

PSDE 5006- Daniel Evans- Assignment 1

by Daniel Evans

FILE	11942_DANIEL_EVANS_PSDE_5006-_DANIEL_EVANS-_ASSIGNMENT_1_165369_1770769366.DOCX (1.24M)		
TIME SUBMITTED	29-JAN-2019 08:28AM (UTC+0000)	WORD COUNT	2718
SUBMISSION ID	99958567	CHARACTER COUNT	10624



colegsirgâr

PSDE 5006- Electronics II- Assignment 1

Daniel Evans

Submission Deadline: 29.01.2019

Word count: 2743

Contents

1. Executive Summary.....	3
2. Research into advanced digital design techniques (800)	4
3. Design method, choices and calculations (700).....	6
4. Conclusions and recommendations.....	23
5. Reference list and bibliography	24
6. Annexe for diagrams, data sheets, results tables etc	25

1. Executive Summary

This assignment involves the design and construction of a traffic light sequence. This complex sequential logic circuit will be required to control all three lights and will require the lights to change to red once the wait button is pressed, before returning to green. Research has been carried out in order to obtain information and theories in relative fields. A circuit overview allowed us to identify what each sub circuit was required to do along with a visual aid of a state diagram. Following development of the circuits using state table/diagrams, karnaugh maps and deriving the Boolean expression. The circuit could then be built using multisim software. The traffic light sequence functioned as intended and all three-sub circuits were built to specification and tested using logic analysers.

2. Research into advanced digital design techniques (800)

Combinational logic

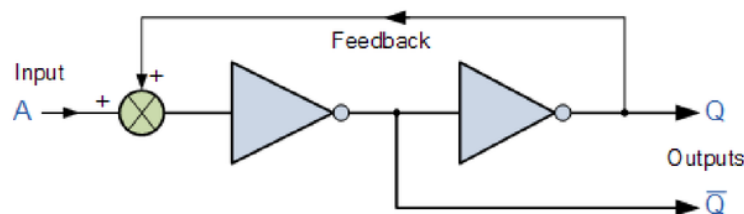
According to Electronics-tutorials.ws (2019), combinational logic circuits output is given by their current input state. Therefore, it can be said that combinational circuits are memoryless. This means that any changes made to the circuits input will have an immediate effect on the output. Combinational logic circuits consist of NAND, NOR and NOT logic gates which are used to make a switching circuit.



electronics-tutorials.ws (2017)

Sequential logic

"Sequential logic circuits are able to take into account their previous input state as well as those actually present, a sort of "before" and "after" effect is involved with sequential circuits." (Electronics-tutorials.ws, 2017). According to Electronics-tutorials.ws (2019), the state of the output can be influenced by three factors. The "present input", the "past input" and the "past output". Therefore, it can be said that sequential logic circuits have some kind of memory as the past input/output has an effect on the present output



electronics-tutorials.ws (2017)

Finite state machine

According to Finite-State Machines: Explanation & Example (2019). Finite state machines are used in order to model some simple circuits. Finite state machines use sequential logic and therefore the circuit has some memory and logic, it can also change its state in order to produce an output. A finite state machine will change state every clock cycle. They also help give the sequential logic some structure allowing easy fault finding due to its easy-to-follow structure. Finite state machines typically use binary sensors, pushbuttons or switches as inputs. The current condition of the state machine is described by the state, these are limited due to the amount of memory the state machine has. The state will be determined by the history of inputs. The outputs are connected to the finite state machine, however they are not the same. Finite-State Machines: Explanation & Example - [Accessed 04 November 2017]].

FPGA's

According to FPGA-Basics (2019), FPGA is an acronym for 'field programmable gate array'. It is a silicon device, which contains reprogrammable digital circuits. They consist of a large number of logic blocks (up to 1000's per FPGA) and the circuitry between them is completely re-configurable. Therefore, the logic blocks in an FPGA are totally configurable dependant on the application you need.

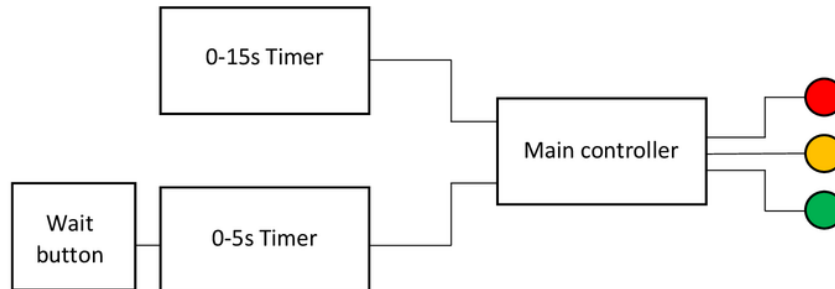
ASIC's

According to ASIC- Basic (2019), ASIC is an acronym for 'application specific integrated circuit'. It's a chip designed for a specified application. It contains logic blocks, which are connected by circuitry. ASIC's are very hard to copy from and can be hidden away due to the size and weight of them. Making them an idea solution for consumer products.

3. Design method, choices and calculations (700)

Designing the circuit

This design is going to consist of three sub circuits which include a main controller, 0-5 second timer and a 0-15 second timer which are linked together in order to illuminate the red, amber and green lights in the desired sequence. These sub circuits will be connected as shown in the diagram below.



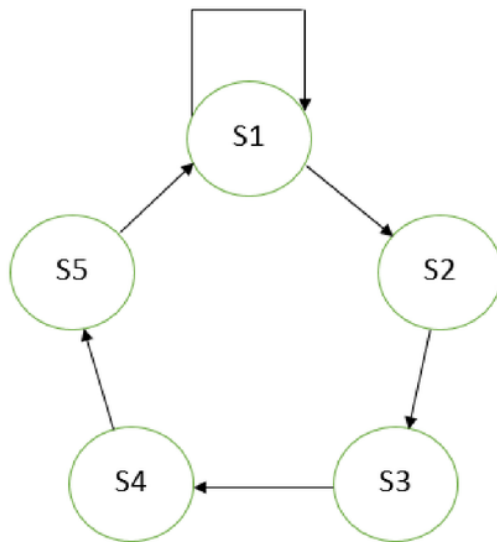
5 second timer design

The purpose of the 0-5s timer is to allow 5s from when the wait button is pressed before the lights begin to change. The first stage of the design process for this timer was to allocate states, a state diagram can then be designed in order to physically show the order in which the states will operate. From this information, a state table can then be created in order to design the karnaugh maps and derive the Boolean expression

States

State 0 = Count 0
State 1 = Count 1
State 2 = Count 2
State 3 = Count 3
State 4 = Count 4
State 5 = Count 5

State diagram



Inputs

B=0 (Wait button not pressed)

B=1 (Wait button pressed)

Outputs

W=0 (Timer counting or button not pressed)

W=1 (Timer has counted to 5s)

State table

QA= 4 in binary

QB= 2 in binary

QC= 1 in binary

Decimal	Next		Current			Next						Output W
	B=0	B=1	Qa	Qb	Qc	B=0			B=1			
						Qa ⁺	Qb ⁺	Qc ⁺	Qa ⁺	Qb ⁺	Qc ⁺	
S0	S0	S1	0	0	0	0	0	0	0	0	1	0
S1	S2	S2	0	0	1	0	1	0	0	1	0	0
S2	S3	S3	0	1	0	0	1	1	0	1	1	0
S3	S4	S4	0	1	1	1	0	0	1	0	0	0
S4	S5	S5	1	0	0	1	0	1	1	0	1	0
S5	S0	S0	1	0	1	0	0	0	0	0	0	1
S6	X	X	1	1	0	X	X	X	X	X	X	X
S7	X	X	1	1	1	X	X	X	X	X	X	X

Karnaugh Maps & Boolean expressions

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

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

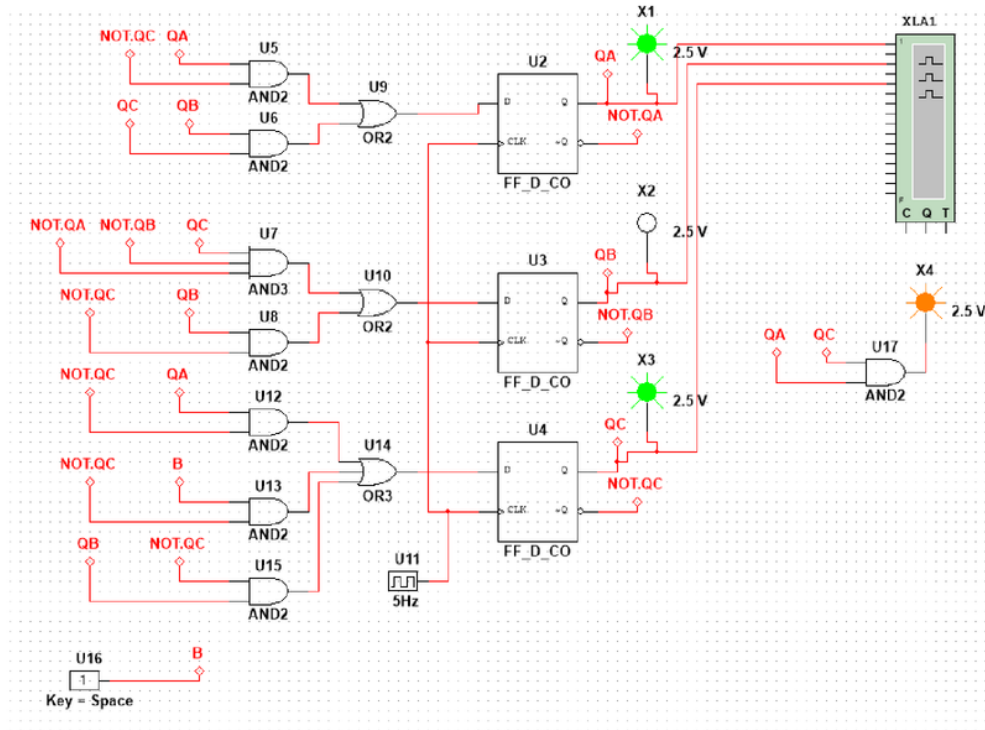
W

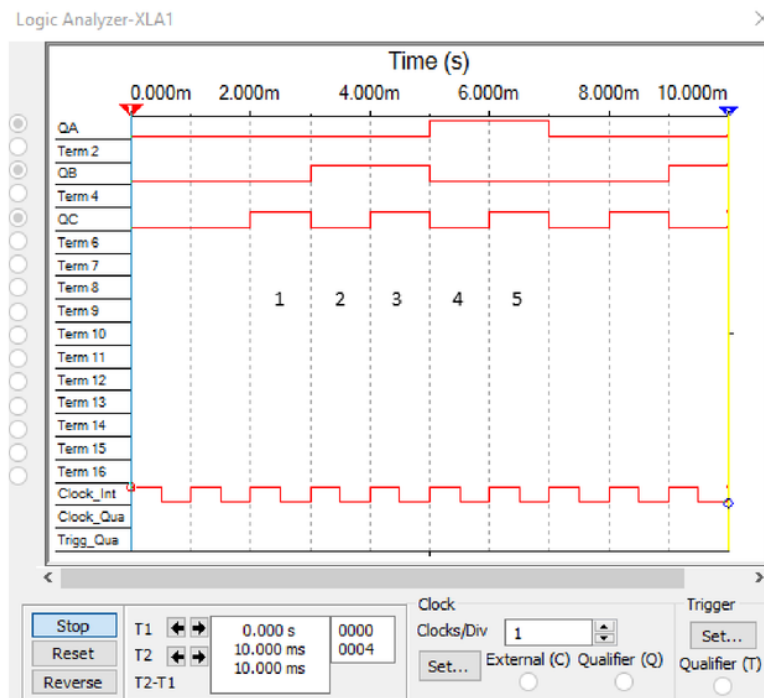
QBQC WQA	00	01	11	10
0	0	0	0	0
1	0	1	X	X

$$W = QAQC$$

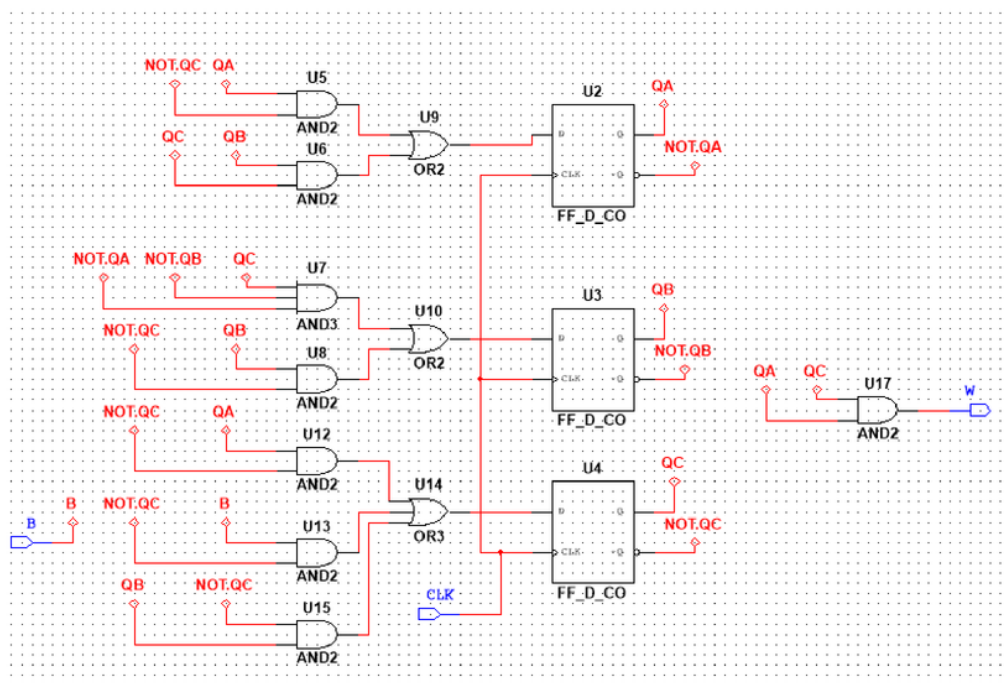
Multisim design

From the Boolean expressions shown above, circuits can now be built and tested using multisim. In order to simulate a person pressing the wait button an 'interactive digital constant' was used. A logic analyser is then used in order to test the circuit ensuring the binary count is correct. An orange LED was used to show that output 'W' was set when the timer reached 5s. The circuit and test are shown below.





A sub circuit can be made for the final design with the required inputs and outputs included.



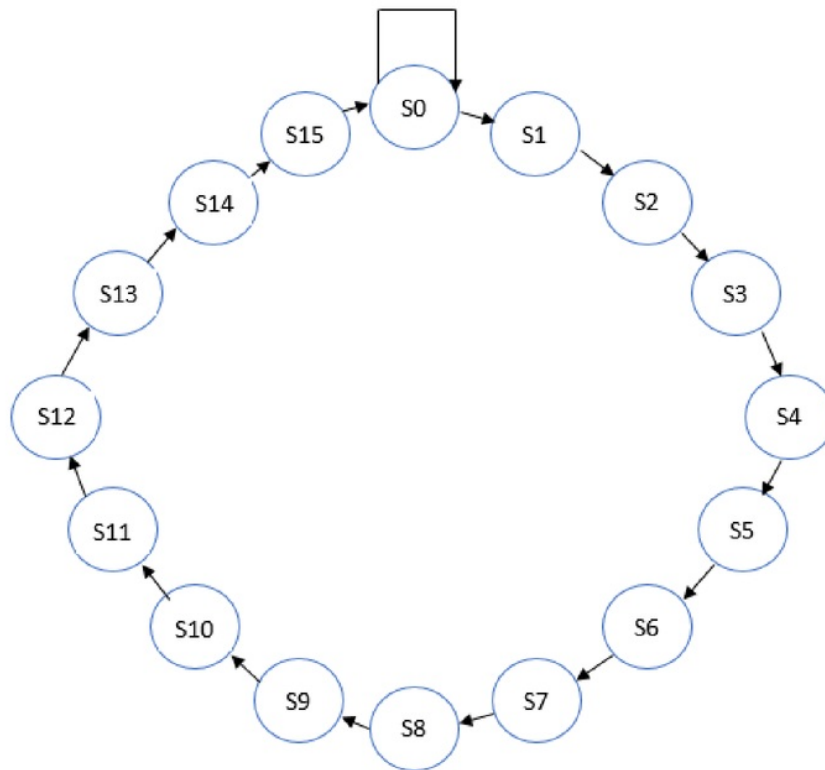
15s second Timer design

The purpose of the 15s second timer is to allow the person to cross the road when the light has changed to red. After 15s of the red light being illuminated, this circuit will produce an output to the main controller to continue the sequence.

States

State 0 = Count 0
State 1 = Count 1
State 2 = Count 2
State 3 = Count 3
State 4 = Count 4
State 5 = Count 5
State 6 = Count 6
State 7 = Count 7
State 8 = Count 8
State 9 = Count 9
State 10 = Count 10
State 11 = Count 11
State 12 = Count 12
State 13 = Count 13
State 14 = Count 14
State 15 = Count 15

State diagram



Inputs

R0 = Do nothing (When amber or green light is illuminated)

R1 = Start counting (when red light is illuminated)

Outputs

T0= 0 (red light not illuminated or timer is counting and has not reached 15s)

T1= 1 (when timer=15s)

State table

QA= 8 in binary

QB= 4 in binary

QC= 2 in binary

QD= 1 in binary

Decimal	Next		Current				Next								Output T
							R=0				R=1				
	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

QA⁺

R=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

$$QA^+ = A\bar{B} + A\bar{C} + A\bar{D} + \bar{A}BCD$$

R=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

QB⁺

R=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

$$QB^+ = B\bar{D} + B\bar{C} + \bar{B}CD$$

R=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

QC⁺

R=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

$$QC^+ = \bar{C}D + C\bar{D}$$

R=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

QD⁺

R=0

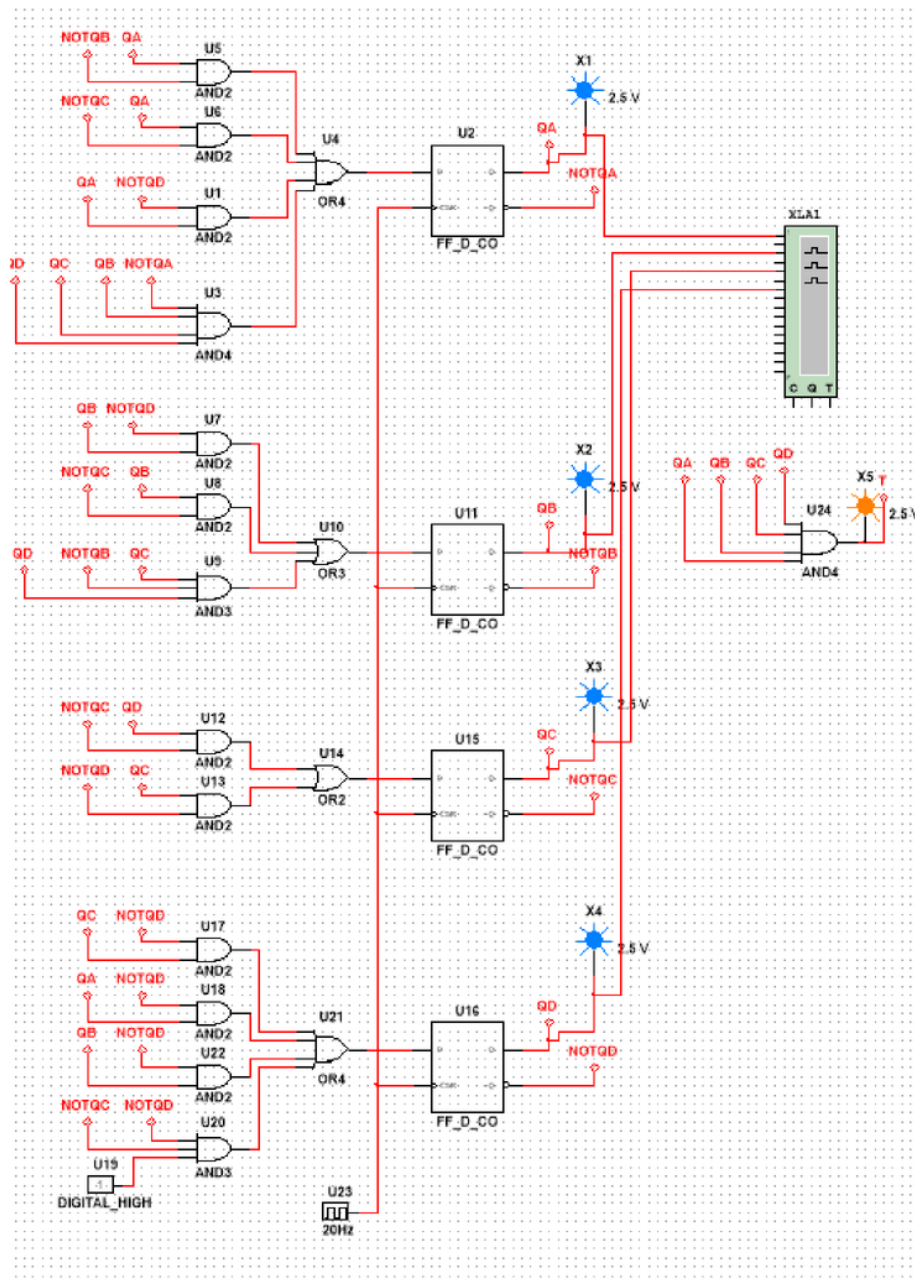
QBQC BQA	00	01	11	10
00	0	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

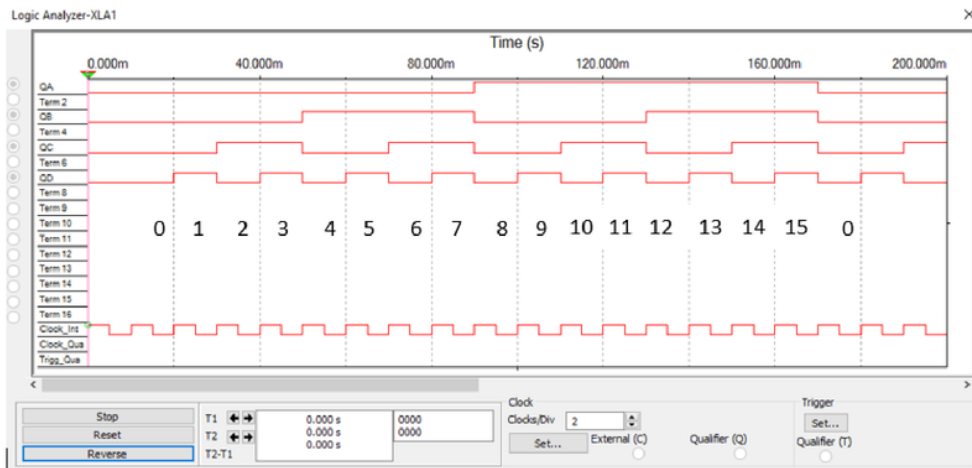
$$QD^+ = A\bar{D} + B\bar{D} + C\bar{D} + R\bar{D}$$

R=1

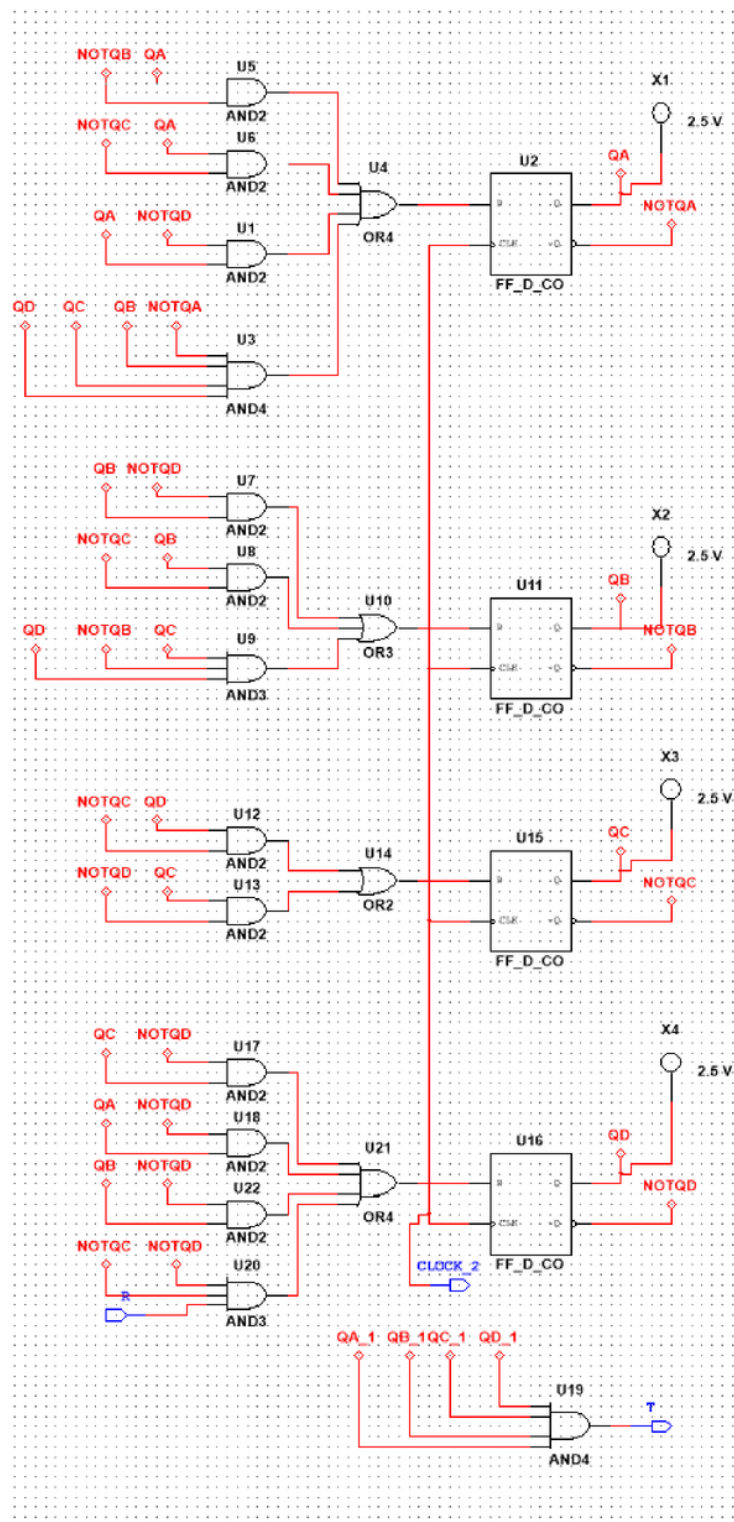
QBQC BQA	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

From the Boolean expressions shown above, The circuit can now be built and tested using multisim. In order to simulate the red light being on an 'interactive digital constant' was used. A logic analyser is then used in order to test the circuit ensuring the binary count is correct. An orange LED was used to show that output 'T' was set when the timer reached 15s. The circuit and test are shown below.





A sub circuit can be made for the final design with the required inputs and outputs included.



Main controller

The purpose of the main controller is to illuminate the red, amber and green lights when required. The main controller will be connected to both the 5s and 15s timer along with the red, amber and green light.

State Allocation

Step 0 (S0) – Green

Step 1 (S1) – Amber

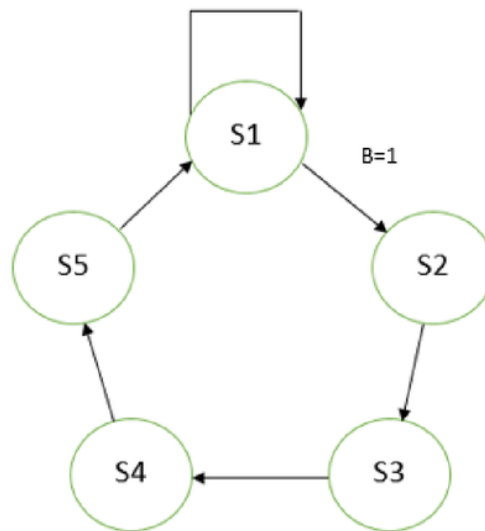
Step 2 (S2) – Amber

Step 3 (S3) – Red

Step 4 (S4) – Red / Amber

Step 5 (S5) – Red / Amber

State diagram



State table

Current	Next				Current		Next												Outputs																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
	Next						W=0						W=1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
	W=0		W=1		QA	QB	QC	T=0			T=0			T=0			QA+	QB+	QC+																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
								QA+	QA+	QA+	QA+	QA+	QA+	QA+	QA+	QA+																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
	T=0	T=1	T=0	T=1				QA	QB	QC																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															

Karnaugh maps & boolean expressions

QA⁺

B=0

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

$$QA^+ = A\bar{C} + TBC$$

B=1

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

QB⁺

B=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	1	1

$$QB^+ = \bar{A}\bar{B}C + \bar{T}B + B\bar{C}$$

B=1

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

QC⁺

B=0

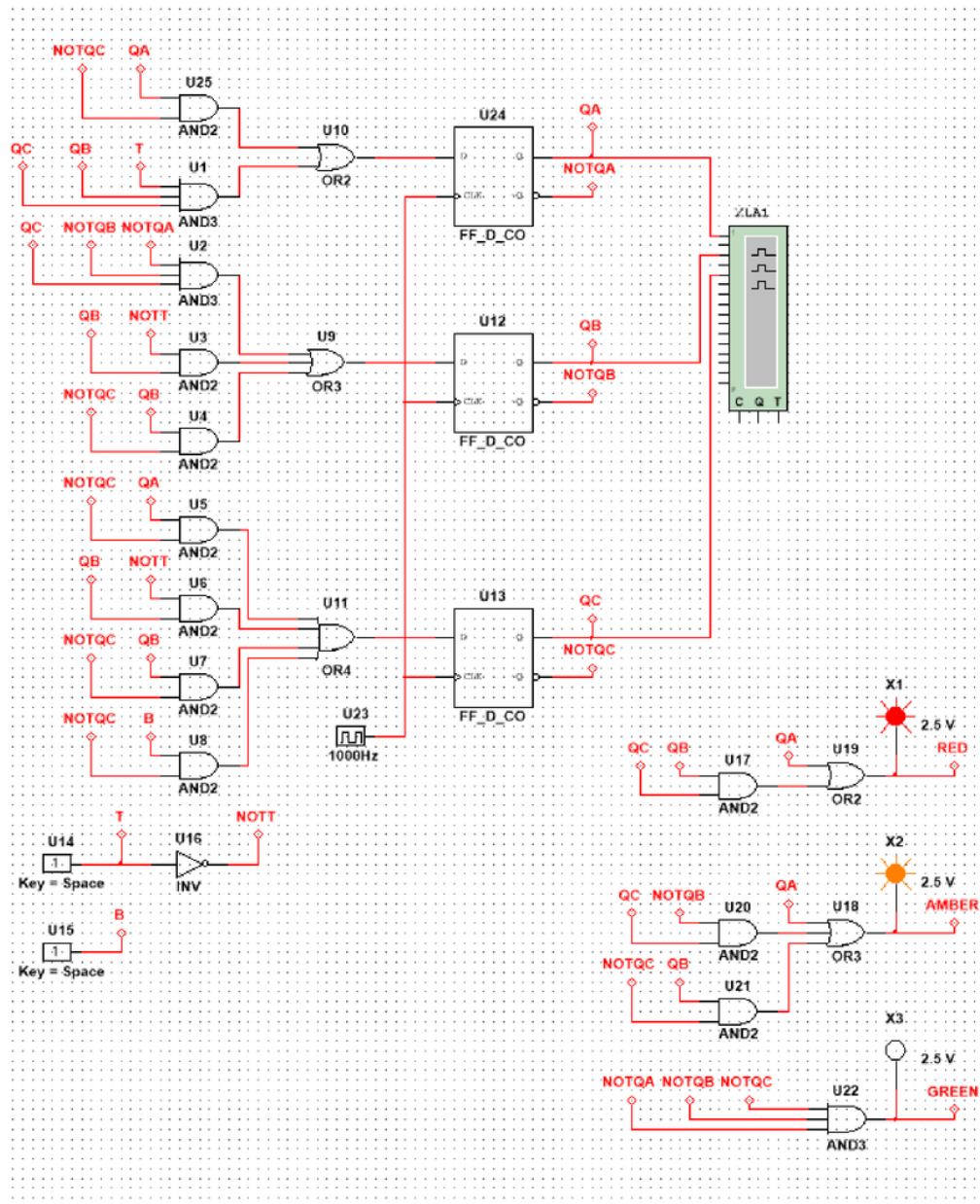
QBQC BQA	00	01	11	10
00	0	0	1	1
01	1	0	X	X
11	1	0	X	X
10	0	0	0	1

$$QC^+ = A\bar{C} + \bar{T}B + B\bar{C} + B\bar{C}$$

B=1

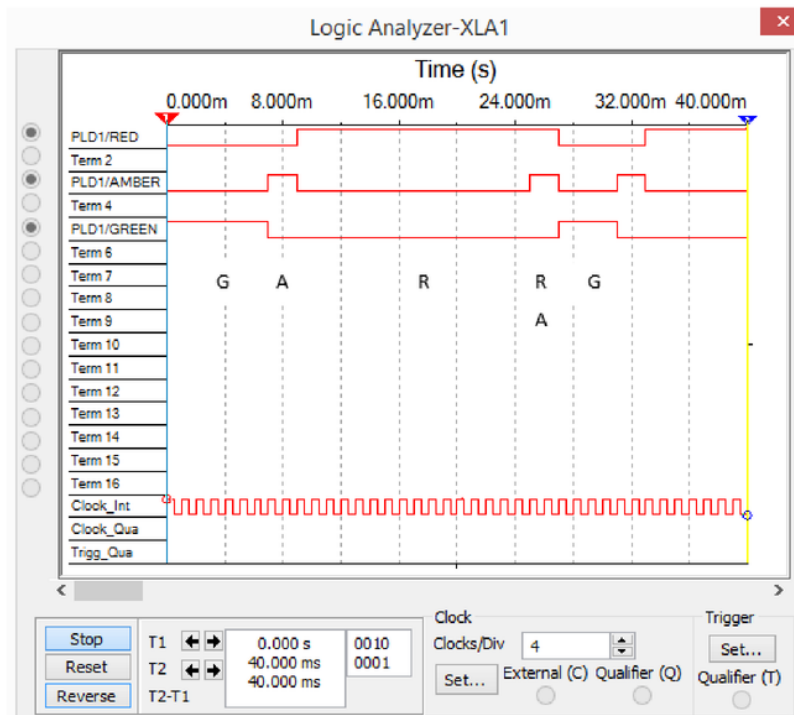
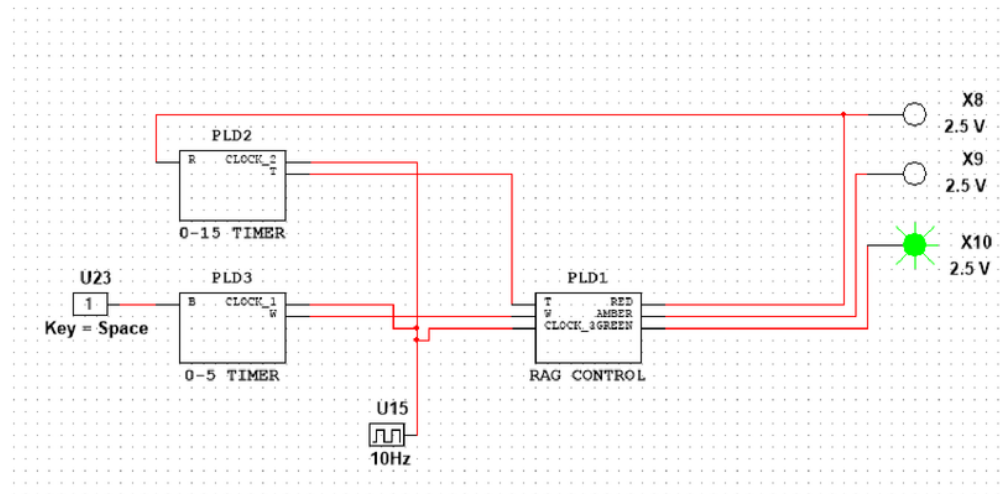
QBQC BQA	00	01	11	10
00	0	0	1	1
01	1	0	X	X
11	1	0	X	X
10	0	0	0	1

Multisim



RAG

This circuit contains three sub circuits. The 5s timer, 15s timer and the main controller. By connecting these three subcircuits, it is possible to drive the lights in the correct sequence.



4. Conclusions and recommendations

To conclude, the purpose of this task was to “apply the theory involved in designing complex electronic circuits and demonstrate an understanding of the need to promote sustainable development” and to “critically analyse the design and performance of complex electronic circuits and adapt designs to meet new applications”, both of these objectives have been met during this assignment.

During background research was carried out in the following categories

- differences between combinational and sequential logic
- The use of a Finite State Machine as a design tool
- The relative merits of FPGA's and ASIC's

Using this information, it has allowed a circuit to be designed for the task specification. These circuits have been tested using logic analysers. It is possible to download the software designed on multisim to a FPGA such as the nexus 4 board. This assignment has confirmed that the research is correct and that the theories gathered are correct.

One potential recommendation that could make this circuit work better is to have a pressure pad by the wait button. Therefore, if the wait button is pressed accidentally then it will detect that no one is waiting to cross and ignore the wait button command.

5. Reference list and bibliography

Citation list

Electronics-tutorials.ws. (2017, 10 22). COMBINATIONAL LOGIC CIRCUITS USING LOGIC GATES. *Basic Electronics Tutorials. (2017). Combinational Logic Circuits using Logic Gates. [online] Available at: http://www.electronics-tutorials.ws/combination/comb_1.html* [Accessed 20 January 2019]

YouTube. 2017. *Finite-State Machines: Explanation & Example - YouTube.* [ONLINE] Available at: <https://www.youtube.com/watch?v=6oe1Tmg9rjM>. [Accessed 20 January 2019]

YouTube. 2017. FPGA Basics - YouTube. [ONLINE] Available at: <https://www.youtube.com/watch?v=L2wsockKwPQ>. [Accessed 22 January 2019]

YouTube. 2017. What is ASIC? - YouTube. [ONLINE] Available at: <https://www.youtube.com/watch?v=cwPytnBSCKk>. [Accessed 22 January 2019]

6. Annexe for diagrams, data sheets, results tables etc



1300 Henley Court
Pullman, WA 99163
509.334.6306
www.digilentinc.com

Nexys4 DDR™ FPGA Board Reference Manual

Nexys4 DDR rev. C; Revised April 11, 2016

1 Overview

The Nexys4 DDR board is a complete, ready-to-use digital circuit development platform based on the latest Artix-7™ Field Programmable Gate Array (FPGA) from Xilinx®. With its large, high-capacity FPGA (Xilinx part number XC7A100T-1CSG324C), generous external memories, and collection of USB, Ethernet, and other ports, the Nexys4 DDR can host designs ranging from introductory combinational circuits to powerful embedded processors. Several built-in peripherals, including an accelerometer, temperature sensor, MEMs digital microphone, a speaker amplifier, and several I/O devices allow the Nexys4 DDR to be used for a wide range of designs without needing any other components.



The Nexys4 DDR

The Artix-7 FPGA is optimized for high performance logic, and offers more capacity, higher performance, and more resources than earlier designs. Artix-7 100T features include:

- 15,850 logic slices, each with four 6-input LUTs and 8 flip-flops
- 4,860 Kbits of fast block RAM
- Six clock management tiles, each with phase-locked loop (PLL)
- 240 DSP slices
- Internal clock speeds exceeding 450 MHz
- On-chip analog-to-digital converter (XADC)



The Nexys4 DDR also offers an improved collection of ports and peripherals, including:

- | | | |
|-------------------------|---|--|
| • 16 user switches | • 16 user LEDs | • Two 4-digit 7-segment displays |
| • USB-UART Bridge | • Two tri-color LEDs | • Micro SD card connector |
| • 12-bit VGA output | • PWM audio output | • PDM microphone |
| • 3-axis accelerometer | • Temperature sensor | • 10/100 Ethernet PHY |
| • 128MiB DDR2 | • Serial Flash | • Four Pmod ports |
| • Pmod for XADC signals | • Digilent USB-JTAG port for FPGA programming and communication | • USB HID Host for mice, keyboards and memory sticks |

The Nexys4 DDR is compatible with Xilinx's new high-performance Vivado® Design Suite as well as the ISE® toolset, which includes ChipScope™ and EDK. Xilinx offers free WebPACK™ versions of these toolsets, so designs can be implemented at no additional cost. The Nexys4 DDR is not supported by the Digilent Adept Utility.

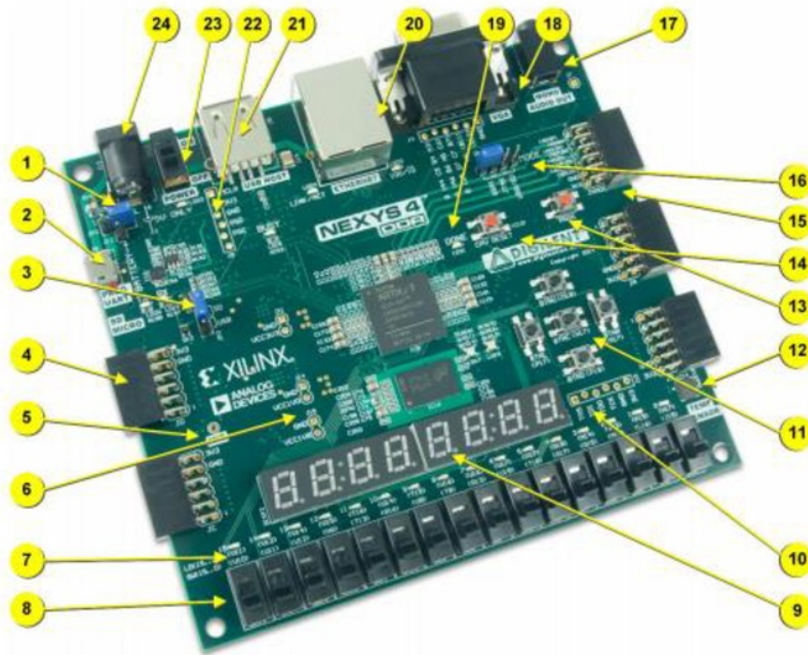


Figure 1. Nexys4 DDR board features.

Callout	Component Description	Callout	Component Description
1	Power select jumper and battery header	13	FPGA configuration reset button
2	Shared UART / JTAG USB port	14	CPU reset button (for soft cores)
3	External configuration jumper (SD / USB)	15	Analog signal Pmod port (XADC)
4	Pmod port(s)	16	Programming mode jumper
5	Microphone	17	Audio connector
6	Power supply test point(s)	18	VGA connector
7	LEDs (16)	19	FPGA programming done LED
8	Slide switches	20	Ethernet connector
9	Eight digit 7-seg display	21	USB host connector
10	JTAG port for (optional) external cable	22	PIC24 programming port (factory use)
11	Five pushbuttons	23	Power switch
12	Temperature sensor	24	Power jack

A growing collection of board support IP, reference designs, and add-on boards are available on the Digilent website. See the Nexys4 DDR page at www.digilentinc.com for more information.

PSDE 5006- Daniel Evans- Assignment 1

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

50/100

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

GRADING FORM: ASSIGNMENT RUBRIC 2018

DANIEL EVANS

50

ENGAGEMENT WITH LITERATURE



There was some evidence of further reading and research and your interpretation and understanding of the additional research was OK. The sources referenced were insufficient at this level given the amount of potential research areas for this assignment and relied on a small selection of mainly inappropriate and non-authoritative sources. You also need to include more in-text citations using the Harvard reference system

3

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 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.

COGNITIVE AND INTELLECTUAL



Your analysis of the task was generally satisfactory, both in class and as shown in this assignment, however it did tend more towards descriptive rather than analytical. You have demonstrated some analytical skills and your approach to the task was logical. You wrote a short conclusion that demonstrated some skills in analysing quantitative and qualitative data, however this was generally basic in nature. Further work – in your conclusion, you should relate the results to the original specification and to accepted theory and provide a detailed critical analysis of this.

15

PRACTICAL APPLICATION



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

10

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.

TRANSFERABLE SKILLS FOR LIFE



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.

6