

LEVEL 4 ASSESSMENT SPECIFICATION

Student name:	Paul Miller	Student ID number:	
Programme:	HNC Electrical/Electronic Engineering		
Module:	Electronics 1		
Module code:	PSDE4004	Contribution to Overall Module Assessment (%):	25%
Lecturer:	Jason Pointer	Internal Verifier:	Steve Roberts
Assignment Title:	Digital	Word count (or equivalent):	1500
Submission deadline:	11/04/19	Return date of provisional marks & written feedback:	
Submission method:	All written assessments, where practical and possible, must be submitted via Turnitin unless otherwise instructed by the Lecturer. (Please DO NOT put this assessment specification into Turnitin or it will match many similarities with other students' submissions.) Alternative submission method (if applicable): Late submission of the assessment will result in a late penalty mark. Penalties for late submission: Up to one week late, maximum mark of 40%. Over one week late, 0%. Only the Extenuating Circumstances Panel may approve a change to submission dates.		
Academic honesty / referencing:	Academic honesty is required. In the main body of your submission you must give credit to authors on whose research and ideas your work is based. Append to your submission a reference list that indicates the books, articles, etc. that you have used, cited or quoted in order to complete this assessment.		

Module Learning Outcomes (from module syllabus)

1. Demonstrate an understanding of semi-conductor components and characteristics and their application in analogue and **digital** electronic circuits.
2. To design, simulate, build, test and evaluate analogue and **digital** electronic circuits using information gained from datasheets.

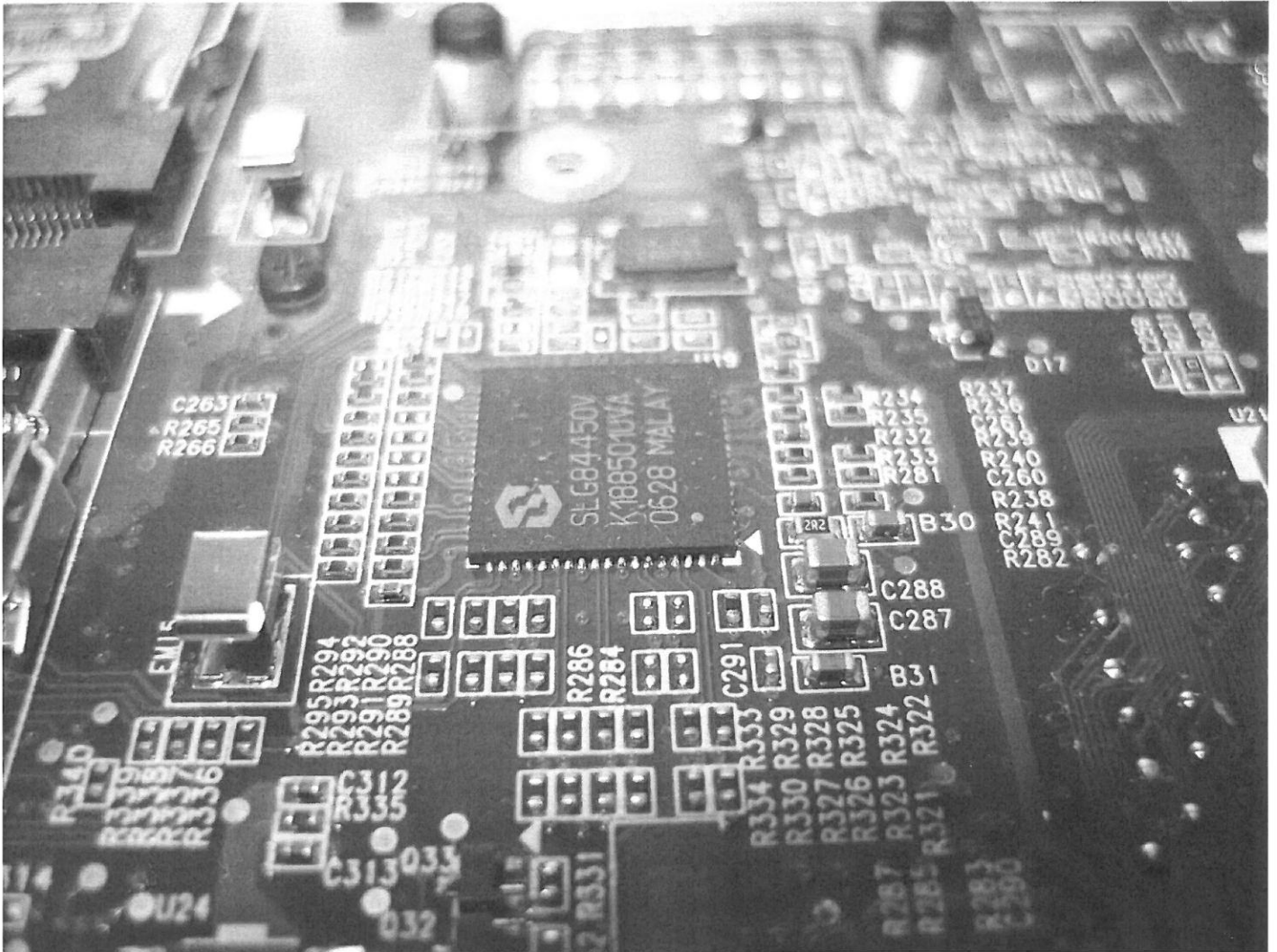
STUDENT FEEDBACK FORM

This section details the *extent* to which the assessment criteria are demonstrated by you, which in turn determines your mark. The marks available for each category of skill are shown. Lecturers will use the space provided to comment on the achievement of the task(s), including those areas in which you have performed well and areas that would benefit from development/improvement.

Generic Assessment Criteria	Marks available	Marks awarded
1. Engagement with Literature Skills	10	2.5
Little evidence of reading including a lack of referencing to any source material. Information must have been gathered for debouncing.		
2. Knowledge and Understanding Skills	30	10.6
Achieved 5% for the combinational logic. Achieved 66% for the sequential logic. Completely missing the work to cover the underlying concepts and principles of combinational logic but had been produced in the lessons. Have only showed the simulation work for combinational logic. Showed competent knowledge of the basic underlying concepts and principles sequential logic, very good design work for the counter.		
3. Cognitive and Intellectual Skills	10	3
Largely descriptive work, with superficial evaluation of the underlying concepts and principles, tending towards description.		
4. Practical Application Skills	40	18
Achieved 30% for the combinational logic. Achieved 60% for the sequential logic. A good application of practical and simulation work to obtain most of the required results but there were some gaps.		
5. Transferable Skills for Life and Professional Practice	10	5
Mostly coherent, organised and accurate work, in a suitable structure and is for the most part clearly expressed.		

Assessment Mark (Assessment marks are subject to ratification at the Exam Board. These comments and marks are to give feedback on module work and are for guidance only until they are confirmed.)	Late Submission Penalties (tick if appropriate)		39%
	Up to 1 week late (40% Max)		
	Over 1 week late (0%)		

Assignment 2



Paul Miller (141589)

Task 1 - Combinational Logic

This should be written in a past tense.

Should have included
your own truth table
and simplified



Logic Converter-XLC1

Logic Diagram: XLC1

Logic Function: AB (AND gate output)

Logic Function: $U1$ (NAND gate output, labeled NAND2)

Truth Table:

	A	B	C	D	E	F	G	H	1
000	0	0	0	0	0	0	0	0	0
001	0	0	0	0	0	0	0	1	1

Conversions:

- AND gate symbol $\leftrightarrow$ 1011
- $1011 \leftrightarrow AB$
- $1011 \rightarrow AB$
- $AB \rightarrow 1011$
- $AB \rightarrow$ AND gate symbol
- $AB \rightarrow NAND$

Output: A'

Figure 2

- c) **Figure 3** illustrates a logic gate which was made using **4 NAND** gates, it also shows the truth table for this logic gate and shows the simplified Boolean expression.

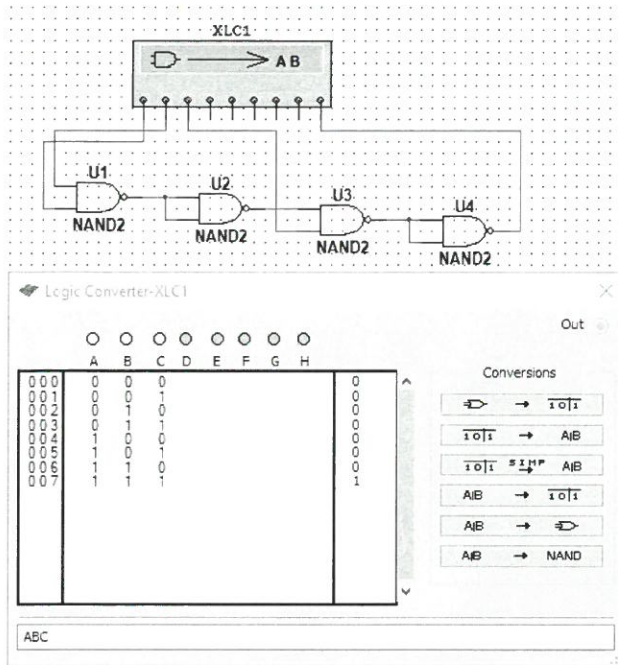


Figure 3

You have not measured the other points in the circuit.

- d) **Figure 4** illustrates a logic gate which was produced by using **3 NAND** gates, it also shows the truth table for this logic gate, and it also shows the simplified Boolean expression. This logic gate works the same way as an **inclusive OR** gate.

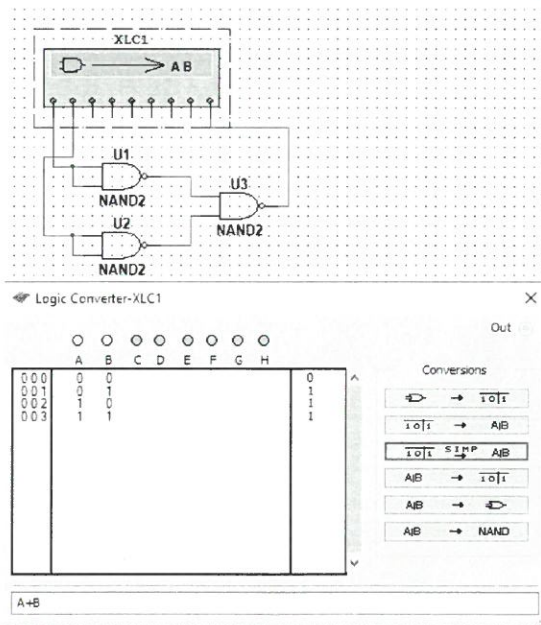


Figure 4

No it is just an OR gate.

- e) **Figure 5** illustrates a logic gate which was produced using **NAND** gates, it shows the truth table and shows the simplified Boolean algebra. This logic gate works as **XOR** gate.

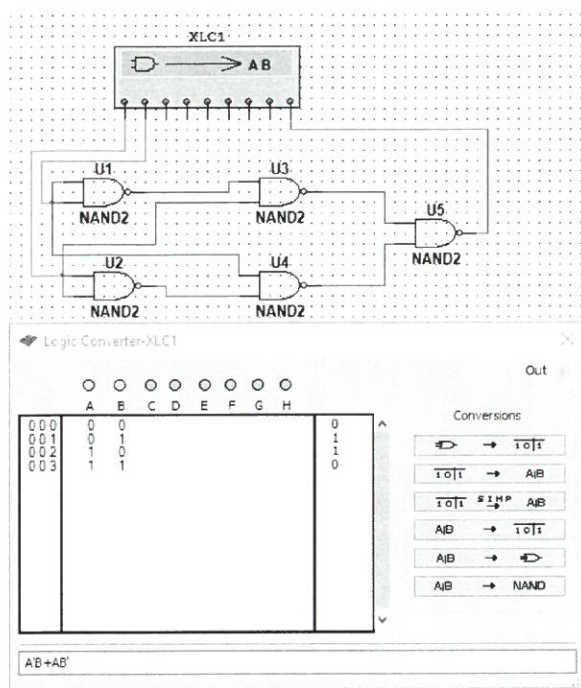


Figure 5

- f) i) The expression given in part F(i) is $[F = A(A + B')]$, The simplified version of this is Boolean algebra is

Where is your answer?

- ii) **Figure 6** illustrates the simplified version of the Boolean algebra expression given, which was $[F = (A+B)(C+A)]$, it shows the simplified expression to be $[BC + A]$, it also shows the logic gate which would be produced from the simplified expression using **NAND** gates.

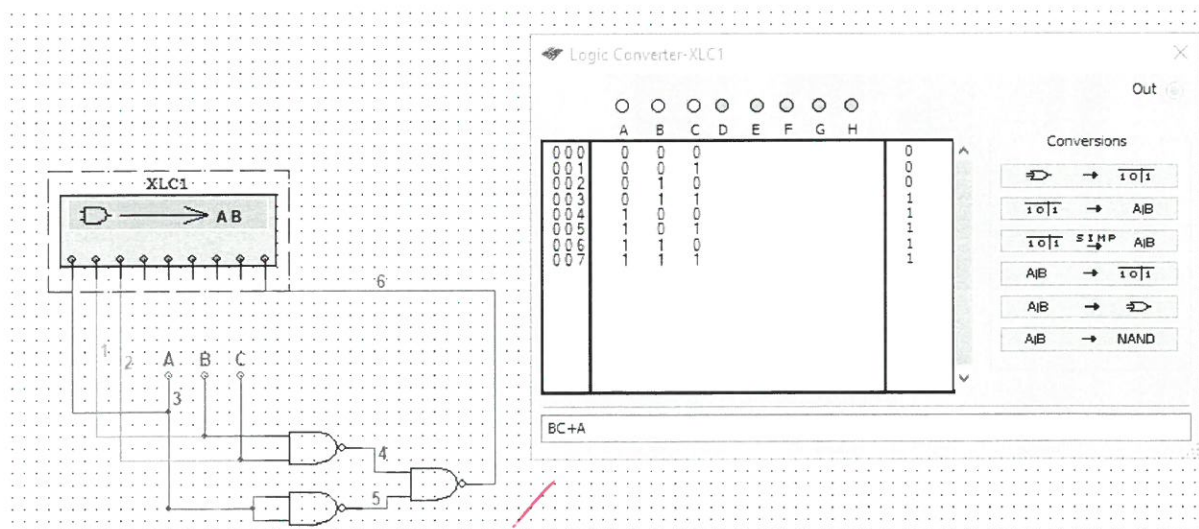


Figure 6

You were supposed to use Boolean algebra to simplify the equation and use DeMorgan to convert the equation to NAND gates.

iii) **Figure 7** illustrates the simplified version of the Boolean algebra expression given, which was $[F = (A + B)C]$, it shows the simplified expression as $[AC+BC]$, it also shows the logic gate which would be produced from the simplified expression using **NAND** gates.

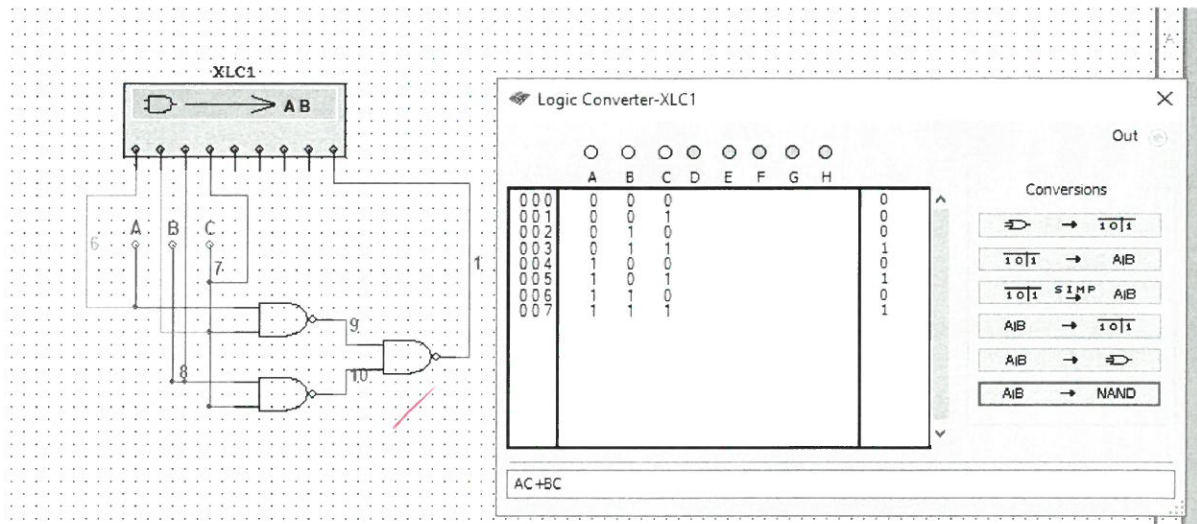


Figure 7

iv) **Figure 8** illustrates the simplified version of the Boolean algebra expression given, which was $[F = (A' + B' + C')(A + B' + C')(A + C)(B + C')]$, it shows the simplified expression as $[AC']$, it also shows the logic gate which would be produced from the simplified expression using **NAND** gates

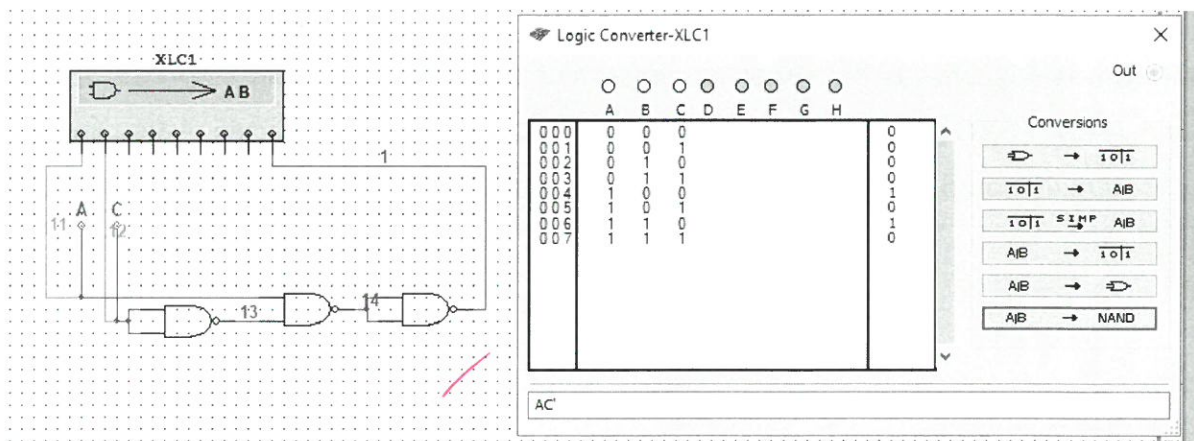
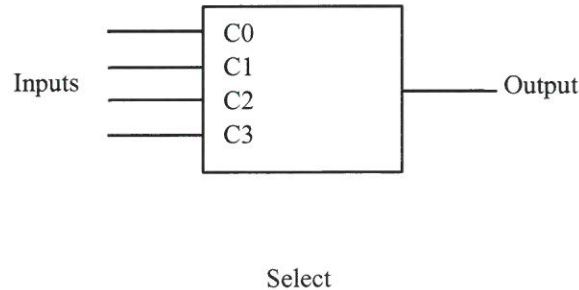


Figure 8

All parts were missing the process of simplifying the equation and DeMorgan.

Task 2 - Multiplexer

An arrangement where several inputs can be connected but only one input, under the influence of a select input, will have access to the output. The block diagram below shows a 4-to-1 multiplexer, produce a truth table and the Boolean equation to representation the multiplexer. Use this information to design the 4-to-1 multiplexer using logic gates and test its function using Multisim.



Task 2:

Figure 9 illustrates the multiplexer which was designed in multisim, the logic converter has been added in the multiplexer circuit as it is used to prove that the multiplexer circuit is working correctly by showing the truth table when a supply voltage from a VCC is added one of the branches.

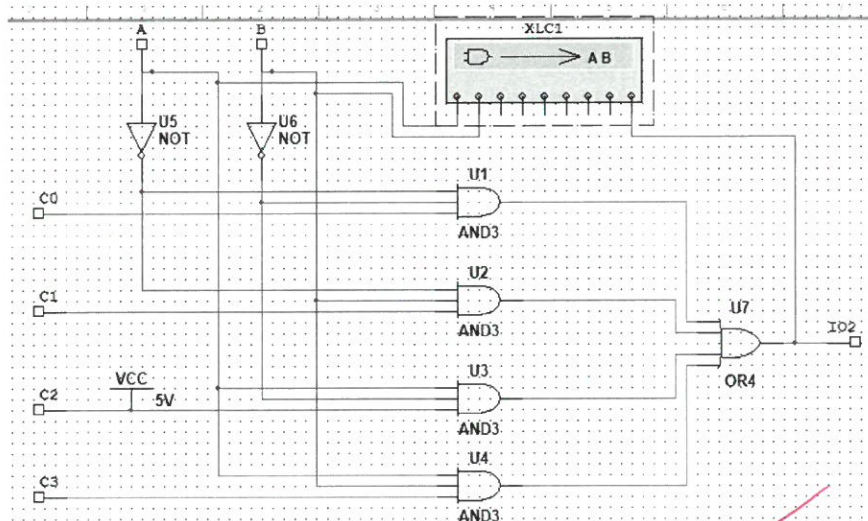


Figure 9

No truth table and Boolean equation

Figure 10 illustrates the circuit **without** the VCC connected, the logic converter is then used to show the truth table of the circuits current state, as the truth table shows, all outcomes are 0, which is correct for not having the VCC connected.

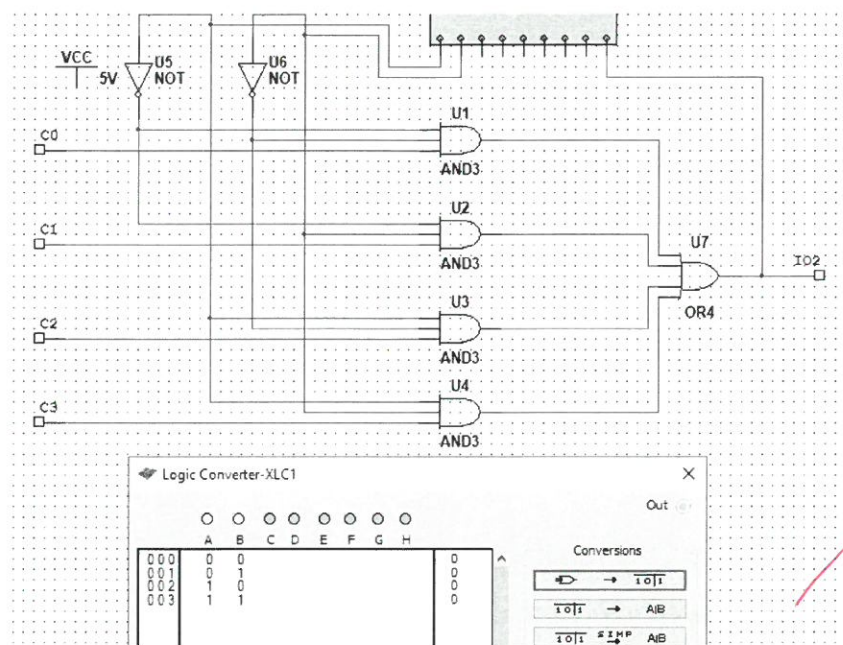


Figure 10

Figure 11 illustrates the circuit **with** the VCC connected, the logic converter is then brought up to prove the current state of the circuit, as the truth table shows in **figure 11**, all outcomes are **logic 1** which proves that the multiplexer is working correctly.

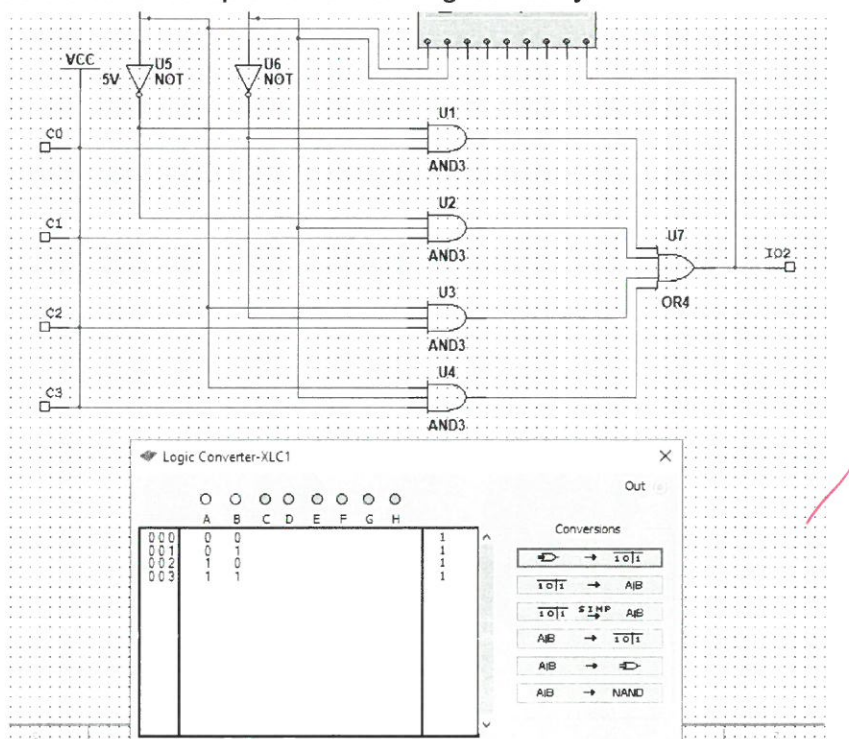


Figure 11

As a final test of the multiplexer circuit, a VCC was connected to two of the branches and the other two were left without a supply, this means the two which the VCC is connected will be **logic 1** and the other two will be **0**, as **Figure 12** illustrates, the above statement is true and the multiplexer circuit works correctly and has been tested.

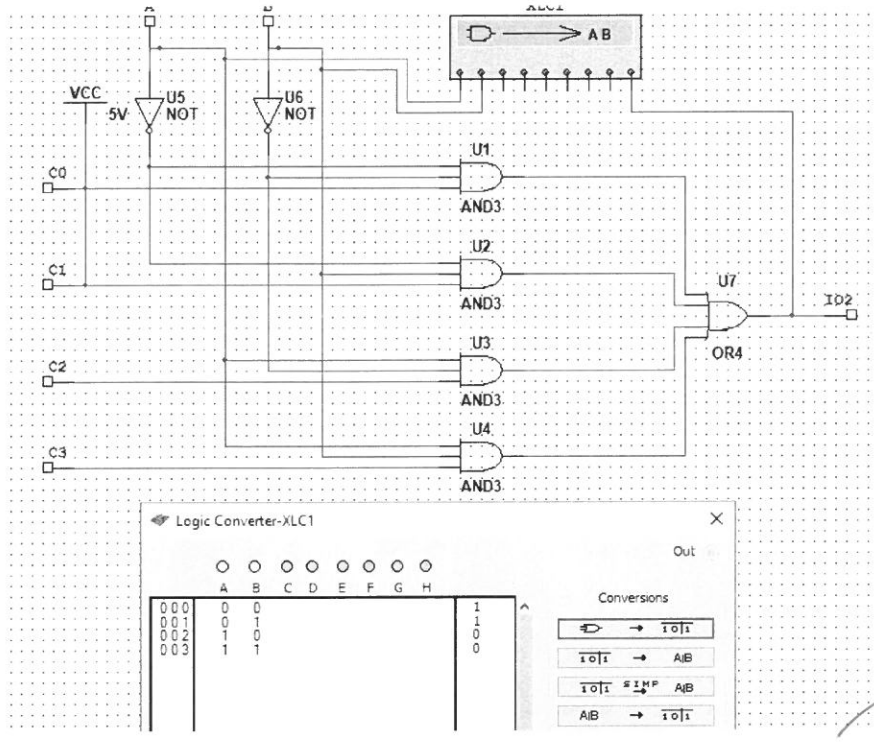


Figure 12

Task 3 - Sequential Logic

The truth table of the D-type and J-K Flip-Flop

- a) Construct the following debouncing circuit, and connect the clock of a D-type flip-flop to the output. Monitoring the clock, Q and $\overline{Q}$ outputs with LEDs and complete **Table 1**.

Table 1 - D-type truth table

Input	Outputs	
D	Q	$\overline{Q}$
0	0	1
1	1	0

- a) **Figure 13** illustrates the D-TYPE flip flop being set at 0, as the figure shows, when the D-TYPE is set at 0, the Q' on the D-TYPE is on, as the 'X2' led shows.

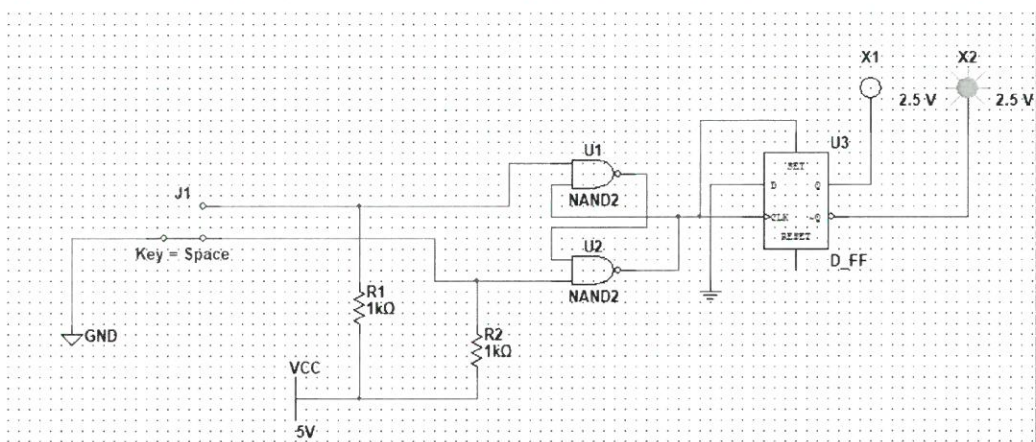


Figure 13

- Figure 14** illustrates the D-TYPE flip flop being set at 1, as the figure shows, when the D-TYPE is set at 1, the Q on the D-TYPE is on, as the 'X1' led shows.

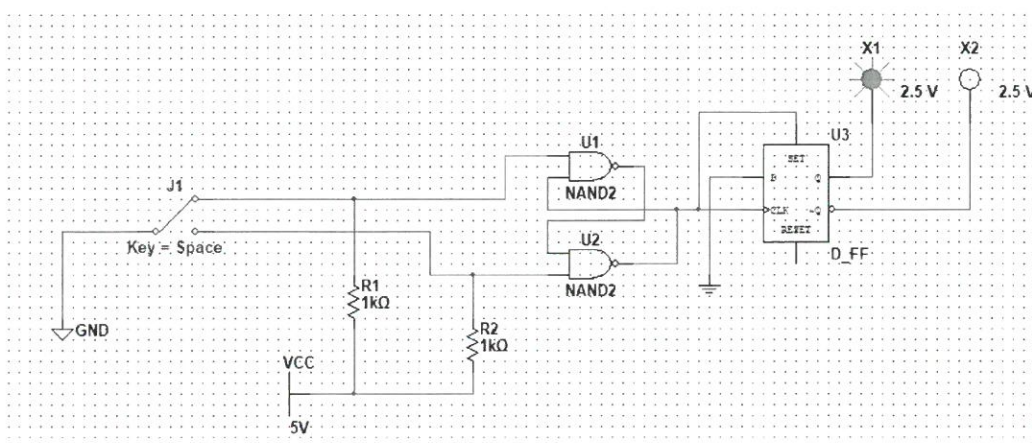


Figure 14

- b) Now connect the clock of J-K flip-flop to the output. Monitoring the clock, Q and $\overline{Q}$ outputs with LEDs and complete table 2

Table 2 – JK Truth Table

Input		Outputs	
J	K	Q	$\overline{Q}$
0	0		
0	1	0	1
1	0	1	0
1	1		

← ? No change.

← ? Toggle

- b) **Figure 15** illustrates the JK flip-flop with the J set to 0 and the K set to 1, as **figure 12** shows, the **LED (X2)** is on which shows Q' is 1 and Q is 0.

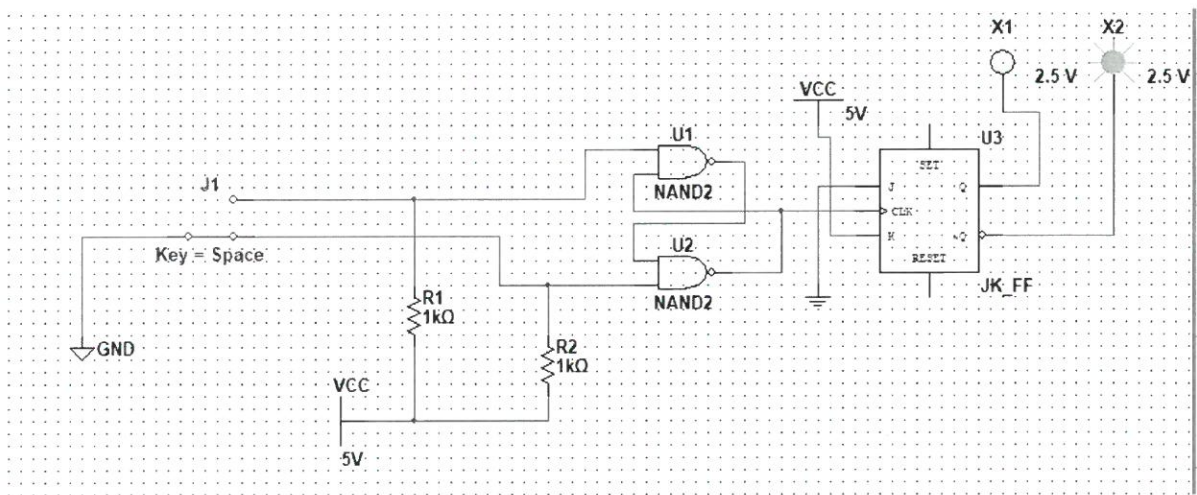


Figure 15

- Figure 16** illustrates the JK flip-flop with the J set to 1 and the K set to 0, as **figure 13** shows, the **LED (X1)** is on which shows Q' is 0 and Q is 1.

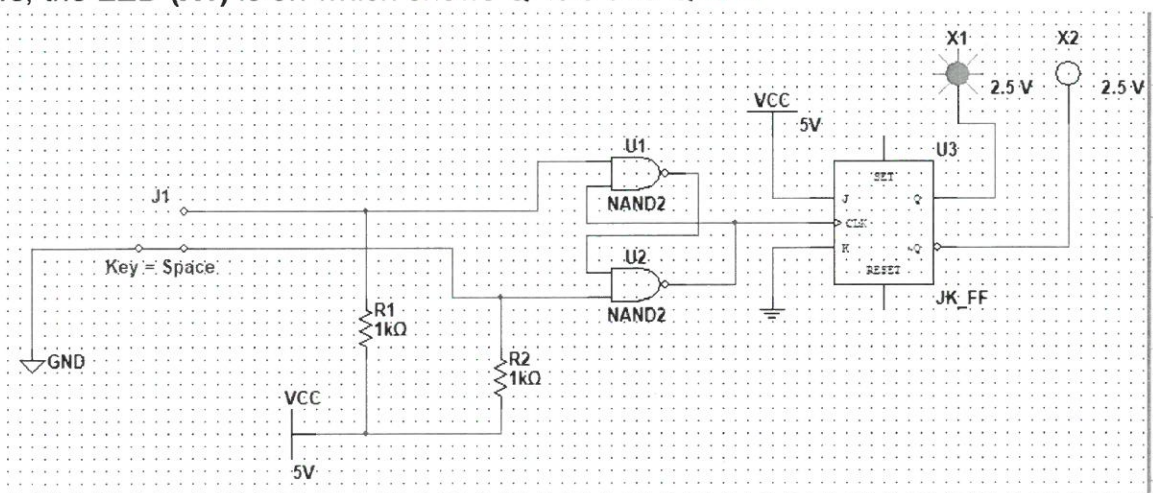


Figure 16

- (c) The cross-coupled NAND gates are for switch debouncing. Why are they needed and how do they work?

Switch debouncing is used to stop electrical components such as counter, from picking up signals which may be generated by accident, this could give a false reading on the counter which will affect the accuracy of the circuit.

Cross couple NAND gates are used, to produce a circuit which can latch on, this means when a signal is generated, it is latched on until the reset is activated, which then will unlatch the first signal and latch it onto the other signal, and example of this is a set reset circuit, the set input is activated which is then latched on allowing a constant signal to be generated, then the reset signal is activated, this will then unlatch the set signal, and latch the signal for the reset.

Task 4 - Using Sequential Logic to Solve Real Problems

Design a synchronous 0-6 counter using D-type flip-flops, the design must include a state table and Karnaugh maps. Once the design has been produced use a suitable simulation package to test the function of the counter and comment upon the results.

Figure 17 illustrates the state table which was produced for the D-type flip flops, this information will then be used to produce the Karnaugh maps, which will provide the information for the connections.

Current			Next			Input		
QA	QB	QC	QA	QB	QC	QA	QB	QC
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	0	0	0	0	0	0

Figure 17

Figure 18 illustrates the first Karnaugh map which was produced for the first D-type flip flop, this will show the logic connections for this D-type flip flop.

	QB QC			
QA	00	01	11	10
0	0	0	1	0
1	1	0	X	X

$$D_A = Q_B Q_C + Q_A \overline{Q_C}$$

Figure 18

Figure 19 illustrates the second Karnaugh map which was produced for the second D-type flip flop, this will show the logic connections for the D-type flip flop.

	QB QC			
QA	00	01	11	10
0	0	1	0	1
1	0	0	X	X

$$D_B = \overline{Q_A} \cdot Q_B \cdot Q_C + Q_B \cdot \overline{Q_C}$$

Figure 19

Figure 20 illustrates the final Karnaugh map which was produced for the third D-type flip flop, this will show the logic connections for this flip flop when producing it on multisim.

	QB QC				
QA	00	01	11	10	
0	1	0	0	1	DC= 'Qc
1	1	0	X	X	

Figure 20

The next step in designing a counter using D-type flip flops is to produce the design on multisim using the information gathered from the Karnaugh maps above, the expressions for **D_A**, **D_B**, and **D_C** show what each D-type counter will be connected to, to ensure it works correctly, **D_A** stands for the first D-type flip flop, **D_B** stands for the second D-type flip flop, and finally **D_C** stands for third D-type flip flop. Using the information:

- **D_A = Q_B.Q_C+Q_A.'Q_C**, this was used to produce the logic gate arrangement in the top left corner for the first D-type flip flop, **Q_B** is taken from Q output on the second D-type flip flop, and it placed in to an **AND** gate with **Q_C**, **Q_C** is taken from the Q output on the third D-type flip flop, the **(+)** represents an **OR** gate, which is what the output from the **AND** gate which **Q_B.Q_C** have produced will go into. The **Q_A.'Q_C** part will create another **AND** gate, this time the output is taken from **Q_A** which is the first D-type flip flop, another output is then taken from **'Q_C**, which is the lower Q output on the third D-type flip flop, and placed into the **OR** gate, the output of this **OR** gate will then be connected to the **D** input on the first D-type flip flop, the **(')** represents a bar, these are clearly shown on the D-type flip flops in **Figure 21**
- **D_B = 'Q_A.'Q_B.Q_C+Q_B.'Q_C**, This was used to produce the logic gate arrangement at the bottom of the page for the second D-type flip flop, **'Q_A.'Q_B.Q_C** are taken from their respective outputs on their D-type flip flops, and placed into a 3 input 1 output **AND** gate, the output from this **AND** gate is then placed into one side of a 2 input **OR** gate, **Q_B.'Q_C** are then taken from their respective outputs on their D-type flip flops and placed into a 2 input 1 output **AND** gate, the output of this **AND** gate is then placed into the other side of the **OR** gate, the output of the OR gate is then placed into the D input on the second D-type flip flop.
- **D_C = 'Q_C**, this was used to connect the **'Q** output of the third D-type flip flop to the D input of the third D-type flip flop, this then completed the circuit.
- A 5V VCC was added to each CLOCK (**CLK**) input on each D-type flip flop, this allow each flip flop to function.

Figure 21 illustrates the completed circuit diagram which was produced using the information obtained from the Karnaugh maps as discussed above.

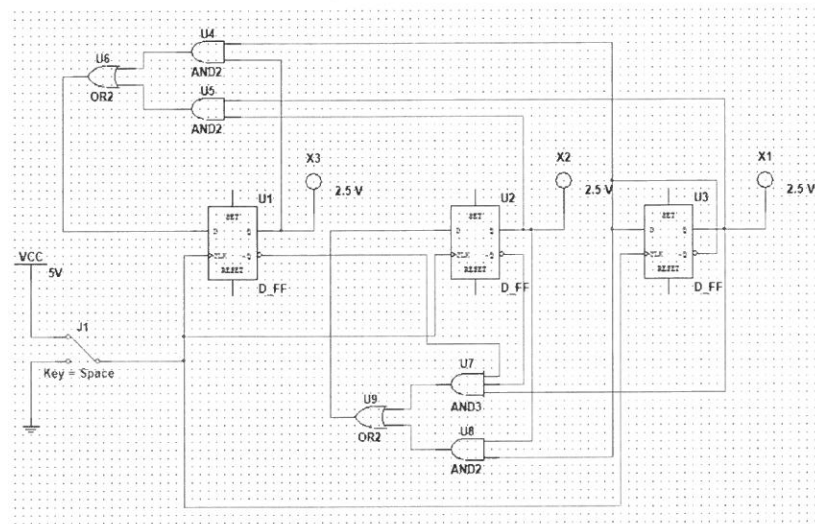


Figure 21

Figure 22 illustrates the counter being simulated and working, this is done by connecting a logic analyser which allows cycles of each D-type flip flop to be seen, so this allows the counter to be tested and checked, as this diagram shows.

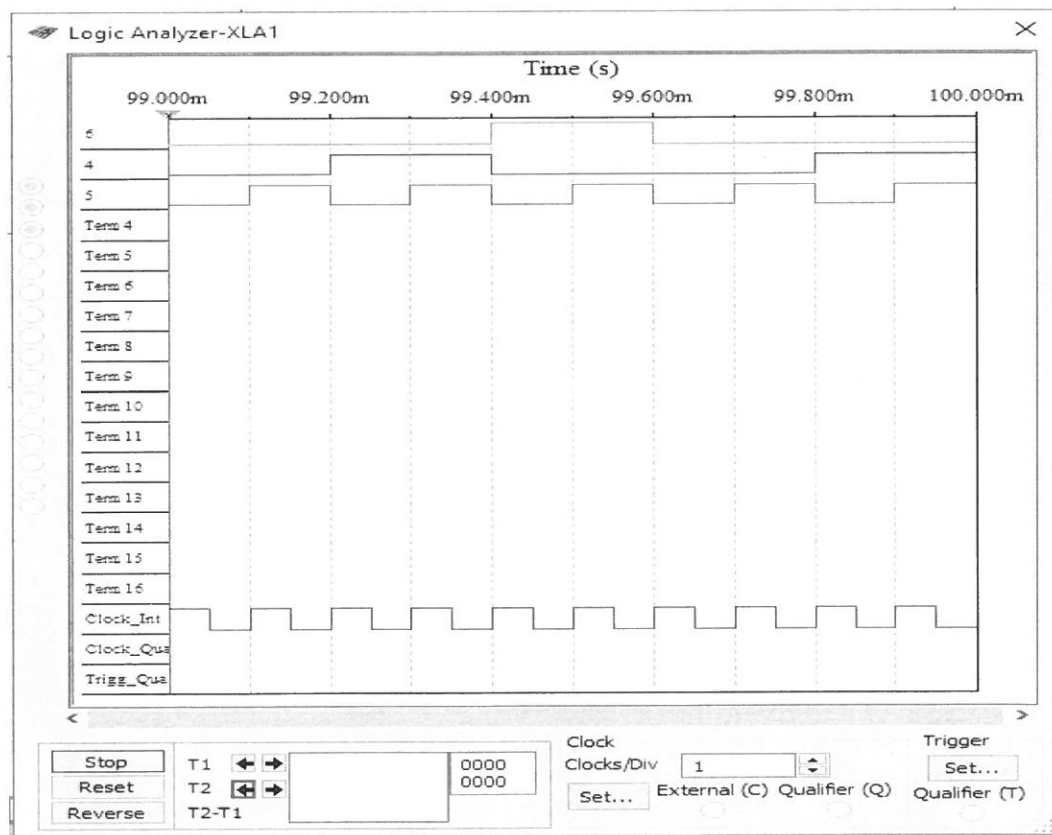


Figure 22

No references.