much more difficult than that of a combinational circuit (Coach, 2019). Combinational circuits also do not contain a clock as they do not require trigger, sequential circuits however are clock dependent as they require triggering. (Aishwarya, 2019) As Combinational Circuits do not have a memory they are generally used for basic Boolean expressions, whereas sequential logic can be used to create memory elements. Sequential logic also contain feedback from the output, however combinational logic does not require this function. The Traffic Light Control System will be made up of three individual Sequential Logic circuits; 0-5 Counter, 0-15 Counter and RAG Controller. (Anon., 2013)

Combinational vs. Sequential Logic

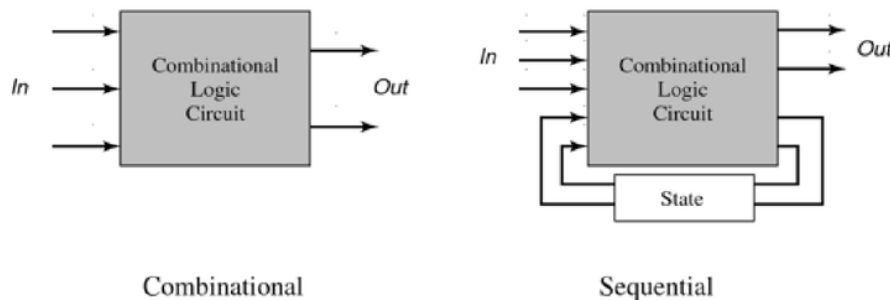


Figure 1 - Combinational vs Sequential Logic (Anon., 2019)

Finite State Machine

A Finite State Machine can be used as a designing tool, when designing and creating circuits. A Finite State Machine is a device that stores previous inputs as well as its current input which then produces an output. There are a series of states within a state machine and the inputs will determine the state of the machine. An example of a finite state machine diagram can be seen in Figure 2, Where S and Q represent the states, B and A are the inputs and the arrows represent the transitions. (Shead, 2018)

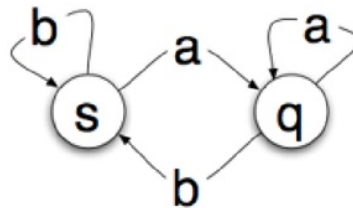


Figure 2 - Finite State Machine Example (Shead, 2018)

The two most common types of finite state machines are the Mealy and Moore machines. A Mealy machine produces an output that depends on both its present state and present input, due to this the Mealy Machine generally has fewer states. A Moore machine however, produces an output purely on its present state, this allows the Moore machine to have more states. Moore Machines require more logic, and this introduces circuit delays, meaning the circuit reacts slower than a Mealy machine (Point, 2018).

FPGA's and ASIC's

Field Programmable Gate Arrays (FPGAs) are semiconductor devices, that allow re-configuration of a circuit, they are made up of configurable logic blocks, which can implement complex logic functions. FPGA's are generally used in the prototyping stages of circuits, because if an error was to occur the circuit can be altered and re-downloaded. FPGA are also relatively cheap for purchase. Due to its ability to allow reprogramming the FPGA is not suited for mass production. FPGA's have poor energy efficiency.

Application Specific Integrated Circuits (ASICs) are circuits that are designed for one individual application, this means that the logic function cannot be changed or altered as the circuit is implemented in silicon. ASIC are beneficial as they improve the speed of the circuit as they are solely designed for one purpose. Due to their application, ASIC can be extremely expensive but are very well suited for mass production. ASIC are much more energy efficient compared to FPGA's and can provide the same functions with less power. (Jayaraman, 2018)

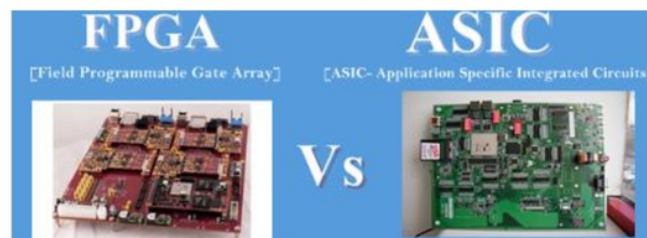


Figure 3 - FPGA vs ASIC (Jangra, 2014)

Sustainable Development

In recent years, sustainable development has been a major factor in all aspects. As electronics have developed the size and energy consumption of components has had a huge influence on sustainable development. The general improvement of electronic components has led to improvements, in sustainability. Modern components have become more reliable, more efficient and have a longer life span, therefor saving the consumer money and energy. An example of this can be seen in the development of solid-state relays, these have been developed and can be used to replace contactors. As there are no moving parts in an SSR, this dramatically reduces the maintenance costs and there for increases the lifetime of the switch. The SSR also has better control than a contactor and therefore reduces energy usages as well as providing PID control. (Energy, 2019)

Developments of Application Specific integrated Circuits (ASIC) have also aided with sustainable development, as have been designed smaller over timer and this in turn reduces the amount of electricity needed, which is beneficial to the environment. (Techopedia, 2019)

Design Method, Choices and Calculations

The task faced was to design a traffic light control system. The first stage in designing the traffic light sequence was to create a 0-5 Counter

0 – 5 Counter

State Allocation

The First stage taken to create the 0-5 counter, was to create a State Allocation Table. This shows the outputs at every state.

Step 0 (S0)	Counts 0
Step 1 (S1)	Counts 1
Step 2 (S2)	Counts 2
Step 3 (S3)	Counts 3
Step 4 (S4)	Counts 4
Step 5 (S5)	Counts 5

Inputs

WB = 0 = Wait Button not pressed.

WB = 1 = Wait Button Pressed

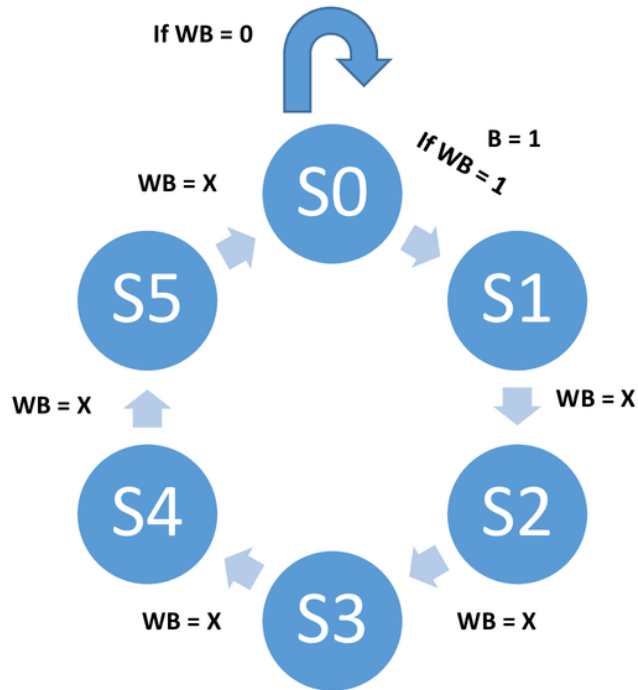
Outputs

W = 0 = Timer counting, or the button has not pressed

W = 1 = Timer equals 5s, which then triggers the sequence.

State Diagram

Using the State Allocation table, the State diagram was created which shows the steps of the sequence of the 0-5 Counter. The Sequence starts when the wait button is pressed.



Finite State Table

The Finite State Table shows what bits are currently on and then what bits are needed for the next step.

QA = 1

QB = 2

QC = 4

			Current			B = 0			B = 1			Output
Current	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	1	1	0	1
S6	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

Karnaugh Maps

The following Karnaugh Maps were produced from the finite state table. Boolean expressions were then derived from the Karnaugh Maps, by obtaining the common variables from each group.

QA⁺

QBQC BQA	00	01	11	10
00	0	0	1	0
01	1	0	X	X
11	1	0	X	X
10	0	0	1	0

$$QA^+ = QA\overline{Q}\overline{C} + QBQC$$

QB⁺

QBQC BQA	00	01	11	10
00	0	1	0	1
01	0	0	X	X
11	0	0	X	X
10	0	1	0	1

$$QB^+ = \overline{Q}A\overline{Q}\overline{B}QC + QB\overline{Q}\overline{C}$$

QC⁺

QBQC BQA	00	01	11	10
00	0	0	0	1
01	1	0	X	X
11	1	0	X	X
10	1	0	0	1

$$QC^+ = QA\overline{Q}\overline{C} + B\overline{Q}\overline{C} + QB\overline{Q}\overline{C}$$

Using the Boolean expressions, the 0-5 Counter Circuit could be designed using Multi-Sim, as seen in Figure 4. In order to, test the circuit, a digital constant was used to mimic the pedestrian button. A Logic analyser was added to see the circuit counting, this can be seen in Figure 5, where QC equals 1, QB equals 2 and QC equals 4.

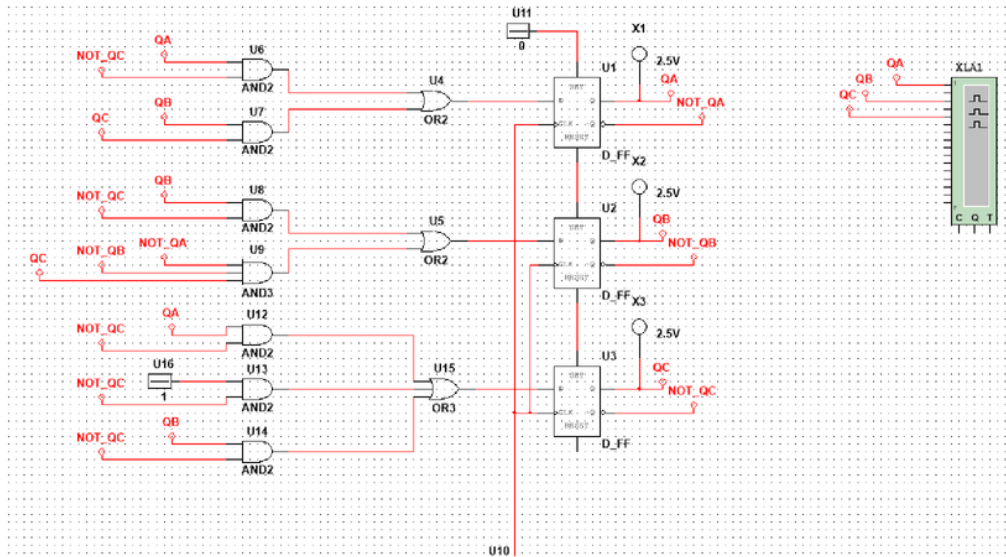


Figure 4 - 0-5 Counter Circuit

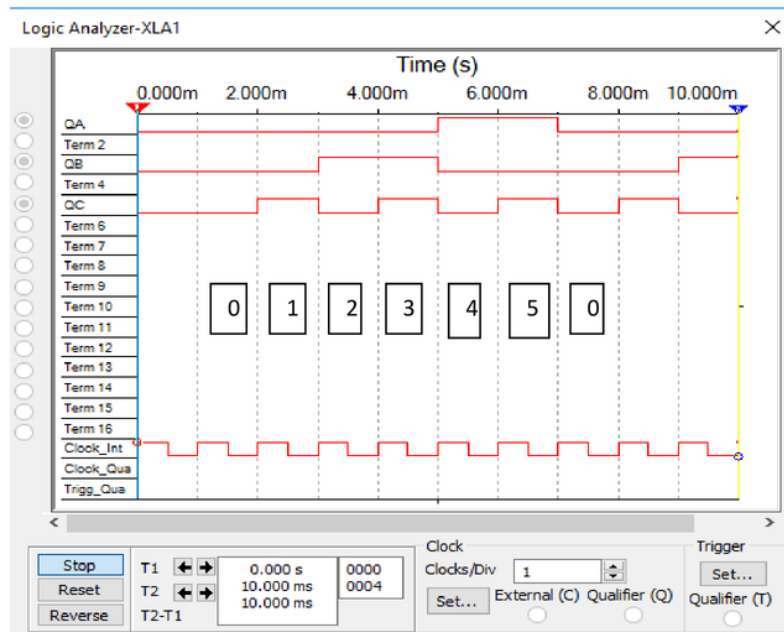


Figure 5 - 0-5 Counter Logic Analyzer

15 Counter

The next stage in order to create our traffic light sequence, was to design and create a 15 second counter.

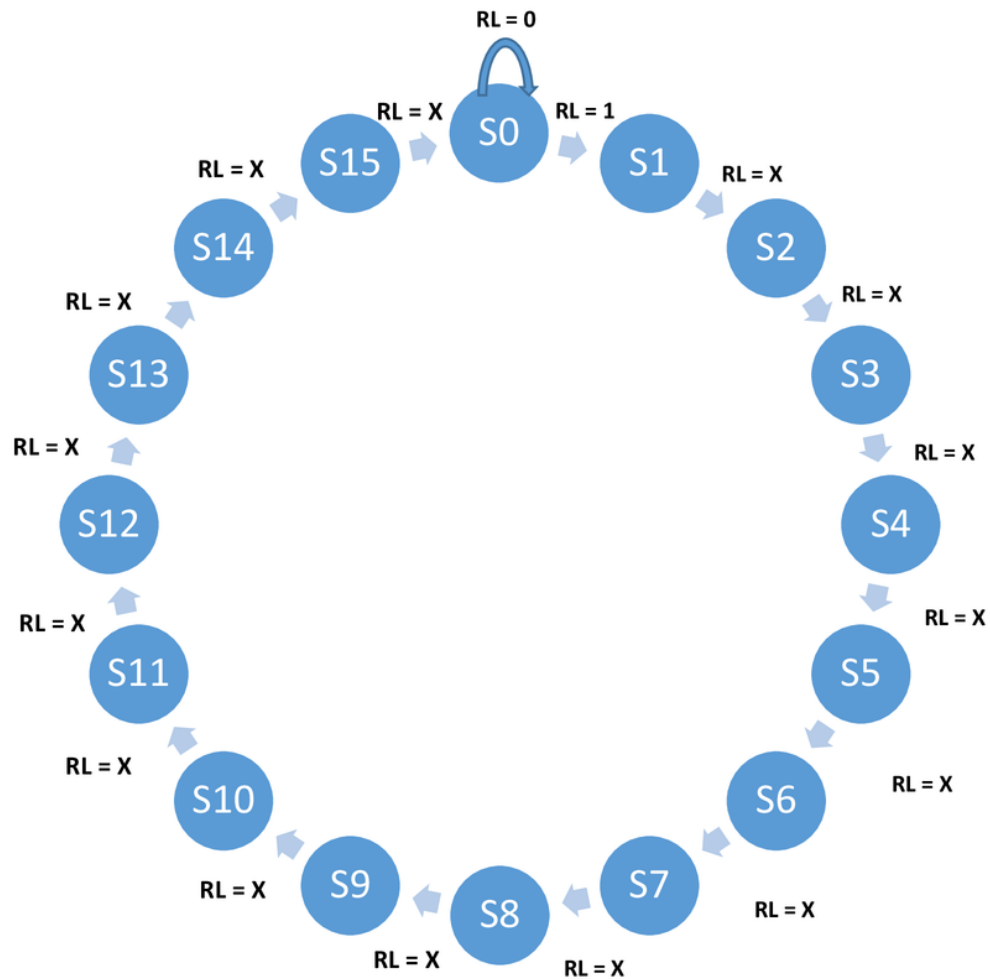
State Allocation Table

This shows the outputs at each step.

Step 0 (S0)	Count 0
Step 1 (S1)	Count 1
Step 2 (S2)	Count 2
Step 3 (S3)	Count 3
Step 4 (S4)	Count 4
Step 5 (S5)	Count 5
Step 6 (S6)	Count 6
Step 7 (S7)	Count 7
Step 8 (S8)	Count 8
Step 9 (S9)	Count 9
Step 10 (S10)	Count 10
Step 11 (S11)	Count 11
Step 12 (S12)	Count 12
Step 13 (S13)	Count 13
Step 14 (S14)	Count 14
Step 15 (S15)	Count 15

State Diagram

The State Diagram was then created for the 15 Counter, the sequence starts when the Red Light is illuminated.



Inputs

Red Light (RL) = 0 (When the amber or green lights are illuminated)

Red Light (RL) = 1 (When the red light is illuminated)

Outputs

Hold (H) = 0 (The Timer is either counting or the red light is not illuminated)

Hold (H) = 1 (The Time equals 15 Seconds, this then triggers the next part of the sequence)

Finite State Table

QD represents binary 1, QC represents binary 2, QB represents binary 4 and QA represents binary 8.

							RL 0				RL 1				Outputs
	RL 0	RL 1	Current				Next								
Current	Next		QA	QB	QC	QD	QA ⁺	QB ⁺	QC ⁺	QD ⁺	QA ⁺	QB ⁺	QC ⁺	QD ⁺	
S0	S0	S1	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
S2	S3	S3	0	0	2	0	0	0	1	1	0	0	1	1	0
S3	S4	S4	0	0	1	1	0	1	0	0	0	1	0	0	0
S4	S5	S5	0	1	0	0	0	1	0	1	0	1	0	1	0
S5	S6	S6	0	1	0	1	0	1	1	0	0	1	1	0	0
S6	S7	S7	0	1	1	0	0	1	1	1	0	1	1	1	0
S7	S8	S8	0	1	1	1	1	0	0	0	1	0	0	0	0
S8	S9	S9	1	0	0	0	1	0	0	1	1	0	0	1	0
S9	S10	S10	1	0	0	1	1	0	1	0	1	0	1	0	0
S10	S11	S11	1	0	1	0	1	0	1	1	1	0	1	1	0
S11	S12	S12	1	0	1	1	1	1	0	0	1	1	0	0	0
S12	S13	S13	1	1	0	0	1	1	0	1	1	1	0	1	0
S13	S14	S14	1	1	0	1	1	1	1	0	1	1	1	0	0
S14	S15	S15	1	1	1	0	1	1	1	1	1	1	1	1	0
S15	S0	S0	1	1	1	1	0	0	0	0	0	0	0	0	1

Karnaugh Maps

The following Karnaugh Maps were produced from the finite state table. Boolean expressions were then derived from the Karnaugh Maps.

QA⁺

RL = 0

QBQC BQA	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	1	0	1
10	1	1	1	1

QA⁺

RL = 1

QBQC BQA	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	1	0	1
10	1	1	1	1

$$QA^+ = QA\overline{QC} + \overline{QA}QBQCQD + QA\overline{QB}$$

QB⁺

RL = 0

QBQC BQA	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	1	1	0	1
10	0	0	1	0

RL = 1

QBQC BQA	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	1	1	0	1
10	0	0	1	0

$$QB^+ = QB\overline{QC} + \overline{QB}QCQD + QB\overline{QD}$$

QC⁺

RL = 0

QBQC BQA	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

RL = 1

QBQC BQA	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$$QC^+ = \overline{QC}QD + QC\overline{QD}$$

QD⁺

RL = 0

QBQC BQA	00	01	11	10
00	0	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

RL = 1

QBQC BQA	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

$$QD^+ = QA\overline{QD} + QB\overline{QD} + QC\overline{QD} + \overline{QD}RL$$

Using the Boolean Expressions, the circuit below was designed, as seen in Figure 6. A digital constant was used to represent the Pedestrian Crossing Button and a logic Analyser, to show the circuit Counting.

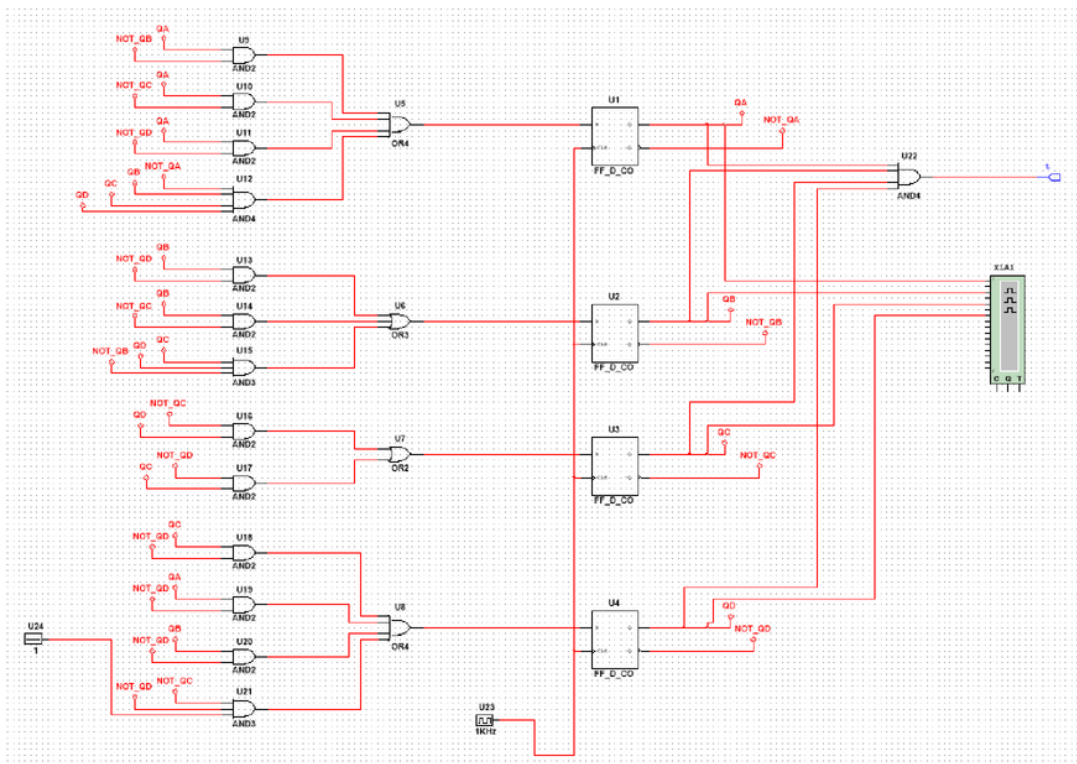


Figure 6 - 0-15 Counter Circuit

The Logic Analyzer proved that the circuit functioned correctly, as it shows the circuit counting from 1 up to 15. On the graph QA represents 8, QB represents 4, QC represents 2 and QA represents 1. This can be seen in Figure 7.

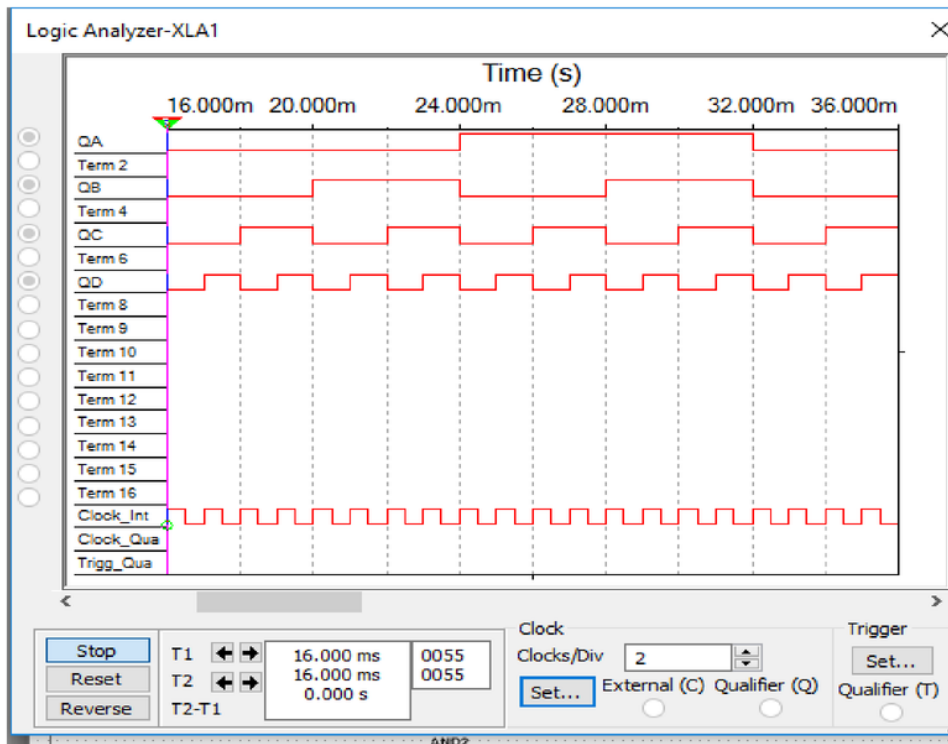


Figure 7 - 0-15 Counter Logic Analyzer

RAG Controller

The next step was to create our RAG controller circuit, which produces an output depending on the flip flops output.

State Allocation Table

A State Allocation table was created to show the outputs at each step.

Step 0 (S0)	Green
Step 1 (S1)	Amber
Step 2 (S2)	Amber
Step 3 (S3)	Red
Step 4 (S4)	Red / Amber
Step 5 (S5)	Red / Amber

Inputs

WB = 0 No P/T or Circuit Counting

WB = 1, The Count is at 5

T = 0, Counting

T = 1, The Count is at 15

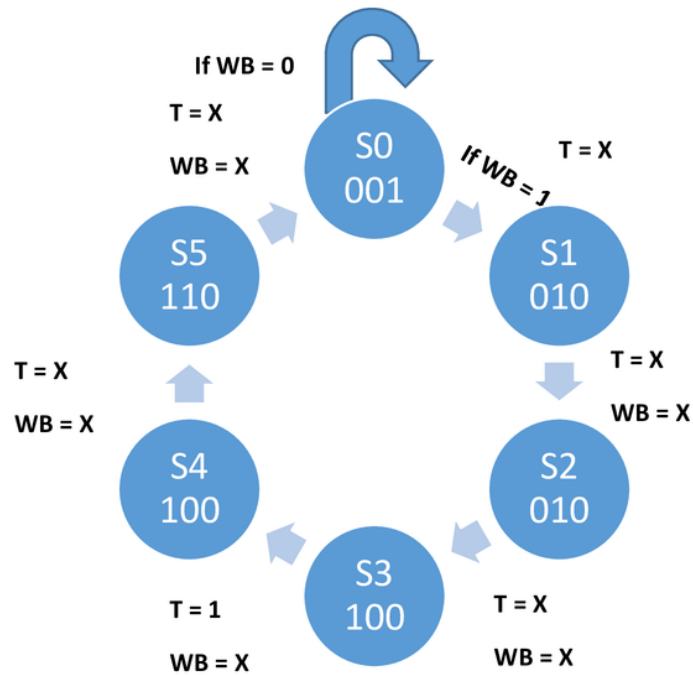
Outputs

R/A/G – 0 = Off

1 = On

State Diagram

Once the State Allocation was designed, I was then able to create my State Diagram for my RAG Controller. A state diagram shows the steps of a sequence. When the Wait Button (WB) is pressed the sequence starts.



Karnaugh Maps

The following Karnaugh Maps were produced from the finite state table. Boolean expressions were then derived from the Karnaugh Maps

QA⁺

RL = 0

QBQC BQA	00	01	11	10
00	0	0	0	0
01	1	0	X	X
11	1	0	X	X
10	0	0	1	0

QA⁺

RL = 1

QBQC BQA	00	01	11	10
00	0	0	0	0
01	1	0	X	X
11	1	1	X	X
10	0	0	1	0

$$QA^+ = QA\overline{QC} + TBC$$

QB⁺

RL = 0

QBQC BQA	00	01	11	10
00	0	1	1	1
01	0	0	X	X
11	0	0	X	X
10	0	1	0	1

RL = 1

QBQC BQA	00	01	11	10
00	0	1	1	1
01	0	0	X	X
11	1	0	X	1
10	0	0	0	1

$$QB^+ = \overline{QA}QBQC + \overline{T}QB + QB\overline{QC}$$

QC⁺

RL = 0

QBQC BQA	00	01	11	10
00	0	0	1	1
01	1	0	X	X
11	1	0	X	X
10	0	0	0	1

RL = 1

QBQC BQA	00	01	11	10
00	1	0	1	1
01	1	0	X	X
11	1	0	X	X
10	1	0	0	1

$$QC^+ = QA\overline{QC} + \overline{T}QB + QB\overline{QC} + WQB\overline{QC}$$

Red

QBQC WQA	00	01	11	10
0	0	0	1	0
1	1	1	X	X

$$QA + QBQC$$

Amber

QBQC WQA	00	01	11	10
0	0	1	0	1
1	1	1	X	X

$$QA + \overline{QB}QC + QB\overline{QC}$$

Green

QBQC WQA	00	01	11	10
0	1	0	0	0
1	0	0	X	X

$$\overline{QAQCQB}$$

pg. 20

Figure 9 shows the Logic Analyzer from the Main Controller. This shows the circuit counting from 1-5. Where QC represents 1, QB represents 2 and QA represents 4.

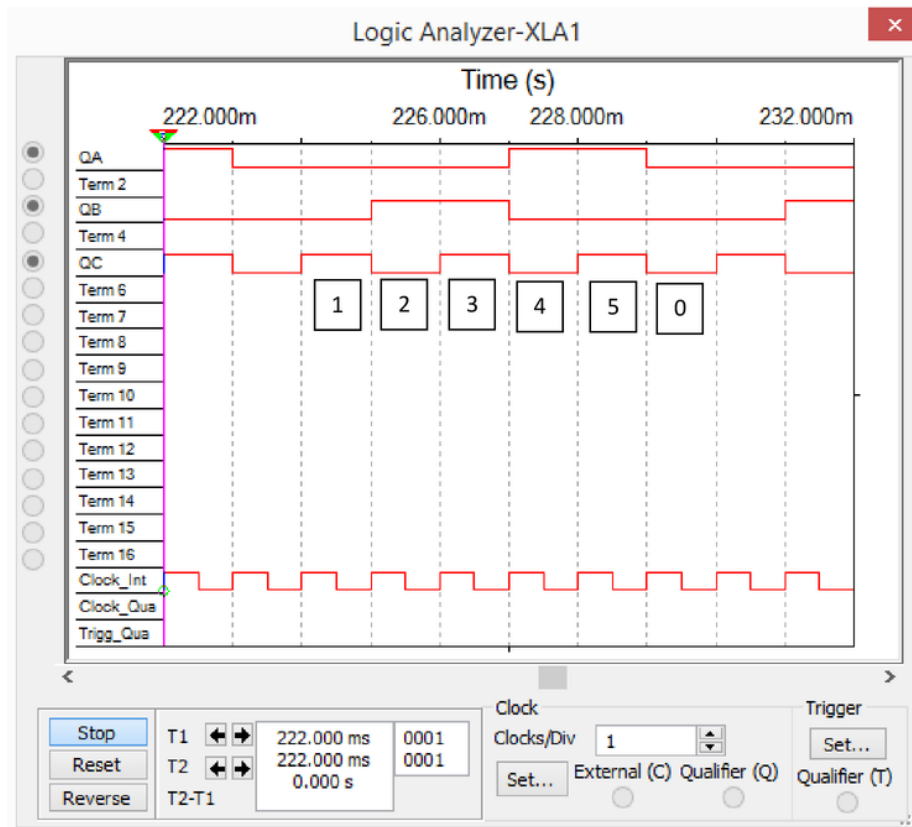


Figure 9 - Main Controller Logic Analyzer

Main Controller

The Main Controller was then built, by making the 0-5 Second, 0-15 Second and RAG circuit into sub-circuits. The Main Controller is used to turn the signals from the RAG controller into the Red, Amber and Green lights. The 5 Second Counter provides the W input into the Main controller and the 15 Second Timer provides the T input, however it gets an input from the Main controller to start counting.

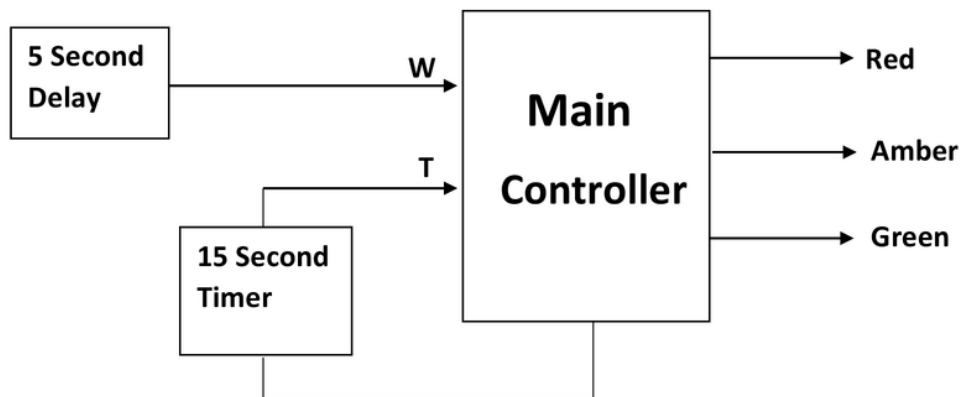


Figure 10 -Main Controller Block Diagram

LED lights were added to the output of the RAG Controller, to represent the Traffic Lights. A Logic Analyzer was also added to show the sequence of the lights. The Circuit can be seen in Figure 11.

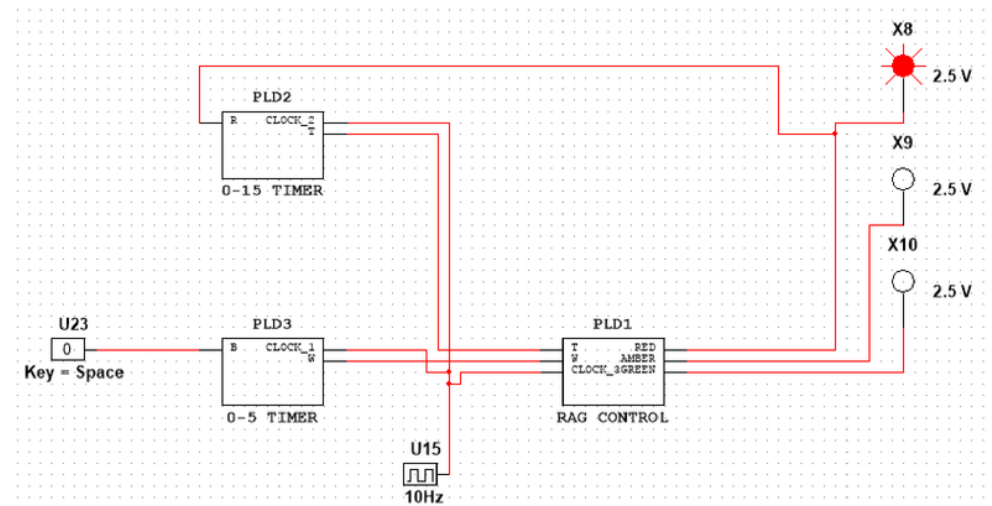


Figure 11 - Main Controller Circuit

Figure 12 shows the Logic Analyzer of the Main Controller circuit. This shows the sequence:

- Red
- Red, Amber
- Green
- Amber
- Red

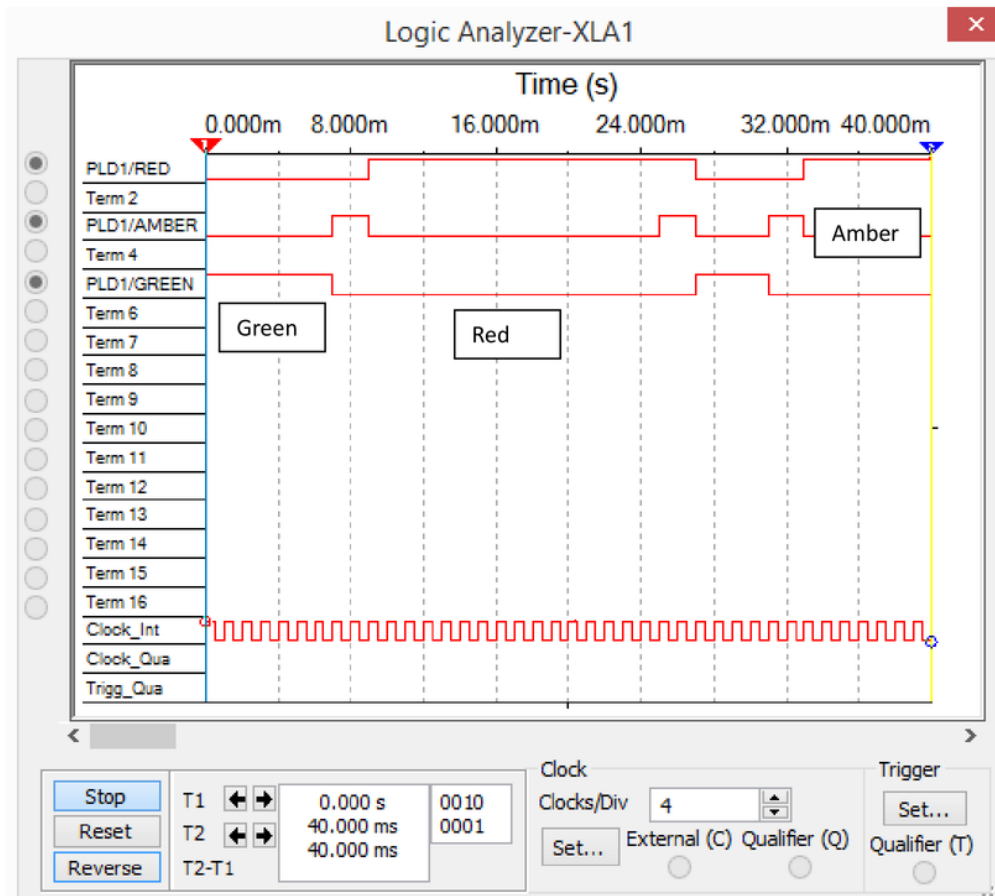


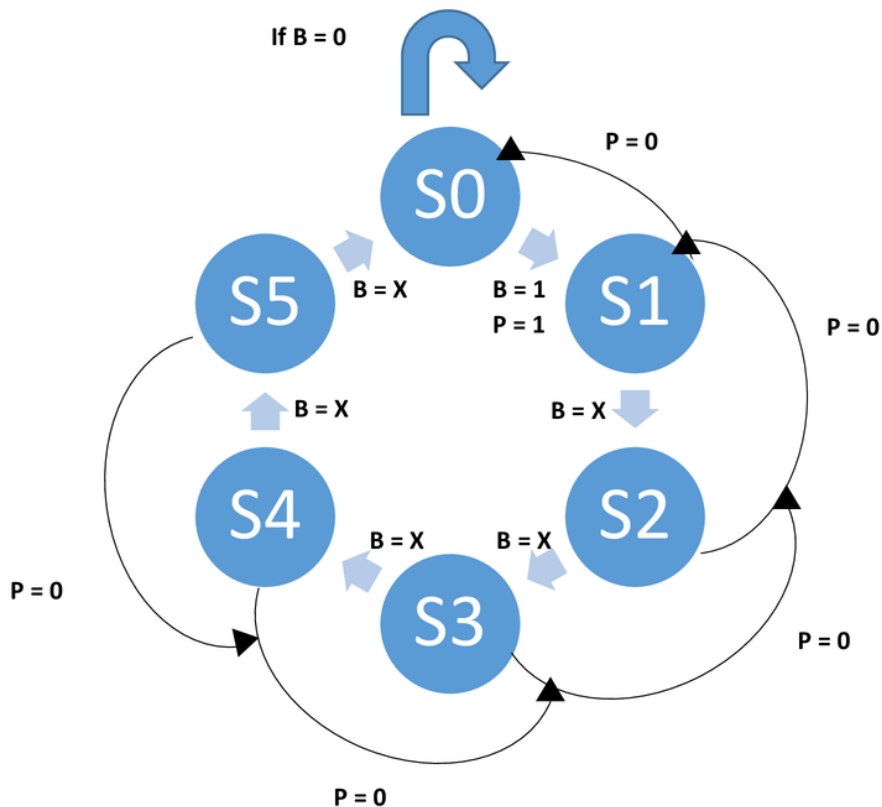
Figure 12 - Logic Analyzer Main Controller

3. Modifying and Improving the Design

Once the Traffic Light Sequence was working, the next step was to modify the design in order to improve the Traffic Light Sequence. The introduction of a pressure plate will improve the system, as once the pressure plate is energised the traffic light sequence will start, turning the traffic light sequence into a Puffin Crossing. The introduction of the pressure plate will also prevent people pressing the traffic light sequence button and walking away.

State Diagram

A State Diagram was created that shows the different states and transitions of the circuit. It can be seen that when the Pressure Plate equals 0, the circuit re-sets back to Step 0 starting the sequence again.



Karnaugh Maps

The following Karnaugh Maps were produced from the finite state table. Boolean expressions were then derived from the Karnaugh Maps

QA⁺

PP = 0

QC QD WB 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

WB = 1

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

$$QA^+ = QA\overline{QD}PP + WB\overline{QC}QDPP$$

QB⁺

PP = 0

QBQC BQA	00	01	11	10
00	0	0	0	0
01	0	0	X	X
11	0	0	X	X
10	0	0	0	0

PP = 1

QBQC BQA	00	01	11	10
00	0	1	0	1
01	0	0	X	X
11	0	0	X	X
10	0	1	0	1

$$QB^+ = \overline{QA}\overline{QC}QDPP + \overline{T}QB + QB\overline{QC}$$

QC^+

$PP = 0$

QCQD WBQA	00	01	11	10
00	0	0	0	0
01	0	0	X	X
11	0	0	X	X
10	0	0	0	0

$PP = 1$

QCQD WBQA	00	01	11	10
00	0	0	0	1
01	1	0	X	X
11	1	0	X	X
10	1	0	0	1

$$QC^+ = QA\overline{QC} + \overline{T}QB + QB\overline{QC} + WB\overline{QC}$$

Conclusion

To conclude, I believe I carried out successful research on understanding the differences between combinational and sequential logic, as well as displaying the use of Finite State Machine as a design tool. I also explored the advantages of both Field Programmable Gate Arrays (FPGA) and Application Specific Integrated Circuits (ASICs), as well as the use of modern electronic components in the promotion of sustainable development.

The Research proved beneficial throughout this assignment as the circuits we were designing were all sequential circuits. All the circuits were designed using a finite state machine, as we created a state allocation table, which established the output on each state. I then created state diagram which shows different states and how inputs affect the state. A Truth table was then established from the state diagram. Karnaugh Maps were derived from truth tables. Boolean expressions are gathered from the Karnaugh maps, and these allow us to design the circuits. It can also be seen through the research that it would be beneficial to test the circuits on a Field Programmable Gate Array (FPGA) rather than an Application Specific Integrated Circuit (ASIC), as they have the ability to be reprogrammed, Also FPGA are also a lot cheaper when prototyping compared to ASIC's. However, once the circuit is finalized and working as expected, it would be better to implement the circuit on a ASIC, as this circuit can only perform one function and so used less energy to provide the same function as a FPGA, also the ASIC's are much cheaper in large batches compared to FPGA's.

Throughout this assignment I developed a Traffic Light Control System, and this was completed by creating three sub-circuits; 0-5 Counter, 0-15 Counter and a RAG controller. These sub-circuits were then all tested individually and combined to create the main controller (Traffic Light Control System). I could then test the final circuit by applying a logic analyzer and LED probes on the outputs, which will represent the traffic lights sequence. The Circuit worked as expected. One recommendation that could be made would be to add a audible bleeper into the circuit, so when the traffic light is green the audible beeper will sound letting pedestrians know its safe to cross.

Reference List

References

- Aishwarya, 2019. *Difference between combinational and sequential circuit*. [Online]
Available at: <https://www.geeksforgeeks.org/difference-between-combinational-and-sequential-circuit/>
[Accessed 23 01 2019].
- Anon., 2013. *Difference Between Combinational and Sequential Logic*. [Online]
Available at: <https://www.differencebetween.com/difference-between-combinational-and-sequential-logic/>
[Accessed 25 1 2019].
- Anon., 2019. *Combinational and Sequential Logic Circuits - Digital Circuits*. [Online]
Available at: <http://ebooksforgate.blogspot.com/2014/01/combinational-and-sequential-logic.html>
[Accessed 25 01 2019].
- Coach, E., 2019. *Difference between Combinational and Sequential Logic Circuit*. [Online]
Available at: <https://electronicscoach.com/difference-between-combinational-and-sequential-logic-circuit.html#KeyDifferences>
[Accessed 25 01 2019].
- Energy, A., 2019. *SCR Power Controllers*. [Online]
Available at: <https://www.advancedenergy.com/globalassets/resources-root/infographics/scr-advantages-infographic.pdf>
[Accessed 22 01 2019].
- Jangra, A., 2014. *FPGA VS ASIC*. [Online]
Available at: <http://siliconmentor.blogspot.com/2014/07/fpga-vs-asic.html>
[Accessed 25 01 2019].
- Jayaraman, R., 2018. *FPGA Vs ASIC: Differences Between Them And Which One To Use?*. [Online]
Available at: <https://numato.com/blog/differences-between-fpga-and-asics/>
[Accessed 23 01 2019].
- Point, T., 2018. *Moore and Mealy Machines*. [Online]
Available at: https://www.tutorialspoint.com/automata_theory/moore_and_mealy_machines.htm
[Accessed 25 01 2019].
- Shed, M., 2018. *Understanding State Machines*. [Online]
Available at: <https://medium.freecodecamp.org/state-machines-basics-of-computer-science-d42855debc66>
[Accessed 26 01 2019].
- Techopedia, 2019. *Application-Specific Integrated Circuit (ASIC)*. [Online]
Available at: <https://www.techopedia.com/definition/2357/application-specific-integrated-circuit-asic>
[Accessed 25 01 2019].

Annexes

PSDE5006 SGEE5000, Electronics II Assignment 2018_19

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

82/100

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

GRADING FORM: ASSIGNMENT RUBRIC 2018

ETHAN HOWELLS

82

ENGAGEMENT WITH LITERATURE

There was evidence of extensive further reading



and research from credible and reliable sources. Your interpretation and understanding of the additional research was very good and demonstrated a deep underlying appreciation of the subject. Your references were consistently applied and appropriate to the task and your reference list was of an appropriate length for this assignment.

8

KNOWLEDGE AND UNDERSTANDING



The knowledge and understanding of the subject demonstrated by this assignment was generally excellent. There were very 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 highly competent critical understanding of the main concepts and expanded your knowledge well beyond that contained in the course notes.

25

COGNITIVE AND INTELLECTUAL



Your explanation of the design process was good and the conclusions showed that you can interpret quantitative and qualitative data, although this was often descriptive. You could have evaluated the results in more detail and specified whether they concurred with theory suggesting reasons for any discrepancies. You have however demonstrated good analytical skills and your approach to the task was logical. Arguments made in the conclusions were mostly valid and demonstrated a good

23

understanding of the topic. Further work – in your conclusion, you should relate the results to the original specification and to accepted theory.

PRACTICAL APPLICATION



Your application of practical skills for this task was generally excellent. You used both the basic and advanced features of the resources appropriately when designing the circuit and you used appropriate advanced options for testing and evaluating the circuit performance. You managed to design an effective circuit that functioned well within the specification and achieved this by rarely asking for assistance thus demonstrating some advanced problem solving skills. Your results were comprehensive and allowed for a detailed conclusion to be written.

17

TRANSFERABLE SKILLS FOR LIFE



In this assignment you have demonstrated skills in working individually and also collaboratively. Your written report was exceptionally well structured and written in a highly coherent and logical way. The report was presented in a professional manner using appropriate language and formatting, good diagrams in appropriate places, good sentence/paragraph structure and clear calculations.

9