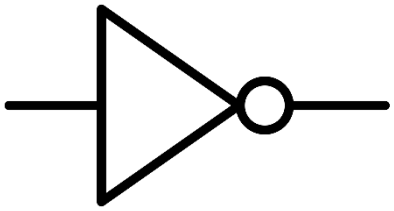
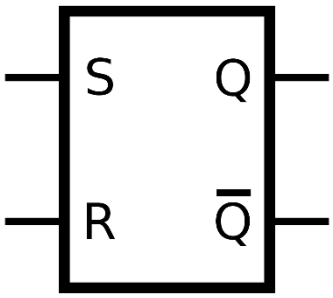




BTEC

2022

Electronic devices and circuits

Digital electronic devices ad circuits

By:

Mohammed Ghalib Al-Hosni | 1615 | 7A

One combinational logic circuit • Two sequential logic circuits

Contents table

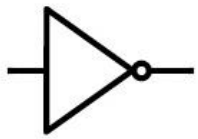
Logic gates

What are logic gates?

Logic gates are digital components, meaning they either output the maximum possible voltage or none, with the reasoning behind whether or not a voltage is outputted being dependent on a Boolean expression that each logic gate possesses, which is an expression that allows logic gates to know the required inputs that have to be provided into their individual pins in order for them to provide a HIGH output (outputting the maximum voltage), with each type logic gate having its own Boolean expression that is used in determining its output relative to its inputs.

Types of logic gates

NOT gate



The not gate takes in only one input and it simply inverts the type of signal that is has been provided, in that if it is provided with a HIGH signal and vice versa. Therefore, the Boolean expression that utilizes is $Y = \overline{A}$

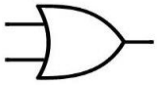
A	Y
0	1
1	0

AND

The AND gate outputs a HIGH signal only when all of the given inputs are HIGH as well, as such, it uses the following as its Boolean expression $Y = A \cdot B$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

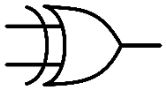
OR



An OR gate checks whether or any of its inputs are HIGH, and if so, the output will also be HIGH, consequently, the Boolean expression used is $Y = A + B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

XOR



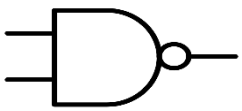
The XOR gate outputs a HIGH signal only when the total number of HIGH inputs is odd, which is indicated using the following Boolean expression: $Y = A \oplus B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

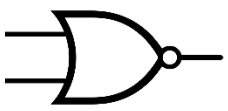
Inverted logic gates

Inverted logic gates are logic gates that combining one of the three major logic gates that were just shown with a NOT gate and is represented by adding an "N" before the original gate's name, and adding a circle in front of the output pin in the logic gate's symbol. If the AND gate were to be taken as an example, the inverse of it would be the NAND gate, and its output properties are the exact opposite to that of AND, meaning wherever an AND gate outputted a HIGH signal, the NAND gate would output LOW, and vice versa. The same principle is utilized for both the NOR and XNOR gates as well.

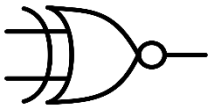
NAND



NOR



XNOR



Logic families

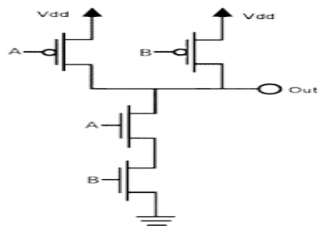
What are logic families?

*Logic families are the different circuits and configuration found within logic gates which determine the Boolean expression that each given logic gate would follow. Said logic families constitute of a number of transistors, diodes, and resistors. Although multiple different logic families can produce the same type of logic gate, each of which utilizes a different methodology when doing so and therefore come with their separate specifications. Overall, around 8 major logic families are available, however, the most prevalent ones seen in the industry are the **CMOS** and **TTL** logic families, and therefore they are the ones that are going to be discussed in the following section.*

Complementary Metal Oxide Semiconductors

Complementary Metal Oxide Semiconductor logic family (abbreviated as CMOS) is a unipolar family, meaning that they utilize transistors which only depend on a single type of charge in carrying electrons through it, with said charge being either excess electronics or electron holes. Therefore, the type of transistors that can be seen withing CMOS circuits are Field Effect Transistors, specifically of type MOSFET, additionally, the complementary in the name stems from the complementary fashion at which the transistors work, in that whenever a transistor turns on the other turns off. The main characteristics of this logic family is that it is possesses great noise immunity and a single CMOS gate is able to transmit its signal to around 50 other logic gates given that they are also of type CMOS. This type of logic can be usually seen in 4000, 74C, 74HXX, and 74AXX ICs.

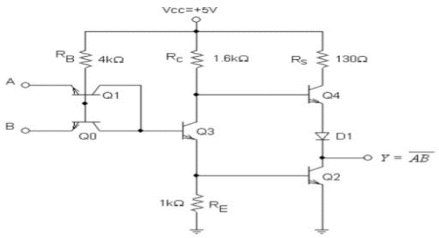
AND gate made using CMOS



Transistor-Transistor Logic

Transistor-transistor logic gate family (abbreviated as TTL) is opposite to that of the CMOS logic family, in that TTL ICs utilize bipolar transistor for their function, meaning the transistors seen within TTL circuits are Bipolar Junction transistors, transistors which use both excess electronics and holes in order for electrons to move through the transistor. The reason that the TTL logic family is called Transistor-Transistor logic is because both the amplification and logic gating processes are done with the use of BJT transistors. The TTL logic family is known for being very immune to electrical damage compared to other logic family on the market, they dissipate a lot of power, and that they possess low output resistance/impedance. This logic family is particularly seen in the 74XX family of ICs, specifically, the 74LS, 74F, 74AS, and 74ALS.

AND gate made using TTL



Characteristics		
Specification	**TTL**	**CMOS**
Supply voltage	5V	3.3V
Propagation	1.5-30ns	1-210ns
Fan out	10	>50
Noise immunity	*Varies but generally strong*	*Average but generally very strong*
Clock rate	35MHz	10MHz

Reference: **Shet, R**. (2019, December 11). *Difference between TTL, CMOS, ECL and BiCMOS Logic Families*. Retrieved from technobyte: https://technobyte.org/logic-families-ttl-cmos-ecl-bicmos-difference/

One combinational logic circuit

Part A – Task i

Construct a schematic diagram and simulate the operation of the logic circuit using standard logic gates.

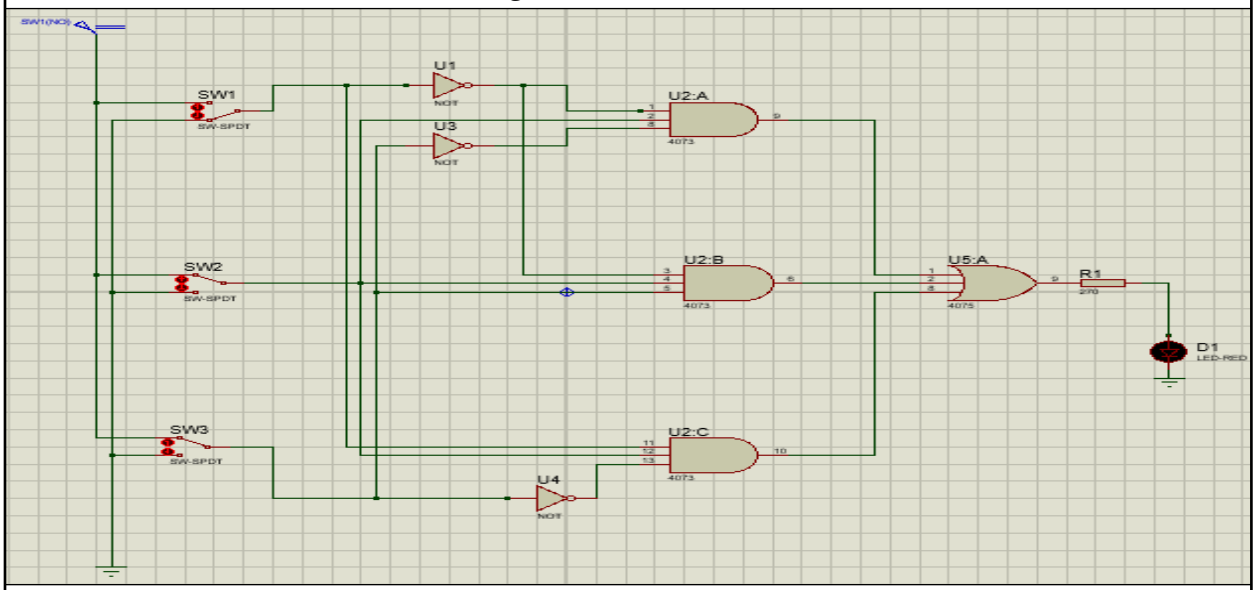
Truth table

The shown truth table is taken from chemical reaction monitoring system, with A being HIGH if the flow rate of a liquid in liters per second is higher than 5, B being HIGH if the temperature is lower than 30°C , and C if the atmospheric pressure is lower than 1 atm.

A	B	C	Y
0	0	0	0
0	1	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

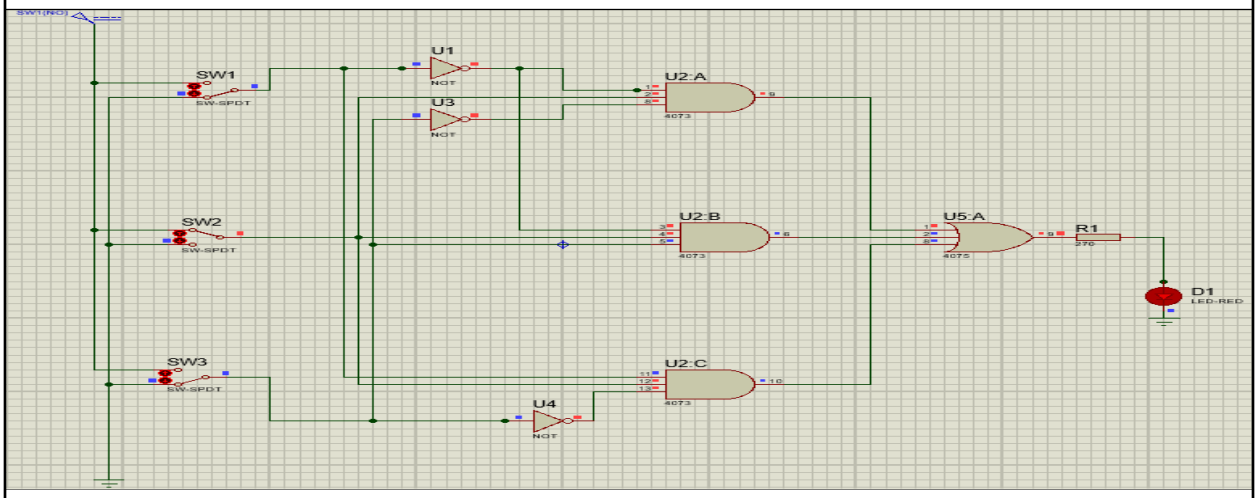
$$Y = \overline{A}.B.\overline{C} + \overline{A}.B.C + A.B.\overline{C}$$

Original circuit schematic



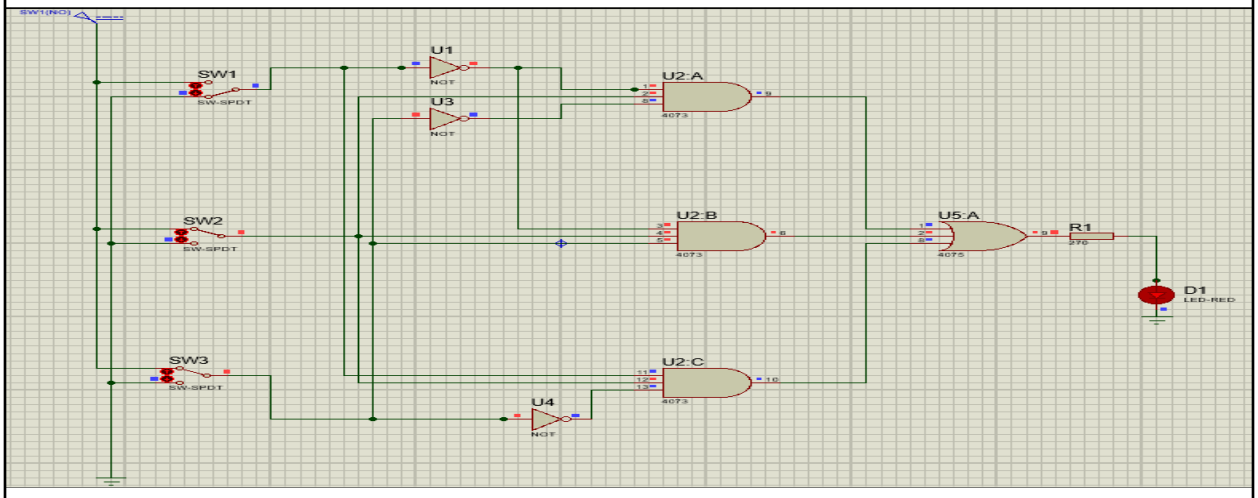
The way that the shown logic gate circuit works is that the main power or signal that goes through them comes from the switches that can be seen on the left. The first switch controls the pin 1 of all the AND gates, with said pin going through a NOT gate for the first and second AND gate. The second switch goes to pin 2 of all AND gates and it does not employ a NOT gate, meaning if the second switch were to be on, the second pin would be HIGH for all given AND gates. Finally, the third switch is connected to pin 3 of all AND gates and it goes through a NOT gate for both the first and third gate. Lastly, all AND gates are connected to an OR gate which powers an LED.

010



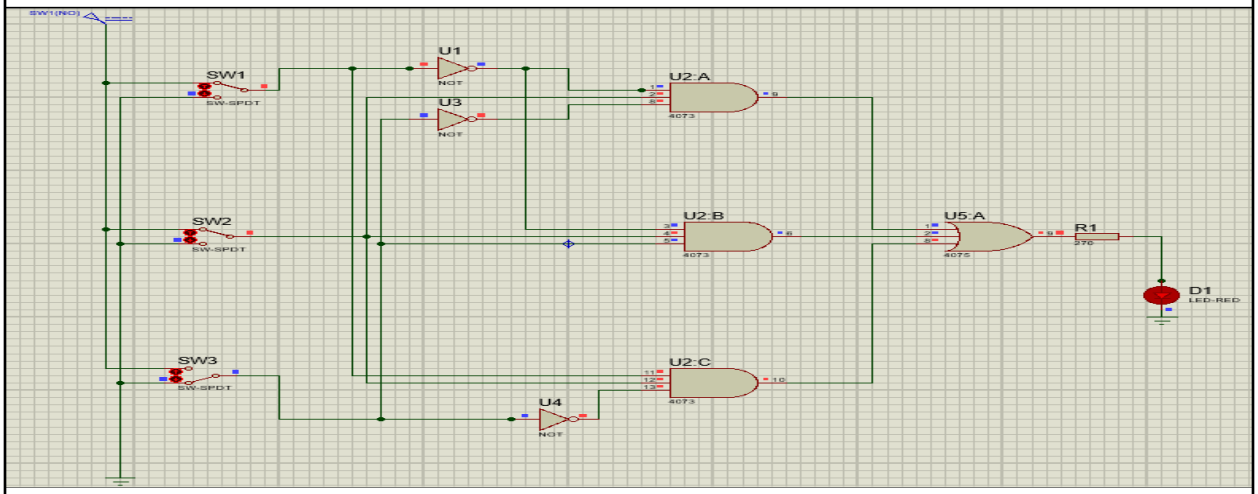
The shown circuit has the first switch in LOW, meaning the signal would be HIGH for pin one in the first and second AND gates as it would initially go through a NOT gate. The second switch is kept in HIGH and therefore – as started earlier – all the second pin for all the given AND gates would be HIGH since switch two does not pass through any NOT gates. Lastly, pin three is kept at LOW which consequently causes pin 3 of the first AND gate to become HIGH as it goes through a NOT gate. Considering all that was mentioned, the first AND gate would be HIGH which in consequence causes the OR gate to be HIGH as well, and the LED to turn on.

011



The first given switch is LOW, therefore the pins 1 for AND gates one and two are HIGH as the signal goes through a NOT gate. The second switch is given as HIGH, and therefore pin 2 of all the AND gates would be high due to the lack of NOT gates that the signal passes through. To end, the third switch is also given as HIGH, however, the signal that passes through the third switch has to go through two NOT gates, with them being attached to pin 3 for AND gates one and three, meaning the HIGH signal would actually only go to pin 3 of AND gate two. To conclude, this means that the given conditions of "011" line up perfectly in order to keep AND gate two HIGH, and due to it, the OR gate would also be HIGH, and consequently the LED would turn on.

110



The first switched is sends in a HIGH signal, although only pin 1 from AND gate three stays at HIGH as the signal passes through a NOT gate for two other AND gates. The second switch is also given as HIGH, and therefore pin 2 of all the given AND gates remain at HIGH as well since – as stated earlier – there are no NOT gates switch two's signal passes through. Finally, the third switch is LOW, however pin 3 for both the first and third AND gates are HIGH due to the signal passing through a NOT gate. To conclude, the given arrangement of signals causes all pins for the third AND gate to be HIGH and in consequence causes the AND gate to output a HIGH signal, and thus causing the OR gate to be HIGH which turns on the LED.

Part A – Task ii

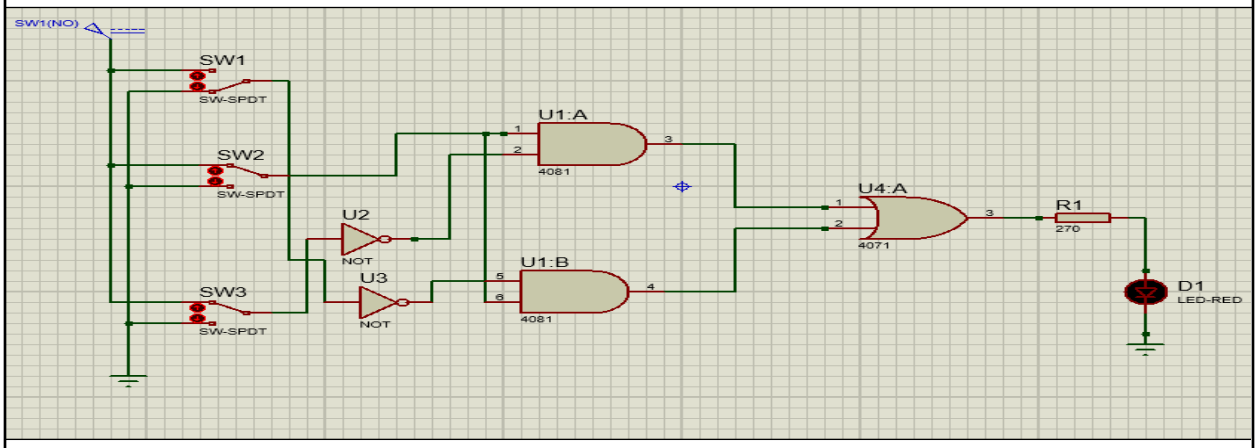
Minimize the logic circuit using a Karnaugh Map, and then contract a schematic diagram, and simulate the operation of the minimized logic circuit, using standard logic gates.

Karnaugh Map

AB / C	00	01	11	10
0	0	1	1	0
1	0	1	0	0

$$Y = B.\overline{C} + \overline{A}.B$$

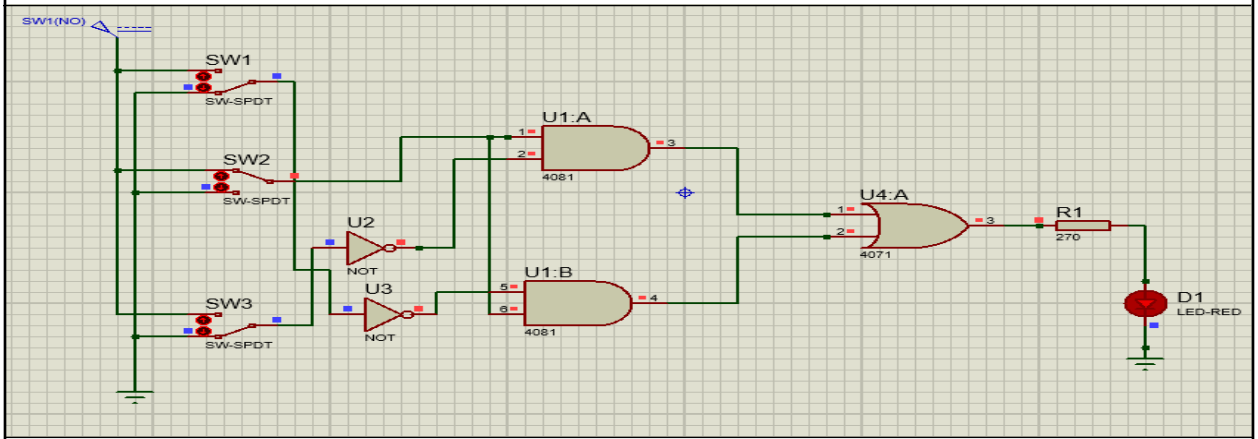
Simplified circuit schematic



As is noticeable, after utilizing the Boolean expression gathered from the Karnaugh Map's solution, the overall complexity and size of the circuit drastically decreased as only two AND gates that contain only two pins are utilized here, however the circuit still retains the same function that the older one had.

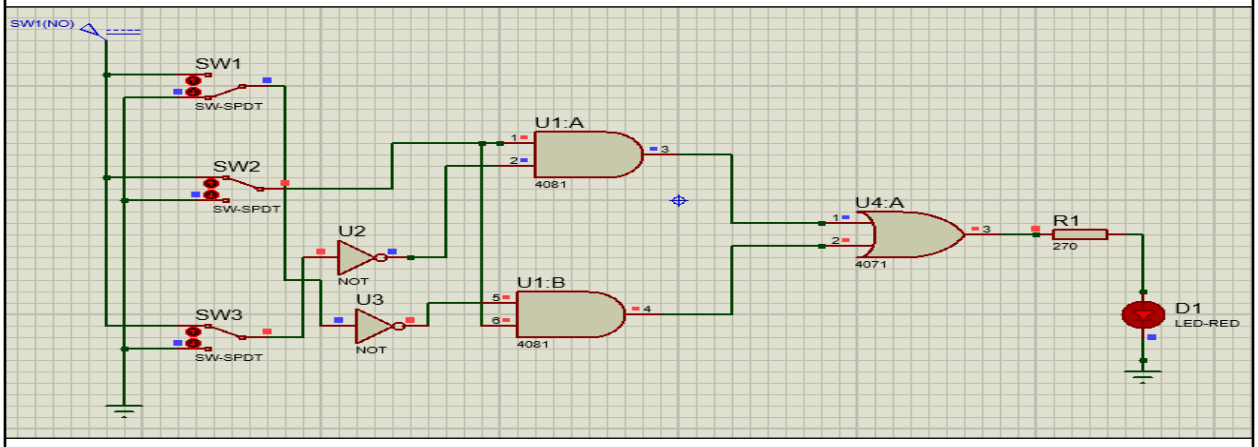
The first switch transfers a signal to NOT gate that is connected to pin 1 of the second AND gate. The second switch's remains as is as it does not go through any NOT gate and it only sends it signal to pin 1 of the first AND gate and pin 2 of the second AND gate. Lastly, the third switch sends a signal to NOT gate connected to the second pin of the first AND gate.

010



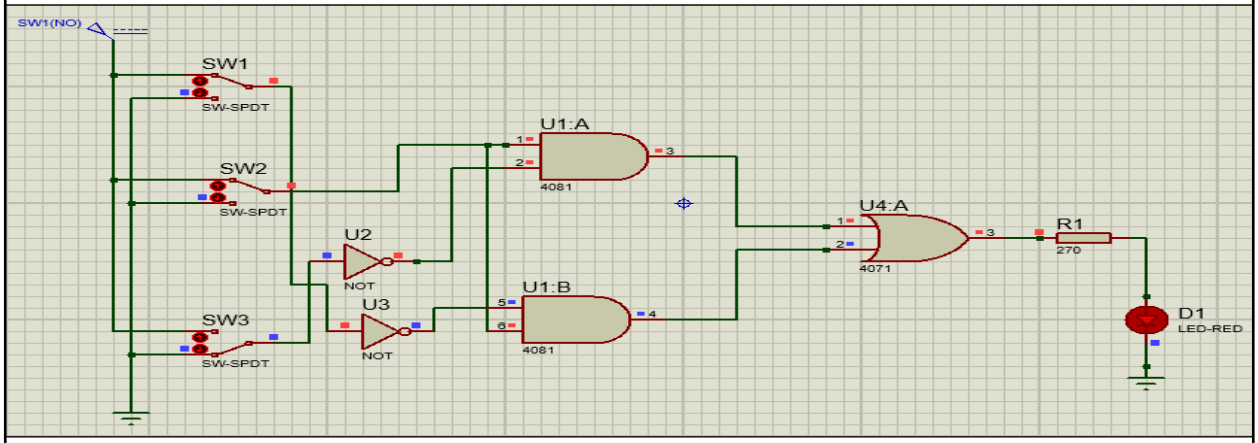
The first switch in the shown circuit provides a LOW signal, and as such pin 1 of the second AND gate is HIGH since the signal goes through a NOT gate. The second switch on the other hand gives a HIGH signal, and therefore both pin one of the first AND gate and pin 2 of the second AND gate are HIGH. Finally, the third given as LOW, however, the signal passes through a NOT and as such pin 2 of the first AND gate is also HIGH. Overall, both AND gates are HIGH and in consequence the OR gate is also HIGH, therefore the LED is turns on.

011



In the shown configuration, the first switch provides a LOW signal, meaning pin 1 of the second AND gate would be HIGH as the signal passes through a NOT gate. The second switch is HIGH and – as stated earlier – it does not pass through any NOT gates and in consequence pin one of the first AND gate and pin 2 of the second AND gate are HIGH. Lastly, the third switch is also HIGH, and as such, pin 2 of the second AND gate is LOW since a NOT gate is positioned between the switch and the pin, meaning the first AND gate outputs a LOW signal while the second one outputs HIGH, and as such the OR gate would be HIGH and the LED will turn on.

110



The shown circuit is the last switch arrangement that provides a final HIGH output. The first switch is HIGH, and therefore pin 1 of the second AND gate is LOW since the signal first passes through a NOT gate. The second switch is also HIGH, and as such pin 1 of the first AND gate is HIGH. Finally, the third switch is kept at LOW, however, it the signal passes through a NOT gate, meaning the second pin of the first AND gate is also going to be HIGH. Altogether, the given configuration makes it so the first AND is HIGH, and as such the OR gate is going to provide a HIGH signal and turn on the LED.

Part A – Task iii

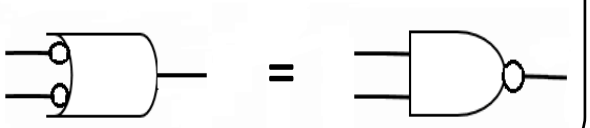
Use De Morgan's Theorems to convert the minimized circuit to use only NAND gates.

De Morgan's theorems

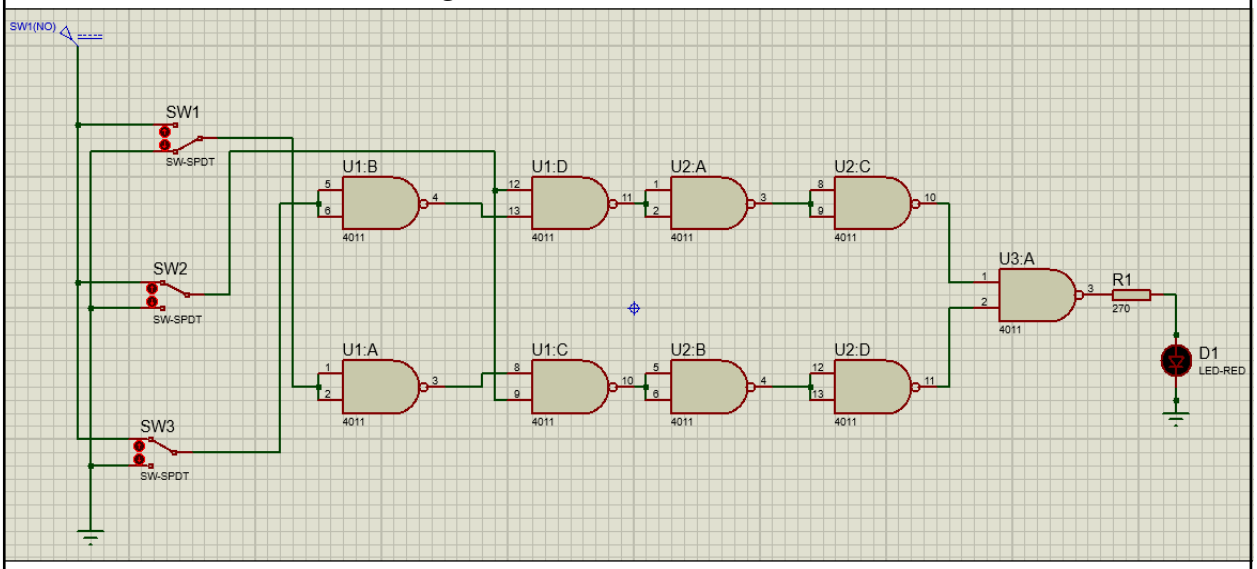
De Morgan's theorems state and represent the correlation between the combination of specific logic gates that can be minimized into one. His initial theorem states that a NOR gate is interchangeable to that of an inverted AND gates, inverted meaning that both pins are connected to a NOT gate. While De Morgan's second theorem states that a NAND gate is equivalent to an OR with inverted inputs.

$$\overline{\mathbf{A}} \cdot \overline{\mathbf{B}} = \overline{\mathbf{A} + \mathbf{B}}$$

$$\overline{\mathbf{A}} + \overline{\mathbf{B}} = \overline{\mathbf{A} \cdot \mathbf{B}}$$

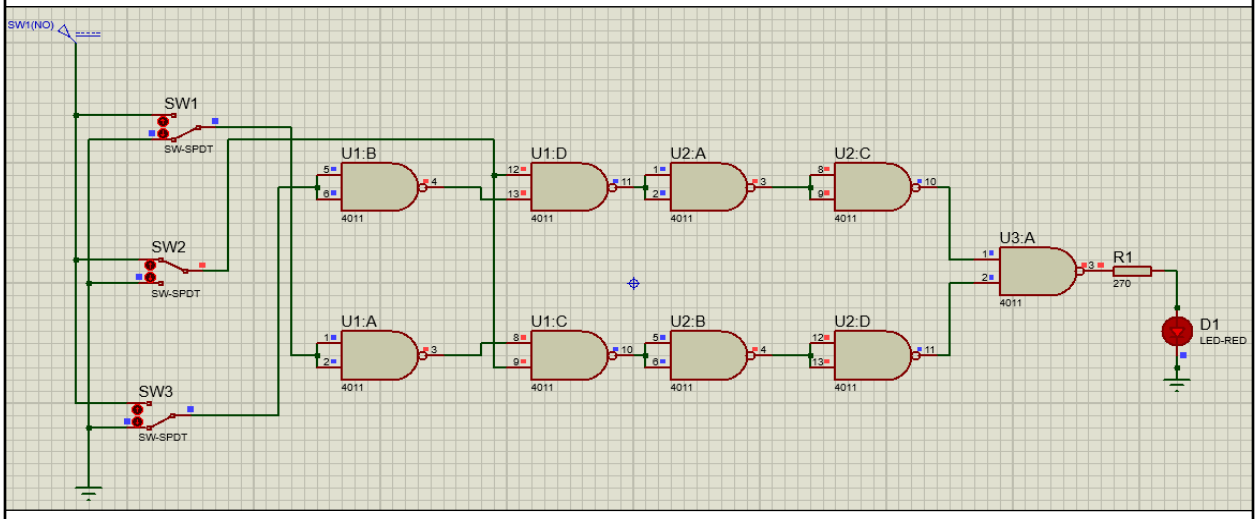


NAND gate combinational circuit schematic



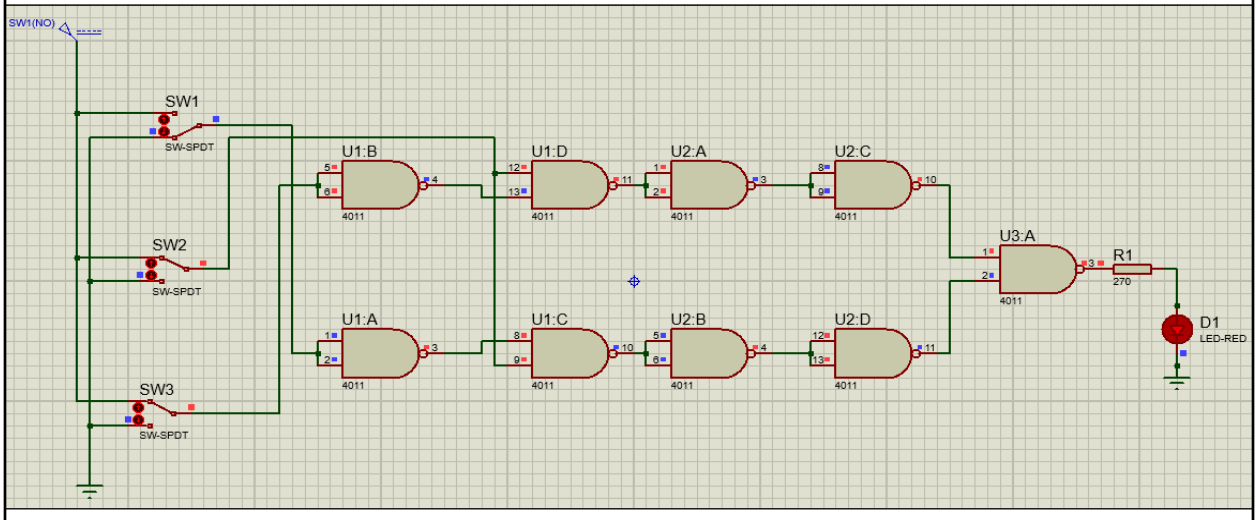
The circuit performs the same exact functions as the ones shown earlier, with the difference lying in that it utilizes De Morgan's theorem by replacing the main OR gate seen in the previous circuit with NAND gates. This works since the shown OR gate has two AND gates which are connected to a NOT gate, which can basically be interpreted as an OR with inverted pins since the stated NOT gate is enough to effect of the AND gate. The main advantage of utilizing De Morgan's theorem in real application is that it decreases the overall number of individual components that would be required, which in consequence decreases the cost of the IC in which the components are going to be placed in. As can be noticed, there is a number of NAND gates which are simply connected to the previous one, this is done to represent how NAND gates might be placed within an IC in a real-life application. In terms of connections, the first switch powers the entire second AND gate and the third switch powers the entire first one, while the second switch goes to the unconnected pins of the third and fourth AND gate. After which, which AND gate simply powers the entirety of the one following it, meaning that if NAND gates 3 and 4 both give off a HIGH signal, the LED will not turn on since the last NAND gate would switch the signal back to LOW → (HIGH, LOW, HIGH, LOW)

010



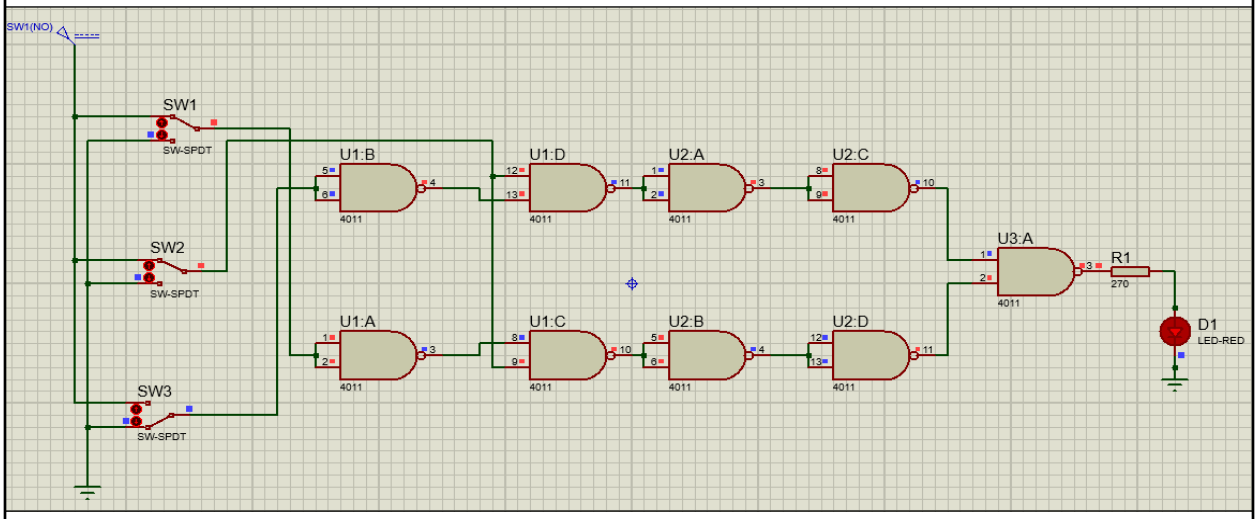
The shown circuit has the first and third switches off, meaning that both the first and second NAND gates would output a HIGH signal as it is NAND, and not an AND gate, however, due to switch 2 being LOW, both the third and fourth NAND gates would output a LOW signal, however, due to the group of NAND gates following gates 3 and 4, the signal would go through a series of changes that causes it to end up being HIGH (LOW, HIGH, LOW, HIGH), and as such, the LED connected to the output of the final NAND gate would turn on.

011



In this circuit, switch one is the only one that does not provide any signal, as such, the second NAND gate is one, and in consequence to that, the fourth NAND gate would receive two HIGH signal due to the signal being inverted. However, the third NAND gate would receive one LOW and one HIGH input, with the HIGH one being from switch two and the LOW from the first NAND gate. Therefore, pins three and four would output opposite signals, meaning the final NAND would receive one HIGH and one LOW signal, and therefore output a HIGH signal and cause the LED to turn on.

110



The shown circuit represent the results when switches one and two are in a HIGH state. Switch one being on cause the second NAND gate to output a LOW signal, while the first one outputs a HIGH signal since the switch supplying it – switch 3 – is LOW. As such, the third NAND gate would provide a LOW signal since both of its pins are HIGH, which comes from switch 2 and NAND gate one. In terms of the fourth NAND gate, it will be providing a LOW HIGH signal since one of its pins – the one being supplied from the second NAND gate – is outputting a LOW signal. And therefore, due to the output signal of both the third and fourth NAND gates being the opposite, the same circumstance as the one that occurred in configuration 011 would occur, in that the final NAND gate would get one pin as HIGH and the other as LOW, which consequently causes it to output a HIGH signal and turn on the LED

Part A – Task iv and v

Build a prototype of your final circuit and verify that it meets the conditions in the truth table, and then compare the results from theory, simulation, and measurement. Also simulate the circuit designed in (iv) to demonstrate how the circuit can shift data from left to right or right to left.

Results

Both simulations and practical circuit resulted in the same circuit configuration, with the truth table also having the same results as the ones deuced from the simulation.

The appendix of the results for the practical build of the 3-bit shift register circuit is provided at the end of the report. (Appendix A)

Section 1 references

Electronics Tutorials. (2022). *DeMorgan's Theorem.* Retrieved from electronics-tutorials: https://www.electronicstutorials.ws/boolean/demorgan.html#:~:text=DeMorgan%27s%20first%20theorem%20states%20that,Complement)%20and%20OR´ed.

Two sequential logic circuits

What are electrical flip flops?

Electrical flip flops are digital devices that utilize logic gates to perform their function. Just like any other digital device, flip flops output only HIGH or LOW signals depending on whether or the input signals that the component receives corelate with a certain Boolean expression. However, the major characteristic of flip flops is their ability to retain and store certain HIGH or LOW signals from prior control inputs, allowing for much further versatility and applications that could utilized the component.

SR Flip flops

Component brief

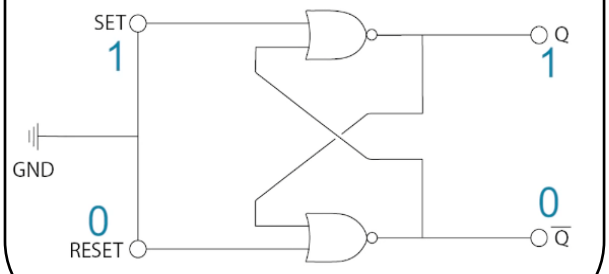
SR flip flops are the rudimentary electrical flip flops in which all other flip flop variations derive from. Although SR flip flops would not be utilized in the following brief, this section will help in setting the basis for which all other flip flops that are going to be discussed will relate back to.

An SR flip flop is comprised of four major pins, Q and QB which are the output pins, and SET and RESET pins which are the control pins that allow the user to manage the output value.

Q & QB	***Set***	***Reset***
The Q and QB (pronounced as Q bar) pins are – as stated earlier – the output seen in any electrical flip flop. The Q pin is the one that is directly affected by both control pins, with the output value being solely dependent on them, however, in terms of the QB pin, it is simply the opposite to that of the Q pin, meaning that if Q were to be in a HIGH state, QB would be in LOW and vise versa.	*The Set pin is used utilized to output a HIGH signal to the Q pin. The way in which this works is that when the Set pin is in its active mode, the Q pin becomes HIGH, however, the major characteristic in the Set pin is that the Q pin retain the HIGH value if SET was active for even one time. Meaning that if SET became inactive after it became active once, Q will remain HIGH.*	*The RESET button is using specifically for resetting and it works in correspondence to the SET pin, in that – as stated – the Q pin remains HIGH even after SET goes back to being inactive, which is when the RESET pin is essential, as activating the RESET pin causes the Q to go back to its default state of being LOW. Activating the pin while Q is on LOW would not cause any change to the value.*

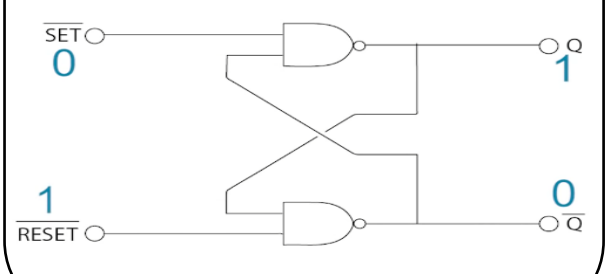
Active-High flip flops

Set and reset are active at HIGH and inactive at LOW and uses NOR logic gates



Active-Low flip flops

Set and reset are active at LOW and inactive at HIGH and uses NAND logic gates

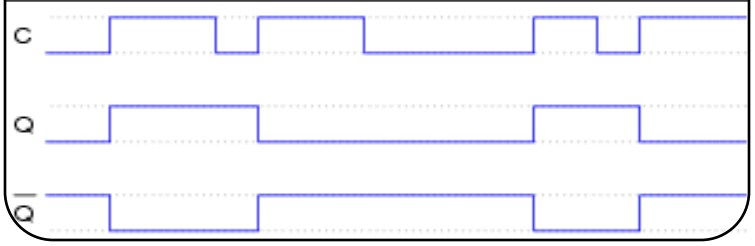


The CLOCK pin

Function

The clock pin – sometimes seen as "CLK or CP" – is one that can be seen in either D-Type or JK-Type flip flops (which are the flip flop types that are going to be discussed in the report). The clock pin acts an alternative method to controlling the output value. The way in which it works is that each time the clock pin is activated, it toggles that Q pin, meaning that the first time the clock pin is seen active, the Q pin will become HIGH, while the second time the clock pin is activated, the Q pin will be LOW, and vice versa for the QB pin. Clock toggles the Q value at its positive edge in Active-High flip flops and the opposite is true in Active-Low flip flops. The Clock pin becomes nullable and is overridden whenever the Set or Reset pins are activated.

Signals graph



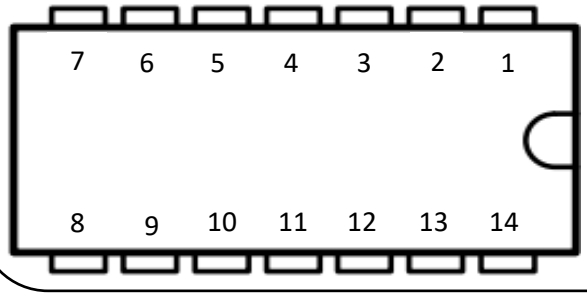
State table

C	0 – 1 – 0 – 1 – 0 – 1 – 0 – 1
Q	0 – 1 – 1 – 0 – 0 – 1 – 1 – 0
QB	1 – 0 – 0 – 1 – 1 – 0 – 0 – 1

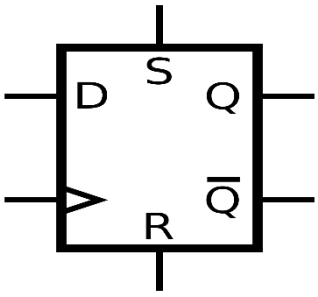
A) 3-bit shift register – D-type flip flops

74HC74 dual D-Type flip-flop with positive edge trigger flip flop

Pinout diagram



Symbol

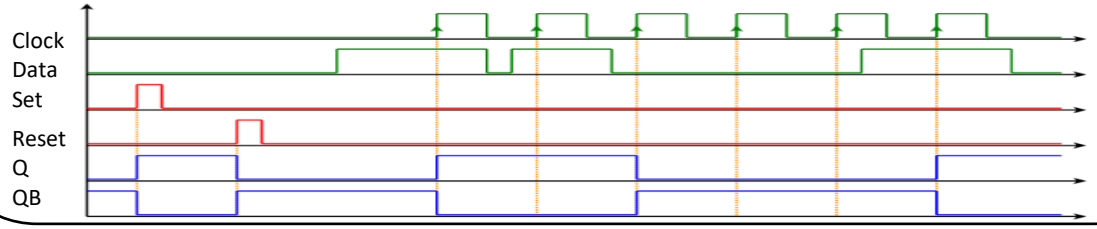


Pinouts

1/13	*RESET: Takes the output back to LOW when it becomes HIGH and overrides the Clock pin*
3/11	*Clock: The clock pins toggles the output pins whenever it becomes HIGH*
4/10	*SET: Causes the output pin to latch onto HIGH whenever Set is HIGH. Is able to override clock*
5/9	*Q: Acts as the output pin. Its value is mainly affected by SET, RESET, CLOCK, and Data*
6/8	*QB: Utilized as an alternative output pin. Its value is always the opposite to that of Q*
14	*Vcc: The pin is connected to the main power source and it's used to turn on the flip flop*
7	*GND: The pin is connected to 0V/ground. Used to turn on flip flop alongside vcc*

2/12: *Data pins*

Pins 2 and 12 – also known as the Data pins – are the defining pins seen in D-Type flip flops, hence the name. The data pins works in correspondence to that of the Clock pin, in that whatever value is seen in the Data would not affect the flip flop whatsoever unless power is going through the Clock pin. What the Data pin does is that it causes the output (Q) to be HIGH whenever it is active and LOW whenever it is inactive, however, it does not do that until the Clock pin is in its positive edge, meaning that if the output is currently at LOW and the Data pin is activated, the Q pin would not output a HIGH signal until Clock is at its positive edge. In precis, if the Data pin is HIGH, Q would also be HIGH, however, Clock pin would initially have to reach its positive edge for the Data signal to be noticed by Q.



Simultaneous activity of Set and Reset

A major flaw that can be seen in D-type flip flops (and SR type) is that when both the Set and Reset pins are in their active states, the Q pin is unable to deduce which state it should be in. Consequently, this causes the Q pin to go into an "invalid" state, in which both Q and QB can be seen having the same signal type, with both being either HIGH or LOW, which is the opposite of the expected function of the Q and QB pins. In addition to that, if the issue reoccurs, it could also lead to the output state for both pins being random, and therefore causing the user to lose control of the component, which could be extremely dangerous in electrical devices with important applications.

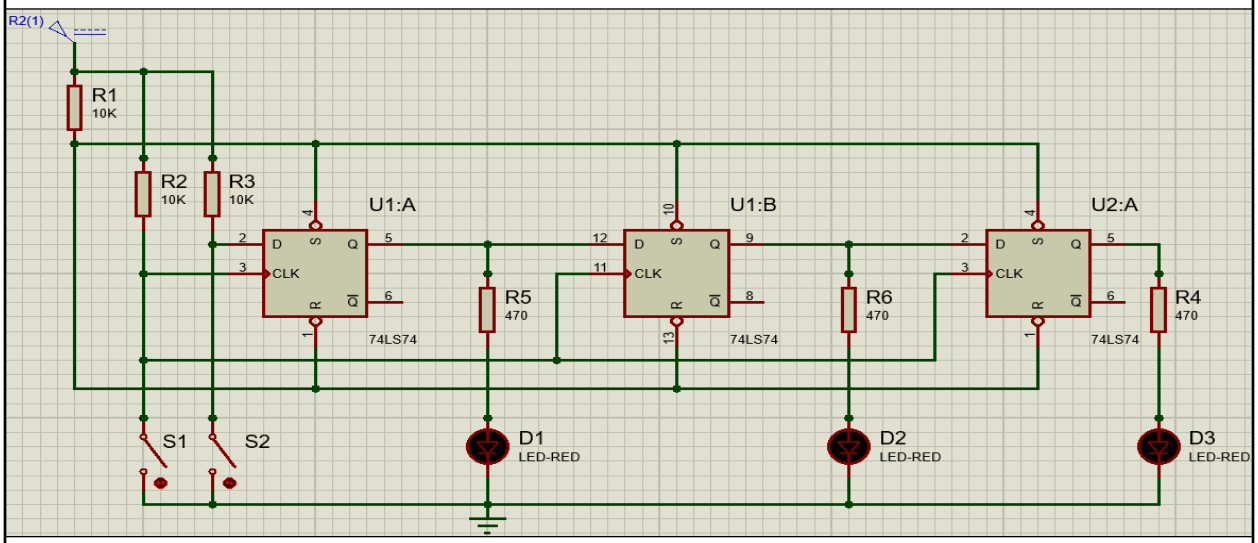
What are shift registers?

A 3-bit shift register a type of circuit that depends on flip flops to operate. The main function of the circuit is to "shift" between one flip to another, in other words, the circuit configurations allows for the activation of a certain number of flip flops to be done in a set chronological order. The stated functionality is done by controlling the clock edge making it so whenever the clock edge is positive, the output of flip flop becomes high, with the output later one being connected to the digital pin of a subsequent flip flop. What this does is that it makes it whenever a flip flip is activated, the second one will be activated as well, and therefore causing the position of the currently active flip flop to "shift" until the clock pin's edge rises ones again to the positive edge.

Part 2A task i

Use the schematic to simulate the behavior of the circuit by setting data values at the D1 input and clocking them through.

3-bit shift register schematic

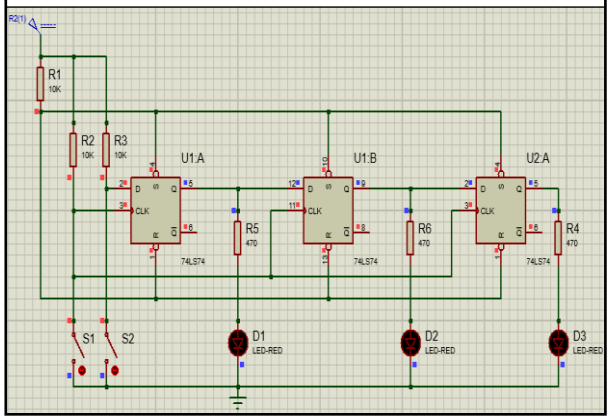


The represented circuit schematic is an example of a 3-bit shift register circuit, 3-bit referring to the fact that 3 flip flops are utilized. In terms of the pin connection for each of the given flip flops, the Data pin of the initial is flip flop is connected to main power source, however, as can be noticed, the Data pins of the subsequent flip flops are connected to the previous flip flop's output pin, which will assist in the previously stated operation of each flip flop activating the subsequent one. The clock of each flip flops is connected to the first switch (S1), meaning whenever the stated switch enables the flow of current, the clock pin will be activated, causing it to turn to its positive edge and activate whatever signal is being sent through the Data pin. Finally, in addition to the output pin being connected to the Data pin of the following flip flop, it is also connected to an LED which will be used to show the operation, however, in practical examples, the LED could be substituted for whatever device is required to operate a given electrical system.

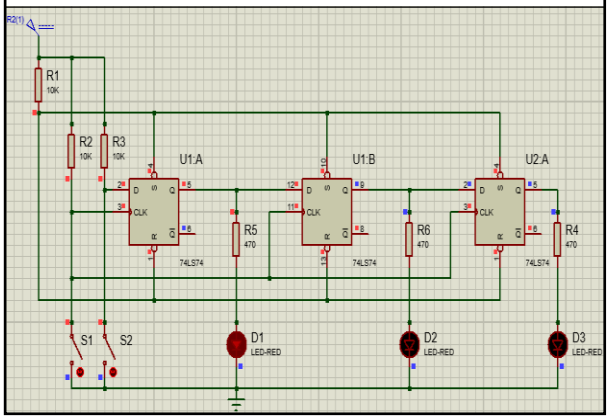
In terms of the operation of the circuit itself, initially, the Data pin of flip flop one is going to be set to HIGH as powers goes through it immediately, after which, when S1 is closed, a signal would go through to the every Clock pin, however only the first flip flop will be affected as it is the only which has a HIGH signal going through its Data pin, which in consequence would cause the output (Q) of the first flip flop to become HIGH, and accordingly the LED will be on and the Data pin of the subsequent flip flop will be HIGH as the output is connected to both of them. Due to the Data pin of the second output becoming HIGH, the following time the switch is closed, it will once again go through all Clock pins, however, this time, Data pin of flip flop two would have a HIGH signal, meaning it would perform the same action that flip flop one did but rather it would do it to pin three. As such, a "shift" in the given flip flops occurs each time the Clock pin turns to its positive edge, and each one of these said shifts is a Pulse Shift. Due to the stated procedure, in this particular circuit each Pulse Shift simply turns on an additional LED.

Pulse shift			
Decimal equivalent	***LED state***		
0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	

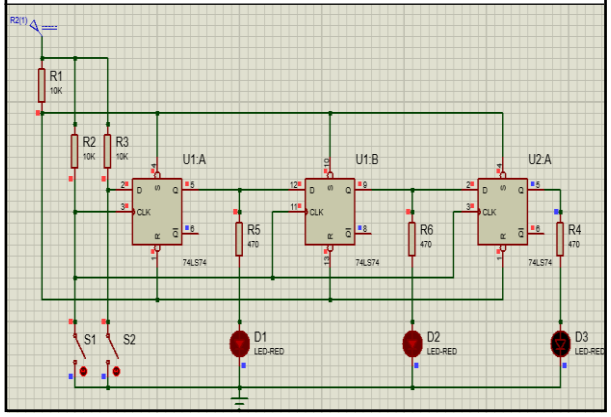
Pulse Shift: 1



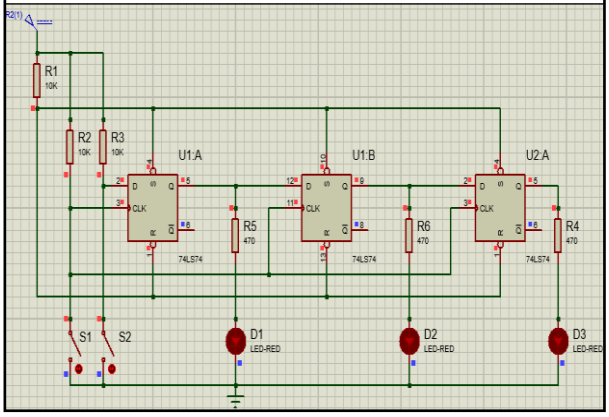
Pulse Shift: 2



Pulse Shift: 3



Pulse Shift: 4



Part 2A task ii

Build a prototype of the circuit and demonstrate that it works as expected, and then compare the results from theory, simulation, and measurements.

Results

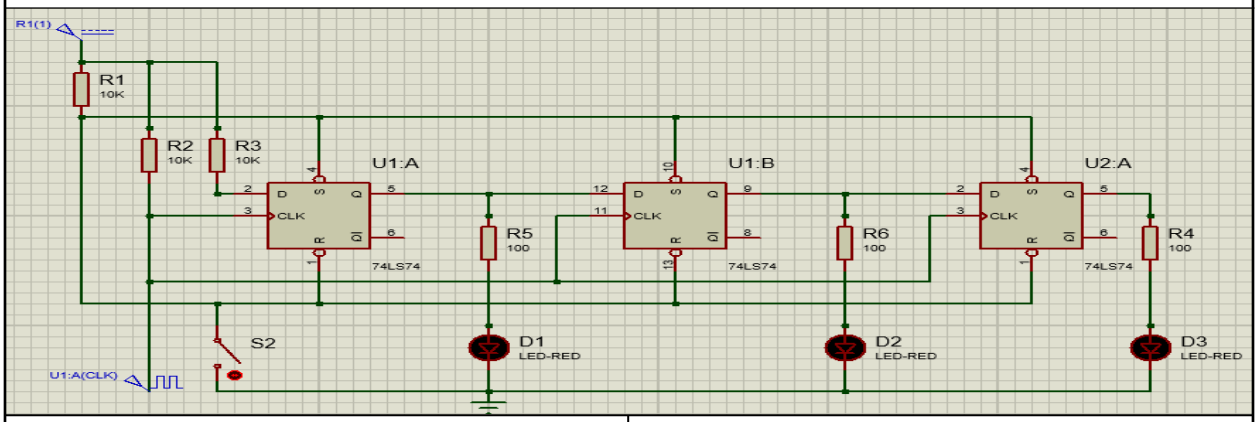
The same results found in the simulation were also gotten during the practical build of the circuit. Although at the time a switch was not available, and a pushbutton had to be used instead, which in certain cases interfered with the function by accidentally allowing two rising edges to pass on to the flip flops and therefore shift twice, however, overall, the same results were deduced and no major comparison would have to be done.

The appendix of the results for the practical build of the 3-bit shift register circuit is provided at the end of the report. (Appendix B)

Part 2A task iii

What practical improvements would you make to the circuit?

Improved circuit

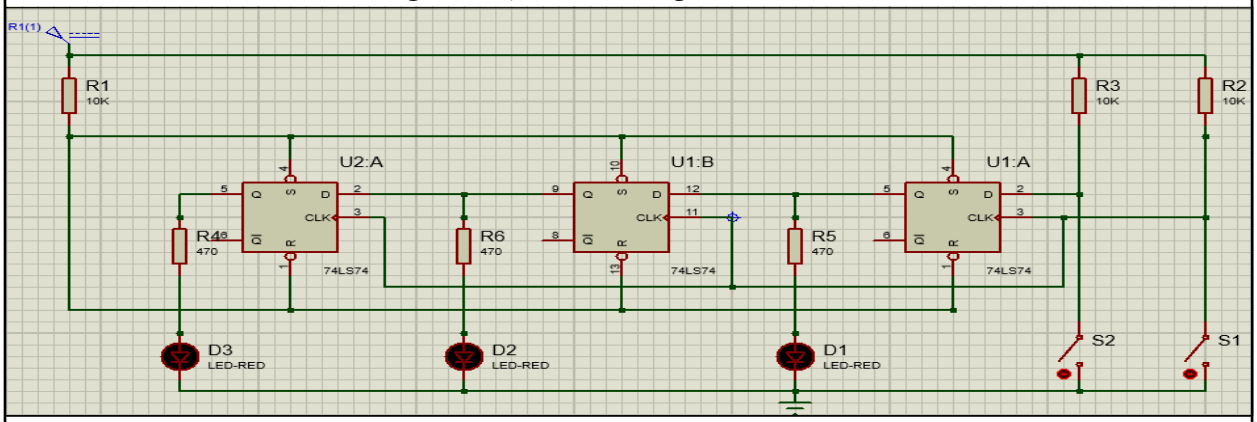


DCLOCK the input signal	***Connect switch 2 to the reset pins***
The replacement of the switch to a DCLOCK assists in simulating how a real-life application using a 3-bit shift register would be. What the DCLOCK specifically does is that it provides a set frequency, and therefore automatically incrementing the Pulse Shift.	*Initially, no way of turning off the LEDs after the shift has increments was possible, as such, a major improvement that was added is connecting the second switch to the reset wire, this way, whenever Switch 2 is closed, a signal would be sent to all reset pins which would turn off all the LEDs.*

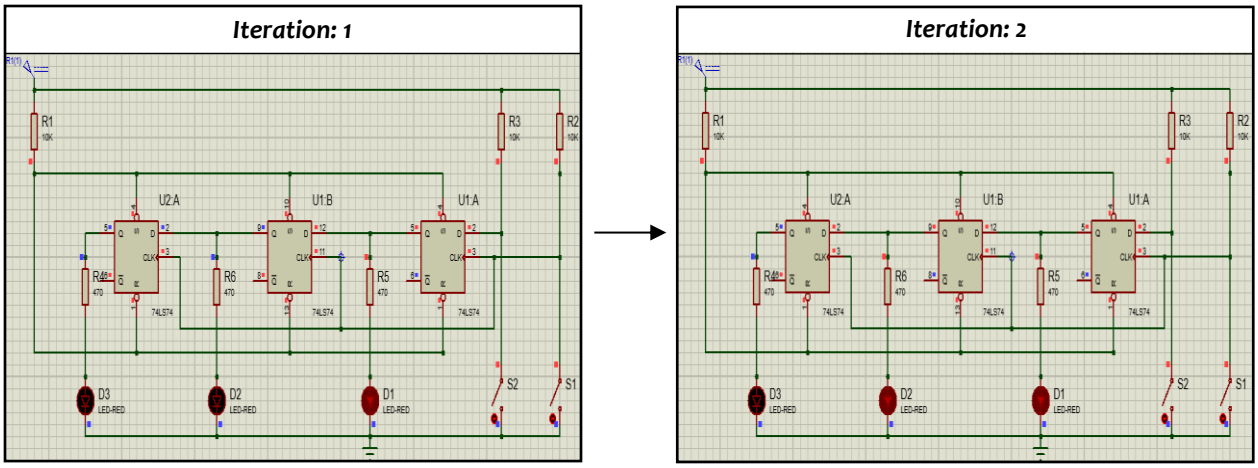
Part 2A task iv

The circuit above shifts data from left to right. Draw a schematic diagram to show how logic gates can be used to enable the data to be shifted from right to left.

Right to left 3-bit shit register schematic



The method utilized in order to flip the phase shift's order to be from right to left rather than the original left to right was by simply rotating the flip flop's so that the clock pin is on the right side, after which also providing the initial signal from the right as well. As such, the initial Data pin that is going to be set to HIGH is going to be the one possessed by the rightmost flip flop, and therefore the initial CLOCK rising edge would specifically affect that flip flop as well, and therefore causing the right LED to turn on first.



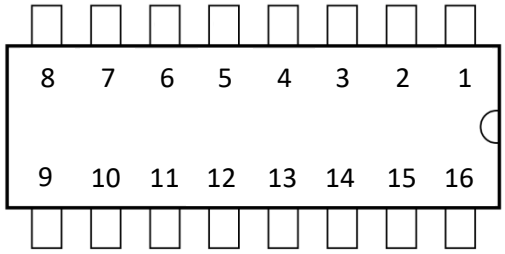
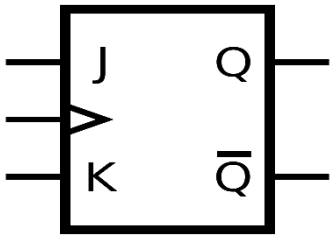
Section 2A References

Element14 (Director). (2019). *How Flip Flops Work - The Learning Circuit* [YouTube video].Retrieved from https://www.youtube.com/watch?v=Hi7rK0hZnfc

Element14 (Director). (2020). *How Shift Registers Work - The Learning Circuit* [YouTube video]. Retrieved from https://www.youtube.com/watch?v=lK_LKo3MwgQ&t=207s

B) 3-bit asynchronous counter – JK-type flip flops

4027B dual JK flip flop with positive edge

Pinout diagram	***Symbol***
	

Pinouts

1/15	*Q: Acts as the output pin. Its value is mainly affected by SET, RESET, CLOCK, J, and K*
2/14	*QB: Utilized as an alternative output pin. Its value is always the opposite to that of Q*
3/13	*Clock: The clock pins toggles the output pins whenever it becomes HIGH*
4/12	*RESET: Takes the output back to LOW whenever its HIGH and overrides the Clock pin*
7/9	*SET: Causes the output pin to latch onto HIGH whenever Set is HIGH. Is able to override clock*
8	*Vcc: The pin is connected to the main power source and its used to turn on the flip flop*
16	*GND: The pin is connected to 0V/ground. Used to turn on flip flop alongside vcc*

5/11: K & 6/10: J

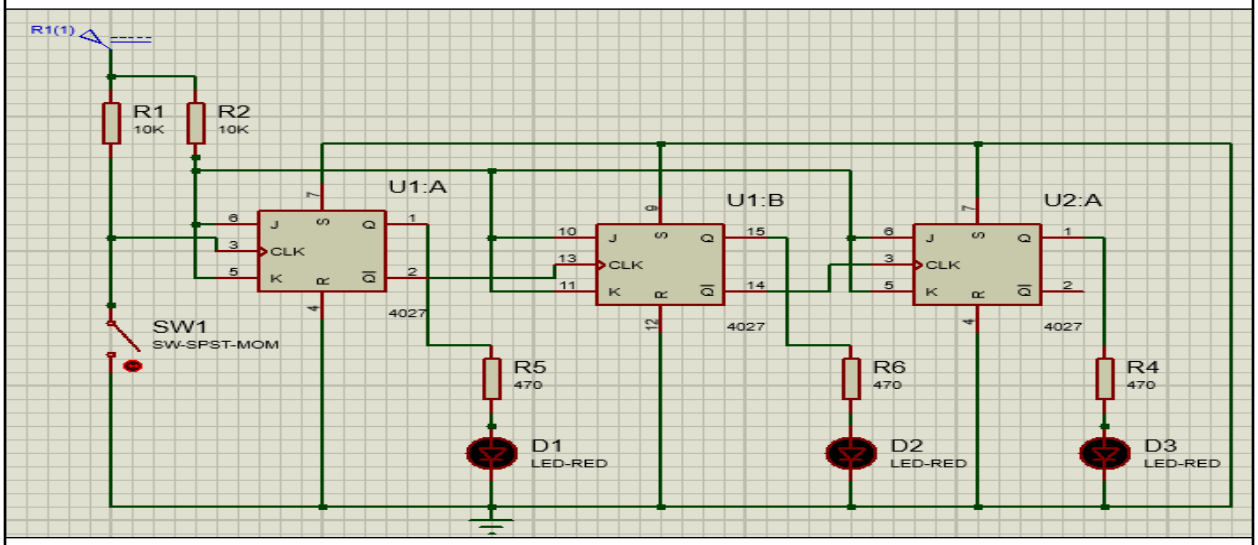
Pins 5, 6, 10, and 11, also known as pins J and K, are the defining pins seen in JK flip flops, they work in the same manner as that to Set and Reset, with J being Set and K being Reset. However, the major difference between the two being the logic gate utilized. In that with the Set and Reset, the pins are connected to two legged AND gates, while J and K pins are connected to two legged NAND gates, with the third pin being the J or K. The main difference that this causes is that – unlike SR or D-Type flip flops – whenever both J and K are high, an invalid state is not activated in the flip flop, but rather this simply toggles Q to the state opposite to its current state, meaning if both J and K became active simultaneously while Q is at LOW, Q would become HIGH, and vise versa if J and K became active while Q is HIGH.

Part 2B task i

Use the schematic to simulate the behavior of the circuit by using S1 as a manual clock.

,

3-*bit asynchronous counter schematic*



The shown circuit represents a 3-bit asynchronous counter, with 3 bit referring to the fact that 3 flip flops are utilized. The specific type of flip flop that is being used here is a JK flip flop. In terms of the circuit layout, a 5 volt power supply runs through a resistor which then simultaneously connect to every J and K pin of the entire circuit, meaning that the output value would toggle each time the clock pin is on its rising/positive edge. The reason the toggling occurs during the rising is due to the specific flip flop that is being used – the 4027B – being of positive edge trigger type. Additionally, the main output pin (Q) of each flip flop is connected to a 470 ohms resistor which powers an LED. However, the secondary output pin, QB, outputs its signal towards the CLOCK pin of the subsequent flip flop, meaning that whenever QB becomes HIGH, the following flip flop's Q pin would be HIGH. Finally, a switch which is connected to the CLOCK pin of the first flip flop can be noticed, which is unlike what was seen in the previously shown circuit, which had the CLOCK pin of each flip flop being connected to one given power supply, however, in this particular circuit, the power goes to the first flip flop's CLOCK pin alone, which is what the "asynchronous" in the name refers to. With regards to the operation of the circuit, initially, all outputs Q pins are kept at LOW, meaning that all LEDs would be off and in addition all QB pins are at HIGH. After which, iteration of the counter will have a set LED state, with said iterations being incremented using the switch powering the initial CLOCK pin, each rise of the initial CLOCK pin's edge will result in the following sequence:

Rising edge 1: *Turning the switch on will send the first flip flop's CLOCK pin to its positive edge, meaning the Q pin will toggle to being on, which consequently will turn the first LED on and cause QB to be LOW.*

Rising edge 2: *The second time the CLOCK pin goes to its rising edge, the first flip flop's Q pin will toggle back to LOW, meaning the first LED will turn off, however, the QB pin will be HIGH, which accordingly triggers the second flip flops' CLOCK pin to become HIGH, and therefore causing its Q pin to be HIGH and the LED to be on.*

Rising edge 3: *During the third rising edge, the first flip flop's output pin will toggle back to HIGH, meaning the first LED will be on. However, this also consequently means that the first flip flop's QB pin – which powers the second flip flop's CLOCK pin – will be in its negative edge, however, due to the flip flops toggling at the rising edge only, the second flip flop's output will not toggle back to LOW as its CLOCK pin simply become LOW, but does not meet the toggling requirements of being at the positive edge, therefore nothing would occur to the second LED, and as such both LED one and two would be on, while currently both the first and second QB pins are at LOW, while the third flip flop never got affected since the second QB pin has not triggered its rising edge.*

Rising edge 4: *During the fourth rising edge of the initial CLOCK pin, the first flip flop's output toggles back to LOW, meaning its QB becomes HIGH, consequently, this causes the same operation to occur for the following flip flop since it had the same configuration as the first flip before the rising of the CLOCK pin. This means that both Q pins are LOW, and that neither first nor second LED is on. However, both QB pins are currently at HIGH, meaning the third flip flop's CLOCK pin has been risen to its positive edge to the second flip flop's QB pin, this consequently causes the third flip flop's Q pin will be HIGH. Overall, this means that the third LED will be the only one on.*

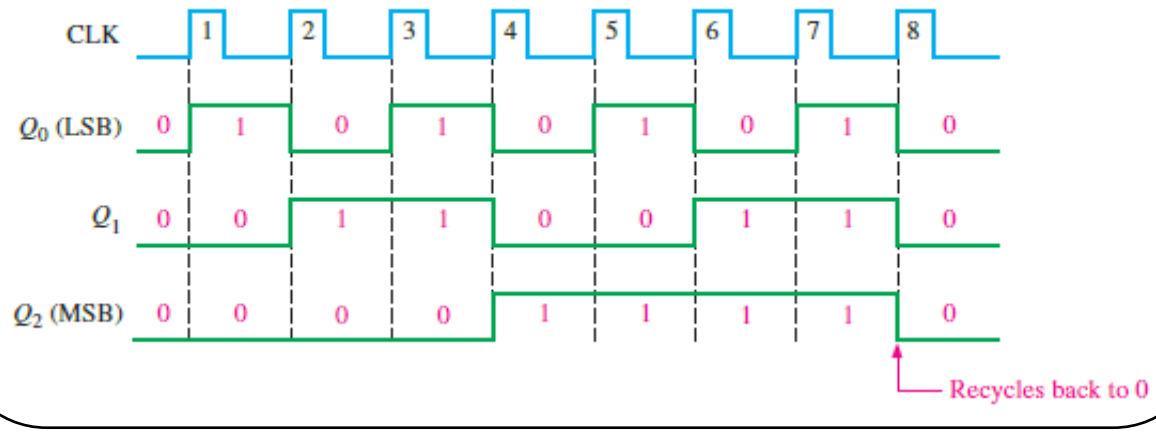
Rising edge 5: *When the initial rises for the first time, the first flip flop will toggle pin's Q pin will toggle back to HIGH, this also means that the signal will not pass on to the following flip flops since the first flip flop's QB pin will be LOW – the opposite to the Q pin – and therefore the second and third LED states will not change, meaning the first and third LEDs will be the ones on.*

Rising edge 6: *For the sixth iteration, the first pin toggles back to LOW, meaning its LED will be off and its QB pin will send its signal to the second flip flop, consequently toggling its output to HIGH and its LED to on. This also causes the second flip flop's QB pin to become LOW, however – just as seen in the third iteration – this does not cause flip flop three's Q pin to toggle to LOW since toggling only occurs during the rising edge, while right now the CLOCK pin simply went to its falling edge. Overall, the second and third LEDs will be the ones seen on.*

Rising edge 7: *During the seventh CLOCK pulse, the first flip flop's output will toggle to HIGH, this means that its QB will be LOW, and therefore no change would occur in the following flip flop's since the second flip flop's CLOCK pin will simply go to its negative edge, meaning no change would occur to the output pin. This means that at this iteration all of the given LEDs will be simultaneously on.*

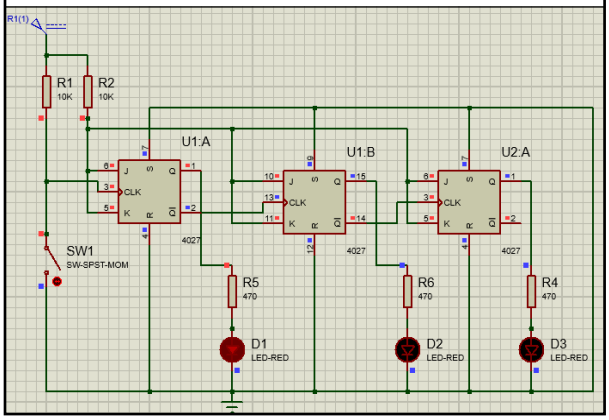
Rising edge 8: *The eighth clock pulse is the final one, which is the case due to the fact that at iteration 7, all LEDs are HIGH, meaning the following clock pulse – clock pulse eight – will cause the first flip flop's Q pin to be LOW, consequently triggering a HIGH pulse from its QB pin, and since the QB pin is connected to the following flip flop's CLOCK pin, the same operation that happened to the first flip flop will happen to the second one, and in consequence also happen to the third one. Overall, this causes all Q pins to become LOW, and therefore turning off all of the LEDs. As a result, this means that the following iterations will repeat the same process as the one just discussed, hence the cycle is repeated.*

Signal graph

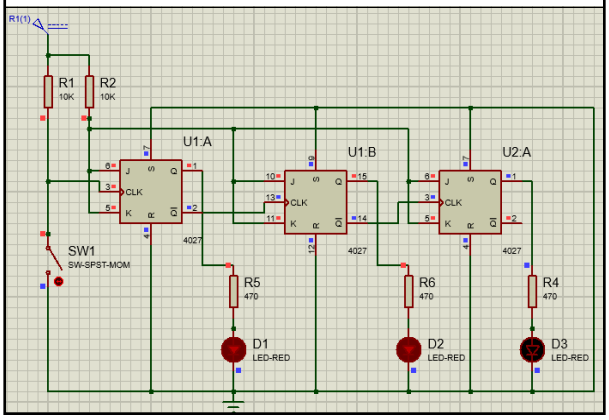


Counter's state			
Decimal equivalent	**LED state**		
0	0	0	0
1	1	0	0
2	0	1	0
3	1	1	0
4	0	0	1
5	1	0	1
6	0	1	1
7	1	1	1
8	0	0	0

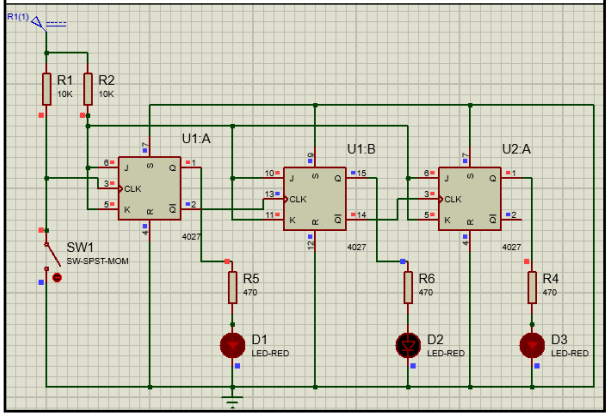
Iteration: 1



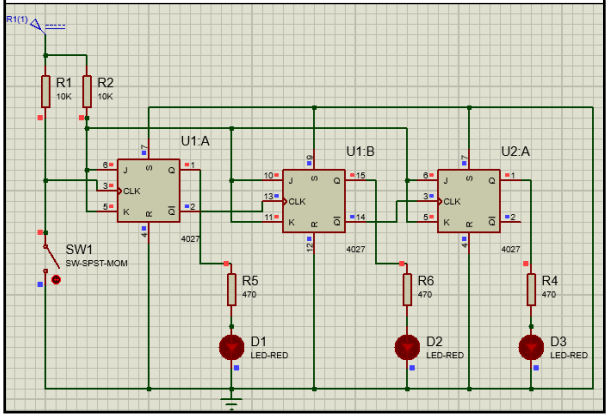
Iteration: 3



Iteration: 5



Iteration: 7



Part 2B task ii

Build a prototype circuit and verify that it works as expected, and then compare the results from theory, simulation, and measurements.

Results

The results of the practical circuit were the same as the ones deduced from the simulation, with the LED states also being given in the same order. As such, no major comparison are to be done.

The appendix of the results for the practical build of the 3-bit shift register circuit is provided at the end of the report. (Appendix C)

Part 2B task iii

What practical improvements would you make to the circuit?

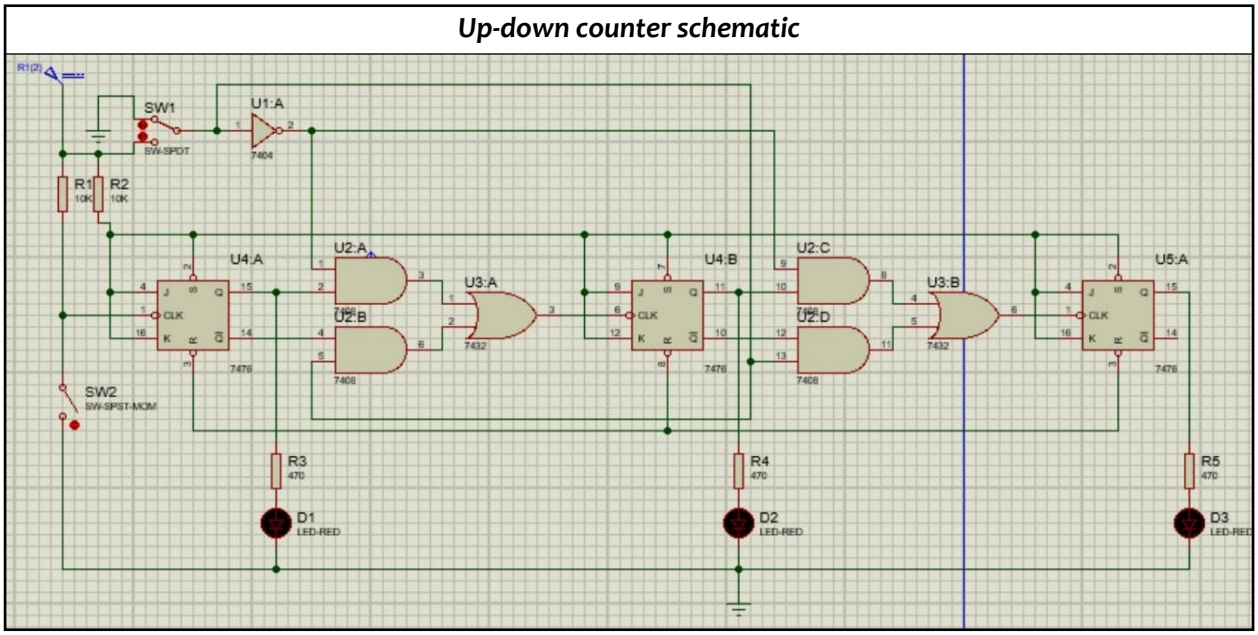
Improved circuit

Possible improvements to be made to the circuit are the same as the ones given for the 3-bit shift register, which were the addition of a DCLOCK, a frequency-based input that provides a set voltage over a certain frequency, which automatically iterates the counter and assists in simulating a more realistic example of the shown circuits that is utilized in practical applications. In addition to that, connecting the second switch to the reset pins would also improve the circuit in that – if required – the circuit operation could be reset back to the first counting iteration.

Part 2B task iv, v, and vi

Draw the schematic diagram to show how logic gates can be added to the up-counter in the diagram to convert it into an up/down counter. Simulate the circuit designed in (iv) and show the results which may include a table. Finally, demonstrate how the circuit can be used as an up or down counter.

Up-down counter schematic



The "up-down" seen in the name of the circuit refers to the fact that the circuit is able to perform but upwards and downwards counting operations, with upwards being what was discussed earlier, in that the outputs depending on the iteration would initially start at 1 0 0 and end at 1 1 1, however, downwards counting does the opposite to that, in that it starts counting at what is generally considered the final iteration – which is 1 1 1 – and the ends at 1 0 0 before repeating the cycle.

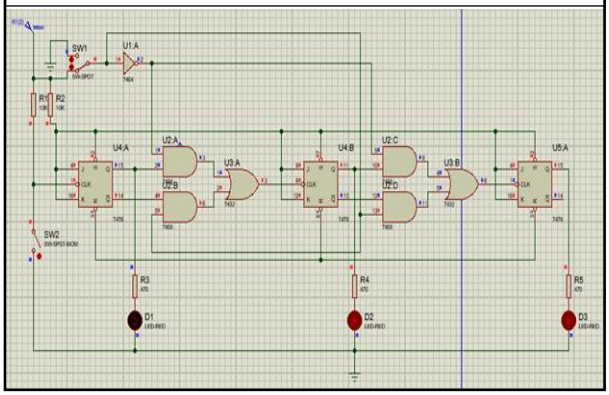
In terms of the circuit's connection configuration, it is relatively similar to the one seen in the initial circuit, with the difference lying in the incorporation of the combinational logic circuit placed in between each flip flop and the additional signal source seen at the top of the circuit, which would given the name of signal "X", which is connected to an SPDT switch, a switch that locks in place until the user turns it back on or off . Regarding the additional power source, it supplies to different wires, one of which includes a NOT gate while the other simply provides the raw signal. Thenceforth, the signal is provided to the stated combinational logic circuit, with the inverted signal going into an AND gate that is additionally being supplied by Q in the other pin, while the raw signal goes into an AND that has its other pin occupied by QB. From there, both AND gates are connected to an OR gate which supplies the subsequent flip flop's CLOCK pin. Furthermore, the same exact configuration is also applied to the second combinational logic circuit that is seen in between the second and third flip flop. It should also be noted that pin Q is still the one providing power to the LEDs in this circuit.

Concerning the circuit's operation, what primarily occurs is that signal X determines whether output Q or QB would be utilized in clocking the following flip flop, in that if switch providing signal X were to be closed, the raw signal would be HIGH while the inverted one would be LOW, which consequently means that the AND gate connected to output Q will have one LOW signal, and therefore it will permanently output a LOW signal even if pin Q were to be HIGH until the switch is opened, and vice versa occurring to the AND gate connected to output QB, in that if QB were to be HIGH, the AND gate will output a HIGH signal, therefore the OR gate would only output HIGH whenever QB is also HIGH, and as such the following flip flop's CLOCK is dependent on QB alone, with the opposite operation occurring if the switch controlling signal X's value were to be open, in that QB's AND gate would be always LOW, while Q's AND gate has the potential of being HIGH if Q were to be HIGH since its other pin has received a HIGH signal as well from signal X's inverted signal.

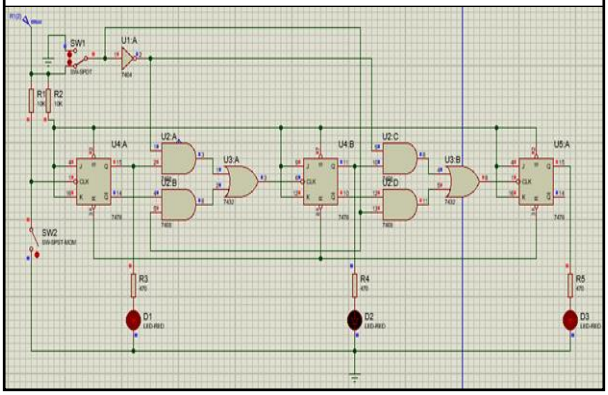
If iteration one was to be taken as an example, during upwards counting, signal X would be HIGH since QB is the output pin required, which can be seen from the initial circuit. During the first rising edge, Q will become HIGH, meaning the first LED would turn on, and QB would be LOW, and therefore the second flip flop's clock pin would not receive any signal since Q's AND does not transmit any signal due to its other pin – which is being provided by signal X – is LOW hence both the second and third LEDs would remain off. However, during downwards counting, the switch sending signal X would be closed, and therefore the inverted signal would be HIGH, meaning here the AND gate connected to output Q would in fact transmit a signal given that Q was also HIGH, accordingly, during the first rising edge, pin Q would be HIGH, which would consequently clock the second flip flop, and therefore cause its pin Q to be HIGH, and due to the second combinational logic circuit following the same configuration, the third flip flop's Q pin would also be HIGH. As such, due to the all LEDs being connected to pin Q, every single LED would be on during the first iteration which during upwards counting would be the seventh and final iteration before the outputs cycle back. And as such, the initial phase has all LEDs as on, and when it comes to the first iteration of the counter, the rising edge would switch the initial flip flop's Q pin back to LOW, and therefore turning off the first LED, however, the second and third one would remain on since during downwards counting QB does not provide its signal to the second flip flop. And as such, the first iteration has the first LED as off while the second and third are on, which is the exact opposite of what was discussed in upwards counting.

Counter's state during downwards counting			
Decimal equivalent	***LED state***		
0	1	1	1
1	0	1	1
2	1	0	1
3	0	0	1
4	1	1	0
5	0	1	0
6	1	0	0
7	0	0	0
8	1	1	1

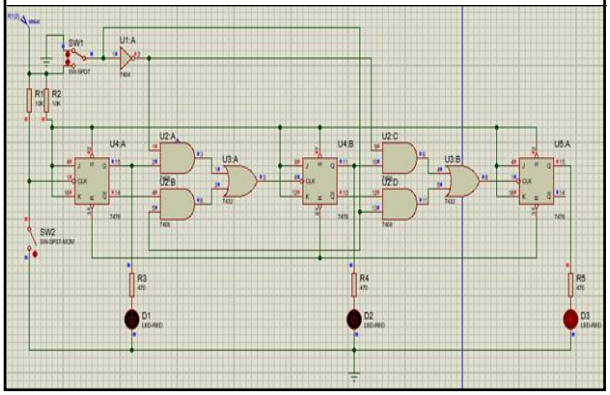
Iteration: 1



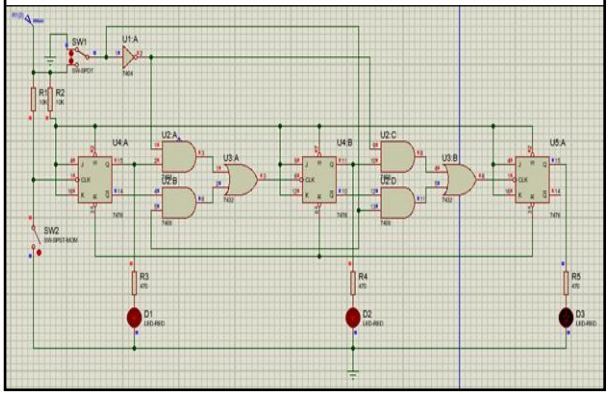
Iteration: 2



Iteration: 3



Iteration: 4



Section 2B references

Neso Academy (Director). (2015). *3 Bit & 4 Bit UP/DOWN Ripple Counter* [YouTube video]. Retrieved from https://youtu.be/5Um3NDvsYjQ

Neso Academy (Director). (2015). *3 Bit Asynchronous Up Counter* [YouTube video]. Retrieved from https://youtu.be/s1DSZEaCX_g

"*Information: the negative reciprocal value of probability.*"

-Claude Shannon

Theoretical pioneer of digital circuitry