

1809811

by Gari Davies

FILE	12817_GARI_DAVIES_1809811_174902_891562297.PDF (3.46M)		
TIME SUBMITTED	27-JAN-2020 11:48PM (UTC+0000)	WORD COUNT	4520
SUBMISSION ID	119211989	CHARACTER COUNT	31615

HNC Electrical and Electronic
Engineering
SGEE5000 – Electronics II
Finite State Machines

Gari Davies

1809811

28/01/2020

Word Count: 2673

Table of Contents

Executive Summary.....	3
Research into advanced digital design techniques.....	4
The differences between combinational and sequential logic.....	4
The use of a Finite State Machine as a design tool.....	5
The relative merits of FPGA's and ASIC's	6
Modern electronic components and designs to promote sustainable development	7
Mobile Phones	7
Design method, choices and calculations	8
5 Sec Delay.....	10
State Diagram	10
State Table	11
Karnaugh Map.....	12
Multisim 5 second timer.....	13
15 Second Timer	15
State Table	16
Karnaugh Map.....	17
Multisim 15 second timer.....	18
RAG Controller.....	20
State Table	21
Karnaugh Map.....	22
Multisim RAG Controller.....	23
Testing, results and analysis (500).....	24
5 Second Timer	24
15 Second Timer	26
RAG Controller	28
Complete Design.....	30
Conclusions and recommendations.....	34
Circuit Adaption	34
State Table	35
Karnaugh Map.....	36
Conclusion	38
Works Cited	39

Executive Summary

In this report we will examine the research into different types of logics, finite state machines, FPGA's and ASIC's

A task was undertaken to build a pelican Traffic control system a design was created with a 5 second delay timer between the pedestrian pressing the wait button and the lights changing, a 15 second timer was also needed to give adequate time for pedestrians to cross and to enable the system to know when to rest the traffic lights for the vehicles to proceed.

The design was derived by using a finite state machine, state table and the results used in a Karnaugh maps to ascertain the logic gates required for each stage of the design process.

When this system was completed and tested to be satisfactory a further improvement was requested to include a pedestrian pressure pad, this was so that a person had to be standing on the pad for the duration of the 5 second time delay in order for the system to activate the traffic lights. The 5 second timer was redesigned using the same methods used in the first design but now including another variable the pressure pad. This redesign was constructed on Multisim and tested and the outcome found to be satisfactory.

Research into advanced digital design techniques

The differences between combinational and sequential logic

The difference between combinational and sequential logic circuit is that the combinational logic circuit consists of only logic gates while the sequential logic circuits consist of logic gates and memory elements.

A crucial factor which differentiates combinational and sequential logic circuit is the state of output elements. The state of the output elements of the combinational circuit depends on only present state of the input elements. On the contrary, the output obtained from sequential circuit depends on the present state of the input as well as past state of the output. (geeksforgeeks, 2020)

Parameters	Combinational Logic Circuit	Sequential Logic Circuit
Definition	It is a logic circuit which generates output based upon present inputs.	It is logic circuit which generates output based on present input as well as past output.
Memory Element	Memory element is not present in combinational logic circuit.	Memory element is present in sequential logic circuit.
Feedback loop	Feedback loop is not present.	Feedback loop is present.
Output obtained	Combination of operands not necessary in same sequence.	Sequence of operands is necessary in sequential circuit.
Examples	Half Adder, Full Adder, Encoder, decoder, multiplexer, demultiplexer .	Finite state machine

Figure 1: Comparison of combinational and sequential logic

The combinational logic circuit comprises of logic gates and thus the output obtained is directly related to the input. There are no feedback elements in case of the Combinational logic circuit.

A combinational logic circuit consists of logic gates such as NAND, NOR and NOT gates.

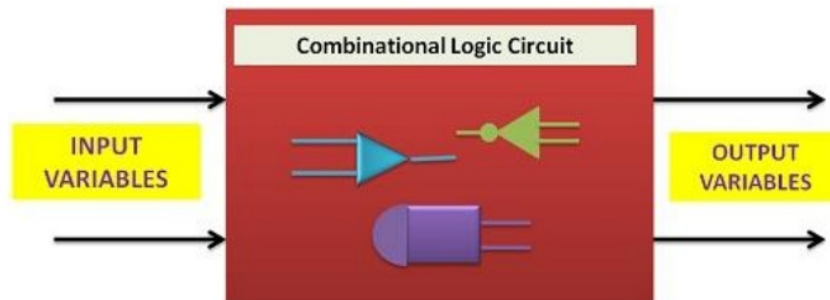


Figure 2: Combinational circuit

The output generated by the sequential circuit depends not only on the present state of the input but also on the previous outputs. The sequential circuit involves usage of feedback loops; with the help of this the state of the previous output is recorded. Therefore, the next output is controlled by the state of the previous output. The memory present in the sequential circuit keeps the track of the output and the output is produced. The most popular example of the sequential circuit is the finite state machine. (geeksforgeeks, 2020)

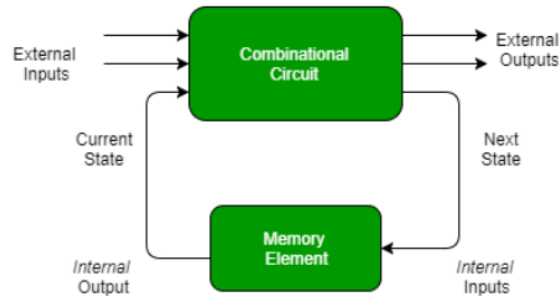


Figure 3: Sequential circuit

The use of a Finite State Machine as a design tool

An FSM can be informally defined as a device that accepts a finite number of different inputs and can produce a finite number of diverse outputs. A fundamental requirement is that the device must have some internal memory that allows it to remember sequences of inputs, so that the output is not only dependent on the combination of input values but also on the order in which they were applied. (Embedded, 2020)

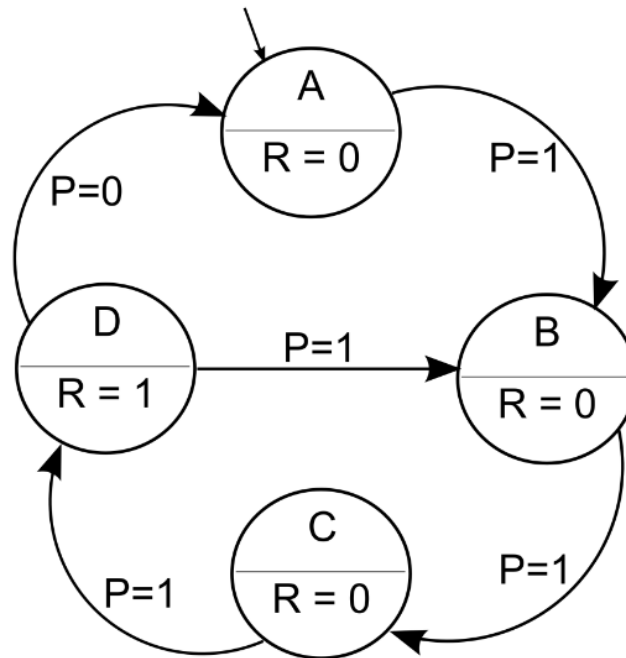


Figure 4: Finite state machine

The relative merits of FPGA's and ASIC's

A Field Programmable Gate Array (FPGA) is a device that is configurable by using a binary image file to implement the required functionality.

A Field-Programmable Gate Array (FPGA) is an integrated circuit silicon chip which has array of logic gates and this array can be programmed in the field i.e. the user can overwrite the existing configurations with its new defined configurations and can create their own digital circuit on field. The FPGAs can be considered as blank slate. FPGAs do nothing by itself whereas it is up to designers to create a configuration file often called a bit file for the FPGA. (Numato, 2020)



Figure 5 - FPGA

An Application Specific Integrated Circuit (ASIC) is a custom designed integrated circuit that is dedicated for one particular application. Both digital and analogue functions can be implemented and with a wide array of process choices available.

ASICs are specific chips used to implement both analogue and digital functionalities in high volume or high performance. ASICs are full custom therefore they require higher development costs in order to design and implement. Unlike the FPGAs chips, they are not reprogrammable and therefore charges incurred to change design



Figure 6 - ASIC chip

On the other side, however, ASICs are much denser, and you can integrate several different functionalities into one chip and therefore offer small size, low power and low-cost solution. (Numato, 2020)

FPGA	ASIC
* Field Programmable Gate Array * FPGA based designs follow a relatively simpler design process Functional Specification --> HDL --> Synthesis --> Place and Route --> Timing Verification --> Download code on FPGA and Verify * Since hardware is already available, no non-recurring expenses (NRE) * Because of above 2 reasons, faster "time-to-market" * Same FPGA can be reprogrammed to suit another application, so reusable and flexible * Standard array design and hence most times, hardware (array cells, transistors and interconnects) goes un-used. Wastage is un-avoidable. * Better suited if the required numbers is less * Relatively larger than an ASIC * Relatively slower than an ASIC (MHz) ** FPGAs are frequently used in ASIC design to emulate the entire ASIC or parts of the ASIC in order to verify design, functionality etc.	* Application Specific IC * Long and complex design process compared to FPGA based design. Refer blog post named "ASIC Design Flow" * NRE involves cost for circuit design and mask design and hence expensive * Above 2 reasons make it a longer "time-to-market" product * Made for a specific application and hence cannot be reprogrammed to suit another application * Full custom design with every transistor and interconnect designed to serve the purpose and so no wastage of hardware * Because of the NRE, ASIC based designs are better if the required numbers are large * For the same functionality, ASICs can be much smaller than an FPGA * Faster than an FPGA (GHz)

Figure 7: Pros and cons of the FPGA's and ASIC's

Modern electronic components and designs to promote sustainable development

Mobile Phones

The mobile phone the electronic symbol of the 21st century is the result of sophisticated engineering of electronic hardware and software. However, as mass-produced electronic consumer items, mobile phones have recently come under scrutiny regarding their sustainability and their potential effects on the environment. With the arrival of two European Directives –on Waste Electrical and Electronic Equipment (WEEE) and on the Restriction of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment (RoHS) the whole life-cycle of mobile phones is under scrutiny.

The great strides that electronic engineering has made in reducing the demands on the Earth's resources in mobile phone manufacture. A typical mobile phone, like many electronic products, consists of 500–1000 components. Most of these components are made up of a large variety of materials and substances that are rare and expensive earth minerals.

The most important environmental issue for a mobile phone, in all the lifecycle phases, is energy consumption. The Printed Circuit Board, Integrated Circuits and Liquid Crystal Display are the components with the highest environmental impacts which have reduced the need for a large number of components which makes their design and manufacture more sustainable. (Engineering, 2020)

Design method, choices and calculations

A specification was written to make sure that the traffic light system would carry out the appropriate functions it needed to meet the specifications.

The specification required a pelican crossing with a WAIT button for pedestrians to cross, with a 5 second delay between pressing the button and the stop sequence initiated for the traffic. When traffic stopped a timer was started to give pedestrians time to cross this was set for 15 seconds and sequence would change lights to give the traffic green light.

In order to design this traffic light system was broken down into 3 separate parts, these being a 5 second timer, a 15 second timer and a Red Amber Green controller for the traffic lights. Each part has been designed by firstly making a Finite state machine as well as separate state tables and Karnaugh maps then the results from these used to build the circuits on the Multisim software.

Once all separate circuits are complete and tested they will be amalgamated together in order to make the traffic light system. These three sub circuits combined will make the Pelican crossing.



Figure 8 - Pelican crossing WAIT Button

Figure 9 below gives a simple representation of what you should be looking for as far as the sequence of the pelican crossing and pedestrian crossing is concerned. As shown when the lights for the traffic are green the Red man is activated. When the Wait light is pressed a timer begins and the green man only shows once the traffic lights turn to red. Another timer begins for the green man and once finished the red man is illuminated and the traffic lights soon turn back to green.



Figure 9 - Pelican Crossing

Figure 10 shows the block layout diagram for the desired traffic light circuit, the wait button

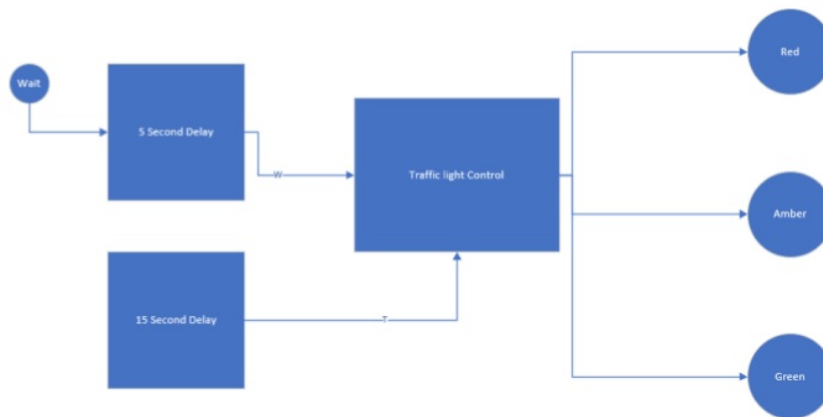


Figure 10 - Traffic Light Layout Diagram

5 Sec Delay

State Diagram

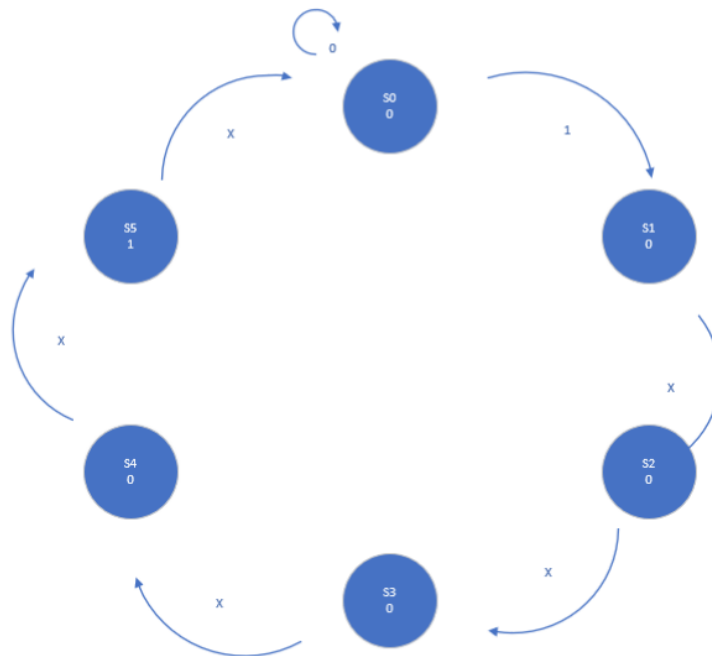


Figure 11 - Finite state diagram 5 second timer

A finite state machine in figure 11 shows the process the 5 second timer will be running through and indicates logics that affect it. The states for the diagram are set below. In order for the state to proceed from so the wait button designation (B) must be pressed if not the timer will stay on so. When the timer completes a rotation to s5 the output (W) will be triggered and go to a state of 1, after it proceed to so and await further input from the wait button.

5 Second Delay

S0 -Reset

S1- 1 Second

S2- 2 Second

S3- 3 Second

S4- 4 Second

S5- 5 Second

Inputs

Wait Button – B

Output

5 Seconds up – W

State Table

In order to design a state table the correct number of states must be selected which is dependent on the specification. This design requires 5 states but a table containing 8 states must be used as the rules require 2,4,8,16 states.

States

1 -2 States

2-4 States

3-8 States

4-16 States

Figure 12 state table below demonstrates how the 5 second timer logic inputs are derived.

Present	Next		Present			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	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	S3	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	S0	S1	1	1	0	X	X	X	X	X	X	x
S7	S2	S2	1	1	1	X	X	X	X	X	X	x

Figure 12 - State Table 5 second timer

Karnaugh Map

In figure 13 it can be seen that the results from the state table have been inputted into the Karnaugh maps and the results derived. A total of 8 separate AND gates and 3 separate OR gates will be required to build this 5 second timer circuit.

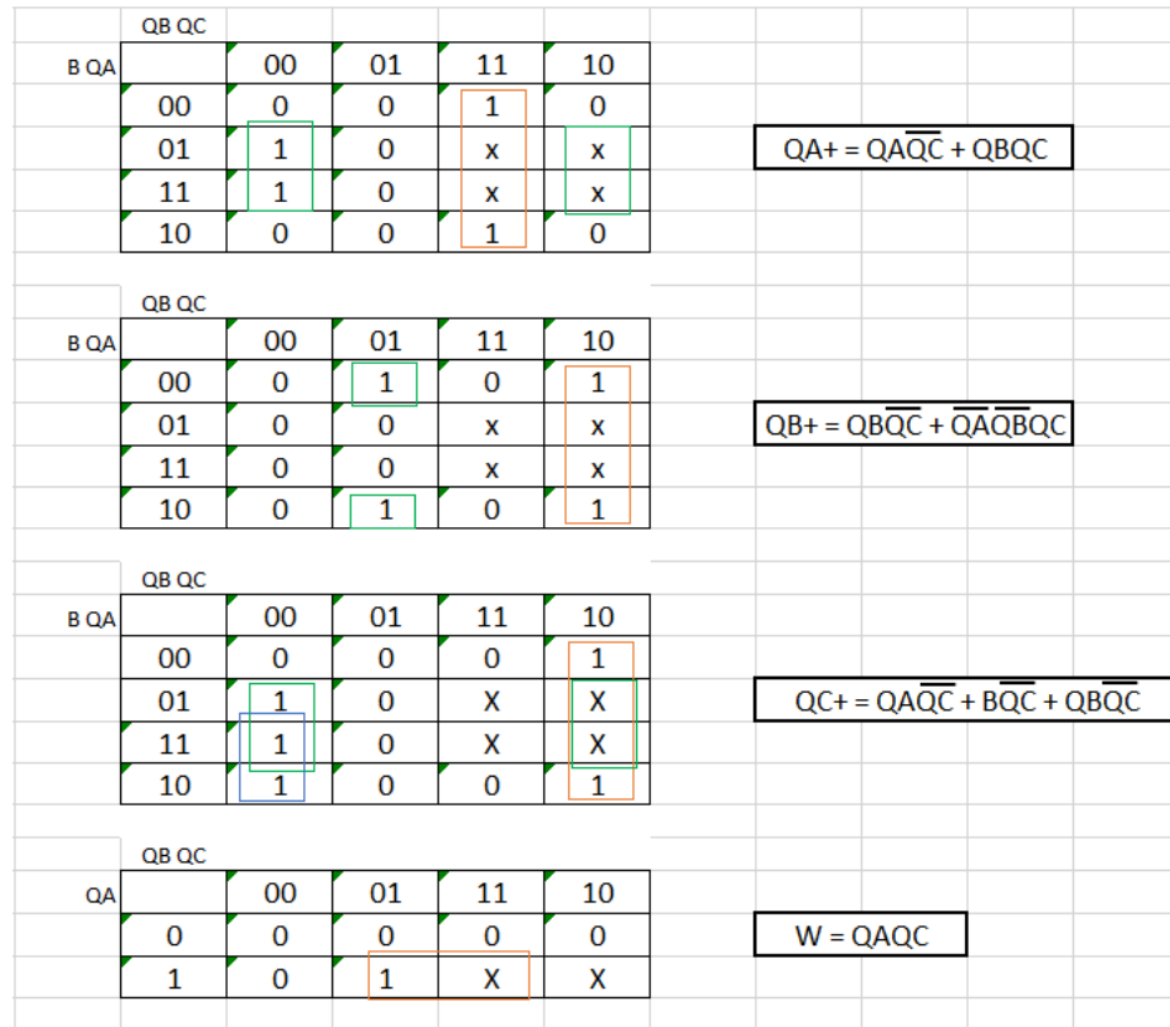


Figure 13 - Karnaugh map 5 second timer

Multisim 5 second timer

Figure 14 shows the results from the Karnaugh maps built on the Multisim software with the gates and correct inputs and outputs for each (QA+,QB+,QC+,W) and the Wait input button (B).

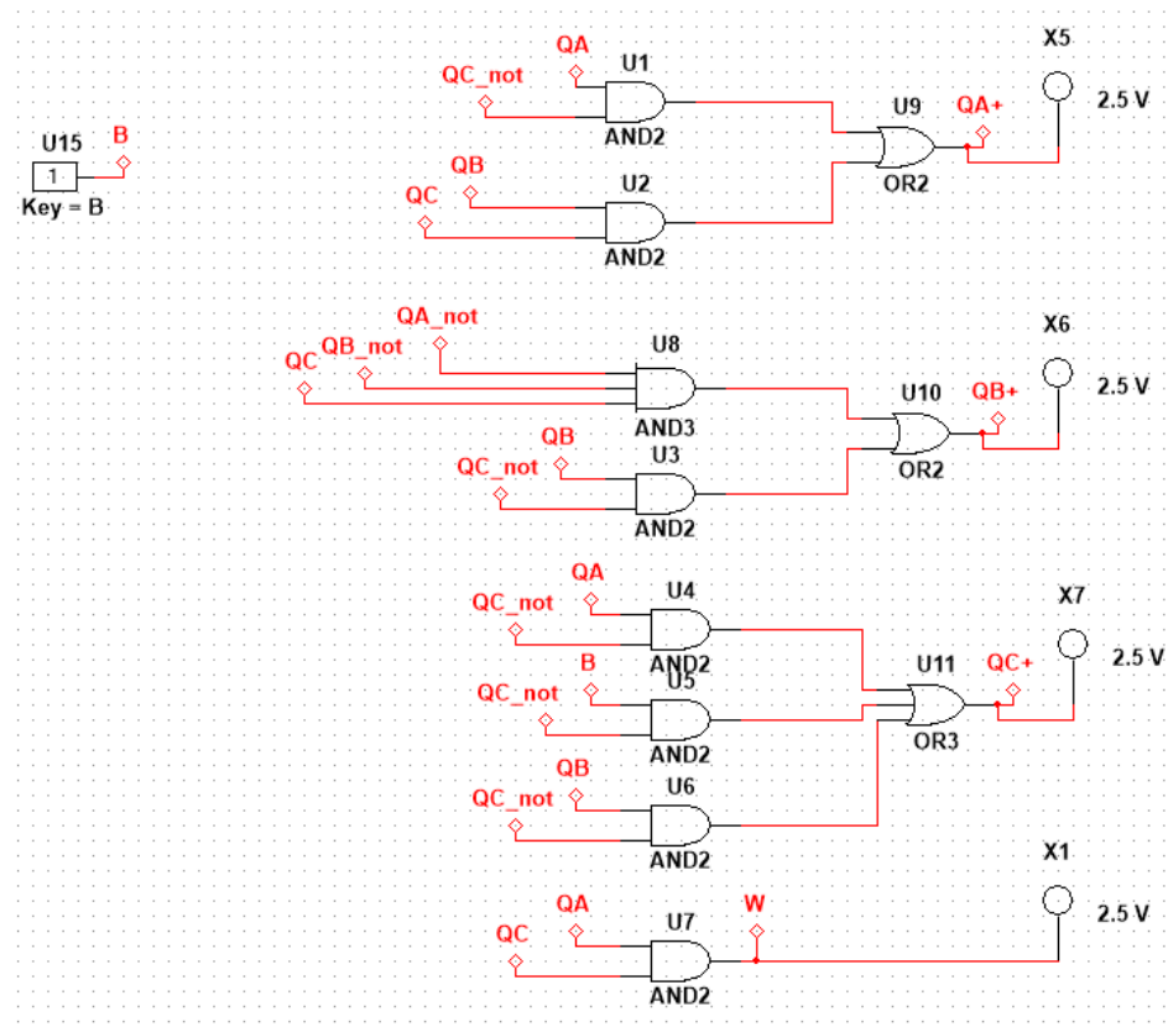


Figure 14 - Multisim 5 second Karnaugh Map gates

Figure 15 shows the design with flip flops added to the circuit, each flip-flop has two inputs, one for the input signal from the OR gate and one input from the clock. The clock input will eventually gain a signal and will go high, when the clock does this the signal from the OR gate does the same and the output results in the probe being energised. The clock will then go low and go high again at the next desired position; this will be according to the input gates.

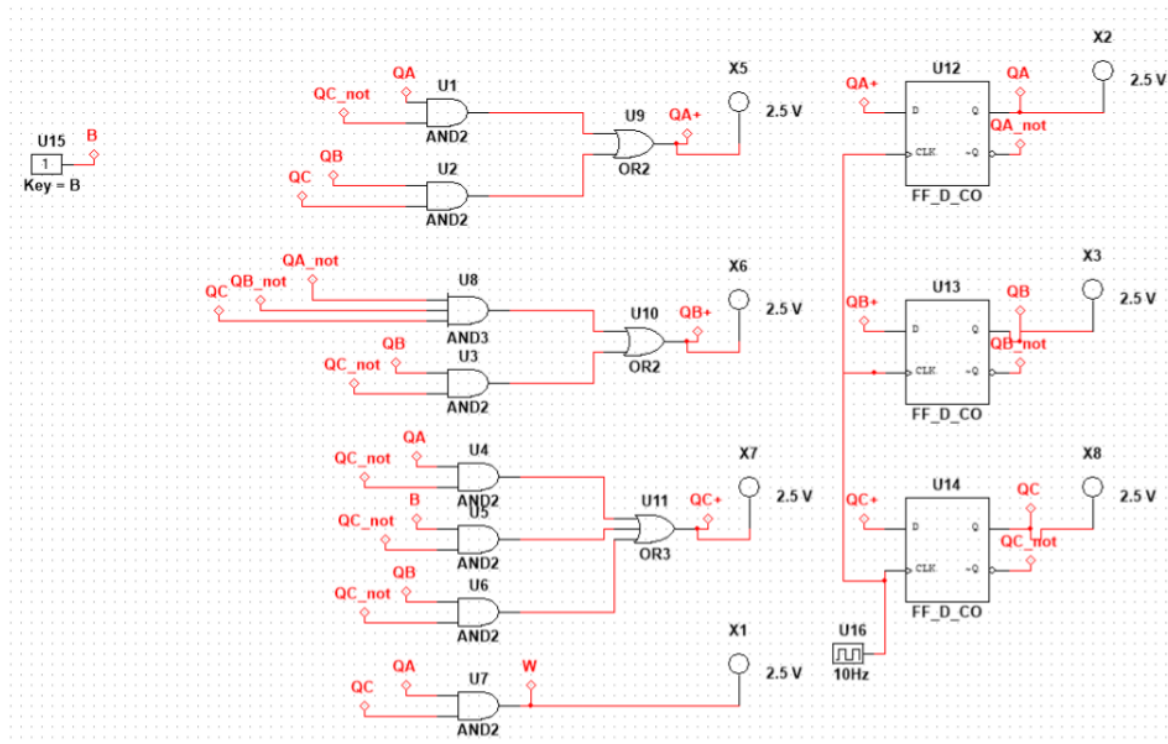


Figure 15 - Multisim 5 second timer

The circuit in figure 16 was changed to PLD mode as seen in figure 9 in order for it to be able to be uploaded to the nexus boards for testing and operation.

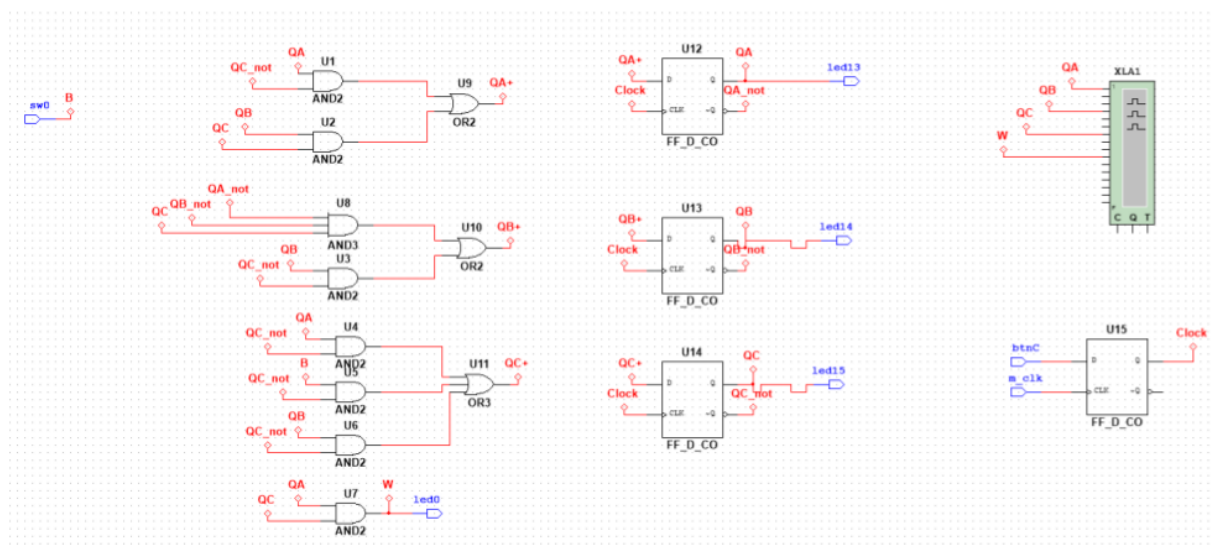


Figure 16 - Multisim 5 second timer PLD Version

15 Second Timer

A finite state machine in figure 17 shows the process the 15 second timer will be running through and indicates logics that affect it. The states for the diagram are set below. In order for the state to proceed from so the red traffic light (R) must be on if not the timer will stay on so. When the timer completes a rotation to s15 the output (T) will be triggered and go to a state of 1, after it proceed to so and await further input from the red traffic light being on.

15 Second Timer

Inputs

Red Traffic Light – R

Output

15 Seconds up – T

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

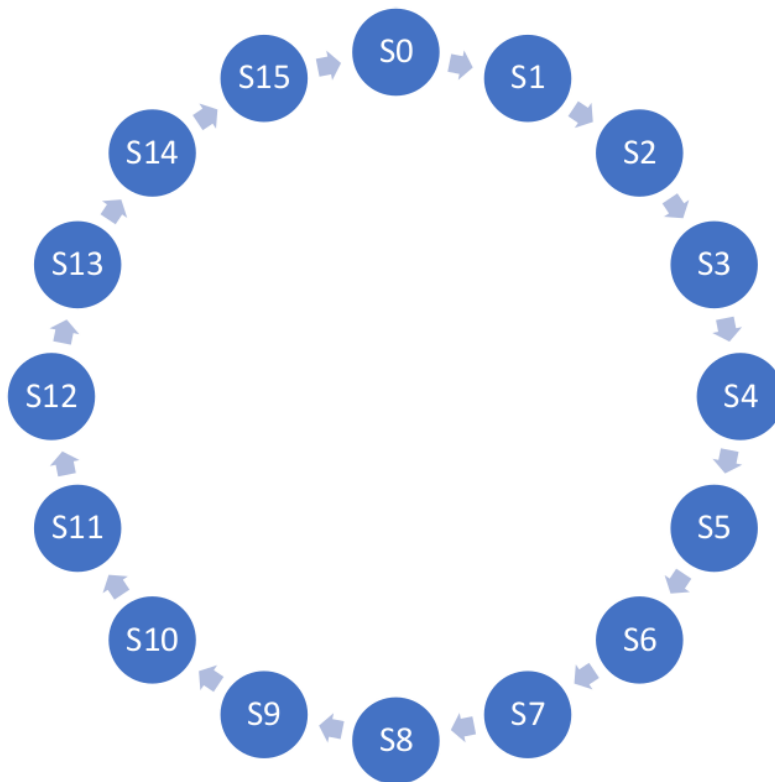


Figure 17 - 15 second Finite state machine

State Table

In order to design a state table the correct number of states must be selected which is dependent on the specification. This design requires 16 states a table containing 16 states can be used as the rules require 2, 4, 8, 16 states.

States

1 -2 States

2-4 States

3-8 States

4-16 States

Figure 18 state table below demonstrates how the 15 second timer logic inputs are derived.

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

Figure 18 - 15 second State Table

Karnaugh Map

In figure 19 it can be seen that the results from the state table have been inputted into the Karnaugh maps and the results derived. A total of 9 separate AND gates and 3 separate OR gates will be required to build this 15 second timer circuit.

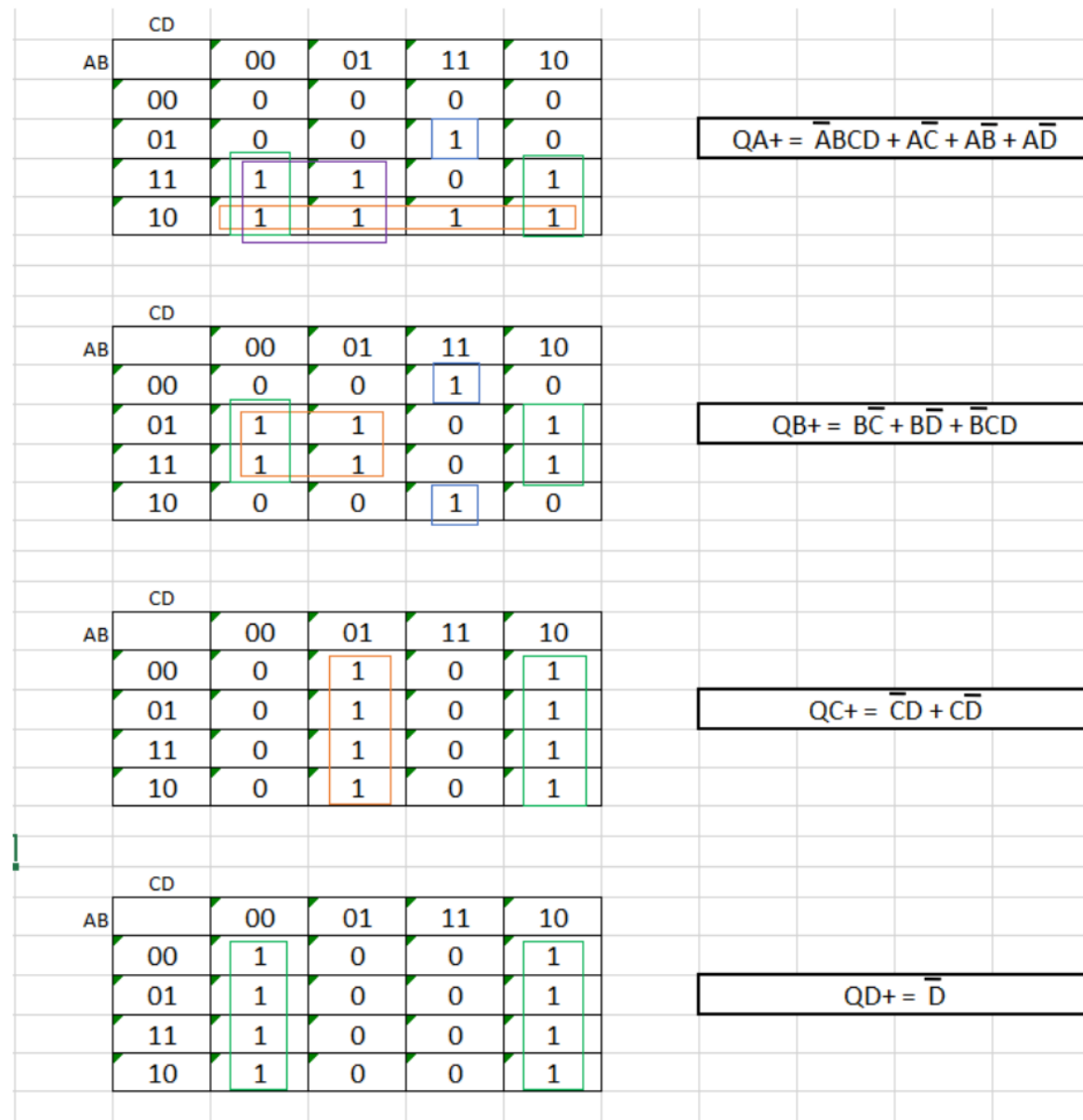


Figure 19 - 15 Second timer Karnaugh maps

Multisim 15 second timer

Figure 20 shows the results from the Karnaugh maps built on the Multisim software with the gates and correct inputs and outputs for each (QA+,QB+,QC+,QD+) and the Red traffic light (RED) input button.

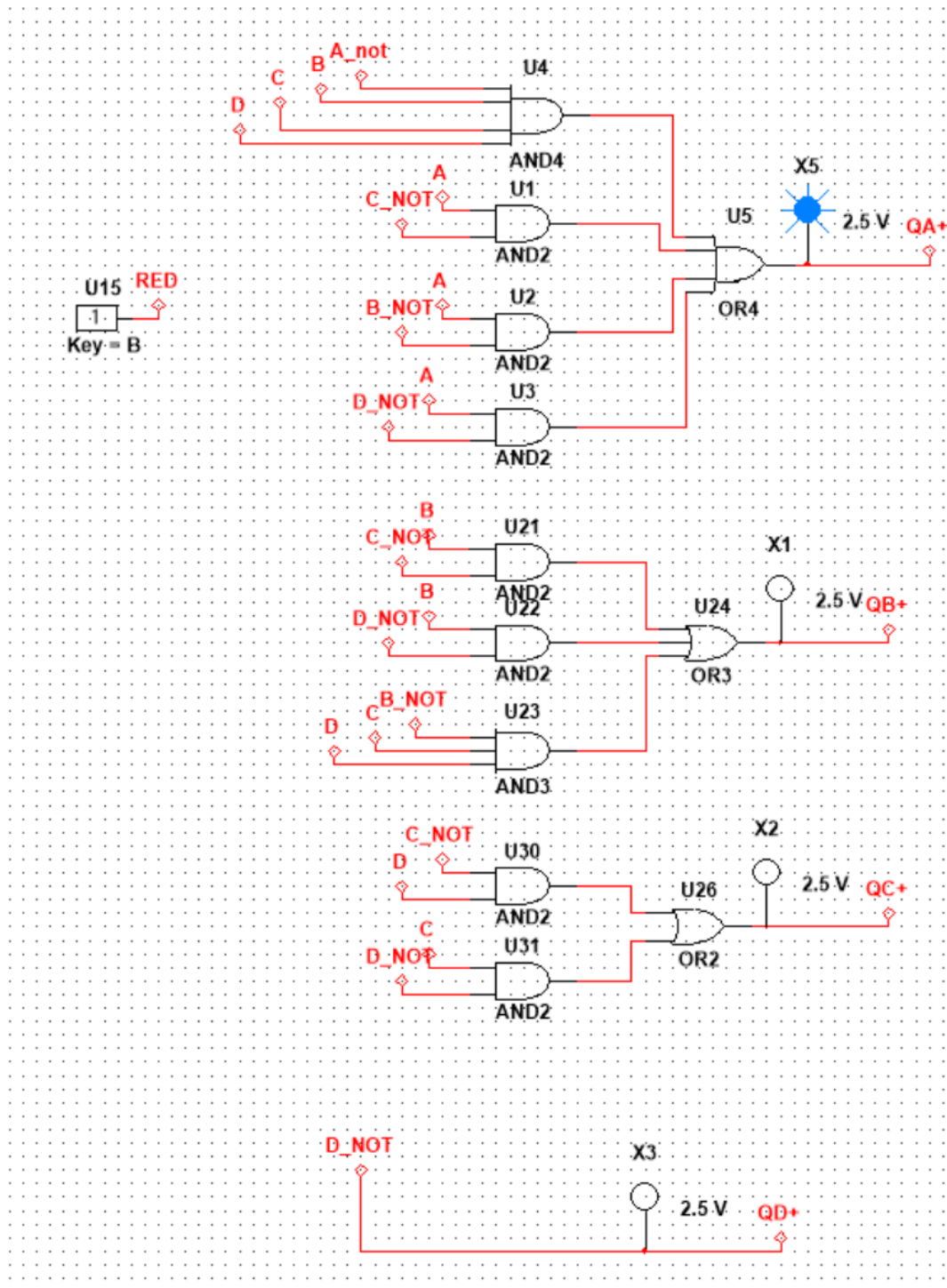


Figure 20 - Multisim 15 second timer

Figure 21 shows the design with flip flops added to the circuit.

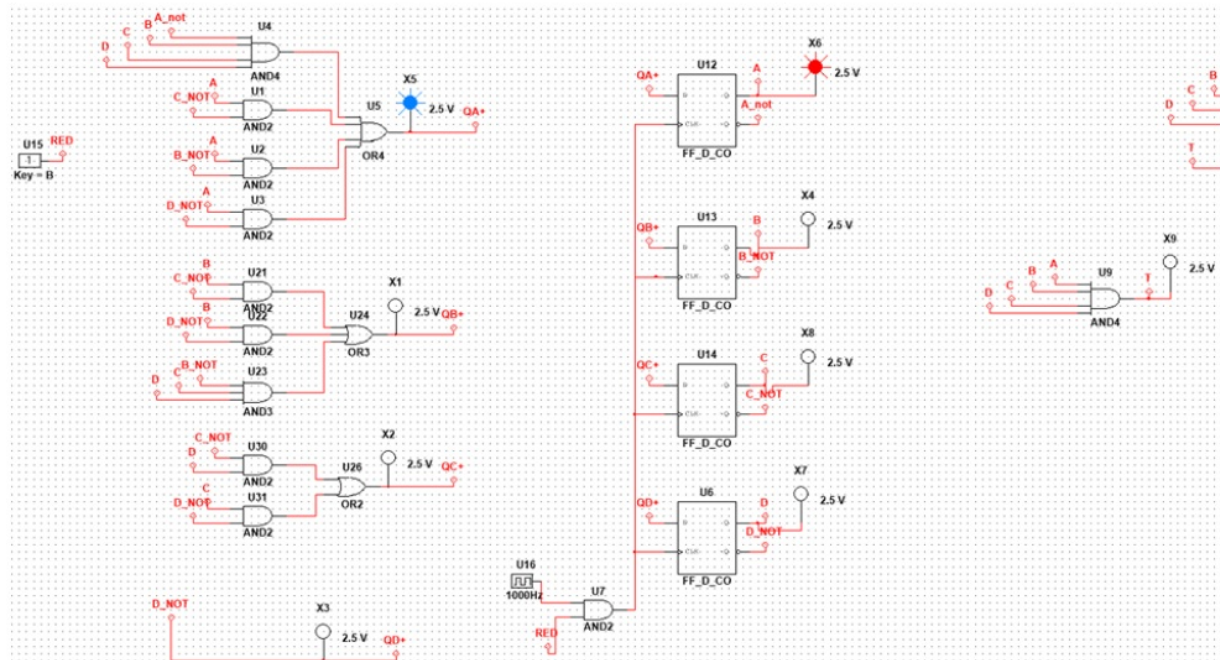


Figure 21 - 15 Second timer with flip flops

The circuit in figure 21 was changed to PLD mode as seen in figure 22 in order for it to be able to be uploaded to the nexus boards for testing and operation.

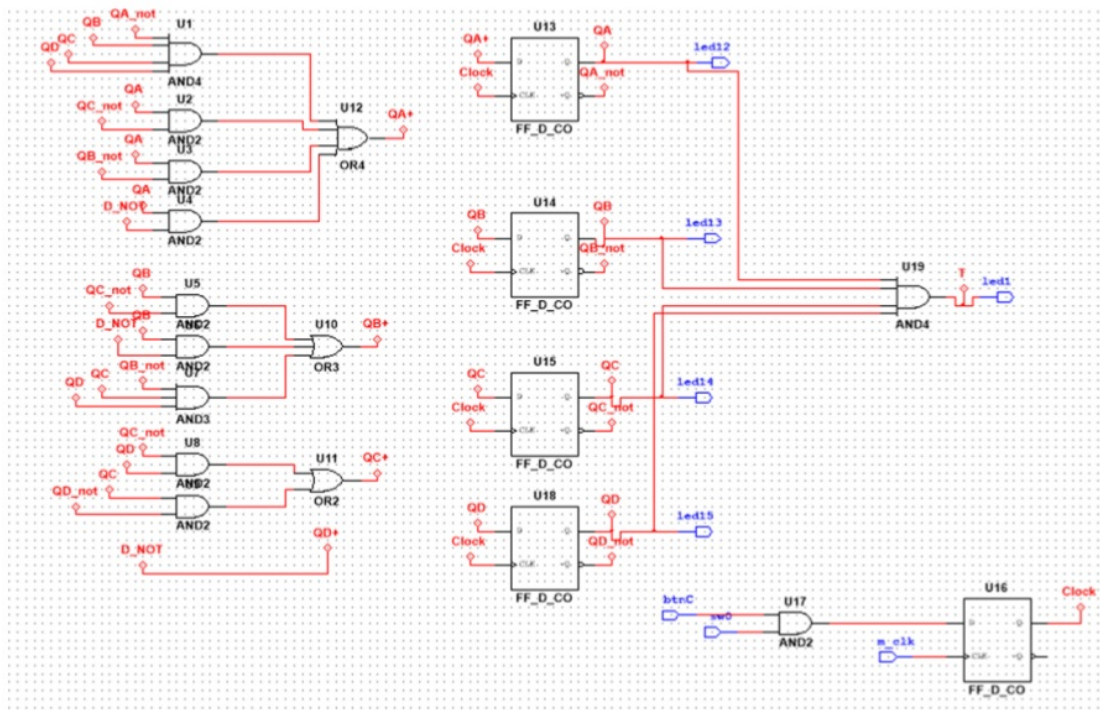


Figure 22 - 15 Second timer PLD

RAG Controller

A finite state machine in figure 23 shows the process the RAG Controller will be running through and indicates logics that affect it. The states for the diagram are set below. In order for the state to proceed from S0 the 5 second timer (W) must be on if not the timer will stay on S0. When the timer completes a rotation to S5 the output (R) will be on which will start the 15 second timer (T) when complete this will trigger (T) and only then will proceed back to state S0, it will then await further input from the 5 second timer (W) being on.

Inputs

5 Second Timer – W

15 second timer – T

Output

Red Traffic Light – R

Amber Traffic Light – A

Green Traffic Light – G

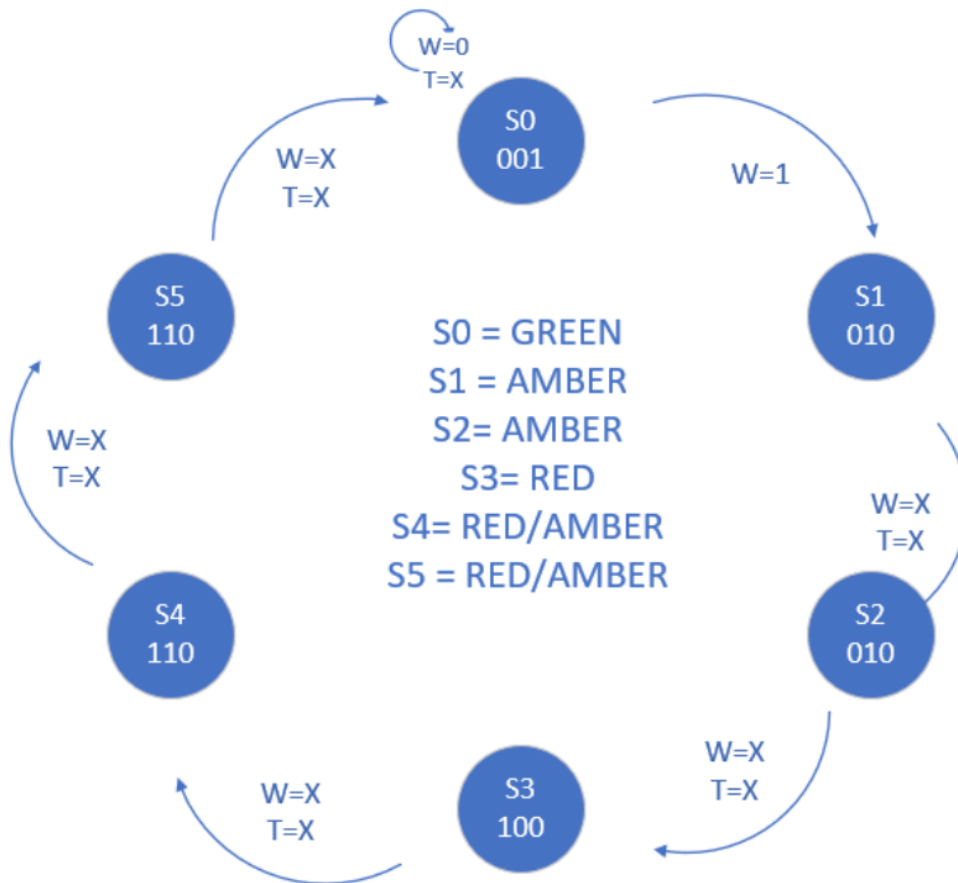


Figure 23 - RAG Controller Finite State Machine

State Table

In order to design a state table the correct number of states must be selected which is dependent on the specification. This design requires 8 states a table containing 8 states can be used as the rules require 2, 4, 8, 16 states.

States

1 -2 States

2-4 States

3-8 States

4-16 States

Figure 24 state table below demonstrates how the RAG Controller logic inputs are derived.

Current	Next				CURRENT			W=0						W=1						OUTPUT			
	W=0		W=1		QA	QB	QC	T=0			T=1			T=0			T=1			R	A		
	T=0	T=1	T=0	T=1				QA+	QB+	QC+	QA+	QB+	QC+	QA	QB	QC	QA	QB	QC				
S0	S0	S0	S1	S1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	G
S1	S2	S2	S2	S2	0	0	1	0	1	0	0	1	0	1	0	0	1	0	0	0	0	1	0
S2	S3	S3	S3	S3	0	1	0	0	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0
S3	S3	S4	S3	S4	0	1	1	0	1	1	1	0	0	0	1	1	0	0	1	0	1	0	0
S4	S5	S5	S5	S5	1	0	0	1	0	1	1	0	1	0	1	1	0	1	0	1	0	1	0
S5	S0	S0	S0	S0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
S6	X	X	X	X	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
S7	X	X	X	X	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Figure 24 - RAG Controller State Table

Karnaugh Map

In figure 25 & 26 it can be seen that the results from the state table have been inputted into the Karnaugh maps and the results derived. A total of 13 separate AND gates and 5 separate OR gates will be required to build this RAG Controller circuit.

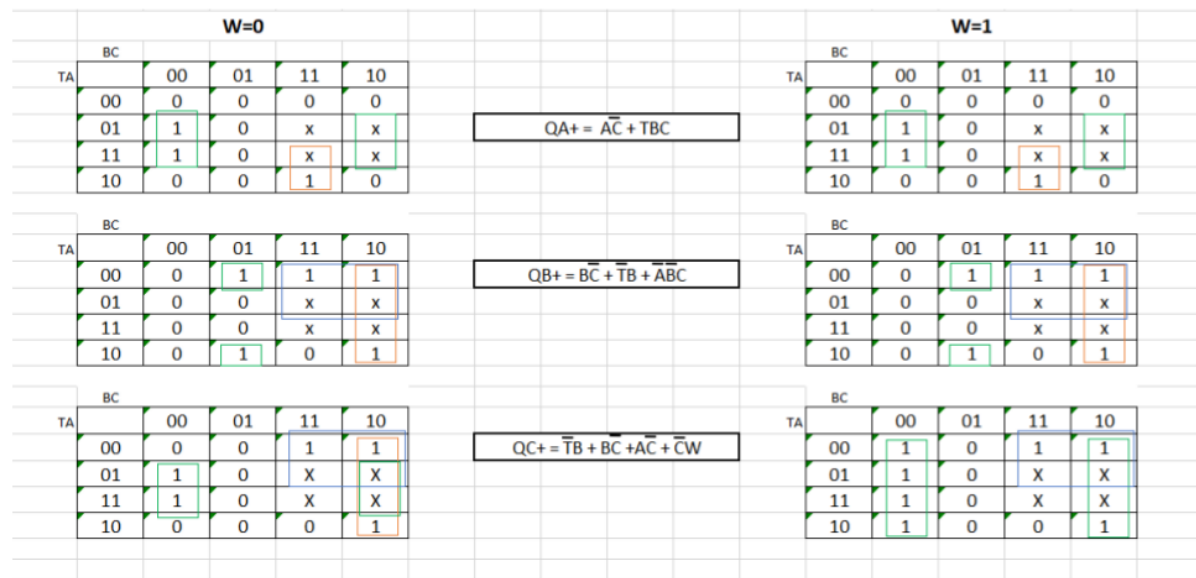


Figure 25 - RAG Controller Karnaugh Maps

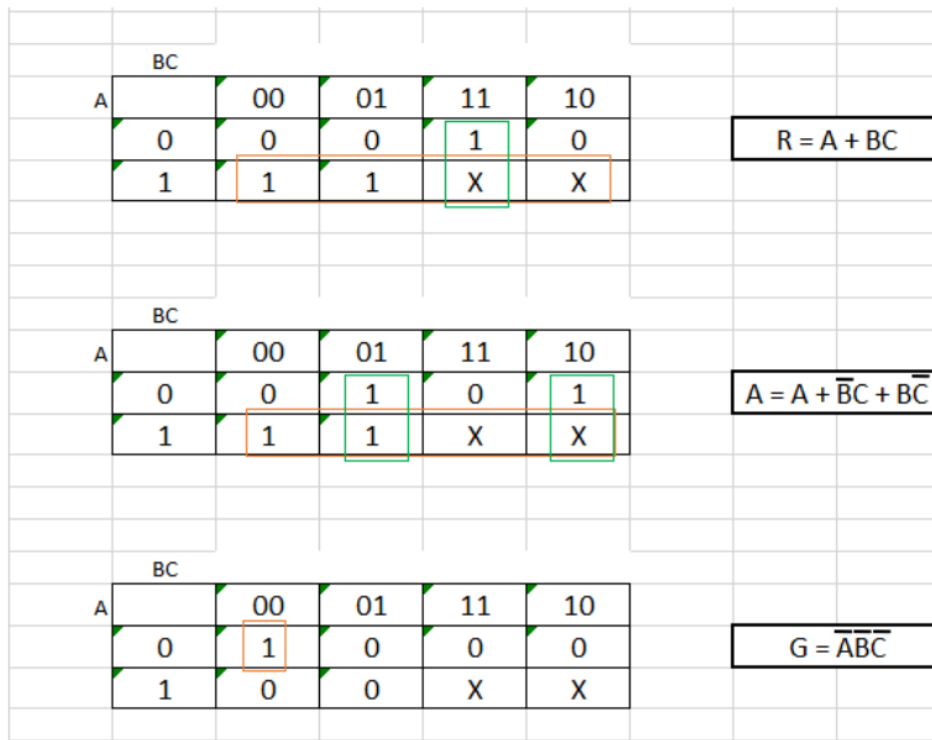


Figure 26 - RAG Controller Karnaugh Maps

Multisim RAG Controller

Figure 27 shows the results from the Karnaugh maps built on the Multisim software with the gates and correct inputs and outputs for each (QA,QB,QC,QD) and the 5 second timer(W) and 15 second timer(T) input buttons.

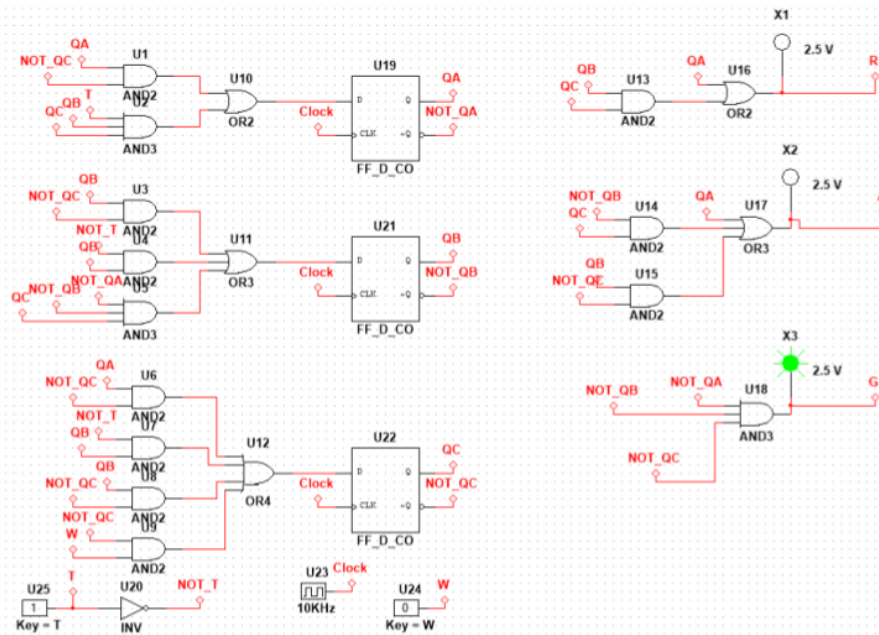


Figure 27 - RAG Controller Multisim

The circuit in figure 27 was changed to PLD mode as seen in figure 28 in order for it to be able to be uploaded to the nexus boards for testing and operation.

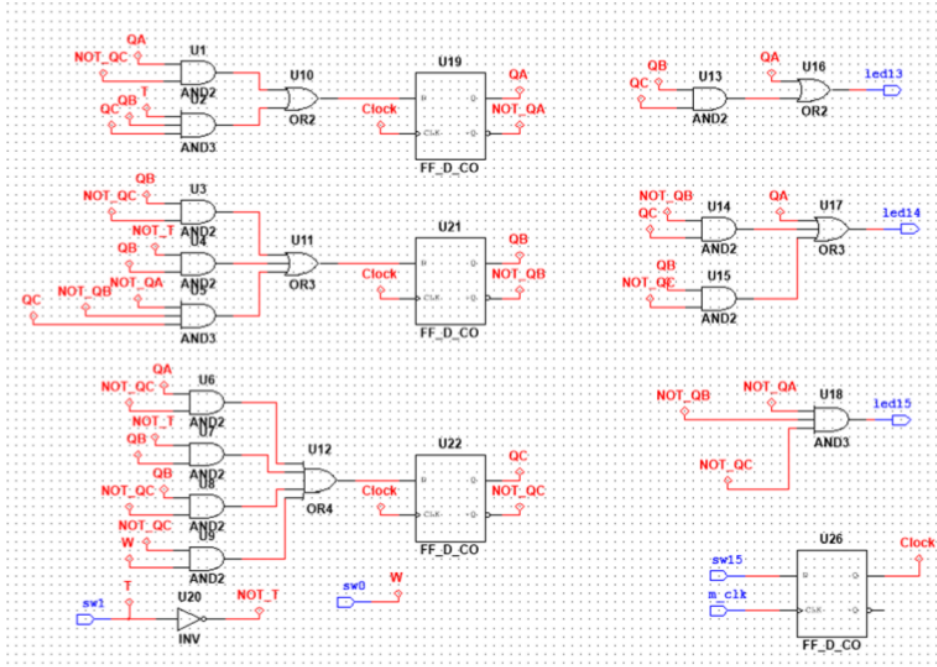


Figure 28 - RAG Controller PLD Multisim

Testing, results and analysis (500)

5 Second Timer

The figure 29 below shows the 5 second timer circuit complete. A logic analyser was connected to the circuit to test the output to see if the result matches the required design parameters. Lighting probes were also connected at several different locations and points on the circuit in order to identify that each stage of the timer was working as intended and designed.

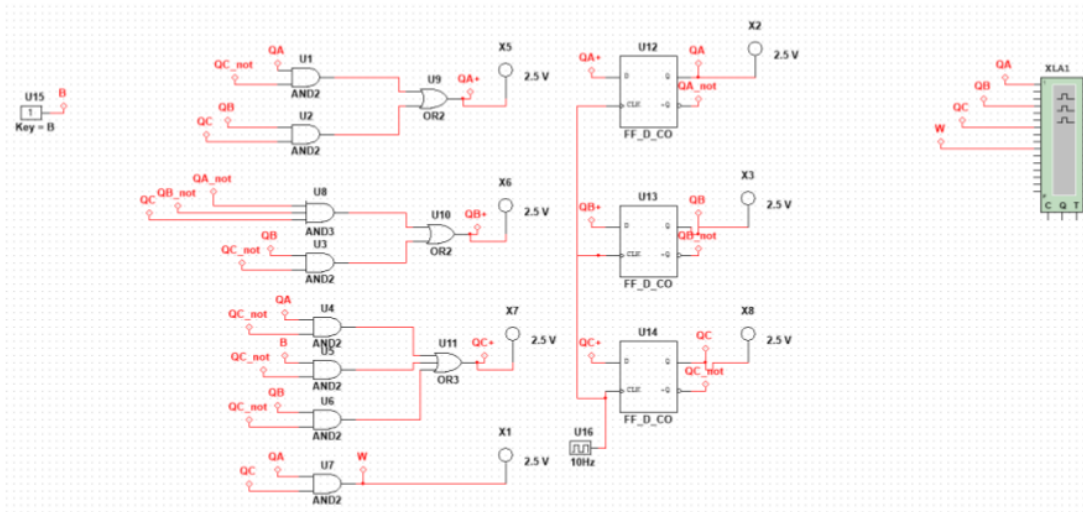


Figure 29 - 5 second timer with logic analyser

The logic analyser was used to monitor the output response of the traffic lights in using waveforms. By connecting the Q output from each flip flop (QA, QB, QC) to the logic analyser and the output (W) and let it run through the sequence the results were captured on a graph and each output producing a different cycle, figure 30 represents this output below. As seen the counter stats at 1 second and counts to 5 seconds at this point it is seen that the output W is triggered as design specifications.

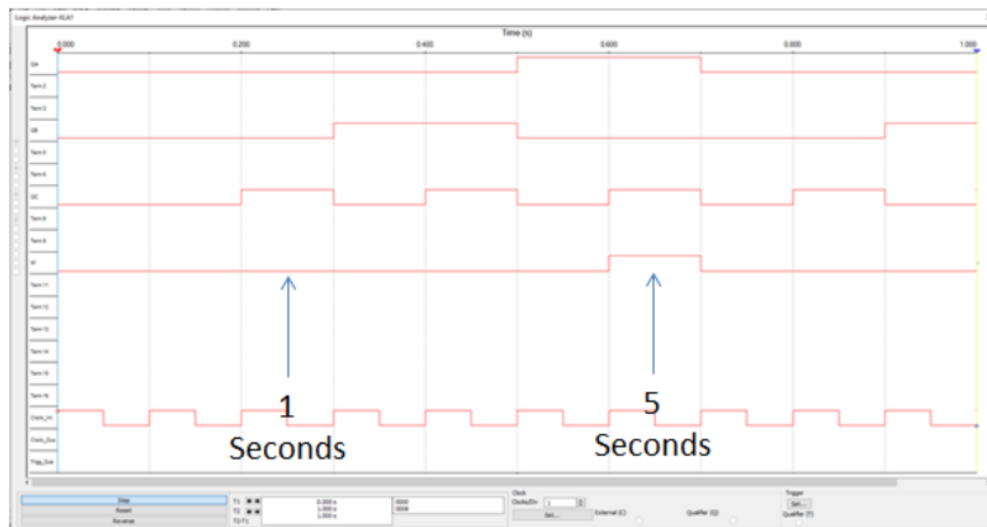


Figure 30 - Logic analyser results for 5 second timer

Figure 31 shows the 5 second timer running through its sequence and the lighting probes lighting up.

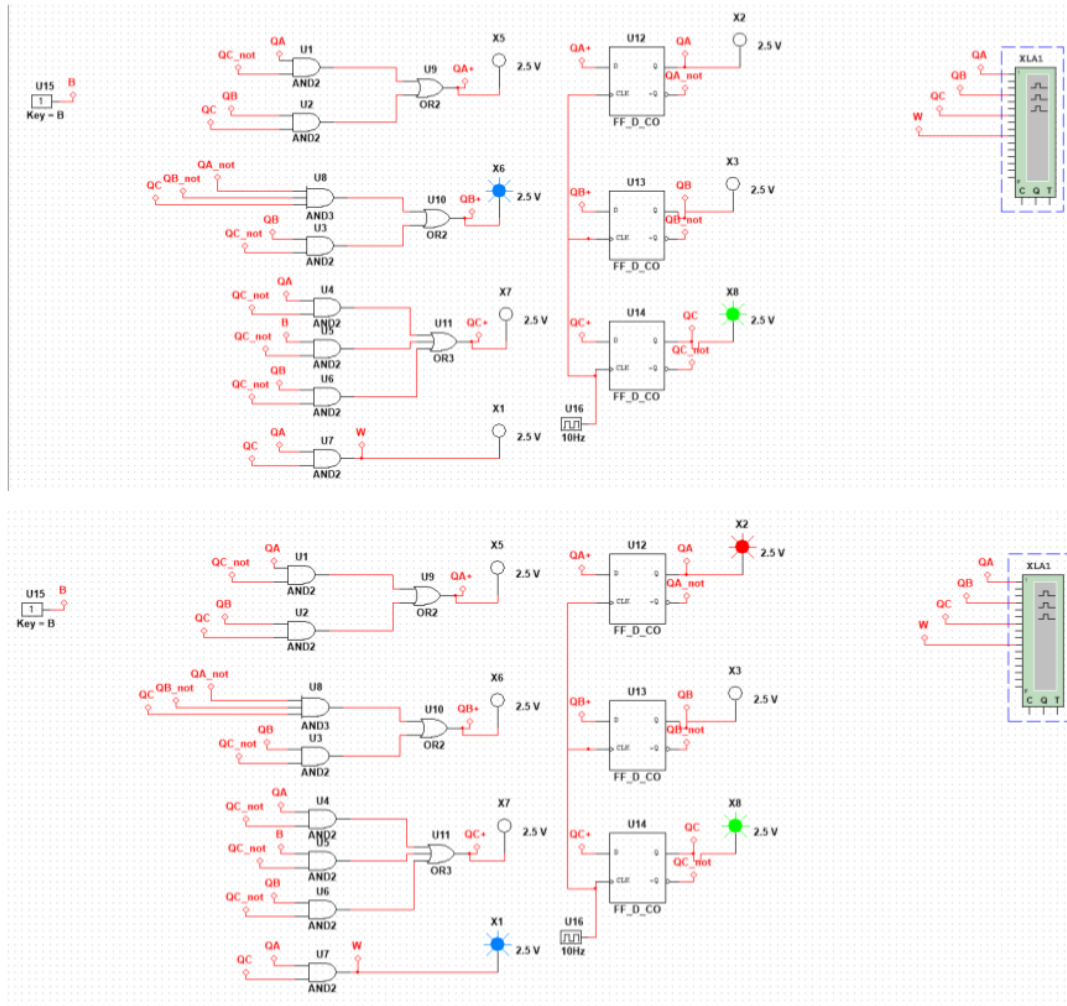


Figure 31 - 5 second timer testing

After this successful test of this 5 second timer the PLD version was uploaded to the nexus board for further testing the results matched the expected outcome and worked successfully on the board.

15 Second Timer

The figure 32 below shows the 15 second timer circuit complete. A logic analyser was also connected to this circuit to test the output to see if the result matches the required design parameters. Lighting probes were also connected again at several different locations and points on the circuit in order to identify that each stage of the timer was working as intended.

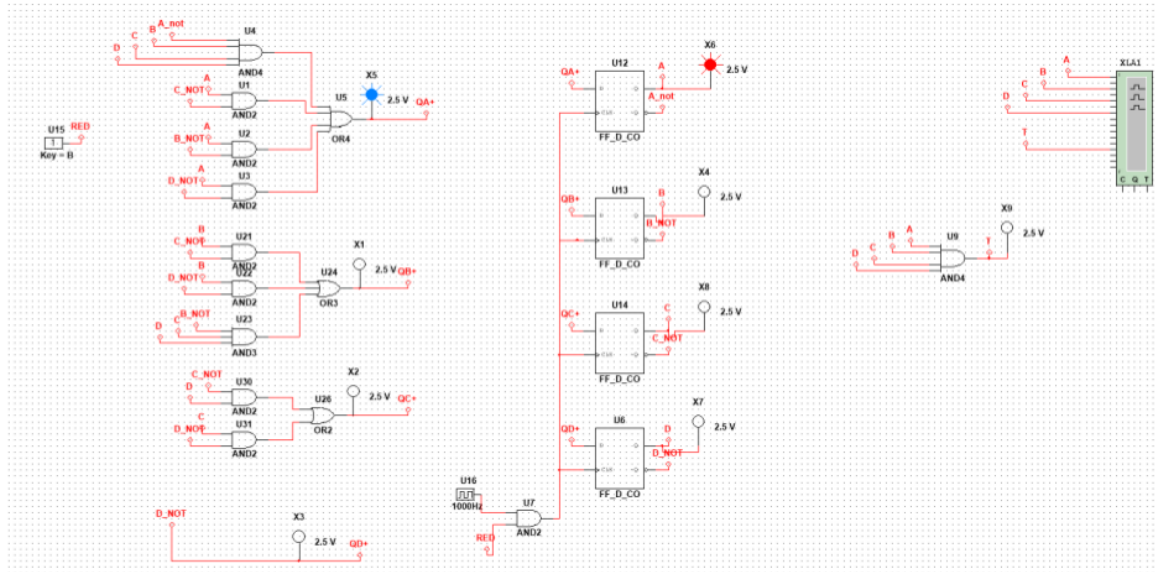


Figure 32 - 15 second timer logic analyser connected

Again the logic analyser was used to monitor the output response of the traffic lights in waveforms. By connecting the Q output from each flip flop (A, B, C, D) to the logic analyser and the output (T) and let it run through the sequence the results were captured on a graph and each output producing a different cycle, figure 33 represents this output below. As seen the counter stats at 1 second and counts to 15 seconds at this point it is seen that the output T is triggered as design specifications.

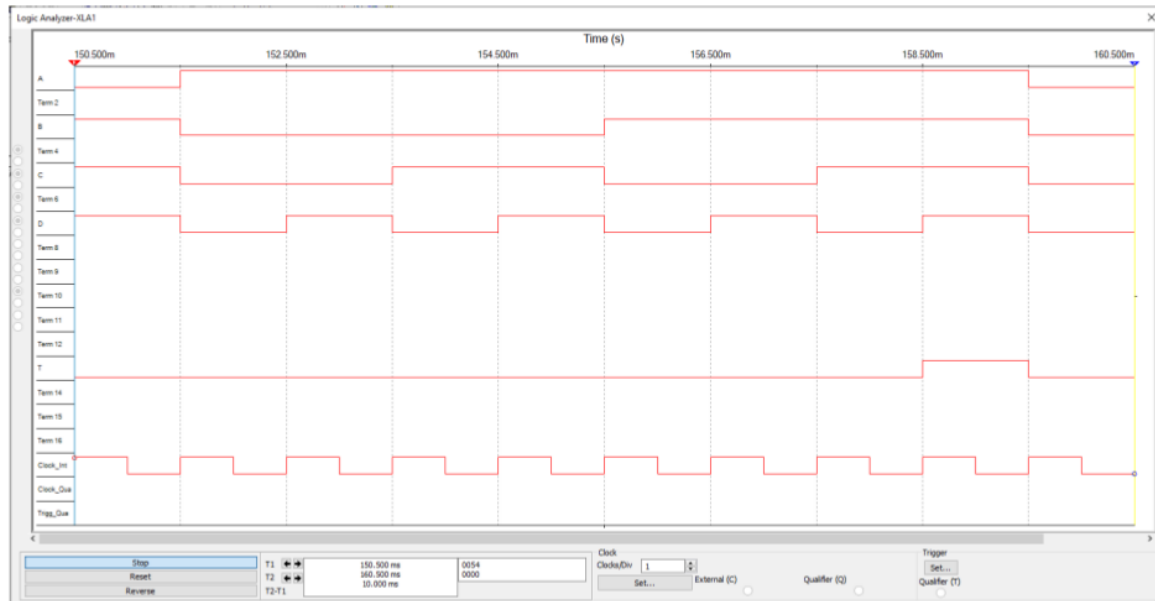


Figure 33 - 15 second logic analyser results

Figure 34 & 35 shows the 15 second timer running through its sequence and the lighting probes lighting up.

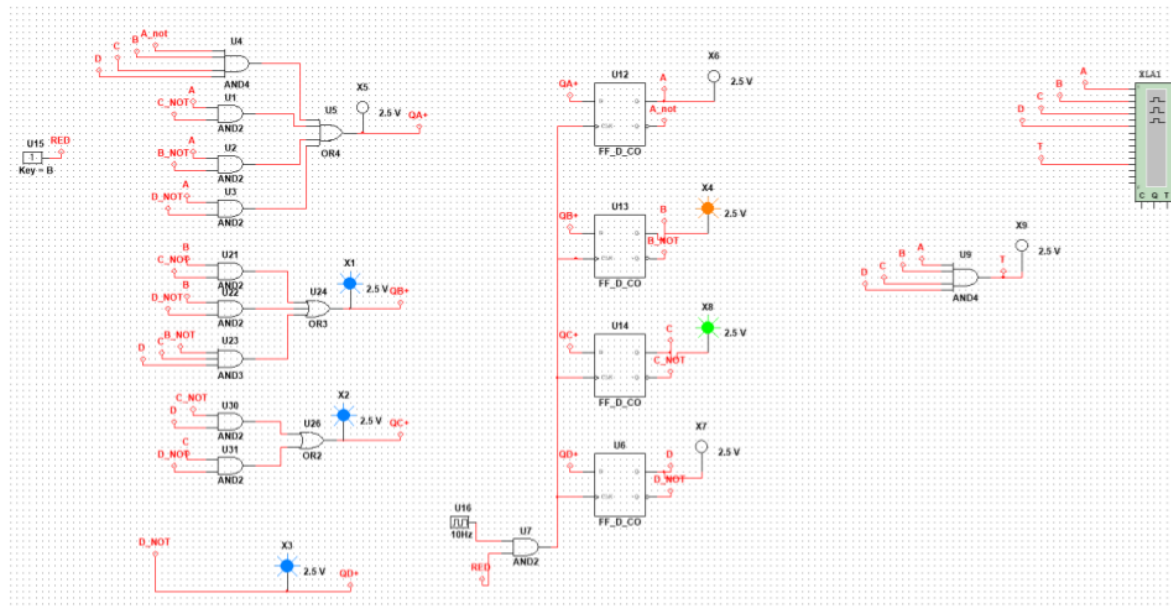


Figure 34 - 15 second timer testing sequence

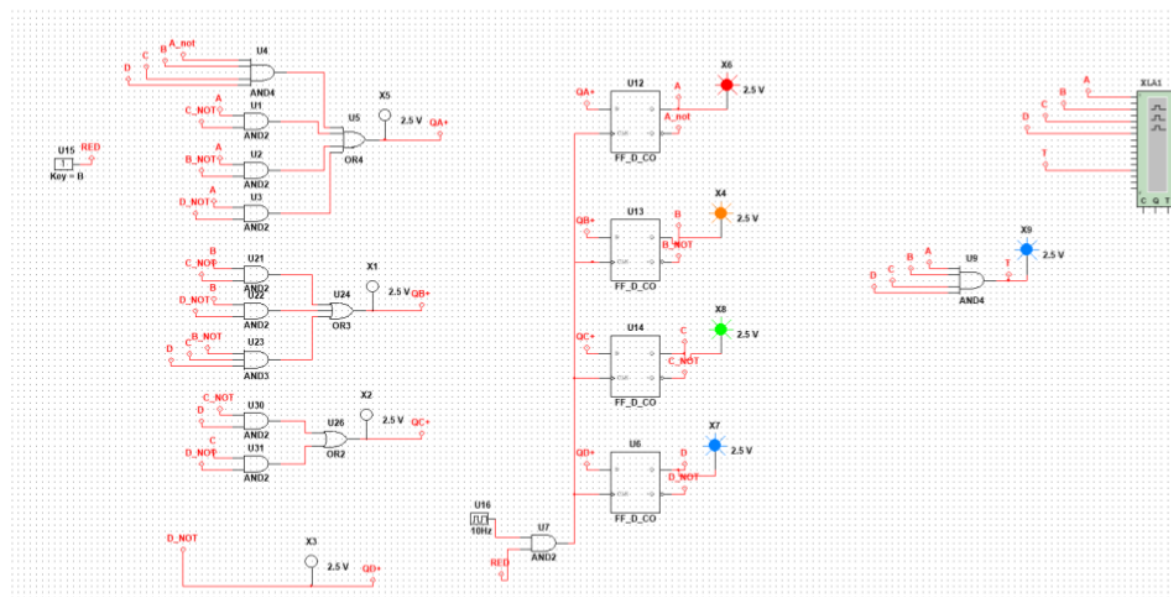


Figure 35 - 15 second timer testing sequence two

After this successful test of this 15 second timer the PLD version was uploaded to the nexus board for further testing the results matched the expected outcome and worked successfully on the board.

The figure 32 below shows the 15 second timer circuit complete. A logic analyser was also connected to this circuit to test the output to see if the result matches the required design parameters. Lighting probes were also connected again at several different locations and points on the circuit in order to identify that each stage of the timer was working as intended.



28

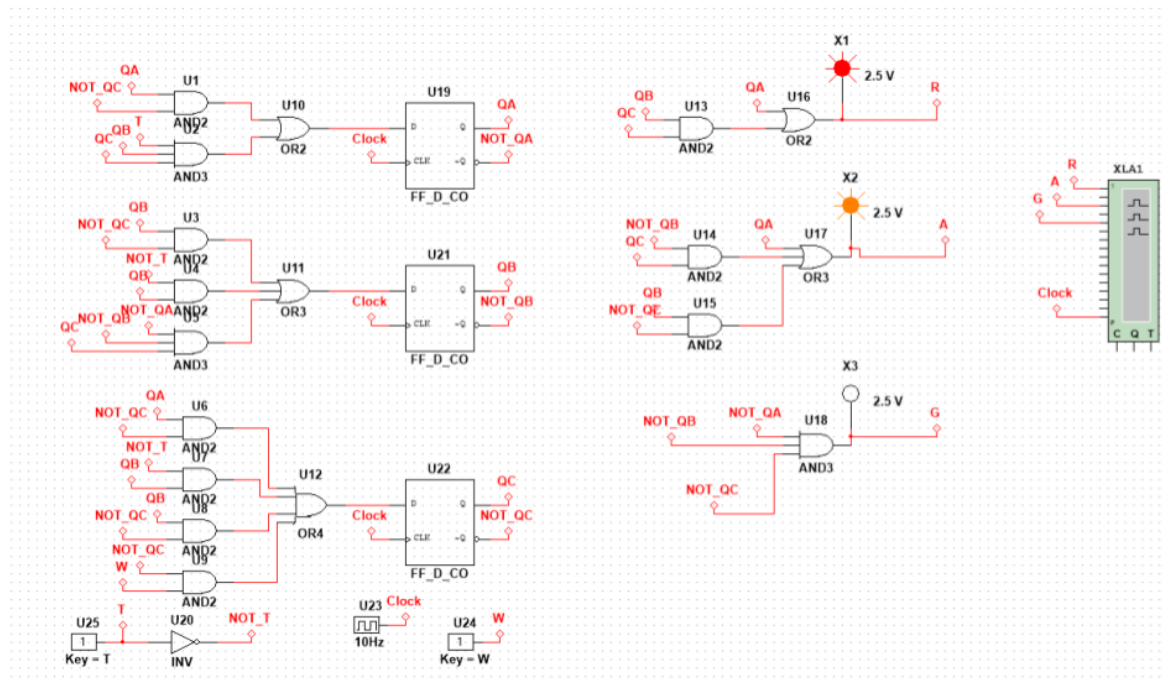


Figure 38 - RAG Controller testing sequence

After this successful test of this RAG Controller the PLD version was uploaded to the nexus board for further testing the results matched the expected outcome and worked successfully on the board.

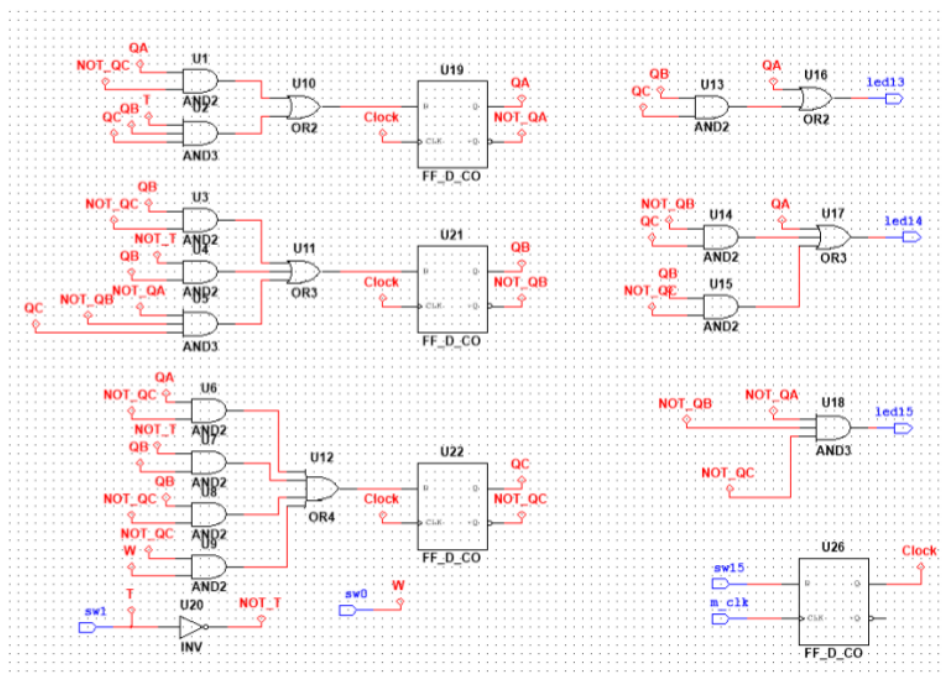


Figure 39 - RAG Controller PLD Mode

Complete Design

All three circuits the 5 second timer, 15 second timer and RAG controller were merged together to produce the fully working traffic control system as seen in figure 40,41 & 42 below.

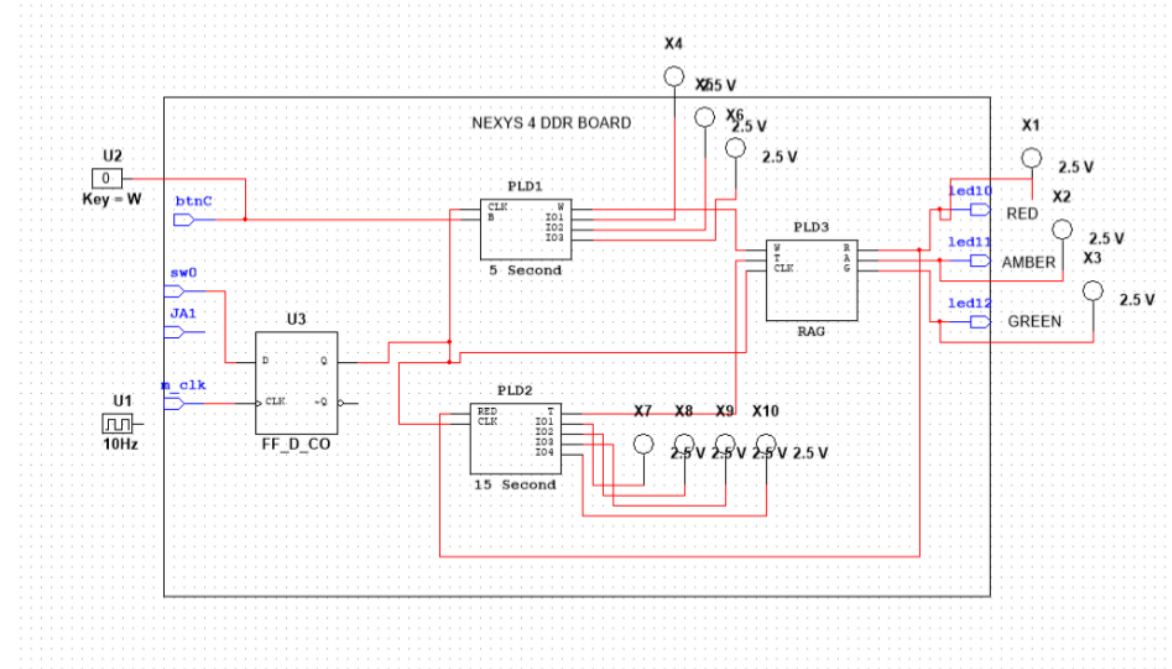


Figure 40 - Traffic control system

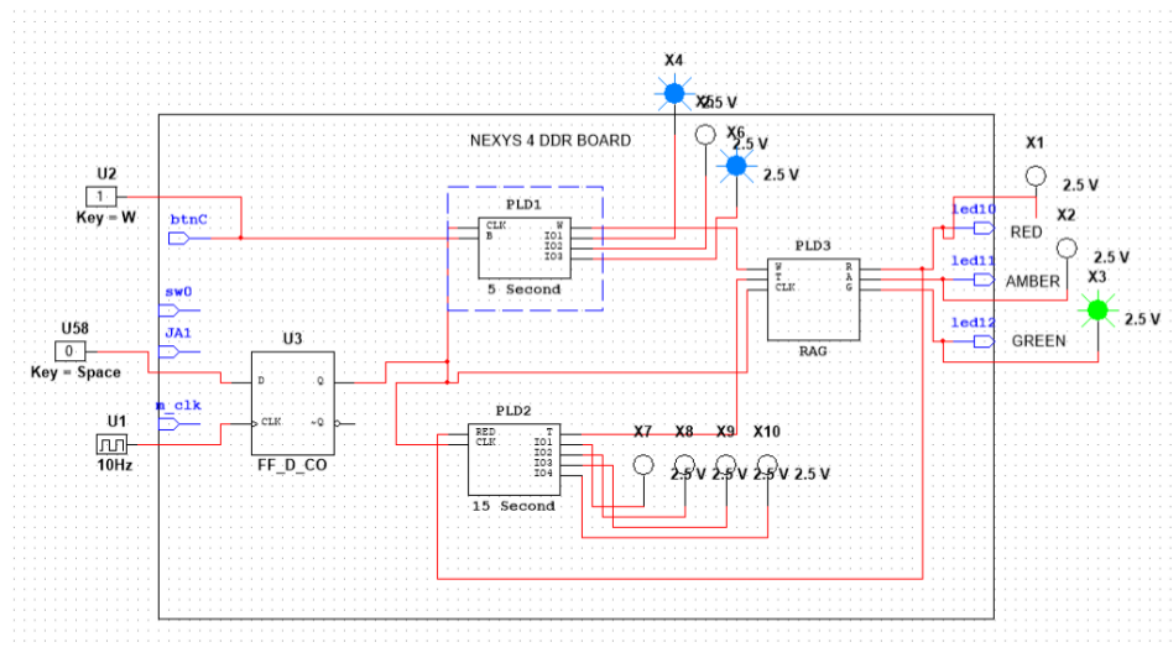


Figure 41 - Traffic control system test

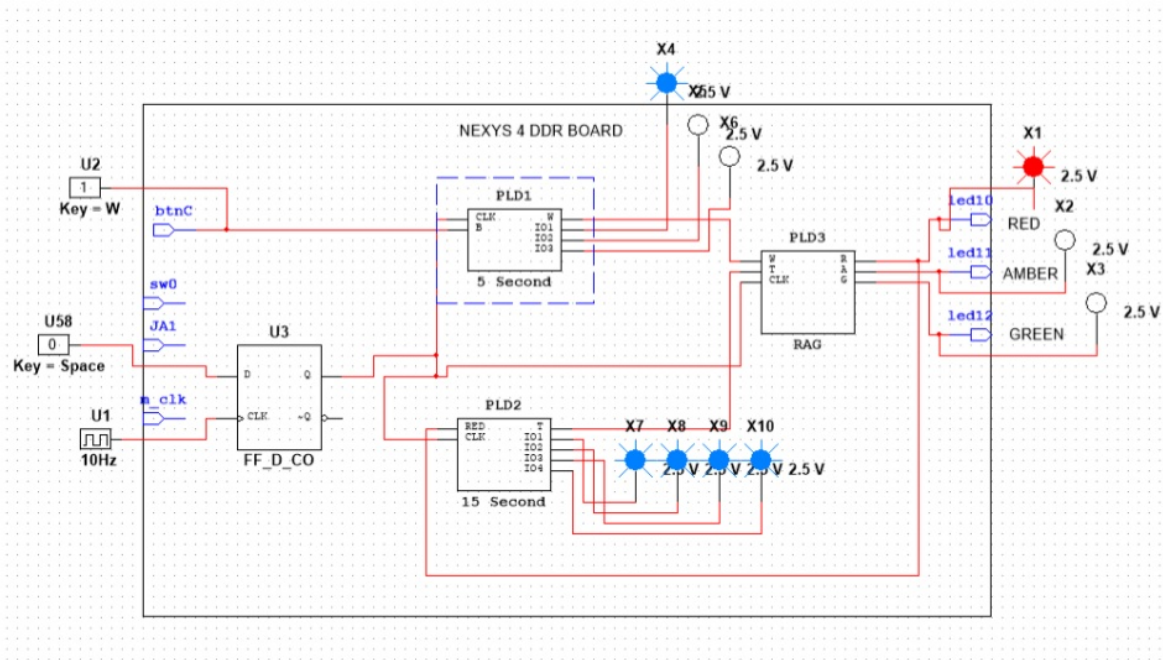


Figure 42 - Traffic control system test

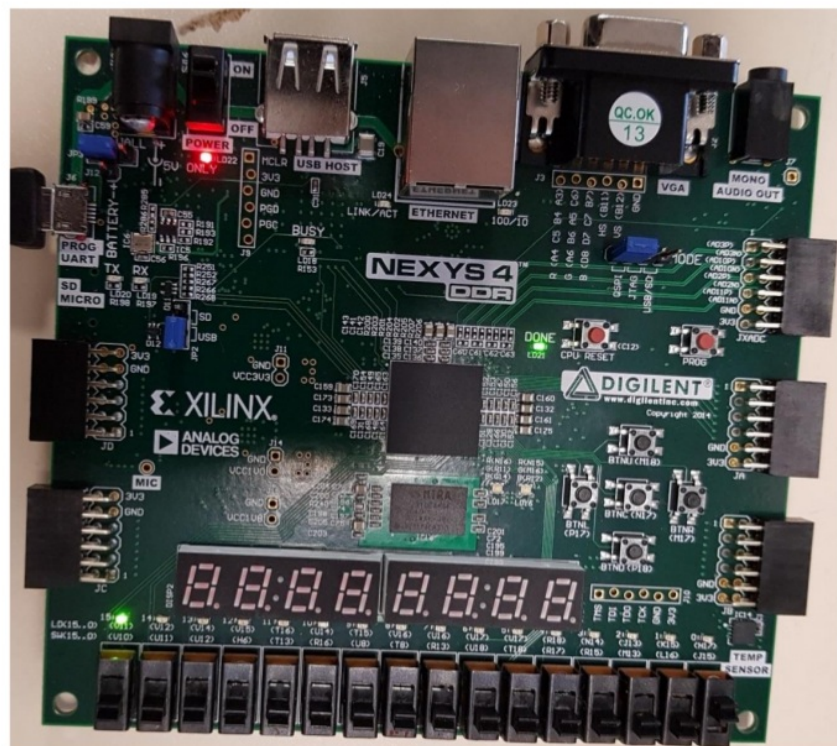


Figure 43 - TCS Nexus board test

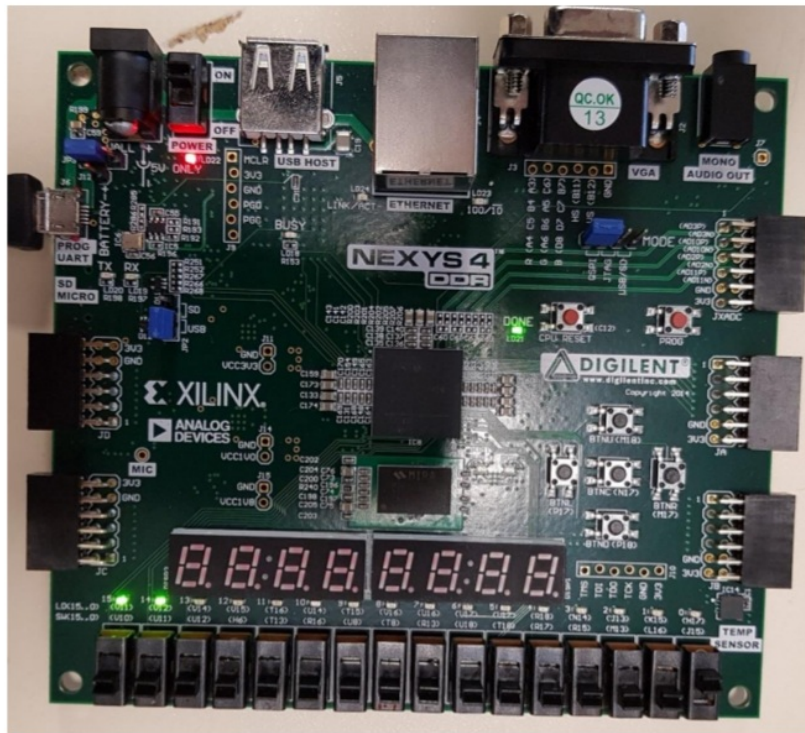


Figure 44 - TCS Nexus board test

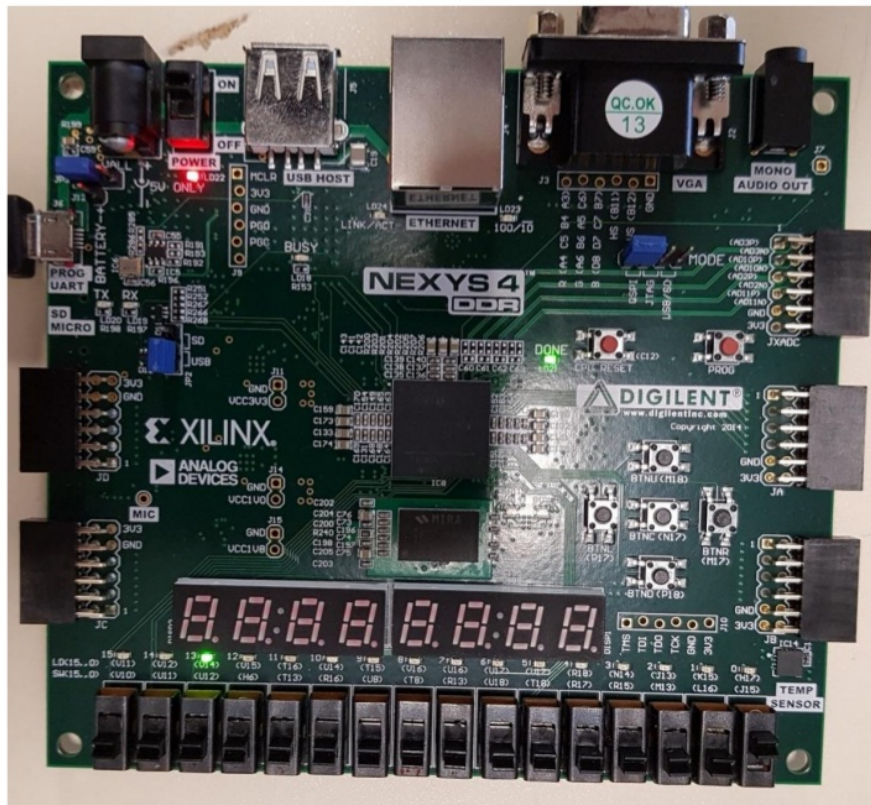
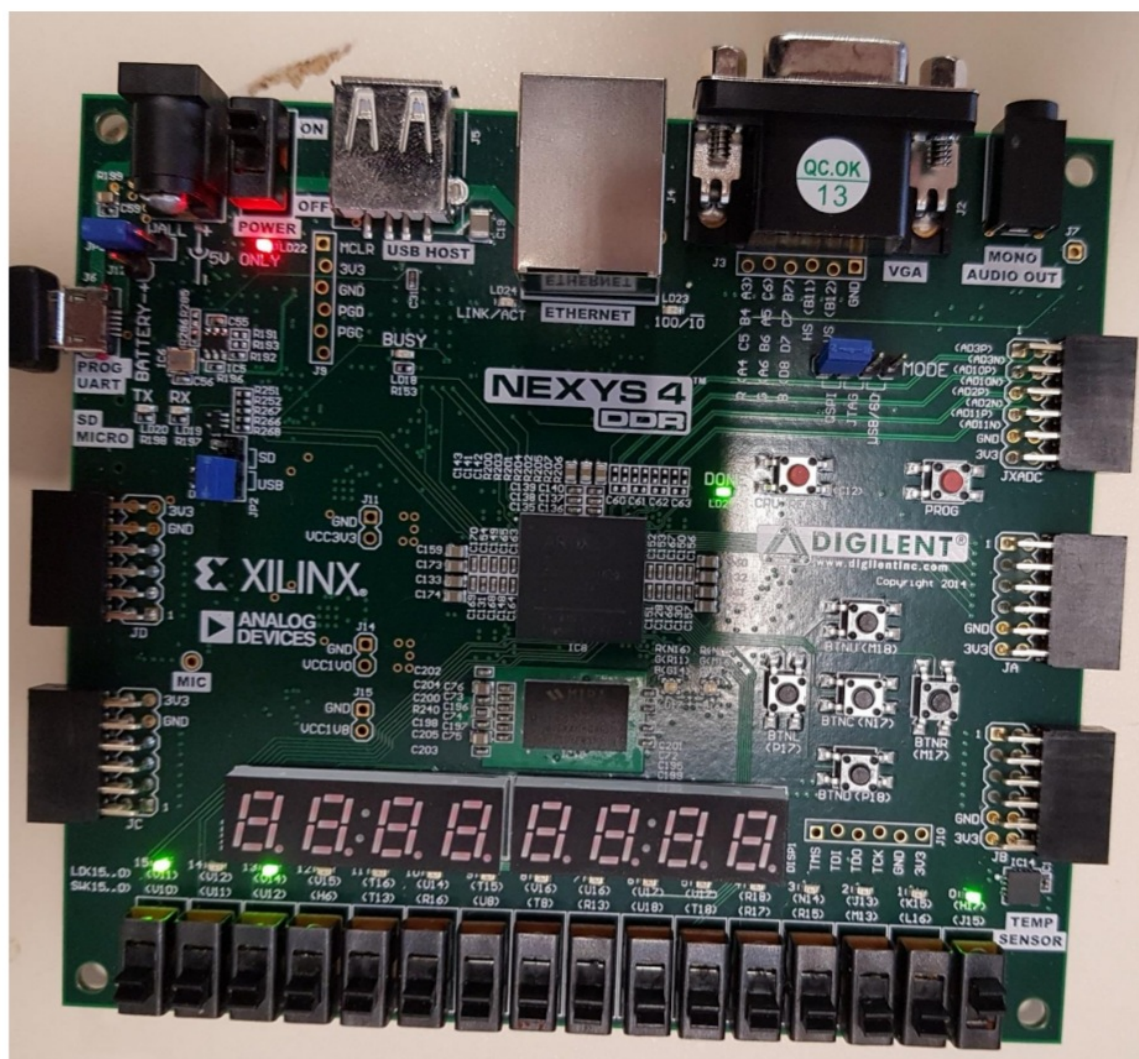


Figure 45 - TCS Nexus board test



Conclusions and recommendations

Circuit Adaption

After completing the traffic control system it was decided to improve the system by adding another input in the form of a pressure pad, a pedestrian pressing the wait button would have to stand on the pressure pad for the duration of the 5 second delay before the lights changed. If the pedestrian left the pad the sequence would stop and return to its original start position waiting for the wait button to be pressed and person on the pressure pad. Figure 47 shows the new layout of the design.

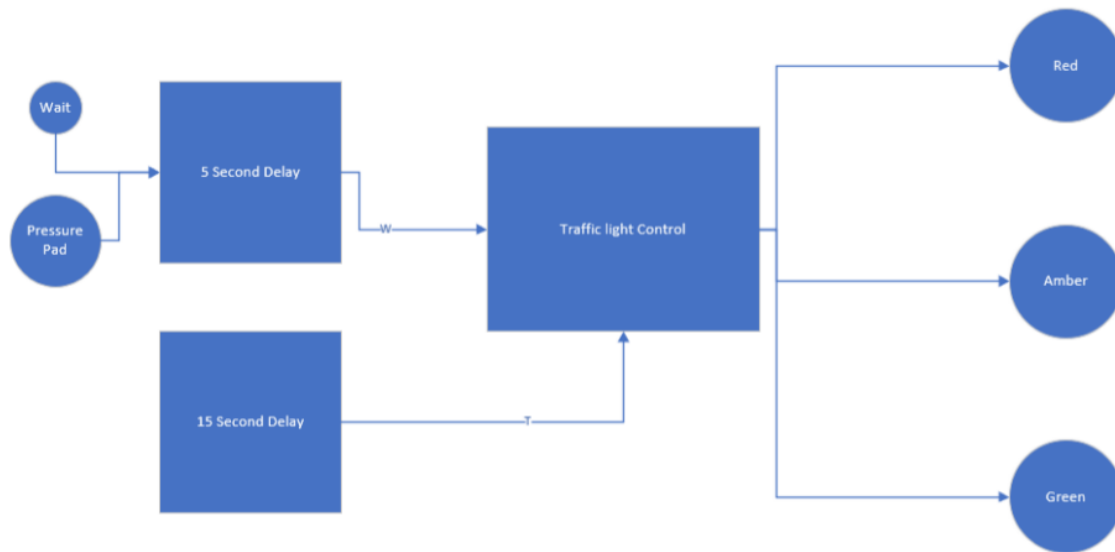


Figure 47 - TCS with pressure pad layout

The finite state machine in figure 48 shows the process the 5 second timer will be running through and indicates logics that affect it. The states for the diagram are set below. In order for the state to proceed from So the wait button designation (B) and Pressure pad (P) must be activated if not the timer will stay on So. When the timer completes a rotation to s5 the output (W) will be triggered and go to a state of 1, after it proceed to so and await further input from the wait button. At every state if (P) return a state of 0 then the count stops and returns to S0.

5 Second Delay

S0 -Reset

S1- 1 Second

S2- 2 Second

S3- 3 Second

S4- 4 Second

S5- 5 Second

Inputs

Wait Button – B

Pressure pad - P

Output

5 Seconds up – W

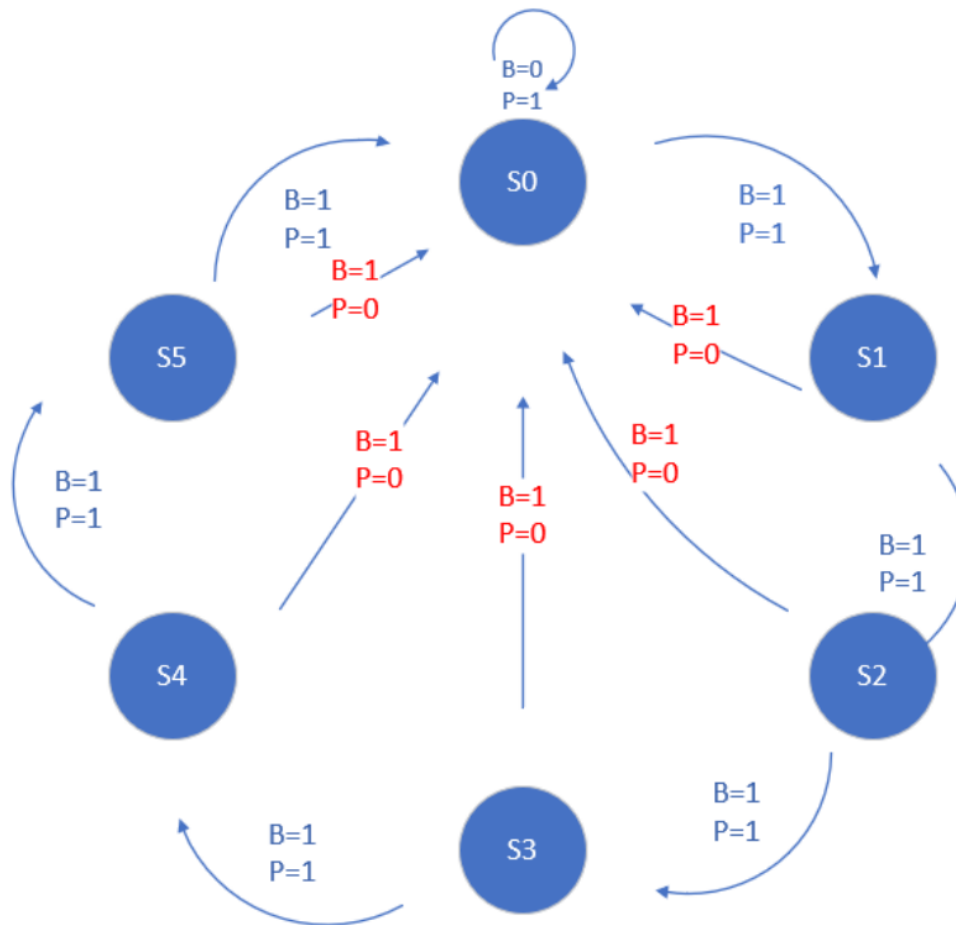


Figure 48 - State machine with pressure pad

State Table

Figure 49 shows the revised state table from the 5 second timer with pressure pad.

Present	B=0				B=1				Present				B=0				B=1				OUTPUT
	P=0	P=1	P=0	P=1	P=0	P=1	P=0	P=1	QA	QB	QC	QA	QB	QC	QA	QB	QC	QA	QB	QC	W
S0	S0	S0	S0	S1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
S1	S2	S2	S0	S2	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0
S2	S3	S3	S0	S3	0	1	0	0	0	1	1	0	1	1	0	0	0	0	1	1	0
S3	S4	S4	S0	S4	0	1	1	1	1	0	0	1	0	0	0	0	0	1	0	0	0
S4	S5	S5	S0	S5	1	0	0	0	1	0	1	1	0	1	0	0	1	0	0	1	0
S5	S0	S0	S0	S0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
S6	X	X	X	X	1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X
S7	X	X	X	X	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X

Figure 49 - State Table

Karnaugh Map

In figure 50 it can be seen that the revised results from the state table have been inputted into the Karnaugh maps and the results derived. A total of 13 separate AND gates and 3 separate OR gates will be required.

B=0					B=1				
	QB QC					QB QC			
P QA	00	01	11	10	P QA	00	01	11	10
00	0	0	1	0	00	0	0	0	0
01	1	0	X	X	01	0	0	X	X
11	1	0	X	X	11	1	0	X	X
10	0	0	1	0	10	0	0	1	0
$QA+ = QAQCB + QBQCB + PQAQCB + PQBQCB$									
	QB QC					QB QC			
P QA	00	01	11	10	P QA	00	01	11	10
00	0	1	0	1	00	0	0	0	0
01	0	0	X	X	01	0	0	X	X
11	0	0	X	X	11	0	0	X	X
10	0	1	0	1	10	0	1	0	1
$QB+ = QBQCB + QAQBQCB + PQBQCB + PQAQBQCB$									
	QB QC					QB QC			
P QA	00	01	11	10	P QA	00	01	11	10
00	0	0	0	1	00	0	0	0	0
01	1	0	X	X	01	0	0	X	X
11	1	0	X	X	11	1	0	X	X
10	0	0	0	1	10	1	0	0	1
$QB+ = QBQCB + QAQCB + PQCB$									
	QB QC					QB QC			
P QA	00	01	11	10	P QA	00	01	11	10
00	0	0	0	1	00	0	0	0	0
01	1	0	X	X	01	0	0	X	X
11	1	0	X	X	11	1	0	X	X
10	0	0	0	1	10	1	0	0	1
$QB+ = QBQCB + QAQCB + PQCB$									
	QB QC					QB QC			
QA	00	01	11	10					
0	0	0	0	0					
1	0	1	X	X					
$W = QAQC$									

Figure 50 shows the new 5 second timer with pressure pad input built into it and shows the circuit in operation with the proper lighting up in sequence.

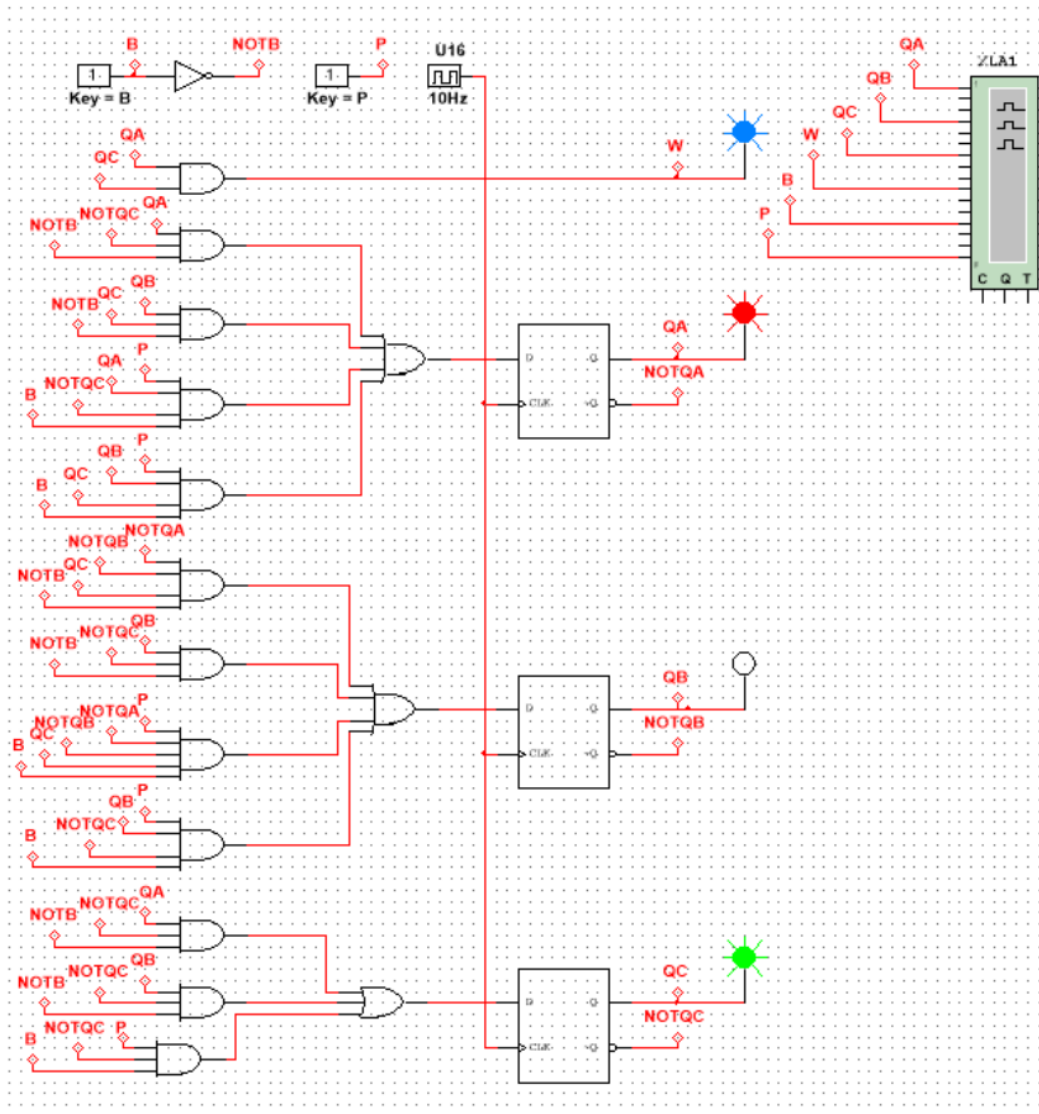


Figure 50 - Adapted design and testing

Figure 51 shows the results from the logic analyser which proves the new adapted 5 second timer work as the revised design with the count stopping when pressure pad is not activated, but with it activated the timer works as previous design.

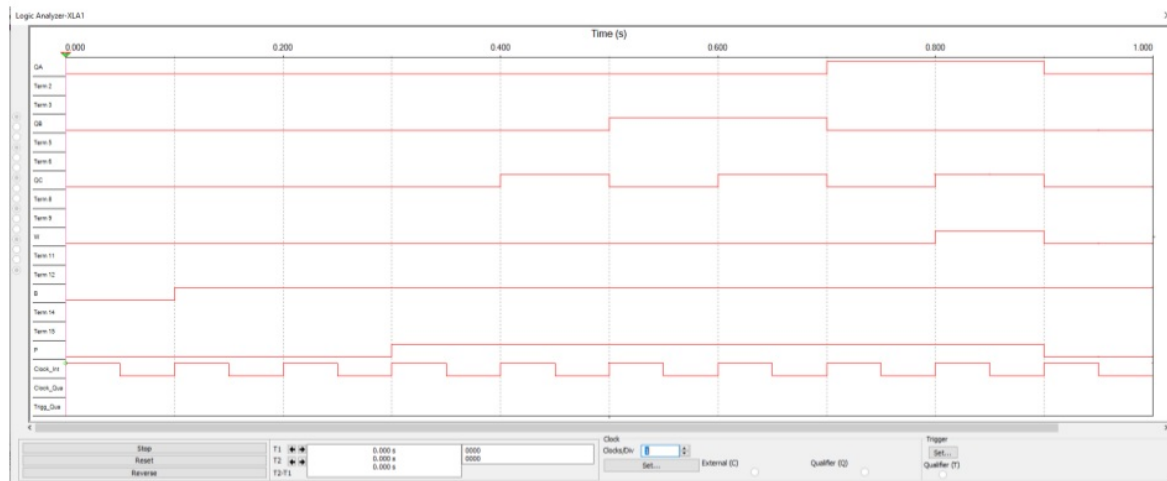


Figure 51 - Logic analyser results

Conclusion

In conclusion the Traffic Control System (TCS) in the first design worked adequately for its design purpose but like any design this can be improved and perfected for better and more efficient use.

By adapting the original design to include a pressure pad in the 5 second timer this has eliminated nuisance triggering of the pedestrian crossing which keeps the traffic moving. To further improve the design a 15 second count down timer display could be included to give the pedestrians a visual indicator of time remaining to cross before traffic lights change as seen in figure 52.

By increasing the efficiency of the design with further inputs the costs for the designs will increase with each further development, the need to balance cost and efficiency is usually the most prominent part of any design process and the traffic light system is no different.



Figure 52 - Pedestrian crossing with count down

Works Cited

Embedded, 2020. *Embedded*. [Online]

Available at: <https://www.embedded.com/using-finite-state-machines-to-design-software/>

[Accessed 05 01 2020].

Engineering, R. A. o., 2020. *Royal Academy of Engineering*. [Online]

Available at: <https://www.raeng.org.uk/>

[Accessed 05 01 2020].

geeksforgeeks, 2020. *geeksforgeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/difference-between-combinational-and-sequential-circuit/>

[Accessed 18 01 2020].

Numato, 2020. *Numato*. [Online]

Available at: <https://numato.com/blog/differences-between-fpga-and-asics/>

[Accessed 12 01 2020].

Figure 1: Comparison of combinational and sequential logic	4
Figure 2: Combinational circuit.....	4
Figure 3: Sequential circuit.....	5
Figure 4: Finite state machine	5
Figure 5 - FPGA	6
Figure 6 - ASIC chip	6
Figure 7: Pros and cons of the FPGA's and ASIC's.....	6
Figure 8 - Pelican crossing WAIT Button	8
Figure 9 - Pelican Crossing.....	9
Figure 10 - Traffic Light Layout Diagram	9
Figure 11 - Finite state diagram 5 second timer.....	10
Figure 12 - State Table 5 second timer	11
Figure 13 - Karnaugh map 5 second timer	12
Figure 14 - Multisim 5 second Karnaugh Map gates.....	13
Figure 15 - Multisim 5 second timer	14
Figure 16 - Multisim 5 second timer PLD Version.....	14
Figure 17 - 15 second Finite state machine.....	15
Figure 18 - 15 second State Table	16
Figure 19 - 15 Second timer Karnaugh maps.....	17
Figure 20 - Multisim 15 second timer	18
Figure 21 - 15 Second timer with flip flops	19
Figure 22 - 15 Second timer PLD.....	19
Figure 23 - RAG Controller Finite State Machine.....	20
Figure 24 - RAG Controller State Table	21
Figure 25 - RAG Controller Karnaugh Maps	22
Figure 26 - RAG Controller Karnaugh Maps	22
Figure 27 - RAG Controller Multisim	23
Figure 28 - RAG Controller PLD Multisim	23
Figure 29 - 5 second timer with logic analyser	24
Figure 30 - Logic analyser results for 5 second timer	24
Figure 31 - 5 second timer testing	25
Figure 32 - 15 second timer logic analyser connected.....	26
Figure 33 - 15 second logic analyser results.....	26
Figure 34 - 15 second timer testing sequence	27
Figure 35 - 15 second timer testing sequence two.....	27
Figure 36 - RAG Controller.....	28
Figure 37 - RAG Controller logic analyser results	28
Figure 38 - RAG Controller testing sequence	29
Figure 39 - RAG Controller PLD Mode.....	29
Figure 40 - Traffic control system	30
Figure 41 - Traffic control system test	30
Figure 42 - Traffic control system test	31
Figure 43 - TCS Nexus board test	31
Figure 44 - TCS Nexus board test	32
Figure 45 - TCS Nexus board test	32
Figure 46 - TCS Nexus board test	33
Figure 47 - TCS with pressure pad layout.....	34
Figure 48 - State machine with pressure pad.....	35
Figure 49 - State Table	35
Figure 50 - Adapted design and testing	37
Figure 51 - Logic analyser results.....	38

FINAL GRADE

GENERAL COMMENTS

Instructor

70/100

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

PAGE 40

PAGE 41

GRADING FORM: CSG-RUBIC

RESEARCH-INFORMED LITERATURE



There was some evidence of further reading and research and your interpretation and understanding of the additional research was satisfactory and demonstrated an underlying appreciation of the subject. Your references still rely on a small selection of mostly credible sources and you need to be more consistent with your referencing conventions e.g. citing diagrams as well as text.

7

KNOWLEDGE AND UNDERSTANDING



The knowledge and understanding of the subject demonstrated by this assignment was reasonably extensive. There were very few gaps in your knowledge of the basic principles and your understanding of the higher level concepts was demonstrated to a limited degree. You have demonstrated some critical understanding of the main concepts and expanded your knowledge beyond that contained in the course notes, however your explanations need to be more detailed and written with a greater emphasis on critical evaluation rather than description.

22

ANALYSIS



Your analysis of the task was generally very good, both in class and as shown in this assignment. You have demonstrated sound critical analysis skills and your approach to the task was logical. You carried out a detailed evaluation of the results and included a good conclusion relating the practical results to researched theory. The conclusions demonstrated very good cognitive skills and the ability to interpret quantitative and qualitative data without error.

19

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.

8