



DIGITAL SYSTEMS

HNC Electrical/Electronic Engineering

PCEE4006
Dr. Rajni Bansal
Electronics

Shay Pallas

Content Page

TASK 1 – BOOLEAN ALGEBRA AND DE MORGAN'S THEOREM	2
i) $Q = ABC + \overline{A}\overline{B}C + A\overline{B}\overline{C}$	2
ii) $Q = ABC + \overline{A}B + A\overline{B}\overline{C}$	3
iii) $Q = (A + B)(A + \overline{B})(\overline{A} + C)$	4
TASK 2 – USING COMBINATIONAL LOGIC TO SOLVE REAL-WORLD PROBLEMS	5
i) Creating Boolean Expression	5
ii) Constructing a Schematic Diagram.....	5
iii) Applying Boolean Algebra	6
iv) Applying De Morgan's Theorem	7
TASK 3 – MULTIPLEXER	8
TASK 4 – SEQUENTIAL LOGIC	10
i) Functionality and Characteristics of Flip Flops.....	10
ii) Timing Diagrams.....	13
TASK 5 – SEQUENTIAL LOGIC.....	16
TASK 6 – LOGIC FAMILIES AND THEIR CHARACTERISTICS	19
REFERENCES	22
TASK 6 – BIBLIOGRAPHY.....	23

Task 1 – Boolean Algebra and De Morgan's Theorem

$$I. Q = ABC + \overline{A}\overline{B}C + A\overline{B}\overline{C}$$

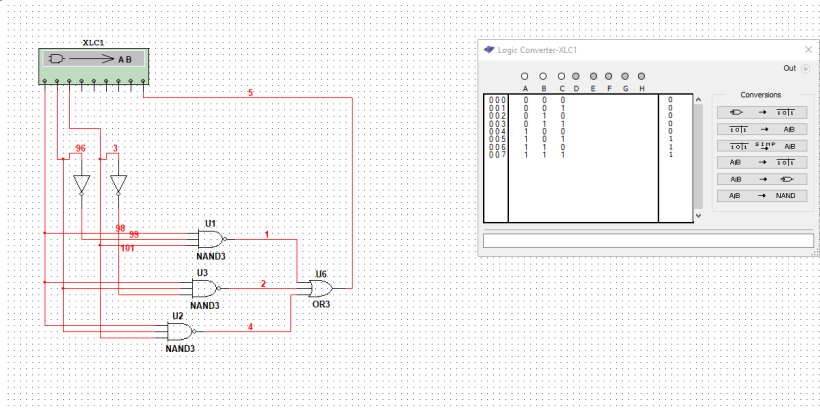


Image 1.1 – Original logic circuit

Logic gate circuit for the Boolean expression provided.

Simplification of original Boolean expression.

$$ABC + AB'C + ABC'$$

$$AC(B + B') + ABC' \text{ - Distributive Law}$$

$$AC(1) + ABC' \text{ - Complement Law}$$

$$AC + ABC' \text{ - Identity Law}$$

$$A(C + BC') \text{ - Distributive Law}$$

$$A(B + C) \text{ - Absorption Law}$$

$$Q = AB + AC$$

Double inversion on both sides to implement De Morgan's theorem. Allowing the original Boolean expression to be resembled with NAND gates.

$$\overline{\overline{Q}} = \overline{\overline{AB + AC}}$$

$$\overline{\overline{Q}} = \overline{A'B' \cdot A'C'}$$

Three NAND gates will be required to implement the simplified logic circuit. The truth table in image 1.1 can be referred to the truth table produced from the NAND gate expression in image 1.2. With comparison both truth tables are identical meaning the conversion is correct.

Commented [RB1]: Not right? Be careful when you expand Not

It should be rather $((AB)') \cdot (AC)'$

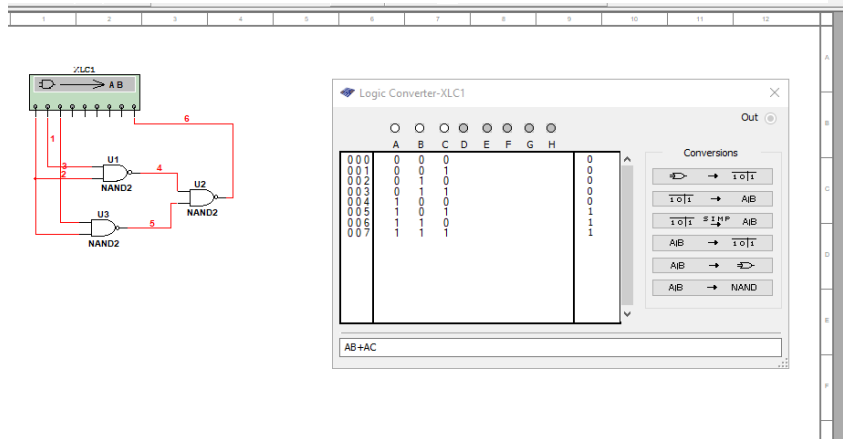


Image 1.2 – Simplified NAND gate circuit.

II. $Q = ABC + \bar{A}B + AB\bar{C}$

$A \cdot B (C + C') + A'B$ – Distributive Law

$A \cdot B (1) + A' \cdot B$ – Complement Law

$A \cdot B + A' \cdot B$ – Identity Law

$B (A + A')$ – Distributive Law

$B (1)$ – Complement Law

$Q = B$

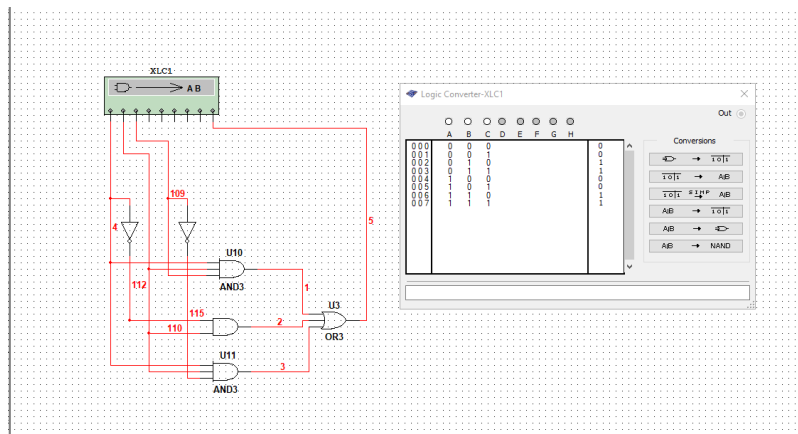


Image 1.3 – Original logic circuit

As the simplified expression is an output of B there is no need to apply the De Morgan's theorem to simplify the circuit as its in its most simplistic form. Image 1.3 is showing the original Boolean expression in a logic format. This circuit highlights the necessary requirements of simplification. As the original Boolean

expression would consist of a significant amount of IC's wasting materials and cost when essentially the output can be connected directly to input B.

$$\text{III. } Q = (A + B)(A + \bar{B})(\bar{A} + C)$$

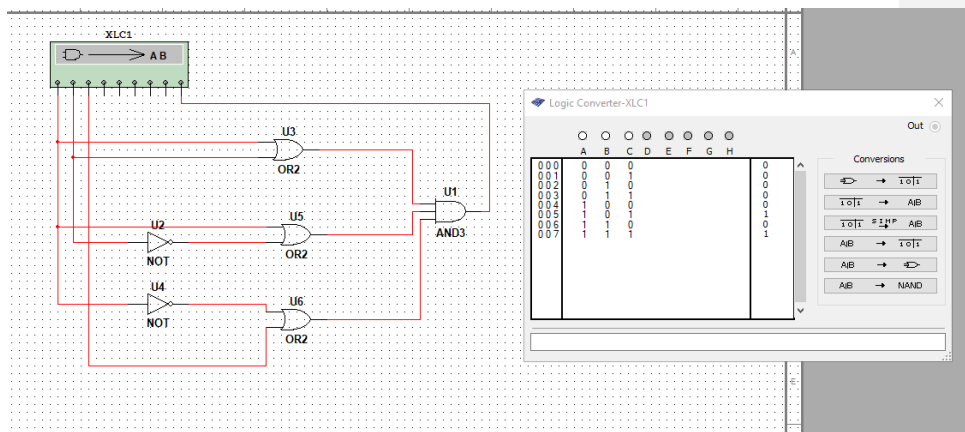


Image 1.4 – Original logic circuit.

$(A(B + B'))(A' + C)$ – Distributive Law

$(A(1)) (A' + C)$ – Complement Law

$A (A' + C)$ – Identity Law

$AA' + AC$ – Distributive Law

$0 + AC$ – Identity Law

$$Q = AC$$

Double inversion on both sides to implement De Morgan's theorem on the simplified Boolean expression. Referring to image 1.4 the truth table is identical to the simplified NAND gate expression shown in image 1.5. Showing the simplified circuit is correct.

$$\overline{\overline{Q}} = \overline{\overline{A.C}}$$

Two NAND gates will need to be used to simplify this circuit.

Commented [RB2]: ? Should be $A+BB'$ and then $A+0 = A$

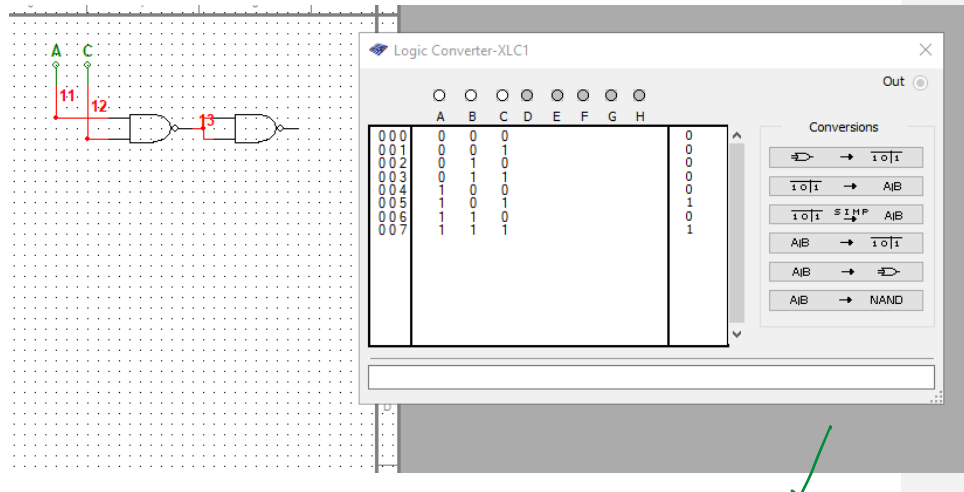


Image 1.5 – Simplified NAND gate circuit.

Task 2 – Using Combinational Logic to Solve Real-World Problems

- I. This truth table has been produced from the given requirements provided within the assignment.

A	B	C	Q
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Table 2.1 – Truth Table for Logic Circuit.

Taking the three combinations which produce a positive output (1). The Boolean expression will be constructed.

$$A'BC' + A'BC + ABC'$$

- II. Using the Boolean expression which has been constructed in section I of this task a logic circuit can be created which is shown in image 2.1. The output / truth table within image 2.1 is the exact truth table which has been constructed previously in table 2.1. Showing the Boolean expression and logic circuit are correct.

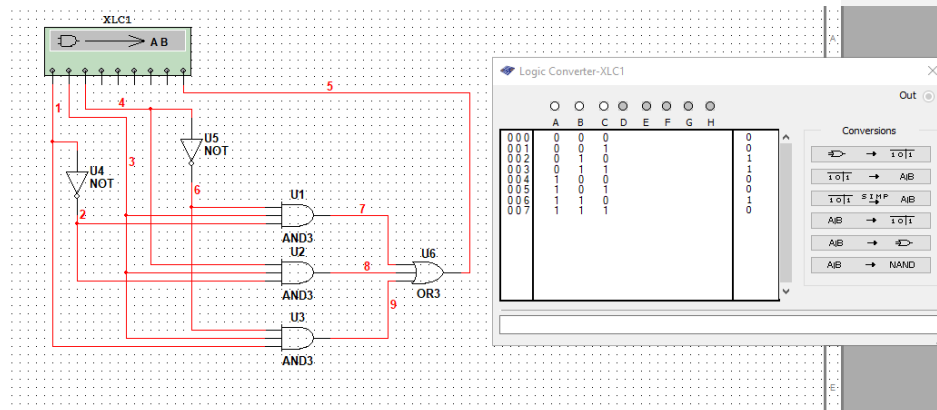


Image 2.1 – Schematic diagram and simulation.

- III. To minimise the logic circuit Boolean algebra will need to be used. The original expression will be used to find a suitable simplified equation. Refer to image 2.2 for the simplified circuit.

$$A'BC' + A'BC + ABC'$$

$$A'B.(C+C') + ABC' - \text{Distributive Law}$$

$$A'B(1) + ABC' - \text{Complement Law}$$

$$A'B + ABC' - \text{Identity Law}$$

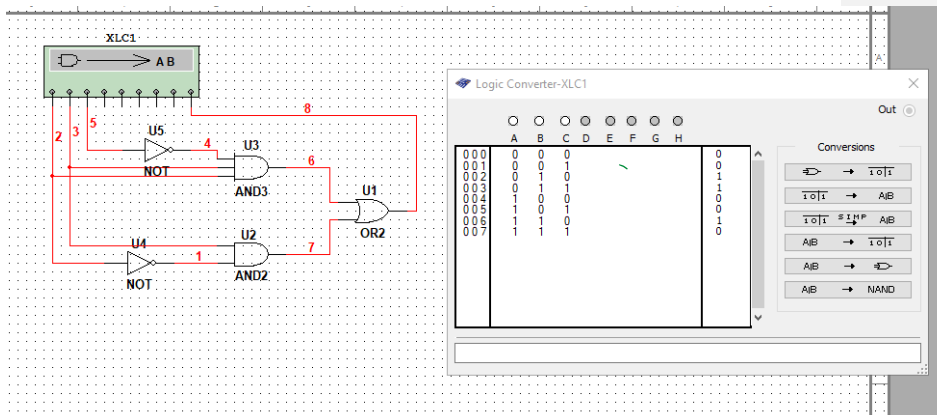


Image 2.2 – Simplified Schematic Diagram.

The truth table in image 2.2 is the same output as the previous simulation and the original truth table created in table 2.1.

- IV. De Morgan's theorem can be applied to the simplified logic circuit to convert the original logic circuit to only NAND gates. (Charles H. Roth Jr. 2010) The theory which has been applied can be seen below.

$$\bar{Q} = \overline{AB + ABC}$$

$$Q = \overline{AB} \cdot \overline{ABC}$$

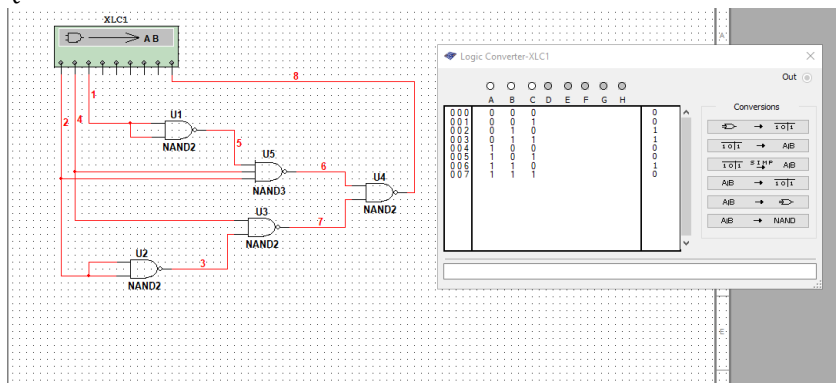


Image 2.3 – NAND Gate Schematic diagram.

The circuit has been simplified and converted to NAND gates through De Morgan's theorem to reduce the cost of the circuit also reducing the complexity. When constructing a logic circuit integrated circuits (ICs) are used. ICs can only store a certain amount of logic gates. Minimising the amount of logic gates allows for minimal IC's to be used, minimising material costs. Now only two IC's will need to be used whilst if the circuit was constructed from the schematic diagram shown in image 2.1 the amount of IC's will be significantly higher. Four two input NAND gates will be in the first IC and the three input NAND gate will be stored in the second IC. "Digital circuits are more frequently constructed with NAND and NOR gates than with AND and OR gates. NAND and NOR gates are easier to fabricate with electronic components and are the basic gates used in all IC digital logic families." (M. Morris Mano. 2006) Throughout industry NAND gates are the universally preferred logic gate to provide a function circuit. This is because cells are designed to have similar drive strengths. Due to the pull up and pull-down resistance of a NAND gates ratio is significantly better than a NOR gate.

Commented [RB3]: Be careful in implementing De-Morgan's as it starts with double inversion for the whole expression.

Commented [RB4]: Text citation requires only sir name and year and should be mentioned before full stop.

Commented [RB5]: Good explanation.

Task 3 – Multiplexer

A truth table has been created to resemble the four input one output multiplexer. The functioning of a multiplexer takes the different input signals to give the desired output. Performing the function of selecting the input on any one of the input lines (C0-C3) then feeding these inputs to the individual output line. Refer to table 3.1.

S1	S0	C0	C1	C2	C3	Q
0	0	1	-	-	-	C0
0	1	-	1	-	-	C1
1	0	-	-	1	-	C2
1	1	-	-	-	1	C3

Table 3.1 – NAND Gate Schematic diagram.

From the truth table which has been created a Boolean expression is able to be formulated. The Boolean expression can be used to create the following schematic diagram. The schematic diagram represents the 4 to 1 multiplexer. Refer to image 3.1, 3.2 and 3.3 to view the circuit in operation.

$$Q = S1'.S0'.C0 + S1'.S0.C1 + S1.S0'.C2 + S1.S0.C3$$

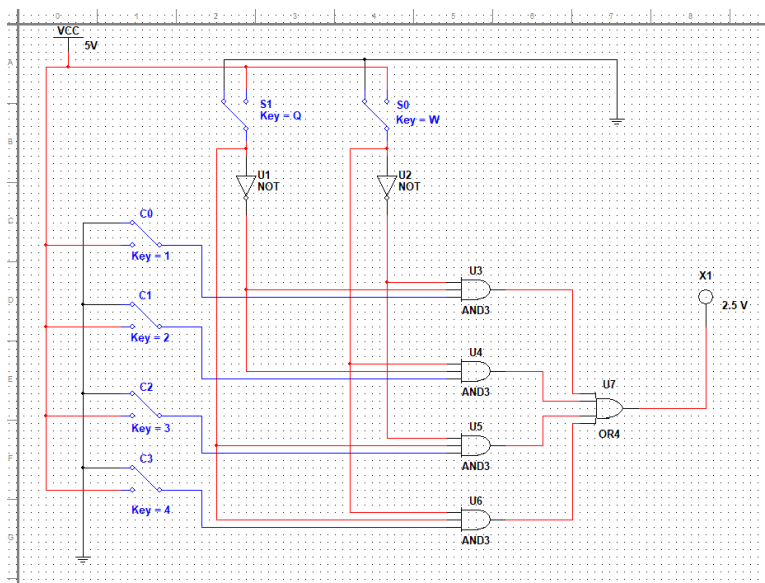
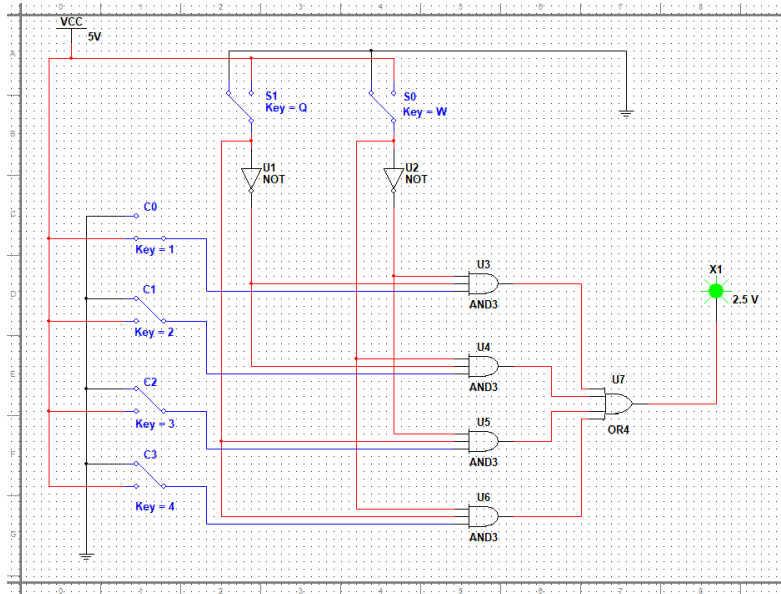
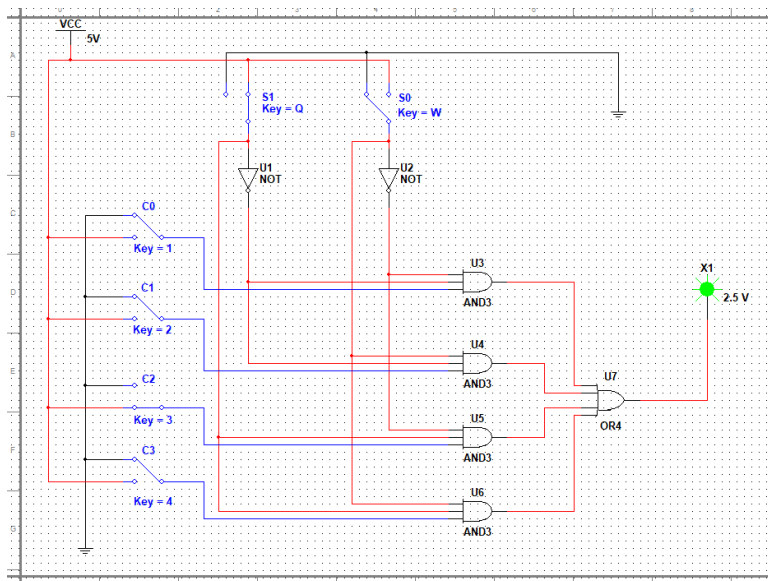


Image 3.1 – Multiplexer Circuit Construction.

*Image 3.2 – Multiplexer Output for C0.**Image 3.3 – Multiplexer Output for C2.*

De Morgan's theorem will be implemented to the Boolean expression to convert all logic gates in the previous screenshots to the required NAND gate format.

$$Q = S1'.S0'.C0 + S1'.S0.C1 + S1.S0'.C2 + S1.S0.C3$$

$$\bar{Q} = \overline{S1'.S0'.C0 + S1'.S0.C1 + S1.S0'.C2 + S1.S0.C3}$$

$$Q = \overline{S1'.S0'.C0} \cdot \overline{S1'.S0.C1} \cdot \overline{S1.S0'.C2} \cdot \overline{S1.S0.C3}$$

This new Boolean expression can be used to create a NAND gate multiplexer circuit which can be seen in image 3.4 below. The circuit can be seen functioning in the correct **manor** as the previous circuit. Refer to image 3.2.

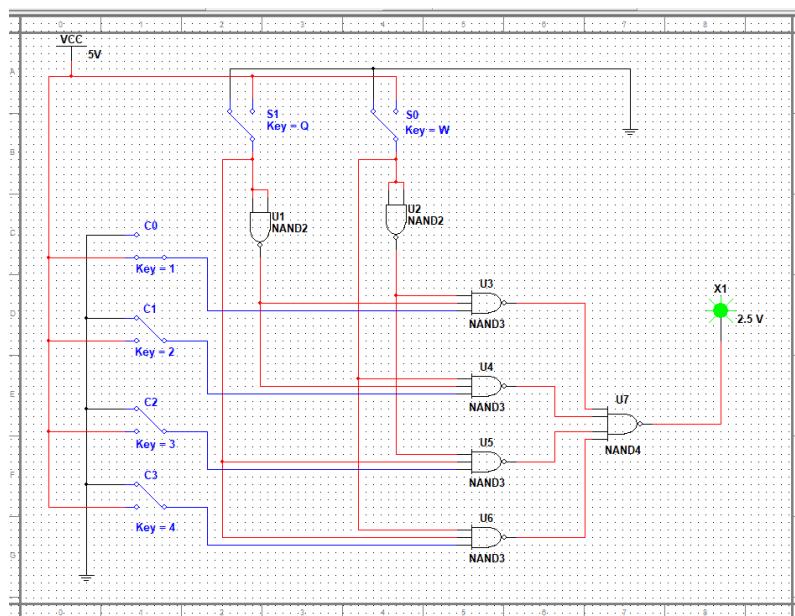


Image 3.4 – NAND Gate Multiplexer **Functioning**.

Commented [RB6]: Spelling?

Task 4 – Sequential Logic

- I) A flip flop circuit has two states (0 and 1). Flip flop circuits are used to store a single bit. Creating a flip flop circuit is obtainable from using only NAND or NOR gates. Having either one or two inputs with two outputs. The outputs are complementary of one and other. Meaning if Q output is equal to 1 Q' must be equal to 0. If the system is reset and Q' is equal to 1 then Q must equal 0 as the two outputs cannot be in the same state simultaneously. If somehow the two outputs become the same value this **violates** the definition of a flip-flop and hence is called an undefined

Commented [RB7]: Good explanation of working of multiplexer, however could relate it with real life applications.

Commented [RB8]: ,

condition. Normally the state of Q is called the state of the flip-flop, whereas the state Q' is called the complementary state of the flip flop." (A. Saba and N. Manna. 2007). A flip flop is a storage cell which allows for information to be latched in between clock pulses. A flip flop circuits functionality is shown in an excitation table rather than a truth table. An excitation table shows the functionality of that specific type of flip flop circuit. "An excitation table, which provides information about what its (flip-flop) input should be if the outputs are specified before and after the clock pulses." (S. Sarkar. 2014).

Commented [RB9]: Should go as (Saba and Manna, 2007)

- D-Type Flip-Flop

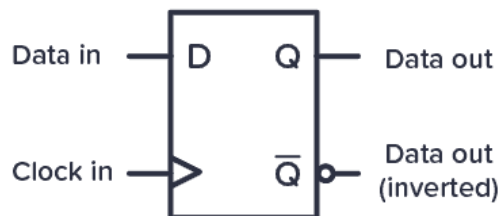


Image 4.1 – D-Type Flip-Flop Symbol

D	QN	QN + 1
0	0	0
0	1	0
1	0	1
1	1	1

Table 4.1 – D-type Flip-Flop Truth Table

- JK Flip-Flop

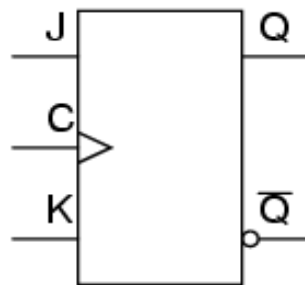


Image 4.2 – JK Flip-flop symbol

J	K	QN	QN + 1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Table 4.2 – JK Flip-flop Truth Table

Within task 4 it is asked to use the characteristics and excitation truth tables of the flip flops to convert JK flip-flop into a D flip-flop. Refer to Table 4.3.

D	QN	QN + 1	J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	1	X	0

Table 4.3 – D type Conversion to JK Flip-Flop.

Karnaugh Map for J

	0	1
0	X	0
1	1	X

$$J = D$$

Karnaugh Map for K

	0	1
0	X	1
1	X	0

$$K = \bar{D}$$

Commented [RB10]: Could show the excitation truth table of JK flip flop as it's being used in the task.

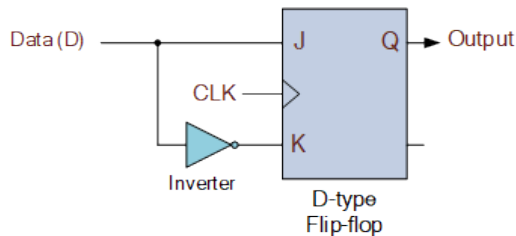


Image 4.2 – D type converted into a JK Flip Flop.

- II) A shift register is a device which uses multiple flip flops combined to shift data. As one flip flop can store 1-bit of data, a shift register allows for can store multiple bits. In this circuit a D type flip flop will be utilised. A D type flip flop output is the same value of the input. If input = 1 output will = 1. The D type flip flops will be connected from one's output to the other one's input. This is what allows the data to be shifted from one flip flop to another. Refer to image 4.3 for a visual representation of the output being transferred to the input of the next flip flop in the shift register.

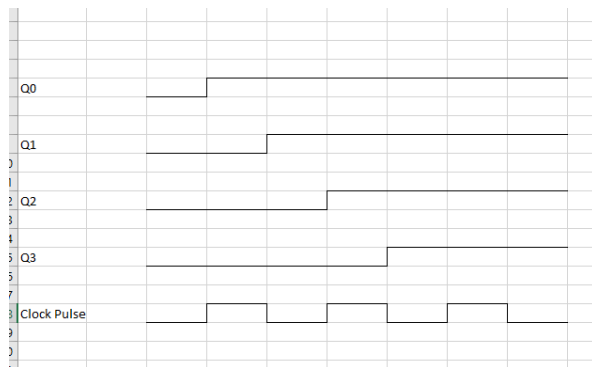


Image 4.3 – clock Pulse screenshot

The bidirectional shift registers which has been created has had various logic gates implemented to allow the shift to be possible both ways. To allow the bits to be shifted left the input needs to be connected to the flip flop located furthest right. To allow this to happen the not gate in the circuit created needs to be implemented. Once the switch is open and the not gate is functioning this allows for the flip flop furthest right to receive the input. This causes AND gates 1, 3, 5 and 7 to become inactive (Refer to image 4.4 for AND gate numbering). This happens because the properties of an AND gate is if one input is equal to 0 the output will always be equal to 0. This allows for the shift left. Refer to image 4.4 this shows the shift progressing left through the shift register.

Commented [RB11]: Sentence structure?

Commented [RB12]: Could also quote this info the truth table format and could be bit detailed about timing diagram showing how clock pulse is being used as reference etc.

Commented [RB13]: ?



Commented [RB14]: Good description of circuit's functionality.



To maintain more control of the output this circuit produces, two switches could have been incorporated into the design. One on Flip flops U1 input and U4 input. This would give the input the capability to be in the on or off state (0 or 1) and not just dependent on the original switching system in the circuit, allowing more control overall. Another improvement would be to connect the reset and -Q pins on all flip flops to ground. Allowing them to constantly remain in an off state. In a non-simulation situation leaving these pins disconnected may cause an issue with results.

Commented [RB15]: Good suggestions for improvements.

After the simulation, a functioning shift register has been created using two 74HC74 (Philips 74HC74 Datasheet) ICs. The circuit was created on a LD-2 logic designer allowing the use of built in LEDs and pulsers to be utilised. When constructing the circuit there were various issues with cable lengths and spacing. To minimise crowding of cabling for overall tidiness the power supply has been split over two rails just by adding a few minor connections to minimise the amount of cable cross overs. The pulser has also been connected to one of the top rails allowing for easy access and multiple connection points maintaining a good standard of work. Image 4.6 is showing the shift register operating in the right direction. Every time PB2 (pulse) is pressed the next LED will operate accordingly. Refer to image 4.7 for the Shift register operating in the opposite direction.

Commented [RB16]: Well done, good to know how you improved on the circuit built.

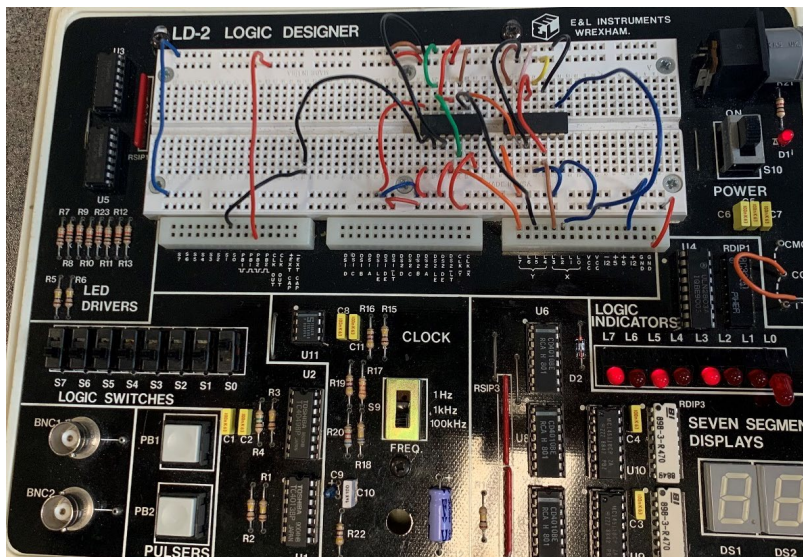


Image 4.6 – Shift Register Practical, Shift Right.

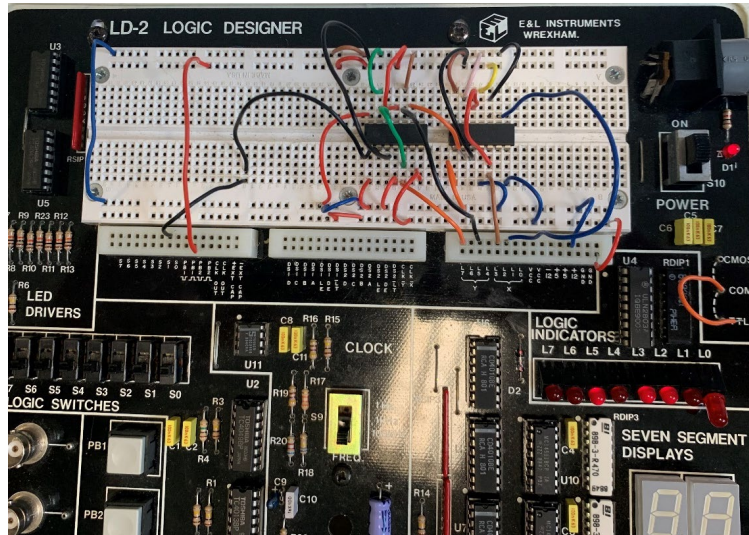


Image 4.7 – Shift Register Practical, Shift Right.

Task 5 - Using Sequential Logic to Solve Real Problems

A 3-bit synchronous binary counter has been designed and created using 3 JK flip flops. As stated this counting system shall only count odd numbers from 0-8 (1, 3, 5, 7). The binary equivalent of these numbers has been identified through binary to decimal conversion and placed into a truth table. This truth table also contains the next state. Refer to table 5.1.

A	B	C	A+	B+	C+
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	1	0	1
1	0	1	1	1	1
1	1	1	0	0	0

Table 5.1 – Next state truth table.

Now that the next state truth table has been completed for all odd values required. To be able to design what section of JK flip Flops need to be connected to each other and external power inputs an excitation truth table is required. Using the Next state truth table and applying the excitation table for JK flip flop this can be achieved. Refer to table 5.2.

A	B	C	A+	B+	C+	JA	KA	JB	KB	JC	KC
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	1	0	X	1	X	X	0
0	1	1	1	0	1	1	X	X	1	X	0
1	0	1	1	1	1	X	0	1	X	X	0
1	1	1	0	0	0	X	1	X	1	X	1



Table 5.2 – Excitation Truth Table.

From the excitation truth tables Karnaugh maps have been produced for JA, KA, JB, KB, JC, KC.

Karnaugh Map for JA

	00	01	11	01
0	0	0	1	X
1	X	X	X	X

$$JA = B$$

Karnaugh Map for KA

	00	01	11	01
0	X	X	X	X
1	X	0	1	X

$$KA = B$$

Karnaugh Map for JB

	00	01	11	01
0	0	1	X	X
1	X	1	X	X

$$JB = C$$

Karnaugh Map for KB

	00	01	11	01
0	X	X	1	X
1	X	X	1	X

$$KB = 1$$

Karnaugh Map for JC

	00	01	11	01
0	1	X	X	X
1	X	X	X	X

JC = 1

Karnaugh Map for KC

	00	01	11	01
0	X	0	0	X
1	X	0	1	X



KC = AB

Refer to image 5.1 for a screenshot showing the JK flip flop counter showing number 3 being counted. Refer to image 5.2 for number 5.

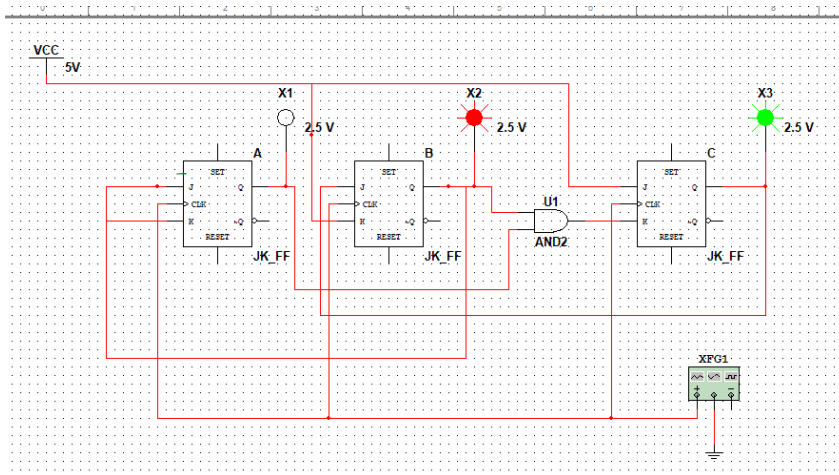


Image 5.1 – JK Flip Flop Counter, Number 3.

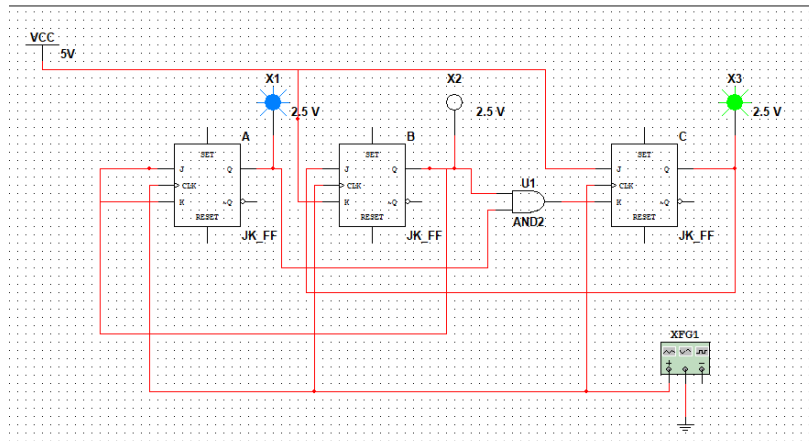


Image 5.2 – JK Flip Flop Counter, Number 5

Commented [RB17]: Could explain the functionality of the circuit by just taking one case into account.

Task 6 – Logic Families and Their Characteristics

Throughout industry logic families are highly significant as all digital systems are built up of ICs. Meaning if all ICs of a digital systems are built up of the same logic family this allows full compatibility. Compatibility with ICs belonging to different families are capable but additional compatibility interfacing techniques must be used. There are two types of logic families which are mainly used throughout industry. Either on their own or the combination of the two. These are Bipolar families and MOSFET families.

Commented [RB18]: Sentence structure

Within these families these are broken down to further subfamilies. These vary due to their performance and characteristics. The families which will be addressed within this report are the TTL (Transistor-transistor logic) sub family from the Bipolar families and the CMOS (Complementary MOS) sub family from the MOSFET families.

The TTL family and CMOS family are built up of various sub families. All the subfamilies have different application and operating values allowing for optimal operation.

Characteristics

Within this section Various characteristics will be viewed for both CMOS and TTL sub families looking at the variations in values.

Propagation delay

Propagation delay is the length of time it takes for a signal to reach its required location. TTL sub families have a propagation delay of 33ns or less varying dependent on the sub family within TTL which is being used. For example, Schottky TTL has a propagation delay of ~10ns. With the CMOS family the delay varies

depending on multiple variables. A variation in input voltage and ambient temperature will cause a variation in the propagation delay. For 74HC00 propagation delay can vary from 7-135ns depending on the surrounding states. Refer to table 6.1 for values.

Commented [RB19]: Interesting findings.

t_{pd}	propagation delay	nA, nB to nY; see Figure 6						
		$V_{CC} = 2.0\text{ V}$	-	25	-	115	135	ns
		$V_{CC} = 4.5\text{ V}$	-	9	-	23	27	ns
		$V_{CC} = 5.0\text{ V}; C_L = 15\text{ pF}$	-	7	-	-	-	ns
		$V_{CC} = 6.0\text{ V}$	-	7	-	20	23	ns

Table 6.1 – Propagation delay. Nexperia.

Power Dissipation

Power dissipation is when logic circuits are consuming varying amounts of power during operations. minimal power consumption is the most desirable. Due to the fact lower power consumption will result in a lower value for power dissipation. Advantages of lower power dissipation is a circuit can then be closely compact reducing circuit sizing. When TTL logic gates output is in a high state it draws current from the power supply. The power dissipation of a TTL is the $V_{CC} \times I_{CC}$. "Power Dissipation in TTL circuits is constant over its range of operating frequencies. For example, the power dissipation of a LS TTL gate is a constant 2.2 mW" [Zeepedia]. With CMOS IC's these can fluctuate slightly in their values for power dissipation. Throughout the sub families they remain rather consistent. For a CMOS transistor power dissipation, a value of 0.1 has been taken from Philips datasheet.

Commented [RB20]: Full stop goes after citation.

Noise Margin

The noise margin for TTL systems is broken down into two margins. The low-level noise margin is the difference between 0.8V and 0.5V (0.3V). The high-level noise margin is the difference between 2.7V and 2V (0.7V). CMOS noise margins vary to TTL because CMOS can operate with a higher input voltage (up to 15V). This means CMOS circuits are capable of a higher noise margin. They can tolerate twice the amount of superimposed noise voltage on the input before errors begin to show in the signals. [AllaboutCircuits].

Current Sinking

Current sinking is when a load is connected to a device in a way where the current flows from the supply through the load into the device. Within a TTL system the sinking current is referring to the maximum amount of current a single pin can contain through the load being connected. TTL device can have a sink current up to 16mA. Some CMOS systems can have a sink current up to 100mA.

Current Sourcing

Current sourcing is when a load is connected to a device in such a way that the device is sourcing the current to the load. This is the amount of current a pin can

Shay Pallas.

Digital Systems Electronics.

supply to an external device. TTL systems are capable of sourcing up to 40μA. With a CMOS circuit with 74HC00 in optimal conditions they can be capable of sourcing 25mA [Texas Instruments].

Commented [RB21]: Overall conclusion could be provided.

References

Charles H and Roth Jr. 2010. *Fundamentals of logic design Sixth edition*. Cengage Learning, Page 47 [Accessed 01/04/2021]

M. Morris Mano. 2006. *Digital Logic and Computer Design*. Page 88 [Accessed: 01/04/2021]

A. Saba and N. Manna. 2007. *Infinity Science Press. Digital Principles & Logic Design*. Page 216. [Accessed: 06/04/2021]

S. Sarkar. 2014. *Foundation of Digital Electronics and Logic Design*. Pan Stanford Publishing, Page 122. [Accessed: 06/04/2021]

Nexperia. November 2015. *74HC00; 74HCT00 Product data sheet. Rev .7* [Accessed: 11/05/2021]

Texas Instruments. January 2021. *SN74HC00 Datasheet*. [Accessed: 11/05/2021]

Zeepedia. *Digital Logic Design*. [DC Supply Voltage TTL Logic Levels Noise Margin Power Dissipation Digital Logic Design Engineering Electronics Engineering \(zeepedia.com\)](#) [Accessed: 21/04/2021]

Philips. *Philips semiconductors integrated circuits data sheet. IC06* [1699902.pdf \(farnell.com\)](#) [05/05/21]

Philips. *Philips Semiconductors integrated circuits data sheet 74HC74: 74HCT74*. [74HC74; 74HCT74 Dual D-type flip-flop with set and reset; positive-edge trigger \(twibright.com\)](#) [Accessed: 20/05/2021]

AllaboutCircuit. *Logic Signals Voltage levels*. [Logic Signal Voltage Levels | Logic Gates | Electronics Textbook \(allaboutcircuits.com\)](#) [Accessed: 11/05/2021]

Bibliography

Andrew Hilton. 2012. *Basic of Logic Design: Boolean Algebra, Logic Gates*. [Accessed: 01/04/2021]

BS 7671:2018. *British Standard, Requirements for Electrical Installation IET Wiring Regulations Eighteenth Edition*. [Accessed: 21/04/2021]

Toggle Flip flop. *T-type flip flop*. [Toggle Flip-flop - The T-type Flip-flop \(electronics-tutorials.ws\)](#) [Accessed: 05/04/2021]

Introduction to Shift Registers. [Introduction to Shift Registers | Shift Registers | Electronics Textbook \(allaboutcircuits.com\)](#) [Accessed: 21/04/2021]

Electrical4U. *Logic Families*. [Logic Families Significance and Types of Logic Families | Electrical4U](#) [Accessed: 21/04/2021]

EE Times. *MOSEFET or Bipolar*. [MOSFET or bipolar, which should you use? | EE Times](#) [Accessed: 21/04/2021]

VLSI System Design. *Propagation delay*. [Propagation Delay of CMOS inverter – VLSI System Design](#) [Accessed: 21/04/2021]

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 re	Marks available	Marks awarded
1. Engagement with Literature Skills	10	
You have used a wide range of literature including books and relevant website and have followed Harvard referencing style with few inaccuracies. However, please take a note of text citation as text citation requires only sir name and year and should be mentioned before full stop. Otherwise, you have done a good job in providing the supporting evidence for your explanations throughout the assessment.	65% 10	6.5
2. Knowledge and Understanding Skills	30	
Your work demonstrates your competent knowledge and understanding of the basic underlying concepts of digital electronics with supporting evidence of reading. You have used appropriate techniques and methods to complete all set tasks given in the assignment. All your work is well expressed and mostly correct with few inaccuracies in implementing De-Morgan's theorem, at times. Your attempt to explain the functionality of bi-direction shift register is worth appreciation. Similarly, you could also explain the functionality of the circuit in task 5 by taking few cases into account. Also, at times, your work lacks the consideration of context of the application- the consideration of the application of each covered concept in real-life situations could have helped you to score more.	67% of 30	20.1
3. Cognitive and Intellectual Skills	10	
You have done an excellent evaluation of underlying concepts and principles, showing your ability to devise an argument from evidence. Your supporting statement for each circuit outcome in task 1 and 2 is well appreciated.	77% of 10	7.7

4. Practical Application Skills	40	
Your all simulated work is correct and well-presented, showing your excellent ability to perform simulations. You have done an excellent evaluation of obtained results by providing the suggestions for the improvement in future. Your work also provides the evidence of different approaches used to problem solving in particular context, especially in task 5. Well done.	80% 40	32
5. Transferable Skills for Life and Professional Practice	10	
You have presented your work well with efficient usage of equation editor and providing a good number of screen shots per task to explain the concept in depth. K-maps and truth tables are also well presented. However, you need to pay attention on the usage of full stops and having a capital letter after full stop. Also, at times, your sentences could be restructured. Overall, your work is structured and well presented, showing your ability to work effectively independently and/or as part of a team.	64% 10	6.4

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)		72.7%
	Up to 1 week late (40% Max)		
	Over 1 week late (0%)		