

2024888

by Ethan Rigby

Submission date: 18-Jan-2022 06:21PM (UTC+0000)

Submission ID: 169510020

File name: 19740_Ethan_Rigby_2024888_179500_758552180.docx (957.31K)

Word count: 2642

Character count: 13290

Course: HNC Electrical and Electronic Engineering
Module: SGEE5000 – Electronics2
Assignment Title: Assignment 1 – Traffic light system
Name: Ethan Rigby
University number: 2024888
Date Submitted: 18/01/2022
Word Count: 2400

- ① BRIEF RESEARCH NEEDED MORE TECHNICAL DETAILS.
- ② DESIGN METHOD WAS EXPLAINED, BUT KARNAUGH MAP GROUPS NOT SHOWN.
- ③ TESTING WAS SHOWN IN THE APPENDIX, BUT ANALYSIS WAS BRIEF.
- ④ CONCLUSIONS WERE BRIEF WITH ONLY A CURSORY MENTION OF HOW A SUSTAINABLE IMPLEMENTATION COULD BE ACHIEVED.
- ⑤ MODIFICATION NOT ATTEMPTED.

Table of Contents

Executive Summary	3
Research into advanced digital design techniques	4
Types of pedestrian crossings	4
Pelican crossing	4
Puffin crossing	4
Toucan	4
Control circuits on traffic light and public crossings	5
FPGAs	5
Description	5
Use in traffic lights/crossings	5
ASICs	5
Description	5
Groups of ASICs	6
Programable ASIC	6
Use within a traffic/crossing light	7
Sustainable electrical elements	7
Design method, choices and calculations	8
0-5 second counter	8
Specification	8
State diagram	8
State table	9
Karnaugh maps with Boolean equation	9
Multisim circuit	10
0-15 second counter	10
Specification	10
State diagram	10
State table	12
Karnaugh maps with Boolean equation	12
Red, amber, and green control	13
Specification	13
State diagram	14
state table	14
Karnaugh maps with Boolean equation	15
Multisim circuit	16
Testing, results and analysis	17

Conclusions and recommendations.....	17
Reference list and bibliography	19
References	19
Annexe	21

Executive Summary

The purpose of this report is to explore the function, reliability and sustainability of the traffic light systems located within the UK. Through design methods and theory-based equations complex circuits will be created to work together and provide the correct sequencing of a full traffic light system. Multiple methods of testing and analysing these circuits will allow the understanding of how the lights work and will allow the implementation of upgrades within the system.

*THIS SHOULD BE A LOT LONGER AND
COVER THE ENTIRE REPORT.*

Research into advanced digital design techniques

Types of pedestrian crossings

Pelican crossing

The pelican crossing is a style of public road crossing system designed using a range of timers. Upon approach the member of public will press a button that tells the system to conduct the crossing sequence. As the green light (green man) appears on the crossing the drivers' lights will appear red. After a period, the amber light will start to flash on the car lights. This means that if there is somebody on the crossing then the vehicles must stay stationary, but if there is nobody on the crossing the vehicles must not move. (King, 2020)

Puffin crossing

A puffin crossings sequence is the same as a pelican's sequence, but its fundamental difference is that unlike the pelican crossing it uses a range of sensors to determine how many people are on the crossing and how fast they move before allowing traffic to continue. If the public cross the road quickly then the green light will appear quicker for the drivers, but if there are more people on the crossing or they are slower then the green traffic will have to wait longer. This system is replacing the pelican crossing due to its ability to increase traffic flow in busy locations. (King, 2020)

Toucan

This crossing can use either a pelican or puffin lighting system. Its unique feature is that cyclists can cross this crossing without the need to dismount their bikes. The green light for the bicycles appears prior to the main green light in order to give the cyclist some time to move preventing them bumping into the public or vehicles. (King, 2020)

Control circuits on traffic light and public crossings

FPGAs

Description

The 'Field Programmable Gate Array' FPGA is a type of integrated circuit with internal hardware blocks that be reprogramed an unlimited time during its lifetime. This provides the user the ability to add modifications to current systems or even completely change the application of the circuit. The FPGAs design consists of thousands of 'Configurable Logic Blocks' (CLB) that are surrounded by programable interconnections between the CLBs. I/O blocks surround the CLBs that connect to external devices (the formation is shown in 'Figure 8'). (Arrow Electronics, 2018)

Use in traffic lights/crossings

FPGAs are heavily used in the creation of a new traffic system due to their volatility. The FPGA allows the user to create a perfect prototype and simulate the system without the need to permanently burn the program onto the circuit. To use them in batch though is rare for a few main reasons. Firstly, when the FPGA is programmed for its permanent use, the program is readable and changeable. This means that the program is stealable to other companies which is not desirable. Also, buying these in batch is more expensive that ordering a large volume of ASICs that have permanently burnt on programs. (WatElectronics, 2019)

ASICs

Description

The 'Application Specific Integrated Circuit' is as stated an integrated circuit. Much like the FPGA it can hold programs to facilitate whatever application is desired. Although, unlike the FPGA when the ASIC is created for its application it can not be reprogramed or edited as the program is burnt onto the circuit. (Electronics Hub, 2020) As ASICs are designed from the root for the application they are higher in cost than the FPGA so they are recommended to buy in bulk. The reason for the desire to have these ASICs over the FPGAs and the reason for the higher cost is due to the lower power these circuits pull as the only elements on this circuit are the ones that are required for the application desired. Also, these circuits are lower in size to the FPGAs due to the lower level of components on them.

Groups of ASICs

ASICs are separated into 3 groups that have different levels of customization available to the programmer. The following groups are described below and shown in 'figure 9'.

Fully custom

The fully customized ASIC is the most efficient and desirable ASIC for a specific application. Its creation is purely devoted around the application it's needed for. This type of ASIC can take as long as eight weeks to be designed but benefits with heightened performance, higher flexible and smaller in physical size. (Elprocus, 2022)

Semi-custom

The semi-custom ASIC is designed in a series of blocks. These blocks are standardised library blocks whilst some of them can have a layer of customisation if the specifications require them to be. This type of ASIC is made up of multiple sections, these sections being the following:

Standard cell-based ASIC- use of standardised logic cells such as AND gates, OR gates, flip flops etc. only the placement of these and the interconnections between is customised on the semi-custom ASIC. 'Figure 10' shows the structure of the standard cell-based ASIC

Gate array-based ASIC- made up of repetitive predefined transistors on the silicon wafer the gate array the base cells are the most repetitive cells on the base array. (Elprocus, 2022) the only thing the designer can change is the interconnections.

Programable ASIC

Programable ASIC chip- this is a PLD (programmable logic device) the internals are made of logic gates, latched, flip flops etc, this gives the designer full programable ability within the specifications of the PLD. This is the most customisable part of the ASIC (Elprocus, 2022)

Use within a traffic/crossing light

The ASIC is used in the final design of the traffic system. These chips are bought in large batches due to the expense of them. When the program is finalized and burnt onto the ASIC it cannot be reprogrammed or the program can not be read or copied of it. The ASIC will never be used in the prototypic stages of the traffic system due to the expense of having to keep buying new ones to re-test.

Sustainable electrical elements

The use of LED strips within traffic lights is becoming the new normal protocol, this is due to their extended lifetime and the energy efficiency that comes with the use of them, they also are more clearly visible to drivers and pedestrians.

The use of the ASIC and FPGA has revolutionised the traffic system by making the controls a lot smaller in size, more energy efficient and easier to replace/install.

BRIEF RESEARCH, YOU COULD HAVE LOOKED
INTO THE TECHNICAL ASPECTS OF VARIOUS
CROSSINGS. DIAGRAMS WOULD HELP.

Design method, choices and calculations

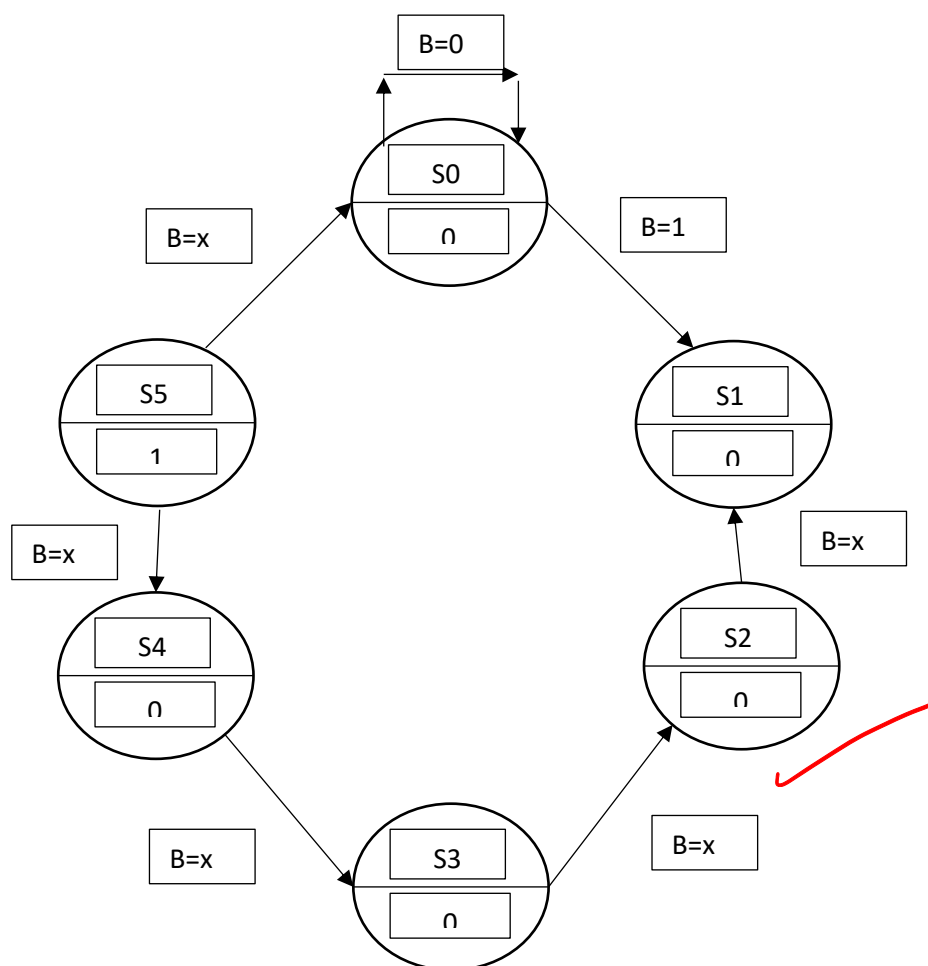
0-5 second counter

Specification

- ❖ Count 0-5 seconds once without repeating after signalled.
- ❖ Start when B=1
- ❖ Make W=1 when the count is 5

State diagram

The state diagram shows the outputs occurring in order after each input is met. The whole design of the traffic light system starts at the state diagram.



State table

The state table will take information from the state diagram and create a series of 1's and 0's that can be taken and turned into equations.

CURRENT STATE	NEXT STATE		CURRENT			B=0		NEXT		B=1		OUTPUT
	R=0	R=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	1
S5	S0	S0	1	0	1	0	0	0	0	0	0	0
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

Karnaugh maps with Boolean equation

The information from the state table created these Karnaugh maps, by pairing the values in the maps using the appropriate rules equations can be formed, these equations tell a designer exactly how the circuit should be created.

	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
$QA+=QA \text{ NOT } QC + QB \text{ QC}$					

	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
$QB+=QB \text{ NOT } QC + \text{ NOT } QA \text{ NOT } QB \text{ QC}$					

Groups not shown.

	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
	$QC = QB \text{ NOT } QC + QA \text{ NOT } QC + B \text{ NOT } QC$				

USE SUBSCRIPT FOR CLARITY.

Multisim circuit

QA

In 'figure 1' we see the 0-5 second counter, this is created using a range of logic gates and flip flops in which the placement is decided by the Boolean equations. A manual input switch is used to represent the input coming into the counter that starts the sequence. In 'figure 1' the system is still in the testing stage so there is an internal clock added to the flip flops to function. In the final design this was removed and connected to an external clock as well as the input 'B'.

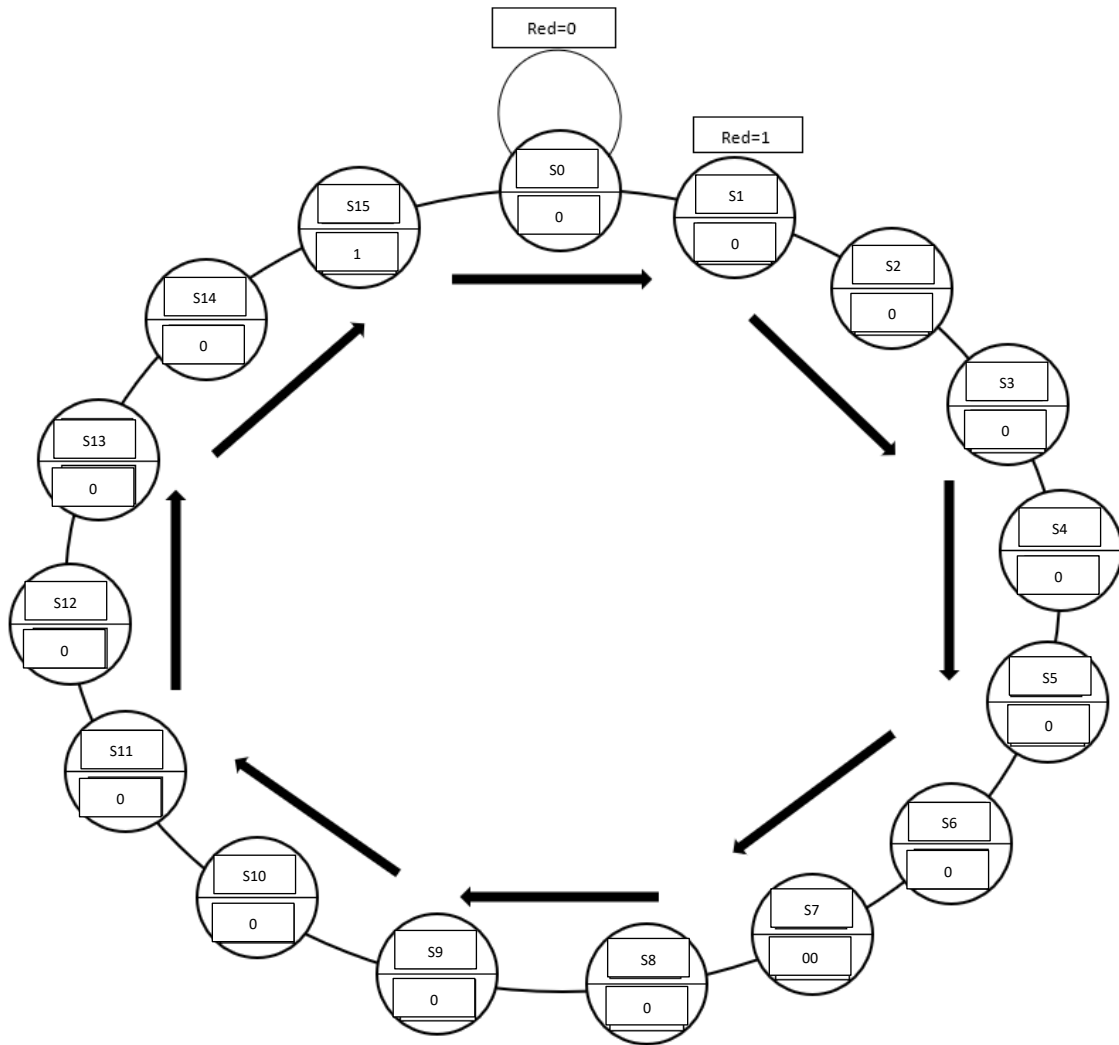
In the 0-15 second timer

0-15 second counter

Specification

- ❖ Count from 0-15 seconds without repeating
- ❖ input R is given to start the counting
- ❖ When count reaches 15 output T=1

State diagram



State table

CURRENT STATE	NEXT STATE		CURRENT				R=0		NEXT				R=1		OUTPUT
	R=0	R=1	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	1	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 with Boolean equation

The Karnaugh maps are designed in pairs as the left represents when the R=0 and right represents when R=1. They interlink in grouping to give the Boolean equation.

QAQB	QCQD				QAQB	QCQD			
	00	01	11	10		00	01	11	10
00	0	0	0	0	00	0	0	0	0
01	0	0	1	0	01	0	0	1	0
11	1	1	0	1	11	1	1	0	1
10	1	1	1	1	10	1	1	1	1
$QA+= QA \text{ NOT } QB + QA \text{ NOT } QC + QA \text{ NOT } QD + \text{ NOT } QA \text{ QB } QC \text{ QD}$									
QAQB	QCQD				QAQB	QCQD			
	00	01	11	10		00	01	11	10
00	0	0	1	0	00	0	0	1	0
01	1	1	0	1	01	1	1	0	1
11	1	1	0	1	11	1	1	0	1
10	0	0	1	0	10	0	0	1	0
$QB+= QB \text{ NOT } QC + \text{ NOT } QB \text{ QC } \text{ NOT } QD + QB \text{ NOT } QD$									

	QCQD						QCQD				
QAQB		00	01	11	10	QAQB		00	01	11	10
	00	0	1	0	1		00	0	1	0	1
	01	0	1	0	1		01	0	1	0	1
	11	0	1	0	1		11	0	1	0	1
	10	0	1	0	1		10	0	1	0	1
QC+= NOTQC QD + QC NOTQD											

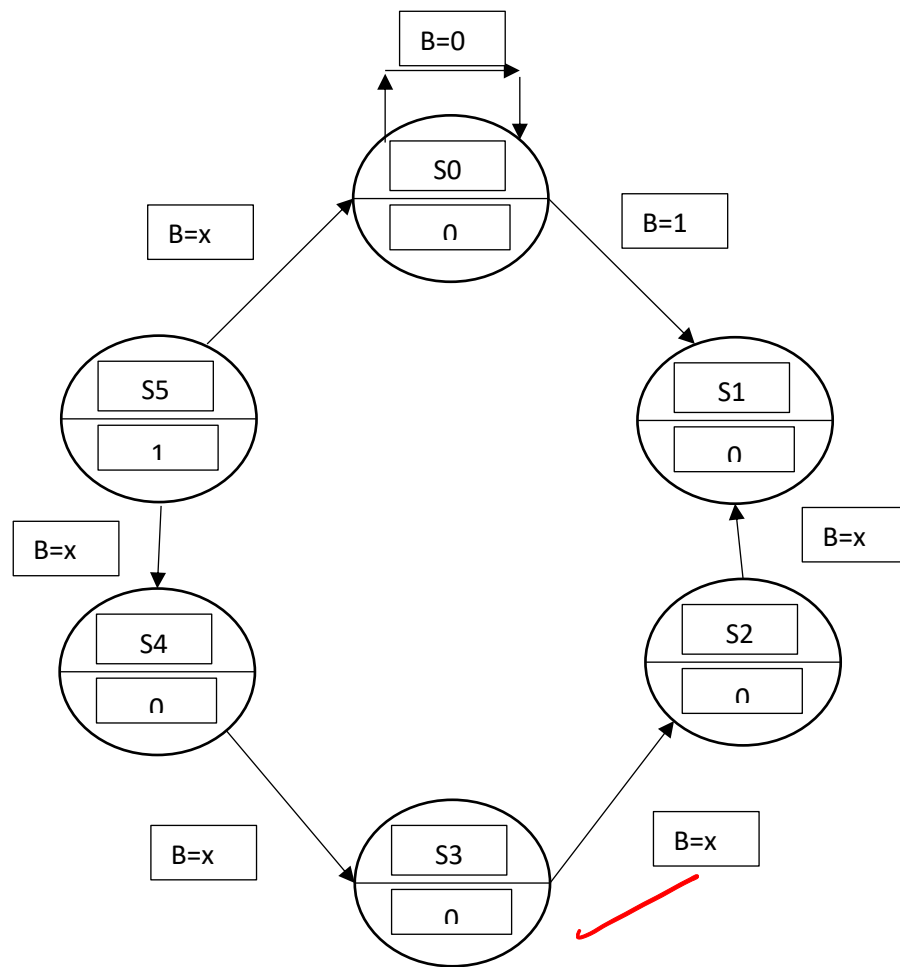
	QCQD						QCQD				
QAQB		00	01	11	10	QAQB		00	01	11	10
	00	0	0	0	1		00	1	0	0	1
	01	1	0	0	1		01	1	0	0	1
	11	1	0	0	1		11	1	0	0	1
	10	1	0	0	1		10	1	0	0	1
QD+= QC NOTQD + NOTQD R + QB NOTQD + QA NOTQD											

Red, amber, and green control

Specification

- ❖ Sequence the red, amber, and green lights using the outputs from the 0-5 and 0-15 second timers.
- ❖ Correctly sequence the red, amber, and green in the right order.

State diagram



state table

CURRENT STATE	NEXT STATE				T=0			T=1			NEXT STATE			T=0			T=1			OUTPUTS		
	W=0	T=0	T=1	W=1	QA+	QB+	QC+	QA+	QB+	QC+	QA+	QB+	QC+	QA+	QB+	QC+	QA+	QB+	QC+	R	A	G
S0 G	S0	S0	S1	S1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1
S1 A	S2	S2	S2	S2	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
S2 A	S3	S3	S3	S3	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0
S3 R	S3	S4	S3	S4	0	1	1	1	0	0	0	1	1	1	0	1	1	0	1	1	0	0
S4 R/A	S5	S5	S5	S5	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	1	0
S5 R/A	S0	S0	S0	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
S6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
S7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Karnaugh maps with Boolean equation

		W=0						W=1			
		QBQC						QBQC			
TQA		00	01	11	10	TQA		00	01	11	10
	00	0	0	0	0		00	0	0	0	0
	01	1	0	X	X		01	1	0	X	X
	11	1	0	X	X		11	1	0	X	X
	10	0	0	1	0		10	0	0	1	0
$QA += QA \text{ NOT } QC + T \text{ QB } QC$											

		W=0						W=1			
		QBQC						QBQC			
TQA		00	01	11	10	TQA		00	01	11	10
	00	0	1	1	1		00	0	1	1	1
	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
$QB += \text{NOT } QA \text{ NOT } QB \text{ QC} + \text{NOT } T \text{ QB} + \text{QB NOT } QC$											

		W=0						W=1			
		QBQC						QBQC			
TQA		00	01	11	10	TQA		00	01	11	10
	00	0	0	1	1		00	1	0	1	1
	01	1	0	X	X		01	1	0	X	X
	11	1	0	X	X		11	1	0	X	X
	10	0	0	0	1		10	1	0	1	1
$QC += QA \text{ NOT } QC + \text{NOT } T \text{ QB} + \text{NOT } QC \text{ W} + \text{QB W}$											

		QBQC			
QA		00	01	11	10
	0	0	0	1	0
	1	1	1	X	X
$R = QA + QB \text{ QC}$					

		QBQC			
QA		00	01	11	10
	0	0	1	0	1
	1	1	1	X	X
$A = QA + \text{NOT } QB \text{ QC} + \text{QB NOT } QC$					

Groups and subscripts

	QBQC				
QA		00	01	11	10
	0	1	0	0	0
	1	0	0	X	X
	G= NOTQA NOTQB NOTQC				

Multisim circuit

In 'figure 1' we see the 0-5 second counter, this is created using a range of logic gates and flip flops in which the placement is decided by the Boolean equations. A manual input switch is used to represent the input coming into the counter that starts the sequence. In 'figure 1' the system is still in the testing stage so there is an internal clock added to the flip flops to function. In the final design this was removed and connected to an external clock as well as the input 'B'.

In the 0-15 second timer shown in 'figure 3' the logic gate and flip flops also made up the circuit in the same way the 5 second counter did, by using the Boolean equations. The 15 second counter used the input from the red light as a signal to start counting. The system is also in testing so the clock is internal to the circuit, this will also be linked to the same clock as the 5 second counter.

The RAG controller takes the outputs from both the 5 second counter and the 15 second counter and activates the lights in the order that it is designed to do. The first part of the circuit takes the outputs and sequences them through the flip flops, the second section takes these and decides which lights are on and for how long they are on. This is also linked to the same clock as the other two circuits.

Testing, results and analysis

On the 0-5 second counter and the 0-15 second counter the use of a logic analysers showed the sequencing of the circuits and allows us to read back through the state table to see if the sequencing is correct. On the 0-5 second counter 'figure 2' shows the results given from the logic analyser, when the line is high it means the value is 1 and when it is low the value is 0. When looking at the counter we can see the states that occur before the output is activated. This output is the signal that is sent off to the RAG controller telling it that the count is 5. The use of indication lights was also done, by setting the internal clock to a low value then it gives time for the indication lights to be seen. This doesn't show as much as the logic analyser, but it does show that the counter is working in a sequence. Much like the 5 second counter the 0-15 second counter is also tested using the logic analyser and the indication lamps. The indication lamps showed the circuit was sequencing, in 'figure 3' we see that the sequencing is in the correct order and the output T is coming on at the correct stage of the sequencing. This means that if the RAG controller is set up correctly then the lights should come on in the correct sequence.

The RAG controller was much easier to see working with the use of lamps as 'figure 5' shows, as the other two circuits were working, they could be used to test the functionality of the RAG circuit. When the circuits are all interconnected the external clock can be used and the external start button. When the button is pressed and released the indication lights for the traffic light sequence in the correct order. This means that the sequence works. A logic analyser was also connected to the R, A, and G outputs to see when they are activated, this is shown on 'figure 7'. The logic analyser showed the order for the lights and when they deactivated, and it matched what was required in the state diagram.

Conclusions and recommendations

After analysing the results given by all the tests on each circuit and on the overall circuit the designing of the system and the theory behind it was correct. With the use of the PLD designs the traffic lights worked efficiently and as requested by the specifications. From the research it can be recommended that the FPGAs should be used throughout the designing and testing stages, this is due to it being reprogrammable. Following this the use of the ASIC will be used now that the system is working correctly and that it should be bought in a large batch to decrease the overall cost.

The full specification of the brief was not reached in this system as the public crossing circuit was not created within the design, this was due to time management errors so this should be organised better next time to allow the full design of the system.

Following the research, the sustainability and reliability of traffic lights are essential to its purpose, modernised methods have increased the reliability by adding LEDs into the bulbs, the LEDs will last a lot longer than the old-style lamps they used to have behind coloured lenses. The use of microcontrollers such as ASICs have also maintained the reliability of these lights, but, they have also reduced the cost of the lights by reducing the energy efficiency of them. The LEDs and the microcontrollers have a lower power draw to original methods of controlling the lights.

Reference list and bibliography

References

Arrow Electronics, 2018. *FPGA Basics: Architecture, Applications and Uses*. [Online]
Available at: <https://www.arrow.com/en/research-and-events/articles/fpga-basics-architecture-applications-and-uses>
[Accessed 14 01 2022].

Electronics Hub, 2020. *Introduction to ASIC Technology | Different Types, Design Flow, Applications*. [Online]
Available at: <https://www.electronicshub.org/introduction-to-asic-technology/>
[Accessed 15 01 2022].

Elprocus, 2022. *Introduction to Application Specific Integrated Circuit (ASIC)*. [Online]
Available at: <https://www.elprocus.com/application-specific-integrated-circuits/>
[Accessed 15 01 2022].

King, G., 2020. *Pedestrian crossings: pelican, puffin, toucan and zebra crossings explained*. [Online]
Available at: <https://www.carwow.co.uk/blog/pelican-puffin-toucan-zebra-pedestrian-crossing>
[Accessed 14 01 2022].

WatElectronics, 2019. *Basic FPGA Architecture and its Applications*. [Online]
Available at: <https://www.watelectronics.com/fpga-architecture-applications/>
[Accessed 14 01 2022].

Annexe

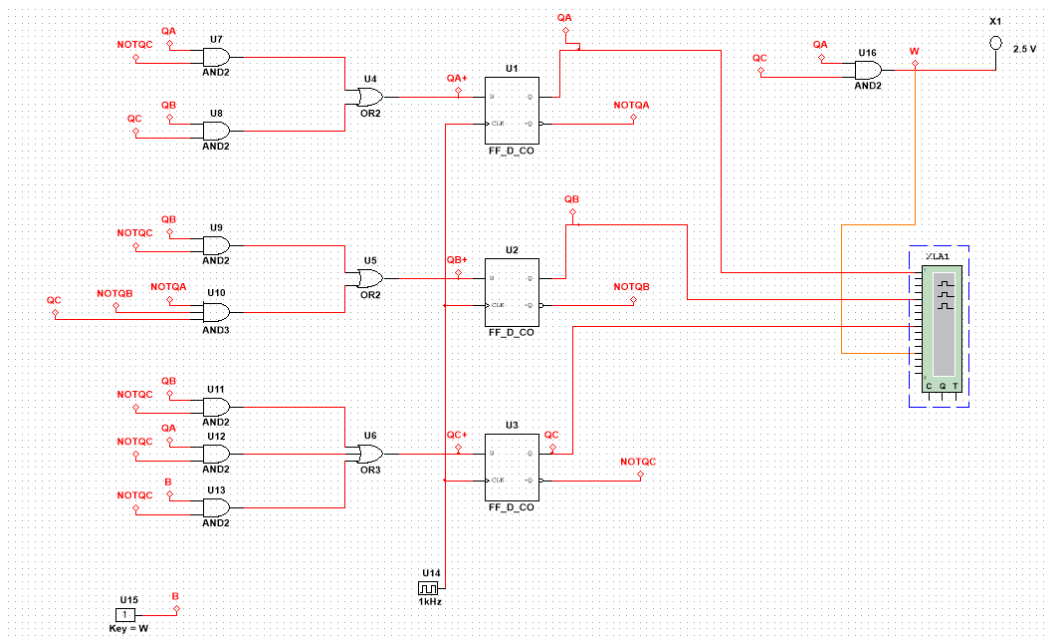


Figure 1- 5s counter

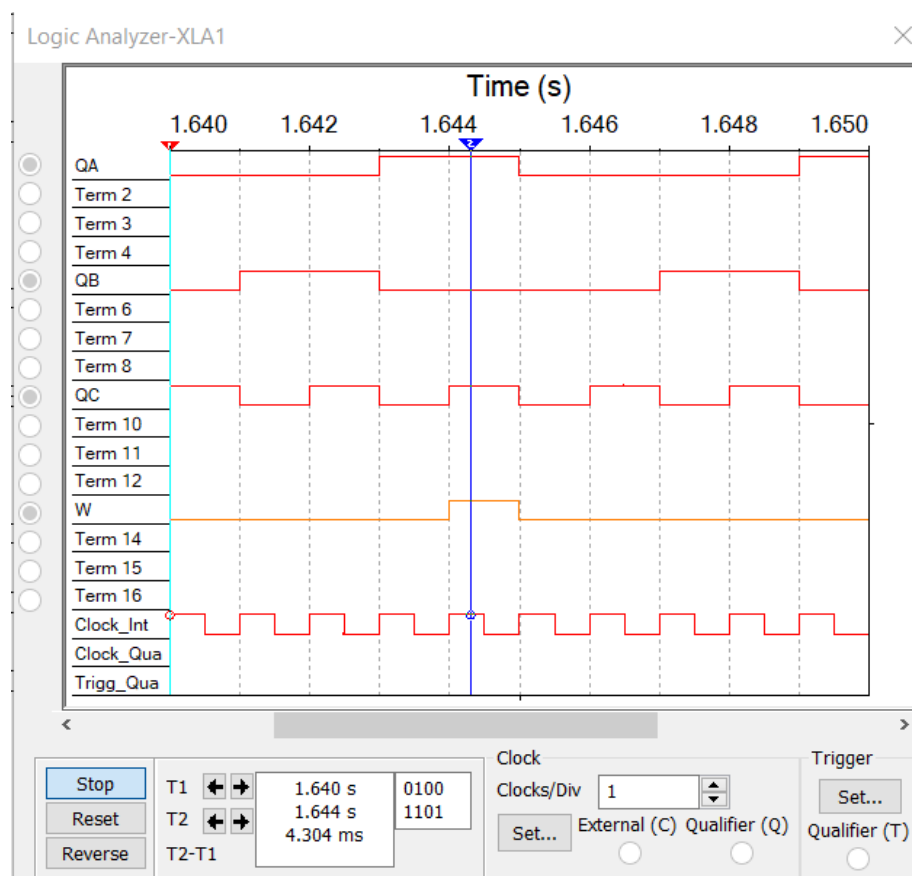


Figure 2- 5s counter

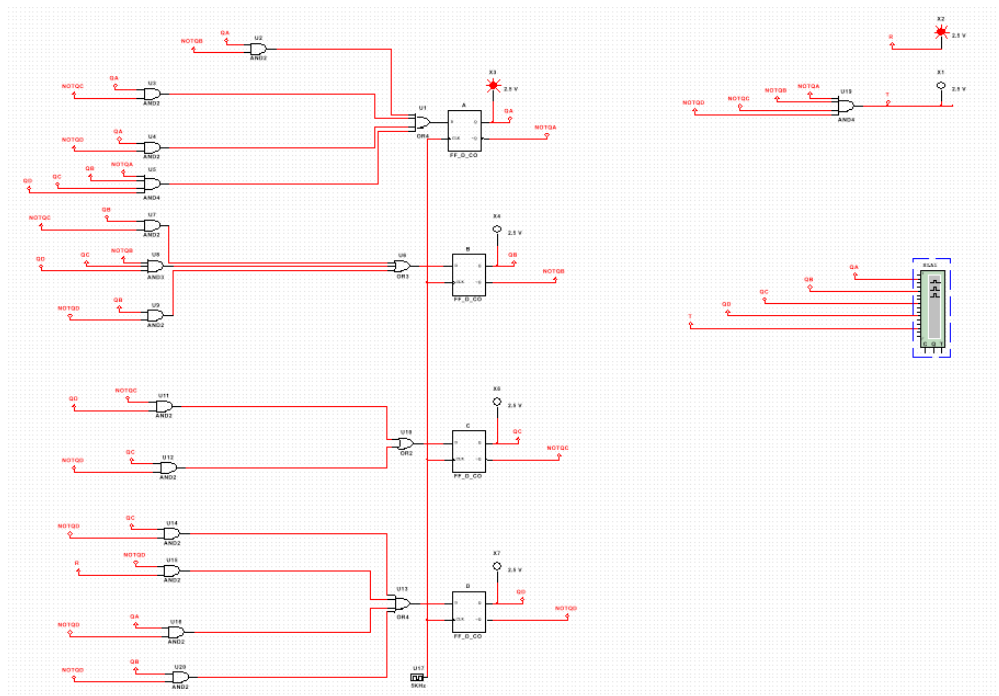


Figure 3- 0-15s counter

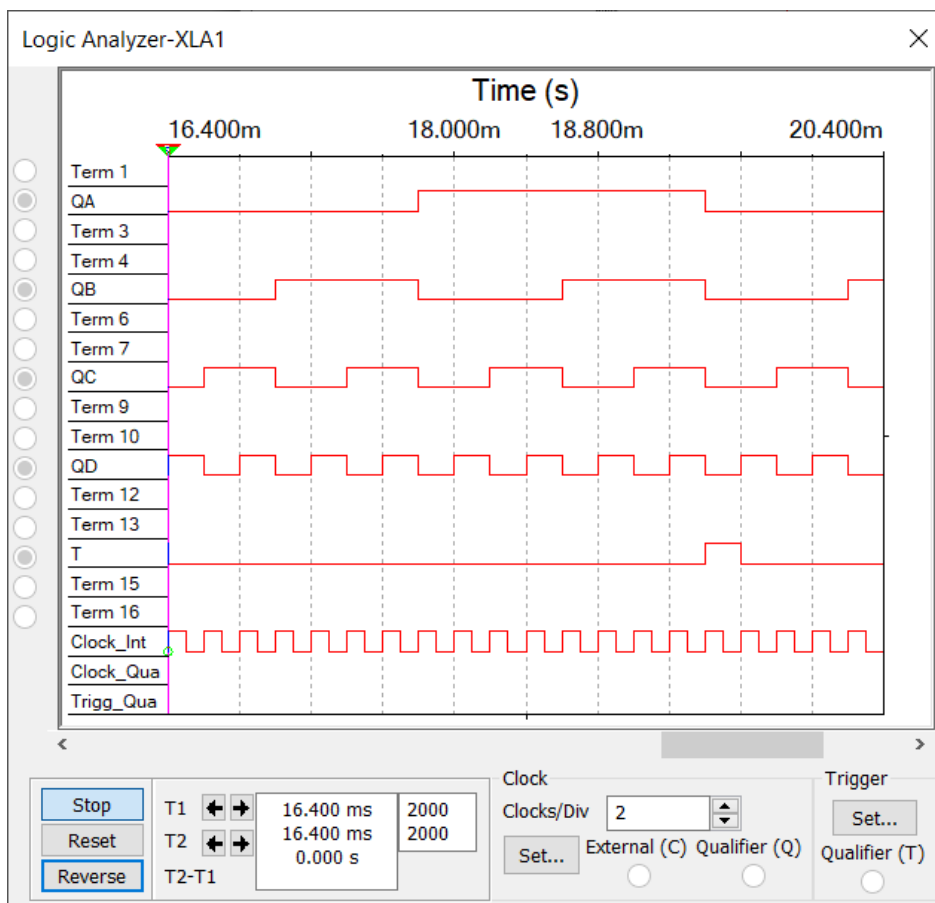


Figure 4- 0-15s counter

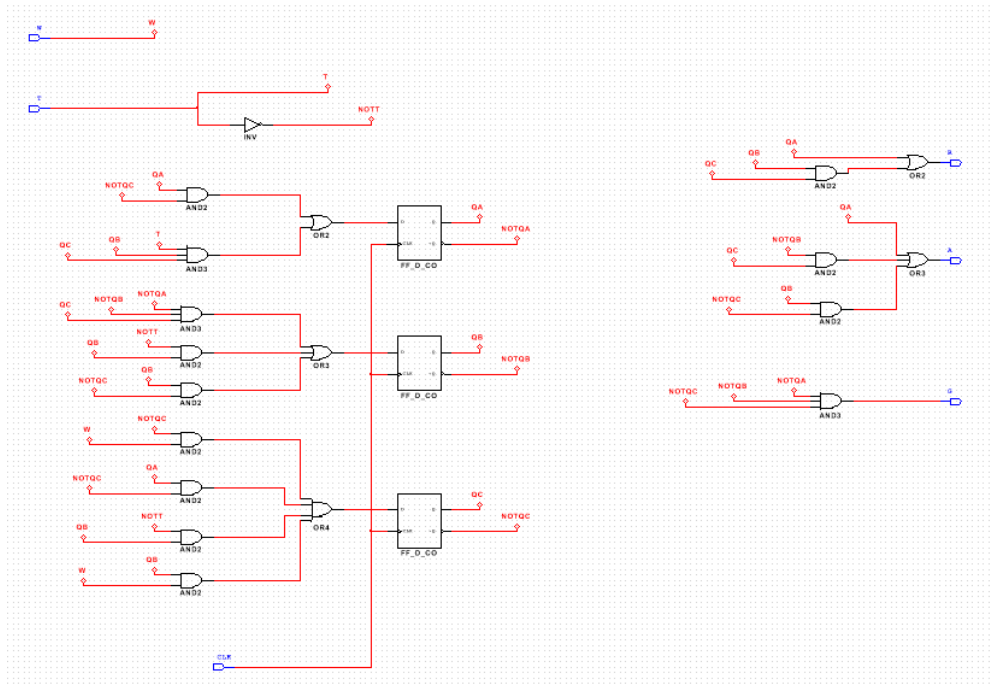


Figure 5- RAG controller

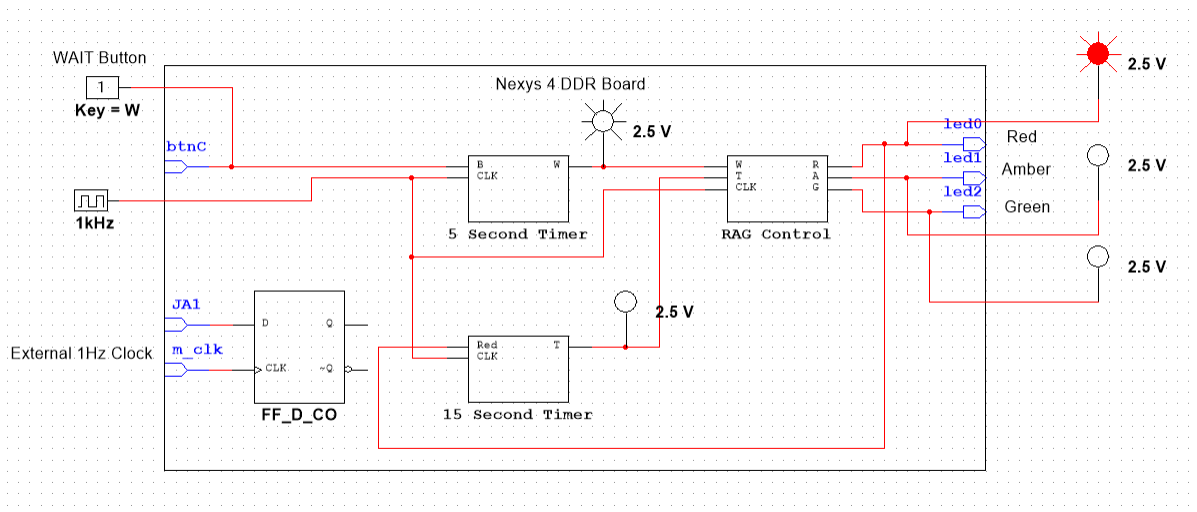


Figure 6- full circuit RAG lights

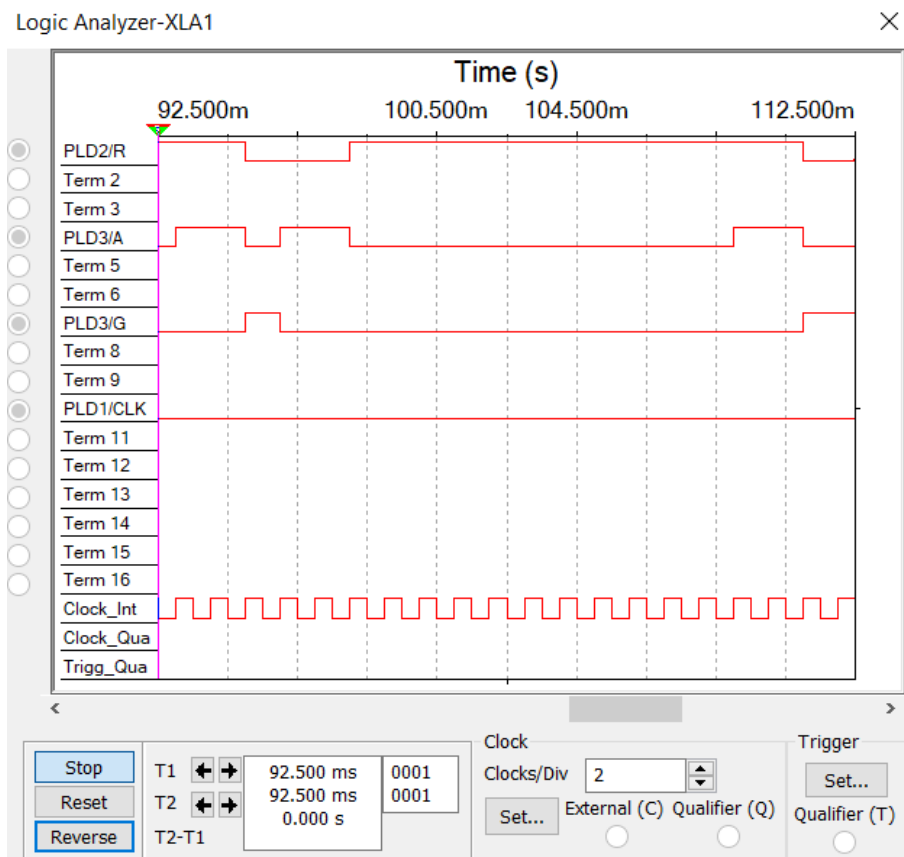


Figure 7- full circuit RAG lights

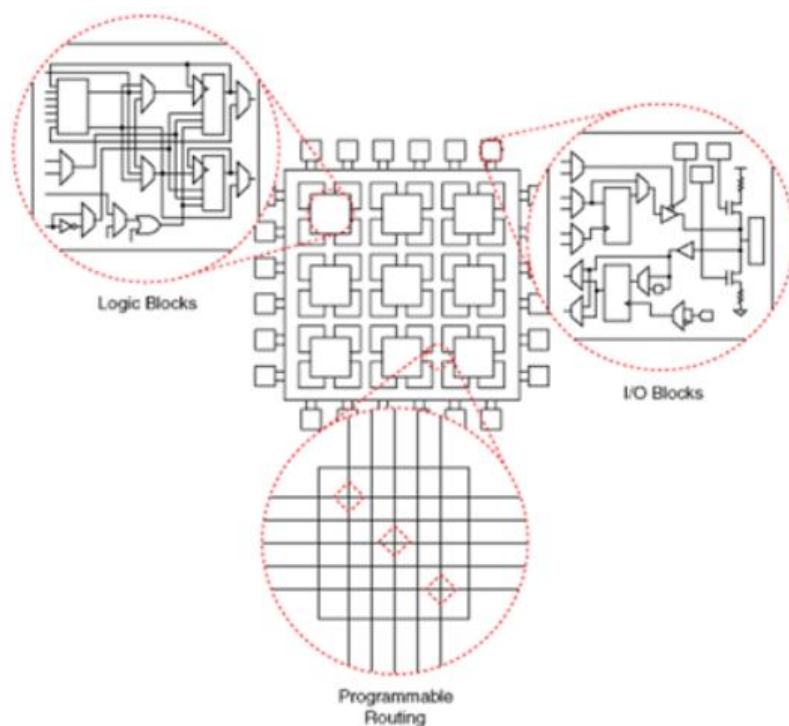


Figure 8- FPGA design

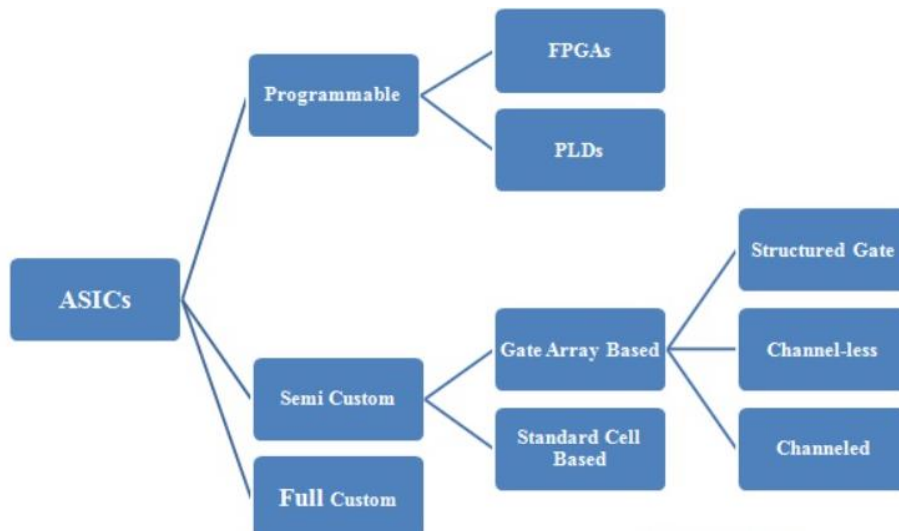


Figure 9- ASIC groups

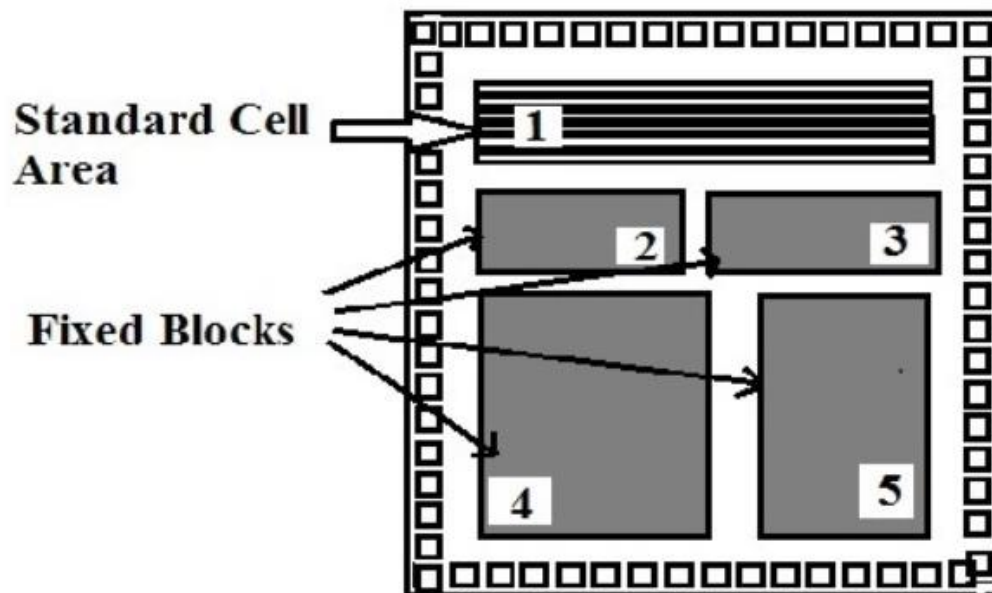


Figure 10- semi-custom ASIC design

FINAL GRADE

GENERAL COMMENTS

Instructor

41 /100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

GRADING FORM: CSG-RUBIC

ETHAN RIGBY

41

RESEARCH-INFORMED LITERATURE



There was little evidence of further reading and research

2

KNOWLEDGE AND UNDERSTANDING



The knowledge and understanding of the subject demonstrated by this assignment was generally adequate. The circuits worked and the explanation of how this was achieved was included, but only to a descriptive level, there was little evidence of any

12

critical understanding. Your report lacked depth and detail at times and didn't really extend your knowledge beyond that contained in the course notes. Further work – Research results need to be in much more detail and written in a more logical and clearer way.

ANALYSIS



Your analysis of the task was mainly descriptive and superficial and lacked a lot of the required detail. Your conclusions were also mainly descriptive and there were several instances of flawed understanding. There was little evidence of critical analysis with most results accepted uncritically as fact and without explaining the underlying principles. You did however demonstrate some basic analytical skills and your approach to the task was logical.

12

PRACTICAL APPLICATION AND DEP.



Your application of practical skills for this task was generally adequate. You used the resources appropriately when designing the circuit and you used appropriate options for testing and evaluating the circuit performance. You managed to design a basic circuit that functioned within the specification and achieved this by only asking for assistance when appropriate thus demonstrating some basic problem solving skills. Your results were adequate and allowed for a reasonable conclusion to be attempted.

10

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