

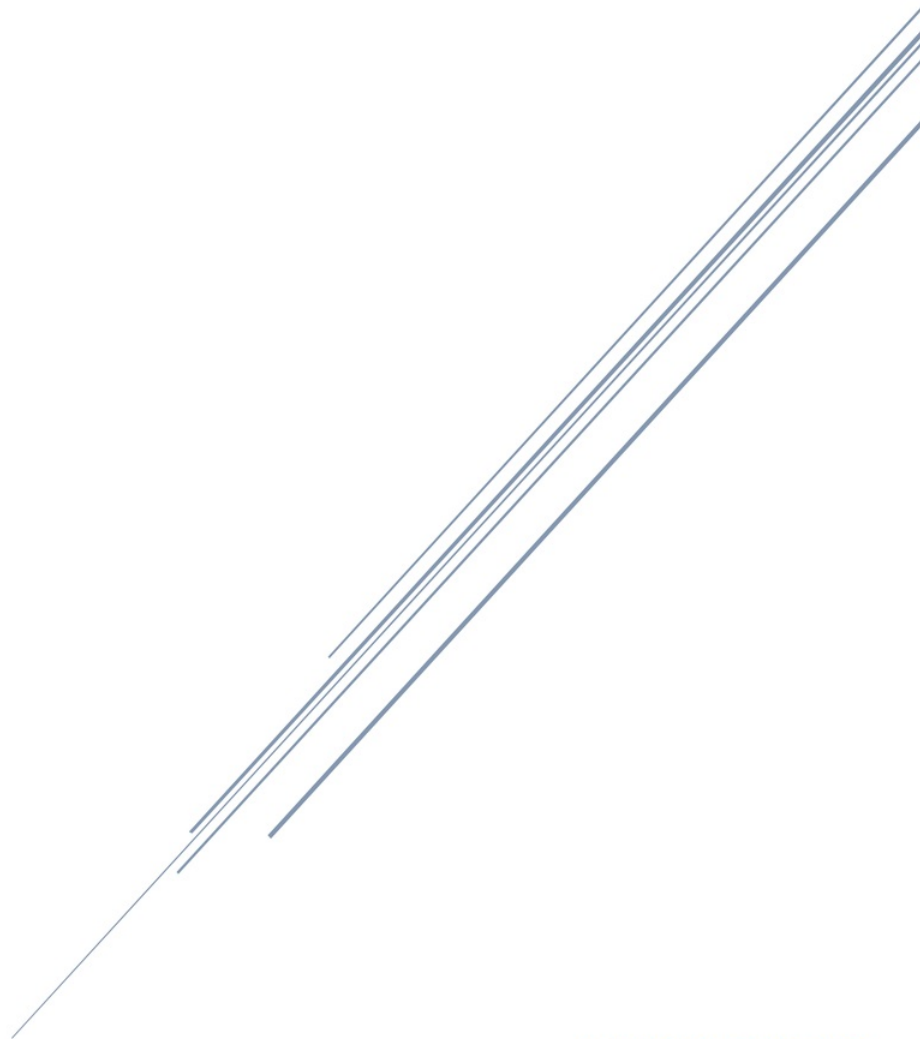
ASSIGNMENT 2

by Trystan Daniels

FILE	11396_TRYSTAN_DANIELS_ASSIGNMENT_2_152849_797741811.DOCX		
TIME SUBMITTED	24-APR-2018 07:34AM (UTC+0100)	WORD COUNT	1703
SUBMISSION ID	87285448	CHARACTER COUNT	10003

HNC ELECTRICAL AND ELECTRONIC ENGINEERING ELECTRONICS 2

PSDE5006 – ASSIGNMENT 2



TRYSTAN DANIELS

Table of Contents

Table of figures	1
1. Introduction	2
2. Research Into Advanced digital design techniques	3
Implementing finite state machines	3
Comparison of moore and mealy	3
3. Design method, choices and calculations	4
Specification.....	4
Design.....	4
5s counter	4
15s counter	6
Lights system controller	9
4. Testing and improving the design.....	10
5. Conclusion	14
6. Reference list	15

Table of figures

Figure 1	4
Figure 2	6
Figure 3	8
Figure 4	9
Figure 5	10
Figure 6	10
Figure 7	11
Figure 8	11
Figure 9	12
Figure 10	12
Figure 11	13
Figure 12	13

1. Introduction

The main aim of this assignment is to look into finite state machines, which will include building a system through the designing stages to downloading it to a Digilent nexys 4DDR FPGA board.

The first task is to research the different methods of implementing finite state machines, looking in particular into moore and mealy machines. Also researching into the different flip-flop types within state machine design.

Next, a system must be designed, tested on multisim and evaluated using the principles of finite state machines to control a pelican crossing. The design must include 3 output lights to simulate the traffic lights, an input "wait" for the pedestrians to press. This system must be fully automatic, run from internal clocks and timers.

The final task will be to carry out a test schedule on the system operation on the Digilent Nexys 4 DDR FPGA board.

2. Research Into Advanced digital design techniques

Implementing finite state machines

A finite state machine can only be in one state at a time, meaning in order for the machine to work it must move from one state to another. These FSM are made up from actions, transitions, events and states. (EETech Media, 2018)

Below is a truth table showing the states and next states, these are used in the designing process.

Present	Next	Present QA/QB/QC	QA+	QB+	QC+
S0	S1	000	0	0	1
S1	S2	001	0	1	0
S2	S3	010	0	1	1
S3	S4	011	1	0	0
S4	S5	100	1	0	1
S5	S0	101	0	0	0
			X	X	X
			X	X	X

Comparison of moore and mealy

A key difference in moore and mealy FSM machines is that a moore machines output depends on the input and the present state, whereas mealy's output depends only on the present state.

Below is a comparison table between these two types of FSM.

<u>Mealy Machine</u>	<u>Moore Machine</u>
Relies on input and current state to choose output	Only relies on current state
Normally tends to react faster to inputs	Usually slightly slower as it requires using further logic to decode outputs.
Fewer state than moore machines normally	Usually a Moore machine has more states than a mealy machine.

(berkeley, 2018)

3. Design method, choices and calculations

Specification

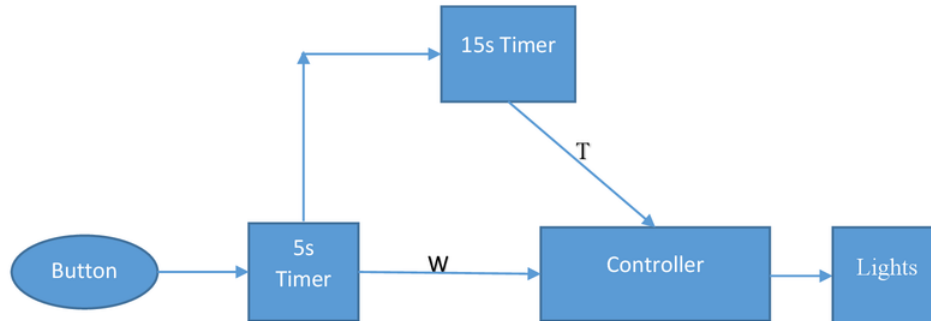


Figure 1

Above in figure 1 is the diagram of the circuit for the pelican crossing. This includes a button to activate the system, which triggers the 5second timer before changing the lights to red. When the red signal is active this triggers the 15s timer. When the 15s timer is up the lights return to green.

The lights order must be RED, RED/AMBER, GREEN, AMBER, RED.

Design

5s counter

Below is the design method used for the 5s counter.

The 5s counter will require 6 states, these 6 are:

S0 – Reset/count 0

S1 - count 1

S2 – count 2

S3 - count 3

S4 – count 4

S5 - count 5

The cycle will start on state 0 and work down the list until state 5 is reached, then reset and sent back to state 0 and carry on the continuous cycle.

Present state table

Present	Next	Present QA/QB/QC	QA+	QB+	QC+
S0	S1	000	0	0	1
S1	S2	001	0	1	0
S2	S3	010	0	1	1
S3	S4	011	1	0	0
S4	S5	100	1	0	1
S5	S0	101	0	0	0
			X	X	X
			X	X	X

The next stage is to do the karnaugh maps to cut the circuit size down as much as you can.

Key: not=A, B, C

QA

	00	01	11	10
0	0	0	1	0
1	1	0	1	1

$$QA+ = BC + AC$$

QB

	00	01	11	10
0	0	1	0	1
1	0	0	0	1

$$QB+ = BC + ABC$$

QC

	00	01	11	10
0	1	0	0	1
1	1	0	0	1

$$QC+ = C$$

From the karnaugh maps the circuit was built on multisim see figure below of the 6 counter circuit.

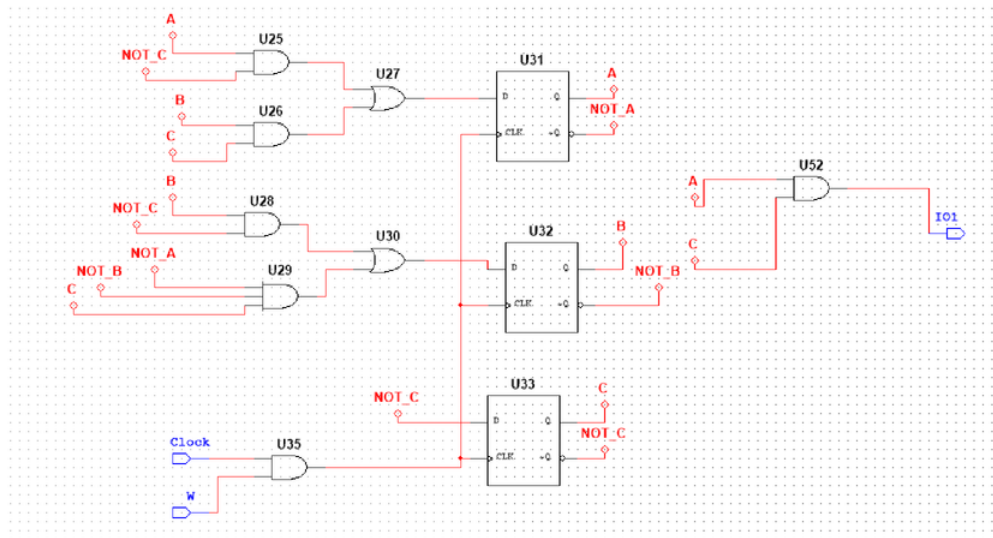


Figure 2

15s counter

Below is the design method used for the 15s counter.

The 15s counter will require 16 states, these 16 are:

S0 – Reset/count 0

S1 - count 1

S2 – count 2

S3 - count 3

S4 – count 4

S5 - count 5

S6 - count 6

S7 – count 7

S8 - count 8

S9 – count 9

S10 - count 10

S11 - count 11

S12 – count 12

S13 - count 13

S14 – count 14

S15 - count 15

		Current				Next			
Current	Next	Qa	Qb	Qc	Qd	Qa	Qb	Qc	Qd
S0	S1	0	0	0	0	0	0	0	1
S1	S2	0	0	0	1	0	0	1	0
S2	S3	0	0	1	0	0	0	1	1
S3	S4	0	0	1	1	0	1	0	0
S4	S5	0	1	0	0	0	1	0	1
S5	S6	0	1	0	1	0	1	1	0
S6	S7	0	1	1	0	0	1	1	1
S7	S8	0	1	1	1	1	0	0	0
S8	S9	1	0	0	0	1	0	0	1
S9	S10	1	0	0	1	1	0	1	0
S10	S11	1	0	1	0	1	0	1	1
S11	S12	1	0	1	1	1	1	0	0
S12	S13	1	1	0	0	1	1	0	1
S13	S14	1	1	0	1	1	1	1	0
S14	S15	1	1	1	0	1	1	1	0
S15	S0	1	1	1	1	0	0	0	0
		X	X	X	X	X	X	X	X
		X	X	X	X	X	X	X	X

Key: not=A, B, C

Qa+

AB\CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	1	0	1
10	1	1	1	1

$$Qa = AC + AD + AB + ABCD$$

Qb+

AB\CD	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

$$Qb = BC + BD + BCD$$

QC+

AB\CD	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$$QC = \overline{C}D + CD$$

QD+

AB\CD	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

$$QD = D$$

The cycle will start on state 0 and work down the list until state 16 is reached, then reset and sent back to state 0 and carry on the continuous cycle.

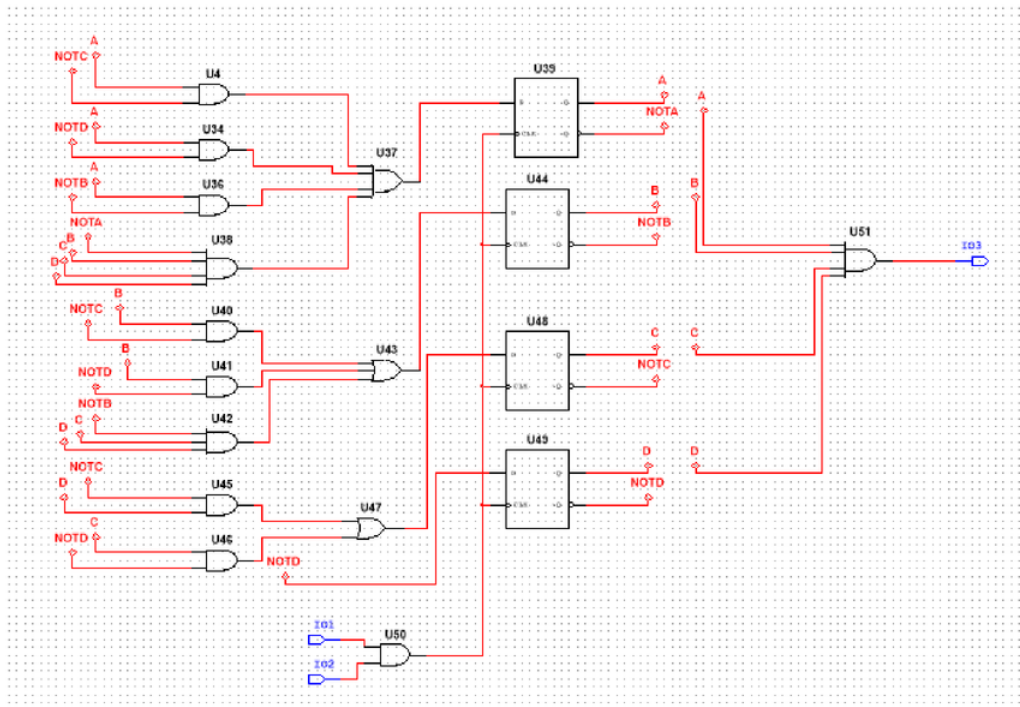


Figure 3

Figure 3 above shows the circuit in multisim for the 15s counter. This is drawn up using the design purpose, the karaugh maps reduce the circuit size to simplify.

Lights system controller

This controller must control the 3 light outputs using the two counter inputs.

This system will have 6 states which are:

S0 – Green

S1 - Amber

S2 – Amber

S3 - Red

S4 – Red/Amber

S5 – Red/Amber

Key: not=A, B, C

Qa+

AB\CD	00	01	11	10
00	0	0	0	0
01	1	0	X	X
11	1	0	X	X
10	0	0	1	0

$Qa = A \text{ NOT } C + TBC$

Qb+

AB\CD	00	01	11	10
00	0	1	1	1
01	0	0	X	X
11	0	0	X	X
10	0	1	0	1

$Qb = ABC + TB + BC$

Qc+

AB\CD	00	01	11	10
00	0	0	1	1
01	1	0	X	X
11	1	0	X	X
10	0	0	0	1

$QC = ABC + TC + BC + B$

R

A\BC	00	01	11	10
0	0	0	1	0
1	1	1	x	X

$$R = A + BC$$

A

A\BC	00	01	11	10
0	0	1	0	1
1	1	1	x	X

$$A = A + \textcolor{red}{B}C + B\textcolor{red}{C}$$

G

A\BC	00	01	11	10
0	1	0	0	0
1	0	0	x	X

$$G = \textcolor{red}{A}BC$$

State table

Current	W=0		W=1		QA+	QB+	QC+	QA+	QB+	QC+
	T=0		T=1							
	T=0	T=1	T=0	T=1						
CURRENT										
T = 0										
S0	S0	S0	S1	S1	0	0	0	0	0	0
S1	S2	S2	S2	S2	0	0	0	0	1	0
S2	S3	S3	S3	S3	0	0	1	0	1	1
S3	S3	S4	S3	S4	0	0	1	0	1	1
S4	S5	S5	S5	S5	1	1	0	1	0	1
S5	S0	S0	S0	S0	1	1	0	0	0	0

QA+	QB+	QC+	QA+	QB+	QC+	Output													
T = 1						T = 0			R			A			G				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

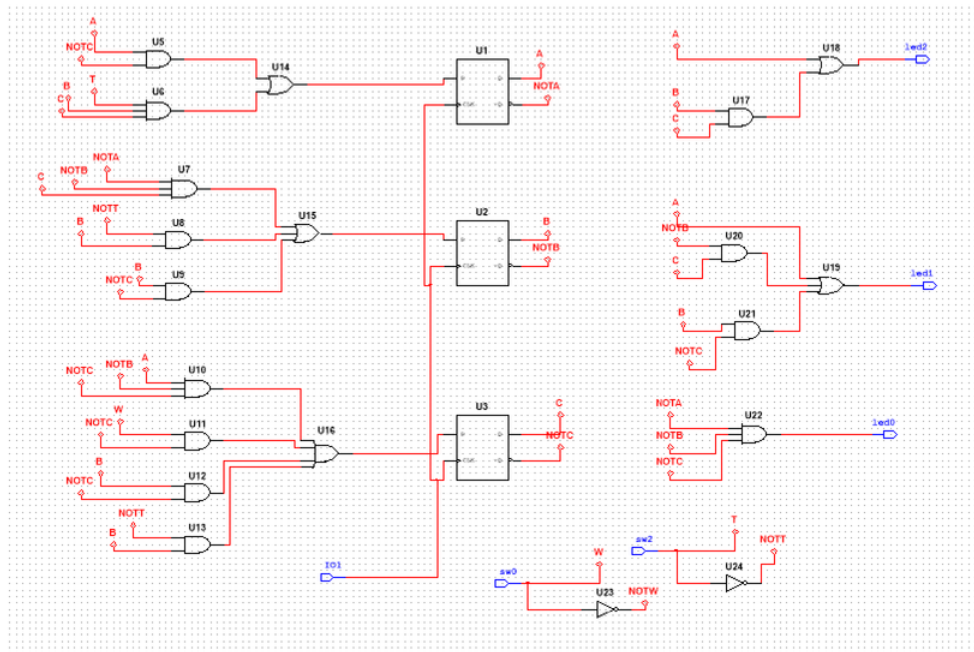


Figure 4

The figure above shows the system controller circuit in multisim, this controller controls the light output from the two counter inputs.

4. Testing and improving the design

6 counter logic analyser results

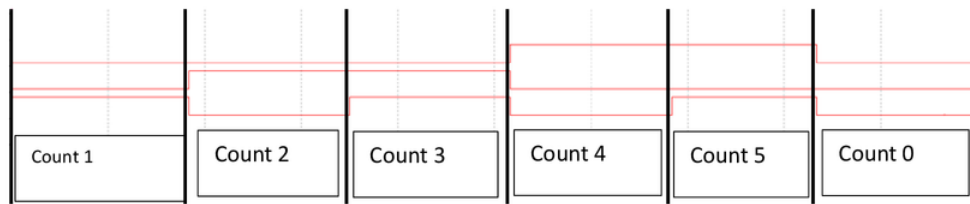


Figure 5

The figure above shows the logic analyser result for the 6 counter as you can see by the labels the counter does work from 0 to 5 as it should.

15second counter test

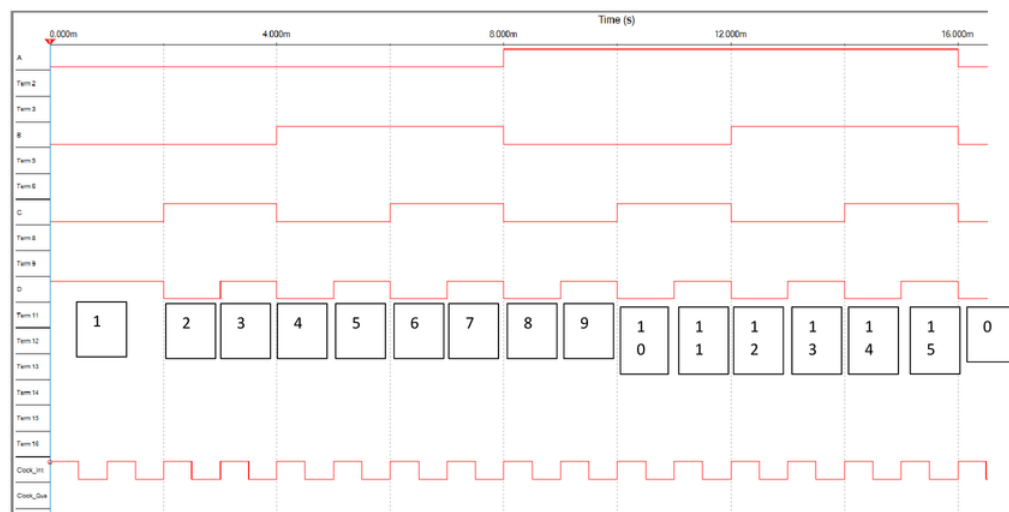


Figure 6

Above in figure the 15s counter is tested to ensure the circuit counts correctly and reset. From the figure it can be seen the counter counts up to 15 and back to 0 with no issues.

System built in multisim

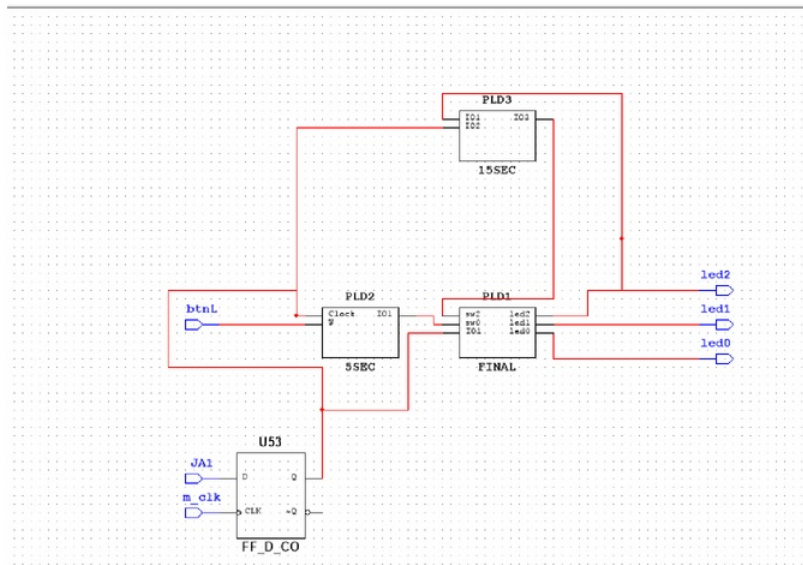


Figure 7

The figure above shows the circuits connected together and put into sub circuits to ensure simplicity. This circuit can be downloaded to the board now as you can see below.

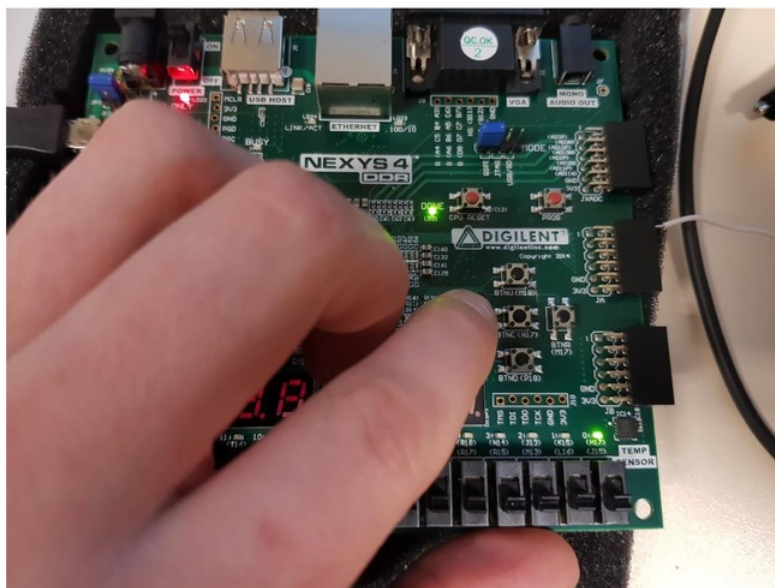


Figure 8

Figure shows the system running on the board as the wait button is pressed. The first LED on meaning green light.

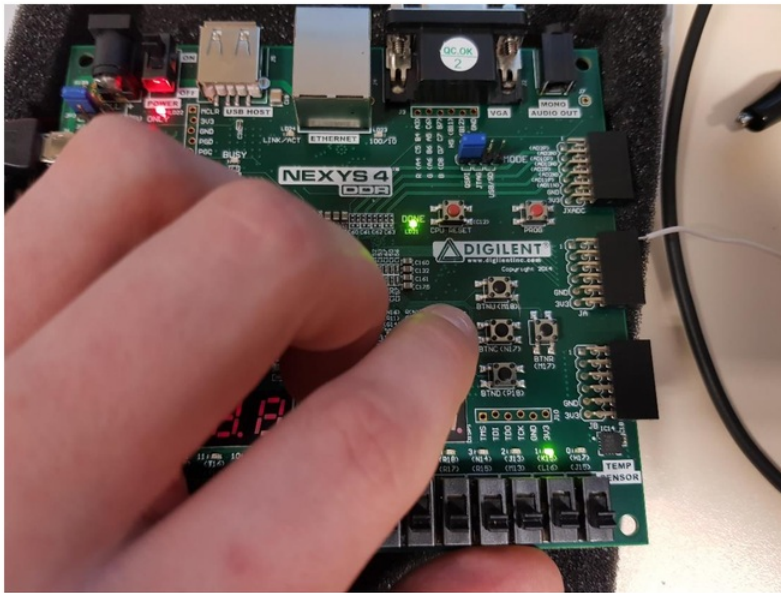


Figure 9

figure 9 show the wait button had been pressed and it waited 5seconds before switching the lights to amber. (LED 2)

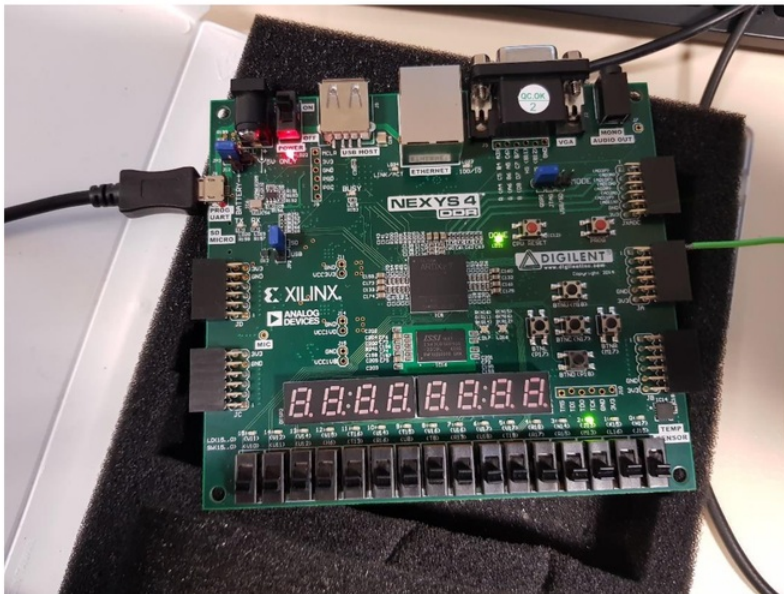


Figure 10

figure 10 shows the lights on red (LED 3), now the 15second timer will start, when the timer is up the lights should return to green in sequence.

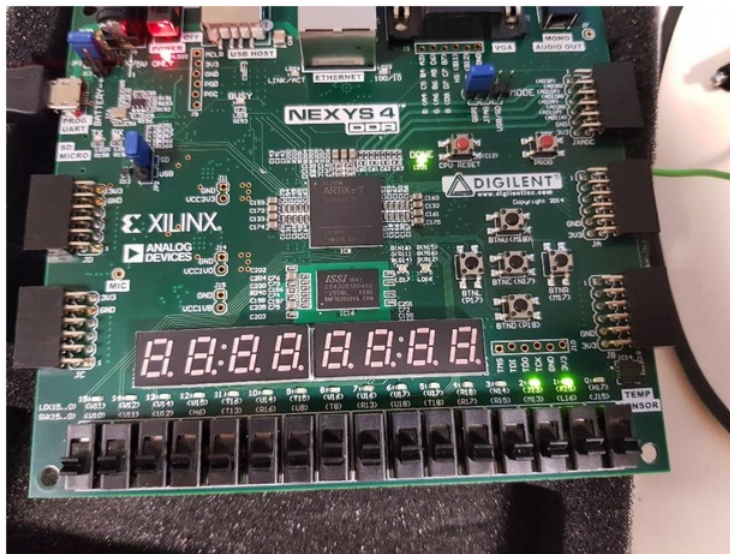


Figure 11

Figure 11 shows the 15s timers was up and the lights turned to red/amber (LED 2 and 3). Now the lights should return to green.

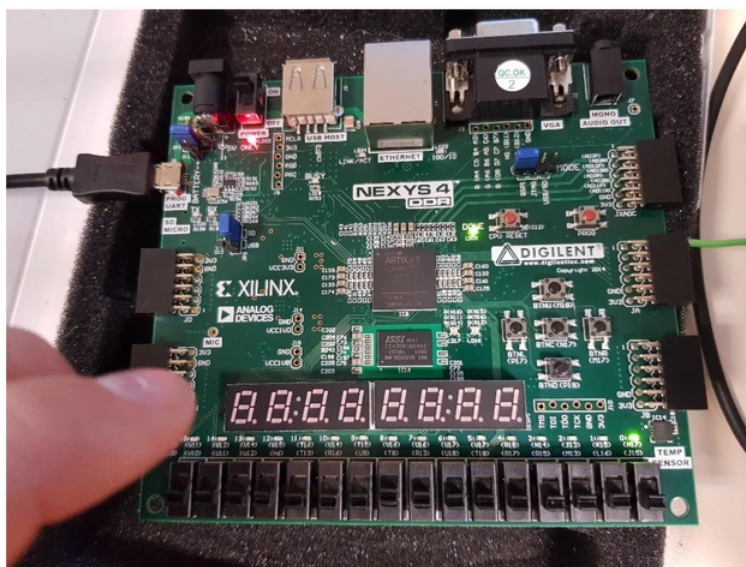


Figure 12

Figure 12 shows the lights returned to green (LED 1). That is the full sequence completed and the system tested successfully.

5. Conclusion

Overall all tasks were successful, gaining key knowledge into finite state machine in the research task to enable to design the system and understand it better. The theory and research does link in and follows the outcomes when designing and testing the system. When designing a system like this it was learnt that no mistake could be done as it effects the outcome of the system, if they're were any the system must be totally redesigned unlike PLC systems. The system testing was successful when downloading and testing with the board as can be seen in figures 9 to 12.

6. Reference list

References

berkeley. (2018, 4 20). *FSM*. Retrieved from Sequential Logic Implementation:
<https://inst.eecs.berkeley.edu/~cs150/fa05/Lectures/07-SeqLogicIIx2.pdf>

EETech Media, L. (2018, 04 20). *Implementing a Finite State*. Retrieved from allaboutcircuits:
<https://www.allaboutcircuits.com/technical-articles/implementing-a-finite-state-machine-in-vhdl/>

ASSIGNMENT 2

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

54/100

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

RESEARCH-INFORMED LITERATURE



There was some evidence of further reading and research and your interpretation and understanding of the additional research was OK. The sources were insufficient at this level given the amount of potential research areas for this assignment.

2

KNOWLEDGE AND UNDERSTANDING



The knowledge and understanding of the subject demonstrated by this assignment was generally good. 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 could have explained in more detail how the circuit evolved. The design process was well explained for the basic circuits, but not for the overall system.

18

ANALYSIS



Your analysis of the task was generally satisfactory, both in class and as shown in this assignment. You have demonstrated some analytical skills and your approach to the task was logical. You carried out a short analysis and a short conclusion. Further work – in your conclusion, you should relate the results to the original specification and to accepted theory.

9

PRACTICAL APPLICATION AND DEP.



Your application of practical skills for this task was generally very good. You used the resources available in MultiSim 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.

18

SKILLS FOR PROFESSIONAL PRAC.



In this assignment you have demonstrated skills in working individually and also collaboratively. Your written report was of an appropriate length, you could however be more efficient with your word count. The report was well presented, with good diagrams in appropriate places, good sentence/paragraph structure and clear calculations.

7