

Digital Systems – Oliver Kucyj

Learner Name: Oliver Kucyj

Course: HNC Electrical and Electronic Engineering Year 2

Module Code: PCEE4006

Module Name: Electronics

Assignment Title: Digital Systems

Table of Contents

Task 1 - Boolean Algebra and De Morgan's Theorem	3
Part 1	3
Part 2	5
Part 3	6
Task 2 - Using Combinational Logic to Solve Real-World Problems	7
Task 3 – Multiplexer	9
Task 4 – Sequential Logic	12
Flip-flops	12
Converting a JK flip-flop into a D-type flip-flop	13
Shift Registers	14
Bidirectional Shift Register.....	15
Shift register build on breadboard	17
Task 5 - Using Sequential Logic to Solve Real Problems	18
Task 6- Logic Families and Their Characteristics	22
References.....	24
Bibliography.....	25

Task 1 - Boolean Algebra and De Morgan's Theorem

Part 1

Below, the steps to simplify this expression are shown.

$$\begin{aligned}
 &A.B.C + A.B.C + A.B.C \\
 &A.C.(B + \bar{B}) + A.B.C - (\text{Distributive Law}) \\
 &A.C.1 + A.B.C - (\text{Complement Law}) \\
 &A.C + A.B.C - (\text{Identity Law}) \\
 &A.(C + B.C) - (\text{Distributive Law}) \\
 &A.(B + C) - (\text{Absorption Law}) \\
 &A.B + A.C - (\text{Distribution Law})
 \end{aligned}$$

DeMorgan's laws can be used to convert the expression to use only NAND gates. Hambley described how these laws work.

"If the variables in a logic expression are replaced by their inverses, the AND operation is replaced by OR, the OR operation is replaced by AND, and the entire expression is inverted, the resulting logic expression yields the same values as before the changes" (2013, p. 382).

This theorem is used below;

$$\begin{aligned}
 Q &= A.B + A.C \\
 \bar{Q} &= \overline{(A.B + A.C)} \\
 Q &= \overline{((A.B).(\bar{A}.C))}
 \end{aligned}$$

Figures 1 & 2 show the original expression and the simplified expression as NAND gates.

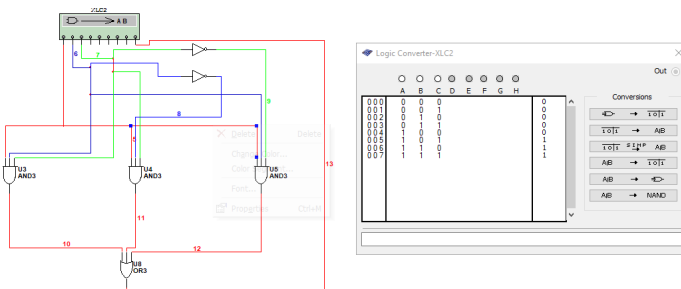


Figure 1 – Simulated logic circuit for $A.B.C + A.B.C + A.B.C$

Commented [RB1]: This happen when you do double inversion.

Commented [RB2]: You have missed not here, though your circuit is right.

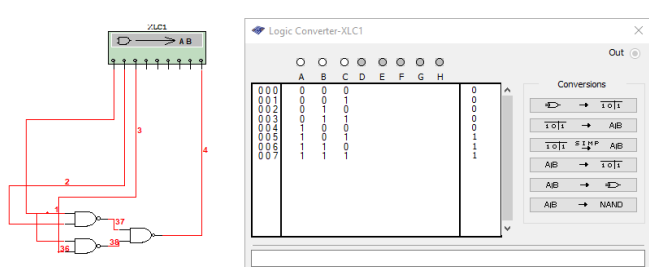


Figure 2 – Simulated logic circuit for $((A.B). (A.C))$

Commented [RB3]: Every evidence is provided and working is shown.

Part 2

Below, the steps to simplify this expression are shown.

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

$$Q = A.B.(C + \bar{C}) + \bar{A}B - (\text{Distributive Law})$$

$$Q = A.B.1 + \bar{A}.B - (\text{Complement Law})$$

$$Q = A.B + \bar{A}.B - (\text{Identity Law})$$

$$Q = B.(A + \bar{A}) - (\text{Distributive Law})$$

$$Q = B.1 - (\text{Complement Law})$$

$$Q = B - (\text{Identity Law})$$

The output to this expression is '1' whenever B is '1', meaning this does not need to be converted to NAND gates. The logic circuit would simply be a line from B to Q. *Figure 3* shows the original expressions logic circuit and truth table. This highlights how important simplification is, as this can be simplified to one line with no logic gates.

Commented [RB4]: Good observation.

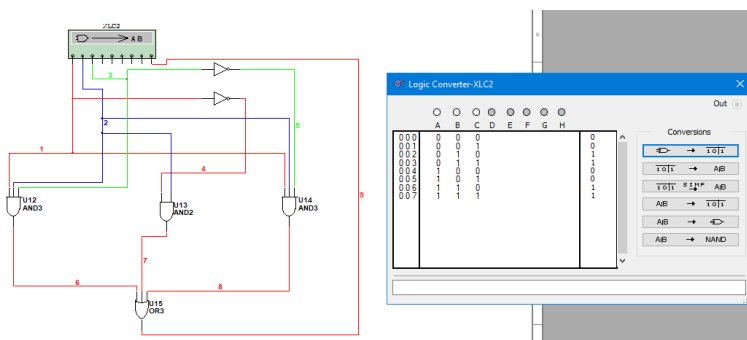


Figure 3 – Simulated logic circuit for $A.B.C + \bar{A}B + A.B.\bar{C}$

Part 3

Below, the steps to simplify this expression are shown.

$$\begin{aligned}
 Q &= (A + B)(A + \bar{B})(\bar{A} + C) \\
 Q &= (A \cdot (B + \bar{B})) \cdot (\bar{A} + C) - (\text{Distributive Law}) \\
 Q &= A \cdot 1 \cdot (\bar{A} + C) - (\text{Complement Law}) \\
 Q &= A \cdot (\bar{A} + C) - (\text{Identity Law}) \\
 Q &= A \cdot \bar{A} + A \cdot C - (\text{Distributive Law}) \\
 Q &= 0 + AC - (\text{Complement Law}) \\
 Q &= A \cdot C - (\text{Identity Law})
 \end{aligned}$$

Commented [RB5]: Should be $A + BB'$ and then $A + 0 = A$

DeMorgan's theorem is then used below.

$$\begin{aligned}
 Q &= A \cdot C \\
 Q &= \overline{(\bar{A} \cdot \bar{C})}
 \end{aligned}$$

Figures 4 & 5 show the original expression and the simplified expression as NAND gates.

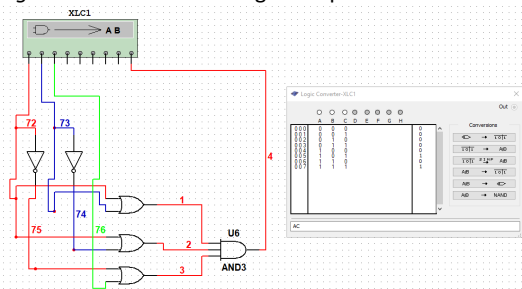


Figure 4 – Simulated logic circuit for $(A + B)(A + \bar{B})(\bar{A} + C)$

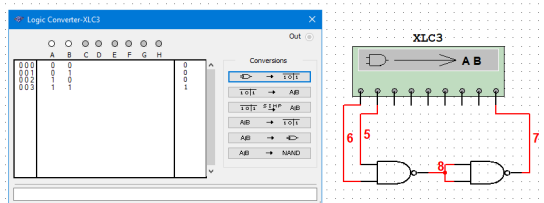


Figure 5 – Simulated logic circuit for $\overline{(\bar{A} \cdot \bar{C})}$

Commented [RB6]: Well done.

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

The Truth Table for the logic circuit controlling the chemical process is shown in *figure 6*.

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

Figure 6 – Truth table for the chemical process logic circuit

From this table, the Boolean expression $Q = \bar{A}\bar{B}\bar{C} + \bar{A}BC + AB\bar{C}$ can be found. *Figure 7* shows the logic circuit and truth table for this expression. This matches the original truth table proving it to be correct.

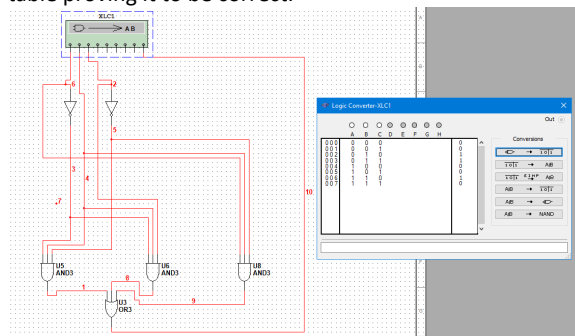


Figure 7 – Logic circuit for $\bar{A}\bar{B}\bar{C} + \bar{A}BC + AB\bar{C}$

This Boolean expression can then be simplified as shown below.

$$\bar{A}\bar{B}\bar{C} + \bar{A}BC + AB\bar{C}$$

$$\bar{A}B.(C + \bar{C}) + AB\bar{C} - (\text{Distributive Law})$$

$$\bar{A}B.1 + AB\bar{C} - (\text{Complement Law})$$

$$\bar{A}B + AB\bar{C} - (\text{identity Law})$$

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

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

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

Figure 8 shows the logic circuit and truth table of the simplified Boolean expression. This truth table is also the same as the truth table in figure 6 showing this is correct.

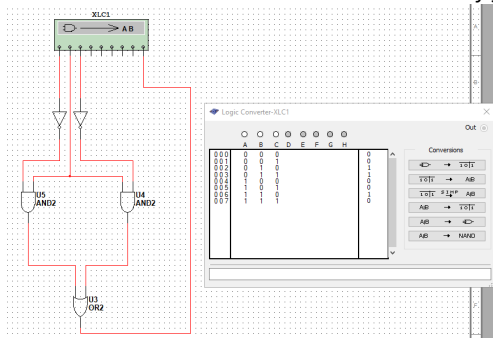


Figure 8 – Logic circuit for $\bar{A}B + B\bar{C}$

DeMorgan's theorem can then be used to convert this simplified expression to NAND gates as shown below. The logic circuit and truth table for this are shown in figure 9.

$$Q = \bar{A}B + B\bar{C}$$

$$Q'' = (\bar{A}B + B\bar{C})''$$

$$Q = ((\bar{A}B) \cdot (B\bar{C}))$$

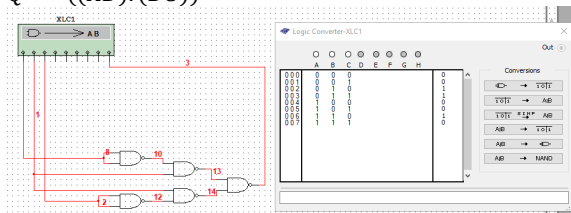


Figure 9 – Logic circuit for $((\bar{A}B) \cdot (B\bar{C}))$

It is important to simplify Boolean expressions to reduce the costs and complexity of the logic circuits. Logic circuits are implemented using Integrated Circuits (ICs) and these ICs have limited functionality. One IC can only implement a limited number of gates/inputs and all of these gates must also be the same logic. Both the original and simplified logic circuits would require 3 ICs. Even though the simplified circuit has fewer gates, there are 3 different logics used so 3 ICs are still required. This is why NAND gates are used universally. They can perform any logic task by simply rearranging the expression. The NAND version of this Boolean expression has 5 NAND gates. One 7400-series IC can have up to 4 2-input NAND gates, so only 2 ICs are required. This saves money and space as only 2 chips are required to produce the same logic.

Commented [RB7]: Clearly shows the good understanding of application of DE Morgan's theorem

Task 3 – Multiplexer

A 4-1 multiplexer has four inputs (C0-C3), however only one is directed to the one output (Q). The two select inputs determine which input is directed to the output (S1 & S0). Bolton wrote “A multiplexer is a circuit that is able to have inputs of data from a number of sources and then, by selecting an input channel, give an output from just one of them” (2015, p. 105).

Multiplexers can be used to reduce the cost of a system which has multiple inputs connected to a PLC or other system which requires the inputs to be converted by an ADC or DAC by eliminating the need for each input to have its own ADC or DAC. In the example from figure 10, each analogue input is sampled in turn and processed by the single ADC. This is a very cost-effective method, and many installations will have a demultiplexer on the other side of the ADC. This would then be connected to the same select inputs as the multiplexer, ensuring each input is directed to the correct output as each is sampled in turn.

Multiplexers have a fast transition time for switching between inputs meaning they are very efficient for this purpose. An example is a TMUX1204, which is a 4-1 analogue multiplexer with a transition time of 14ns. This means each input could be sampled in just 56ns. Both digital and analogue multiplexers/demultiplexers are available.

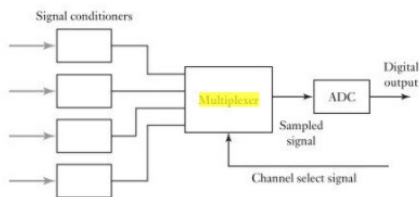


Figure 10 – 4-1 multiplexer with signal conditioners and an ADC (Bolton, W. 2015)

The truth table for a 4-1 multiplexer is shown in figure 11. The ‘X’ means that the input is not selected, so it does not affect the output.

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

Figure 11 – Truth table of a 4-1 Multiplexer

Using this truth table, the Boolean expression below can be found. Essentially, this expression states which input is active under the different select input conditions.

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

Commented [RB8]: Shows the evidence of your reading, however, good practice is to write the year and page no after Author's name.

Commented [RB9]: Good application - shows your deep understanding of multiplexers and demultiplexers.

Using this Boolean expression, the circuit shown in *figure 11* was created to represent a 4-1 Multiplexer. Figures 12-16 show examples of this circuit working correctly according to the truth table.

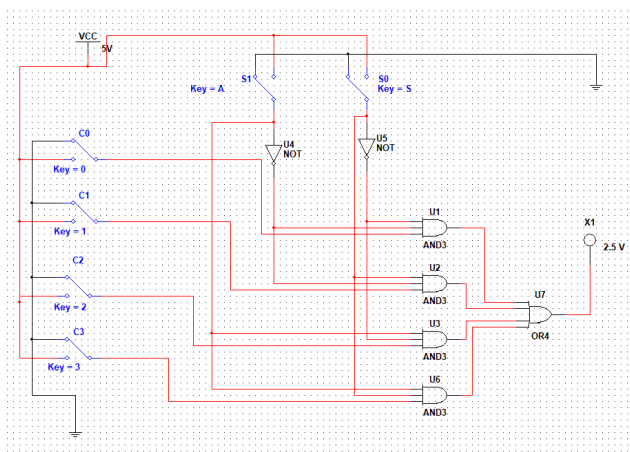


Figure 12 – Circuit to represent a 4-1 Multiplexer

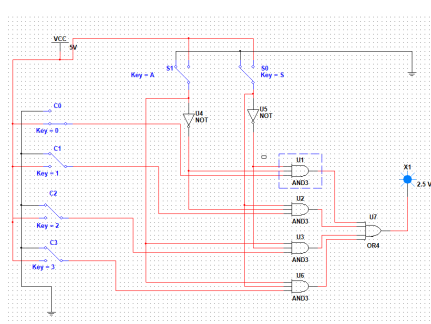


Figure 13

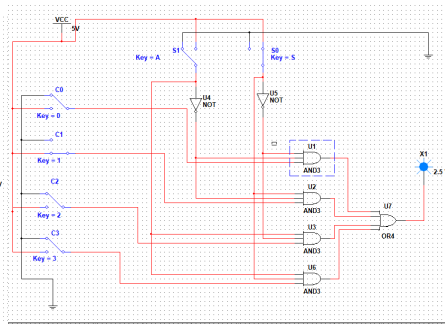


Figure 14

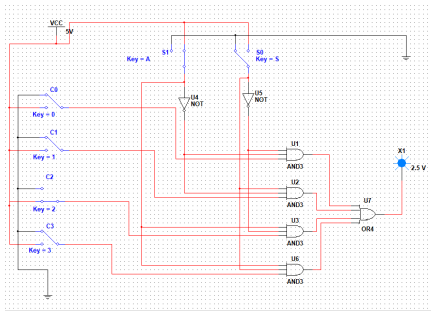


Figure 15

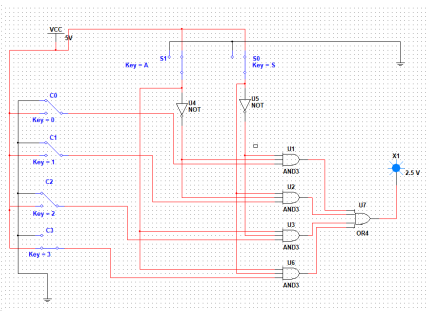


Figure 16

This logic circuit can also be implemented using NAND gates. Below, the steps taken to convert this Boolean expression using DeMorgan's theorem are shown.

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

$$\overline{Q} = \overline{(\overline{S1}.\overline{S0}.C0 + \overline{S1}.S0.C1 + S1.\overline{S0}.C2 + S1.S0.C3)}$$

$$Q = ((\overline{S1}.\overline{S0}.C0).(\overline{S1}.S0.C1) + (S1.\overline{S0}.C2) + (\overline{S1}.S0.C3))$$

This can then be used to create the circuit shown in *figure 16*. The screenshots in *figures 17-21* confirm this circuit's functionality.

Commented [RB10]: Well done, concept is explained in detail

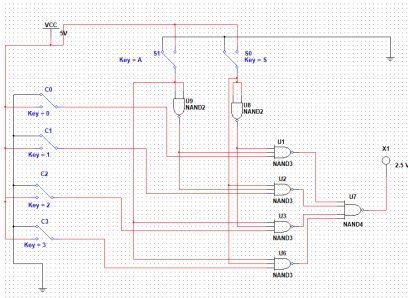


Figure 17 – 4-1 Multiplexer using NAND gates

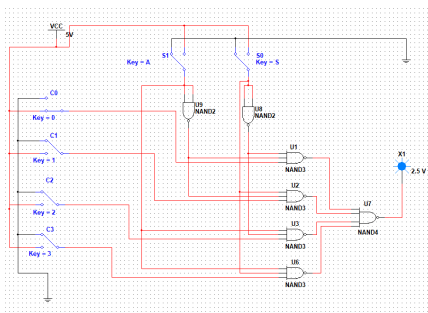


Figure 18

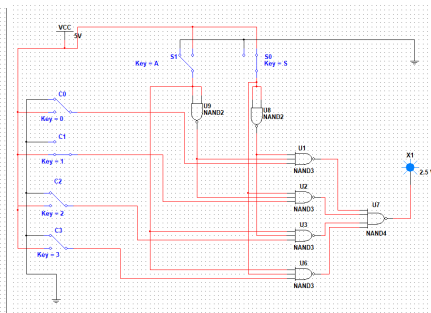


Figure 19

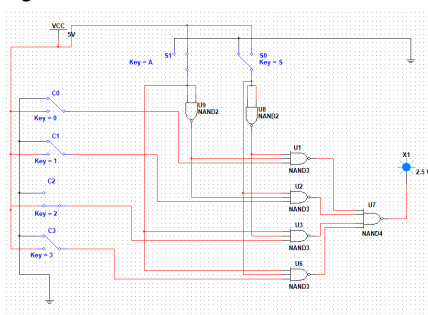


Figure 20

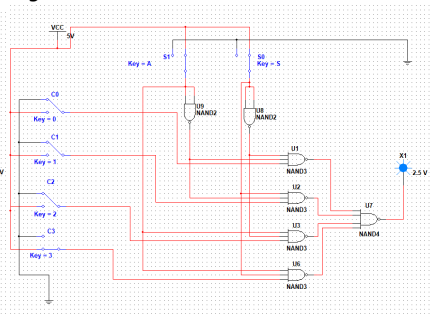


Figure 21

Task 4 – Sequential Logic

Flip-flops

Flip-flops are small components used for sequential logic. They are able to store a single binary digit and change that digit based on the input, the previous output and the clock pulse. They have a set and reset option for overriding stored data regardless of the clock pulse.

Every time the clock pulses, the flip-flop will check the input/output and change the output accordingly. Flip-flops can also be used as latches, as the input is captured at the clock pulse and is held until the next clock pulse. Some flip-flops react to the rising edge of the clock pulse and some react to the falling edge of the clock pulse. Bolton wrote *“The flip-flop is a basic memory element which is made up of an assembly of logic gates and is a sequential logic device.”* (2015, P. 126).

Commented [RB11]: Good intro of flip flops.

Figure 22 shows the truth table, logic circuit and symbol for a D-type flip-flop. Figure 23 shows the truth table for a JK flip-flop and figure 24 shows the logic circuit and symbol for the JK flip-flop.

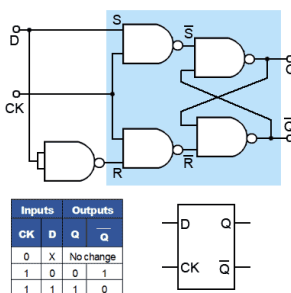


Figure 22 - D-type flip-flop (Learn about electronics)

J	K	Q
0	0	No Change
1	0	1
0	1	0
1	1	Toggles (opposite of previous value)



Figure 23 – JK flip-flop truth table

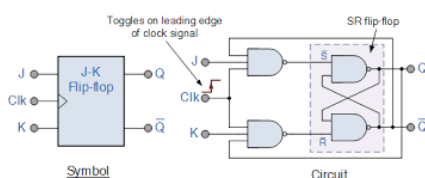


Figure 24 – JK flip-flop symbol and logic circuit (Electronics tutorials)

Converting a JK flip-flop into a D-type flip-flop

To convert the JK flip-flop into a D-type flip-flop the characteristic table for the D-type flip-flop (Figure 25) and the excitation table for the JK flip flop (Figure 26) must be found. They must then be merged. (Figure 27).

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

Figure 25 – Characteristic table for D-type flip-flop

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Figure 26 – Excitation table for JK flip-flop

Q_n	D	Q_{n+1}	J	K
0	0	0	0	X
0	1	1	1	X
1	0	0	X	1
1	1	1	X	0

Figure 27 – Comparison of the two tables

The next step is to find the Boolean expression for the available flip-flop inputs in relation to the required flip-flop. This is done using Karnaugh maps (K-maps). K-maps play a vital role in analysing digital data to find the simplest Boolean expression available. Hughes wrote "Karnaugh maps are a powerful, widely used, graphical method of logic simplification that will always give the simplest sum-of-products form." (2016, P. 543).

Figure 28 shows the K-map for input J and figure 29 shows the K-map for input K.

$Q_n \backslash D$	0	1
0	0	1
1	x	x

Figure 28 – K-map for input J

$Q_n \backslash D$	0	1
0	x	x
1	1	0

Figure 29 – K-map for input K

Using the results from this table, the Boolean expressions below can be found and a D-type flip-flop can be created using a JK flip-flop and a NOT gate, as shown in Figure 30.

$$J = D \text{ and } K = \bar{D}$$

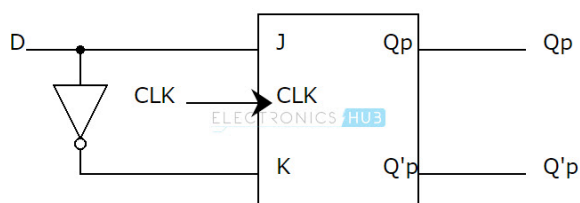


Figure 30 – JK flip-flop working as a D-type flip-flop (Electronics Hub)

Shift Registers

The diagram in the assignment shows a 4-bit serial in parallel out (SIPO) synchronous shift register using D-type flip-flops. The input is stored as a bit on the rising edge of the clock pulse. A high input is stored as '1', a low input is stored as a '0'. Once this bit is stored, all other bits are shifted to the right. Because it is parallel out, all digits are analysed simultaneously. This same shift register could be modified to a SISO (Serial in serial out) system by only analysing the final digit.

This specific shift register outputs a 4-bit binary number as there are 4 flip-flops and each flip-flop can store 1-bit. The shift register is synchronous because one clock pulse is connected to all flip-flops. A shift register with more than one clock pulse connected is called an Asynchronous shift register.

Applying a signal to the set input will override the output to '1' regardless of the clock pulse. Applying a signal to the reset input this will override the output to '0' regardless of the clock pulse.

There is constant power supply connecting to the input, so the truth table and timing diagram shown in figures 31 & 32 can be created. After clock pulse 4, the output will remain 1111 due to the constant high input.

Clock pulse No.	Q ₀	Q ₁	Q ₂	Q ₃
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1

Figure 31 – Truth table showing the output from the shift register

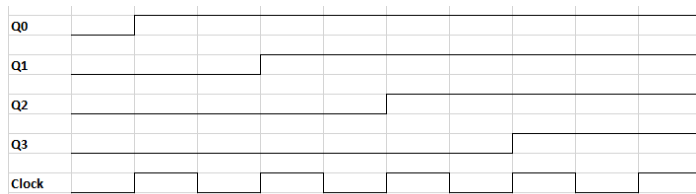


Figure 32 – Timing diagram for the shift register

Commented [RB12]: Good observation.

Commented [RB13]: Could say all the flip flops are clocked at the same time using one external clock to make it more meaningful.

Commented [RB14]: Good understanding of the functionality of the circuit.

Commented [RB15]: Good practice is to have clock input at the top as it is being used as reference.

Commented [RB16]: Good description of the connections.

Commented [RB17]: Good explanation about the working of the circuit.

Commented [RB18]: Good application

Figure 33 – Bidirectional shift register.

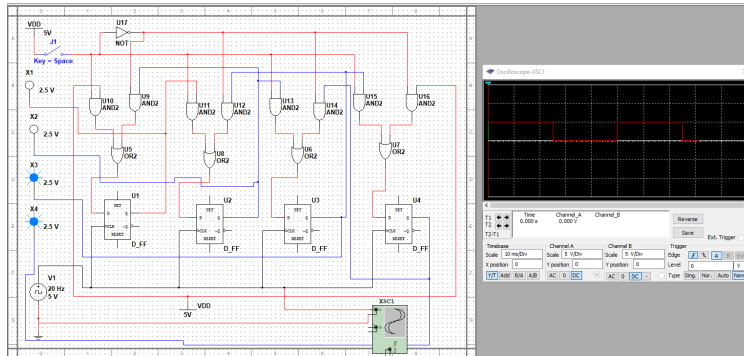


Figure 34

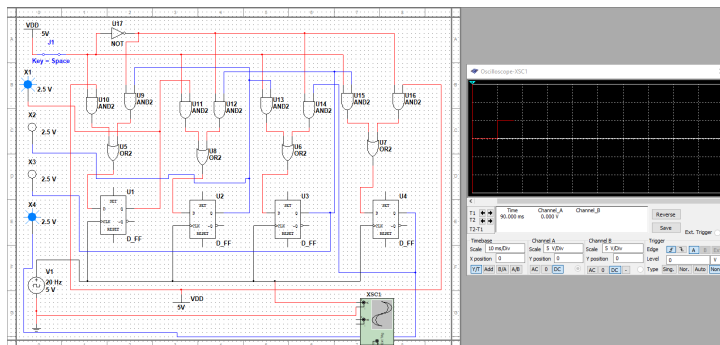


Figure 35

There are two improvements which could be made to this circuit. The first is that a switch could be connected after the 5V input supply. This would enable the input to be switched between a 'high' and 'low' input, adding more functionality. The set and reset pins could also have been connected to the input supply with a N/O switch connected to each one. This would enable the clock pulse to be overridden and the output manually set if required.

Commented [RB19]: Interesting suggestions.

Shift register build on breadboard

A shift register which only counts in one direction was built on breadboard inside a logic designer box using 74HC74 chip, which has 2 D-type flip-flops inside.

This circuit shifts the numbers to the right.

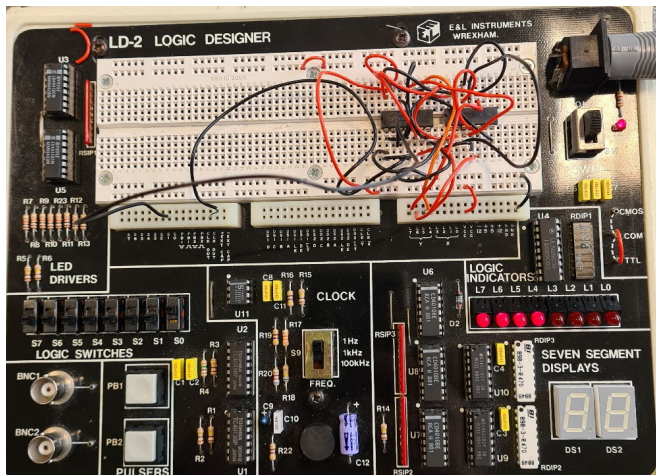


Figure 36 – Shift register

The process for building this circuit went well and it was built quickly. Some difficulties were encountered as the IC chip used was incorrect and time was spent looking at the connections to determine why it was not working. Once the correct chip was used, the circuit worked correctly. The wires could also have been neater, and better colour-coding of the wires could also have made it more presentable. This would also make it easier to make modifications to the circuit.

Commented [RB20]: Evidence of reflective learning.

Task 5 - Using Sequential Logic to Solve Real Problems

This section of the assignment will cover the steps taken to design and build a 3-stage synchronous binary counter using JK flip-flops which will count odd numbers from 0-7. This circuit will require JK flip-flops, a function generator for the clock pulse and a DC voltage source for the high inputs. 3 probes will also be used to identify the output of each flip-flop. Synchronous shift counters are more complex than Asynchronous counters, however they will work faster as one clock pulse is used for all flip-flops. The circuit will be a 3-bit system meaning 3 JK flip-flops will be required.

Commented [RB21]: Good description of the task.

First, the Excitation table for the JK flip-flop must be found. This was used in task 4 and is shown in *figure 26*. The required decimal numbers must also be expressed in binary, as shown in *figure 37*.

Required Decimal number	Binary Equivalent		
	4	2	1
0	0	0	0
1	0	0	1
3	0	1	1
5	1	0	1
7	1	1	1

Figure 37 – Decimal to binary Conversion

In this circuit, the flip-flops are labelled 'A', 'B' and 'C' to make the design stage clearer. Using this and the binary conversion, the next state table can be found. The next state table shows the previous binary number and the next binary number required. (*figure 38*.)

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

Figure 38 – Next state table

By using the next state table and the excitation table for the JK flip-flop, the inputs required to generate the next state can be found. First, the table in *figure 39* is created.

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

Figure 39 – Next state table combined with excitation table of JK flip-flop

The next stage is similar to the flip-flop conversion in task 4. The K-maps must be created for each flip-flop input. This will provide inputs which ensure that under each possible binary number specified, a clock pulse will generate the next number in the sequence. Figures 40-45 show this.

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

Figure 40 – K-map for JA – JA = B

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

Figure 41 – K-map for KA – KA = B

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

Figure 42 – K-map for JB – JB = C

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

Figure 43 – K-map for KB – KB = 1

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

Figure 44 – K-map for JC – JC = 1

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

Figure 45 – K-map for KC – KC = AB

Commented [RB22]: Shows your excellent understanding of designing a counter.

The shift counter shown in *figure 44* was then built, using the results from these k-maps to specify the inputs. A 5V supply was used for the input signal and a function generator was used to provide the clock pulse. As with task 4, probes were used to show the current output. *Figures 45-49* confirm the circuit's functionality.

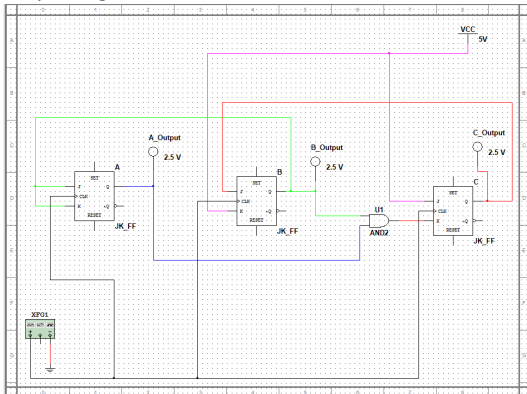


Figure 45 – 3-stage, synchronous shift counter counting odd numbers. Output of 000, equal to 0

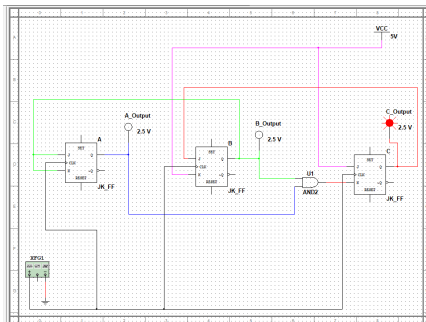


Figure 46 – output of 001, equal to 1

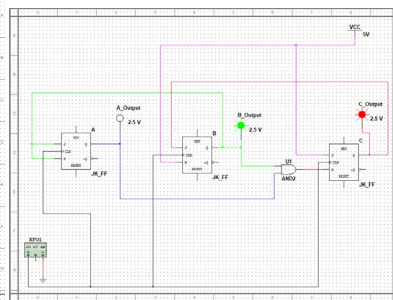


Figure 47 – output of 011, equal to 3

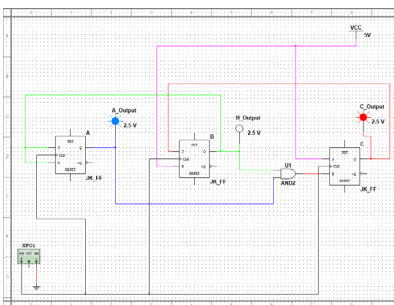


Figure 48 – Output of 101, equal to 5

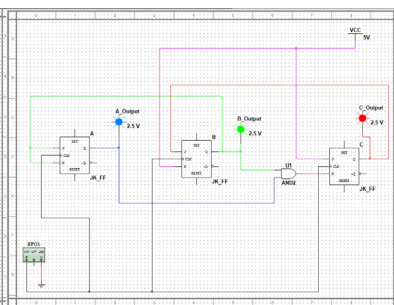


Figure 49 – Output of 111, equal to 7

This circuit works by considering the previous output and the new inputs at every clock pulse to ensure the correct number is returned. The example below shows the process when the current number is 000. The next number should return 001.

On flip-flop A, J & K are both connected to B (0). AQ is also 0. Using the truth table, this is confirmed to return 0.

On flip-flop B, J is connected to C (0) and K is connected to the supply (1). BQ is 0. Using the truth table, this is confirmed to return 0.

On flip-flop C, J is connected to the supply (1). K is connected to an AND gate with A&B. A is 0 and B is 0, so K is also 0. CQ is 0. Using the truth table, this is confirmed to return 1.

This correctly results in an output of 001. The same process is used for all stages of the binary counter.

This circuit could have been improved by connecting a low input to the set and reset pins to eliminate the risk of an error occurring due to a flip-flop being set/reset.

Commented [RB23]: Well done, shows your ability to apply the gain knowledge.

Task 6- Logic Families and Their Characteristics

Logic families play a vital role in designing digital integrated circuits. All ICs are fabricated using BJTs, MOS transistors or a combination of both. ICs which are fabricated in the same way are put into a logic family. All ICs in a single logic family will generally have similar characteristics, with some variation due to the different sub-families. Looking at the different families is important as some ICs may not be compatible with an IC from a different family, and the characteristics will need to be suitable for use with the digital circuit's requirements.

Commented [RB24]: Shows your good understanding of the concept.

There are many different characteristics which define these logic families. In this section of the assignment, the TTL (Transistor-Transistor-Logic) family and the CMOS (Complementary metal-oxide-semiconductor) family will have 5 characteristics compared. The TTL uses bipolar junction transistors and the CMOS uses MOSFETs. Where possible, values from the more modern/advanced sub-families will be used. The TTL sub-family will be the advanced low power Schottky TTL (74ALS00) and the CMOS sub-family will be the 74ACT.

Propagation delay (t_{pd}) – This value represents how quickly a circuit responds to a change in the input. A low value for this is desirable as the circuit will work faster. The value for each is listed below. The TTL has a lower propagation delay, meaning TTL is the better choice in terms of this characteristic.

74ALS00 (TTL) – 4ns (Bolton, W. 2015)

Commented [RB25]: No initials required in text citation.

74ACT (CMOS) – 10ns (Logic Guide, Texas Instruments)

Power Dissipation (p.d.) – This is the voltage drop through the IC and represents the power consumption of the device. This varies depending on the output state. A low value is desirable as the digital circuit will consume less power, allowing more devices to be added with minimal issues. A high p.d. value will also cause the temperature of a chip to increase, which can affect the reliability of the device. The p.d. values of standard devices from both families are listed below:

Commented [RB26]: Good analysis.

74ALS00 (TTL) – 1mW (Bolton, W. 2015)

CMOS – 10nW (Bolton, W. 2015)

Based on this CMOS family is the better choice.

Speed Power Product – This is found using the sum $t_{pd} \times p.d.$ As this is a product of both previously mentioned factors, it provides a good representation of how good the logic gate is, as a low T_{pd} and a low p.d. will result in a low speed power product. However, if one is higher this value is increased. A low value means a better logic gate. The speed power product of TTL chips and CMOS chips are listed below, calculated using the values previously stated. Based on these values CMOS gates are the better choice.

Commented [RB27]: Good attempt.

TTL – $(1.7 \times 10^{-9}) \times (10 \times 10^{-3}) = 17 \text{ Picojoules}$

CMOS – $(10 \times 10^{-9}) \times (10 \times 10^{-9}) = 100 \text{ Attojoules}$

Fan-in and fan-out – This is the number of standard inputs/outputs that can be connected to the gate. Fan-in is the number of inputs for a gate. Fan-out is the number of standard loads that can be driven by the output. CMOS gates can have up to 50 standard loads connected to the output and up to 10 inputs. TTL gates can have up to 10 standard loads connected to the output and up to 14 inputs. Based on this, the CMOS is the better option, unless it is vital that there are more than 10 inputs. (Components101)

Voltage Parameter – This determines the input voltages required for a 'high' or 'low' output and the voltage values that are found at the output under these conditions. Figures 50 & 51 show this for standard TTL and CMOS gates. The voltage parameter for CMOS can also vary depending on V_{cc} . All inputs between the minimum high voltage input and maximum low voltage input should not be used as manufacturers cannot guarantee what the output will be. From these figures, CMOS provides a much more accurate output. However TTL will work with a wider range of inputs. Note, when comparing the 74ACT to the 74AS, they can both register 2-5.5V as a 'high' input. Based on this, and the fact that the CMOS gates have a more accurate output, the more advanced CMOS chips have the edge over TTL chips.

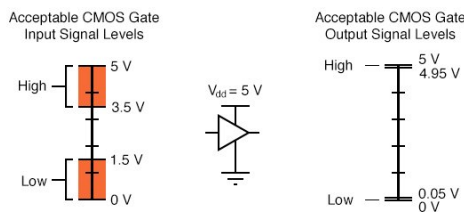


Figure 50 – CMOS Voltage parameters (All about Circuits)

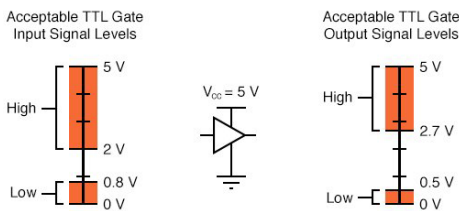


Figure 51 – TTL Voltage parameters (All about Circuits)

Based on these five comparisons alone, the CMOS gates seem the better choice when trying to decide which logic family to choose. The most important characteristics to consider when choosing a logic family are propagation delay and power consumption. Bolton wrote "The main criteria generally involved in determining which logic family to use are propagation delay and power consumption." (2015, p. 120).

Using this information, the logic family of choice will be whichever of these characteristics is most important for the circuit's implementation, as TTL has a lower propagation delay, and CMOS has a lower power consumption.

Commented [RB28]: Strong evidence of reading.

References

Electronics tutorials. *The JK Flip Flop*. [Online].

Available at: https://www.electronics-tutorials.ws/sequential/seq_2.html [Accessed: 14/04/21]

Learn about electronics. *D type Flip-flops*. [Online].

Available at: <https://learnabout-electronics.org/Digital/dig53.php> [Accessed: 14/04/21]

Electronics Hub (2015). *Flip-flop conversions*. [Online].

Available at: <https://www.electronicshub.org/flip-flop-conversions/> [Accessed: 16/04/21]

All About Circuits. *Logic signal voltage levels* [Online].

Available at: <https://www.allaboutcircuits.com/textbook/digital/chpt-3/logic-signal-voltage-levels/> [Accessed: 21/04/21]

Components101 (2019). *An overview of TTL and CMOS ICs and How to Choose Between Them*. [Online].

Available at: [Difference Between TTL and CMOS ICs and How to Choose Between Them \(components101.com\)](https://components101.com/difference-between-ttl-and-cmos-ics-and-how-to-choose-between-them/) [Accessed: 21/04/21]

Texas Instruments (2017). *Logic Guide*. [Online].

Available at: https://www.ti.com/lit/sg/sdyu001ab/sdyu001ab.pdf?ts=1621490066943&ref_url=https%253A%252F%252Fwww.google.se%252F [Accessed: 21/04/21]

Bolton, W. (2015). *Mechatronics: Electronic control systems in mechanical and electrical engineering*. 6th ed. New York: Pearson Education.

Hambley, Allan R. (2013). *Electrical Engineering Principals and Applications*. 6th ed. New York: Pearson Education.

Hughes, E. (2016). *Electrical & Electronic Technology*. 12th ed. New York: Pearson Education.

Bibliography

Wat Electronics (2021). *What is Multiplexer and types of multiplexing techniques*. [Online]. Available at: <https://www.watelectronics.com/what-is-multiplexer-and-types/> [Accessed 18/04/21]

Electronics for u (2020). *Basics and Overview of Flip Flops*. [Online]. Available at: <https://www.electronicsforu.com/technology-trends/learn-electronics/flip-flop-rs-jk-t-d> [Accessed: 20/04/21]

Tutorials Point. *Boolean Algebra*. [Online]. Available at: https://www.tutorialspoint.com/computer_logical_organization/boolean_algebra.htm [Accessed: 23/04/21]

Kuphaldt, T. (2012). *Lessons in Electric Circuits Volume IV*. London: Koros Press.

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	
You have used a wide range of literature including books and relevant website to provide supporting evidence for your explanations throughout the assignment. Referencing conventions are consistent and right, however, at times, text citation needs attention.	68% 10	6.8
2. Knowledge and Understanding Skills	30	
Your work reveals your exceptional understanding and knowledge of the basic underlying concepts such as Boolean algebra, K-maps, De- Morgan's theorem, flip-flops and logic families along with their application in real life situations. You have used appropriate techniques and theories to complete all set tasks, working of each task is clearly shown and is well supported with literature reading. All the calculation for Boolean algebra were clearly laid out with excellent evaluation of results, however, the importance of timing diagrams in sequential logic could be provided in more detail. Your way of explaining the functionality of each circuit by taking few cases into account and relating it with real life applications is commendable. Overall, your work clearly shows your excellent ability to apply the gained knowledge in real-life applications.	83% of 30	24.9
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. Most of your work is supported by a deep evaluation with proper justification.	77% of 10	7.7
4. Practical Application Skills	40	

You have demonstrated an exceptional ability to perform simulations and practical work by verifying the simulated and practical results against your theoretical findings and relating it to the real-world situations. All the practical and simulated results are correct, very well presented and explained. The evaluation of the results supported by the improvements in future demonstrates your exceptional aptitude to interpret and evaluate the practical results for making valid judgements. The context of application is also well considered and insightful. Well done Oliver.	90% 40	36
5. Transferable Skills for Life and Professional Practice	10	
Your work is accurate, coherent, fluent, well-structured and organised, demonstrating your excellent ability of communicating your ideas accurately and reliably with proper arguments in place to support your findings. Usage of equation editor and providing the supporting Multisim screen shot for each task is worth appreciation. Well done.	80% 10	8

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)		83.4%
	Up to 1 week late (40% Max)		
	Over 1 week late (0%)		