

AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh



Title: Design of a 2 to 4 Decoder and a decimal to BCD Encoder.

Experiment No: 06

Date of Submission: 12/08/2021

Course Title: Digital
Logic Design
Laboratory

Course Code: EEE2206

Section: J

Semester: Summer

2020-2021

Course Teacher: Sudipta Das

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* Student(s) must complete all details except the faculty use part.

** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.: 02

Submitted By: Hoque, Fardin, (18-37245-1)

No	Name	ID	Program	Signature
1	Shailee, Nowrin Muhaimin	18-37259-1		
2	Bari, SK Tasnim	18-37201-1		
3	Patwary, Tanjib	18-37230-1		
4	Hoque, Fardin	18-37245-1		
5	Fahim, Ashik Ahmed	18-37269-1		
6				
7				
8				
9				
10				

Faculty use only

FACULTY COMMENTS

	Marks Obtained	
	Total Marks	

Title: Design of a 2 to 4 Decoder and a decimal to BCD Encoder.

Introduction: In this experiment we have learned about the 2 to 4 Decoder, Decimal to BCD Encoder and 4 to 2 Priority Encoder. **Binary Decoders** are another type of digital logic device that has inputs of 2-bit, 3-bit or 4-bit codes depending upon the number of data input lines, so a decoder that has a set of two or more bits will be defined as having an n -bit code, and therefore it will be possible to represent 2^n possible values. The 2-to-4 line binary decoder consists of an array of four AND gates. The 2 binary inputs labelled A and B are decoded into one of 4 outputs, hence the description of 2-to-4 binary decoder. On the other hand, decimal-to-BCD encoder is a digital circuit that has 10 input lines and 4 output lines. The inputs represent the 10 decimal numbers from 0 to 9, where only one input can be active at a time. Encoder and Decoder are very useful in information transmission.

Theory and Methodology: Firstly, encoder is a device that converts motion to an electrical signal that can be read by some type of control device in a motion control system, such as a counter or PLC. The encoder sends a feedback signal that can be used to determine position, count, speed, or direction. A control device can use this information to send a command for a particular function. Encoders use different types of technologies to create a signal, including: mechanical, magnetic, resistive and optical – optical being the most common. In optical sensing, the encoder provides feedback based on the interruption of light. Here we have learned about Decimal to BCD Encoder.

Then Decoder is a device that converts coded input to coded outputs provided both of these are different from one another. The name decoder means translating of coded information from one format into another. So, the input code generally has fewer bits than output code word. The decoder consists of four AND gates which is connected to the output. Here we have learned about De. Here we have learned about the 2 to 4 line decoder.

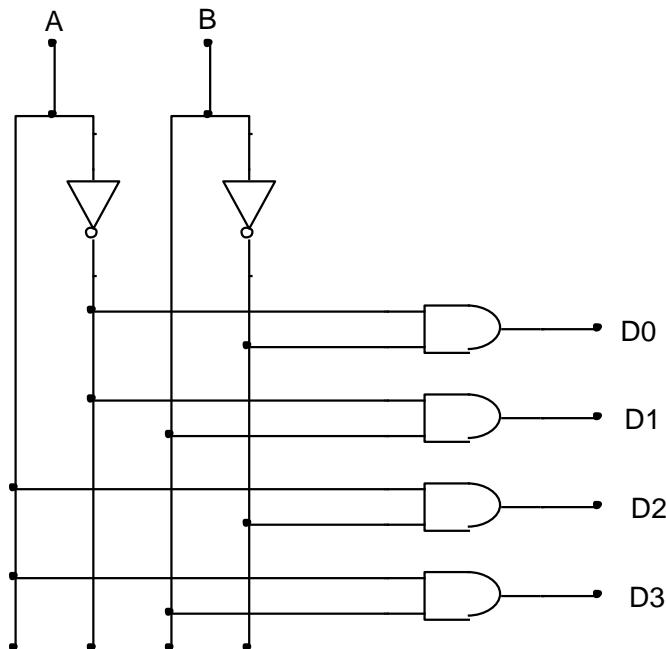


Fig. 1: 2 to 4 line Decoder

Truth table for 2 to 4 line Decoder,

A	B	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

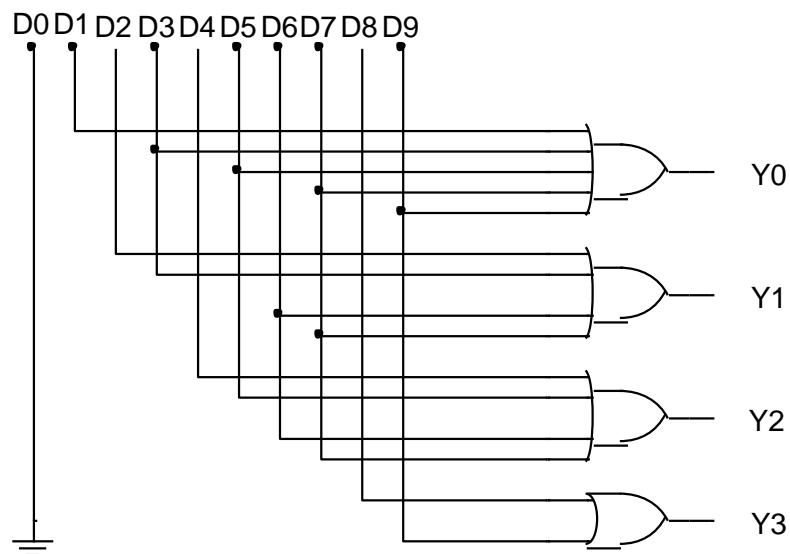


Fig.2: Decimal to BCD Encoder

Truth table for Decimal to BCD Encoder,

Dec.	Y3	Y2	Y1	Y0
D0	0	0	0	0
D1	0	0	0	1
D2	0	0	1	0
D3	0	0	1	1
D4	0	1	0	0
D5	0	1	0	1
D6	0	1	1	0
D7	0	1	1	1
D8	1	0	0	0
D9	1	0	0	1

Priority Encoder: After Encoder and Decoder, we have also learned about Priority Encoder in this experiment. A priority encoder provides n bits of binary coded output representing the position of the highest order active input of 2^n inputs. If two or more inputs are high at the same time, the input having the highest priority will take precedence. The priority encoder consists of 2 not gates, 2 and gates, 2 or gates. There are 4 inputs and 2 outputs in priority encoder.

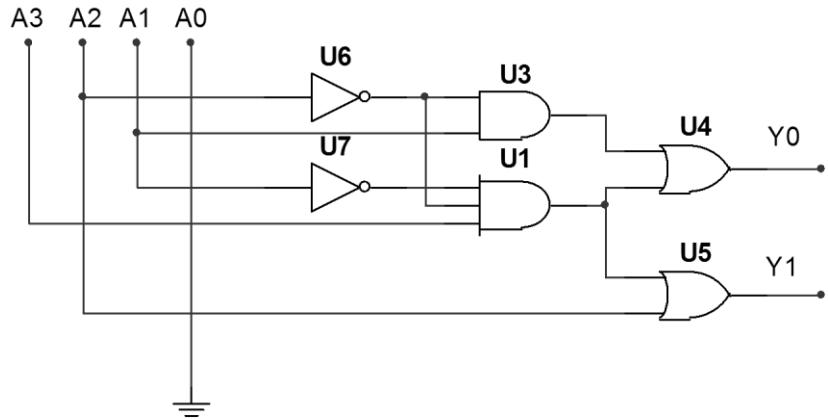


Fig.3: 4 to 2 Priority Encoder

Truth table for Priority Encoder,

A3	A2	A1	A0	Y1	Y0
x	1	x	x	1	0
x	0	1	x	0	1
1	0	0	x	1	1
0	0	0	1	0	0

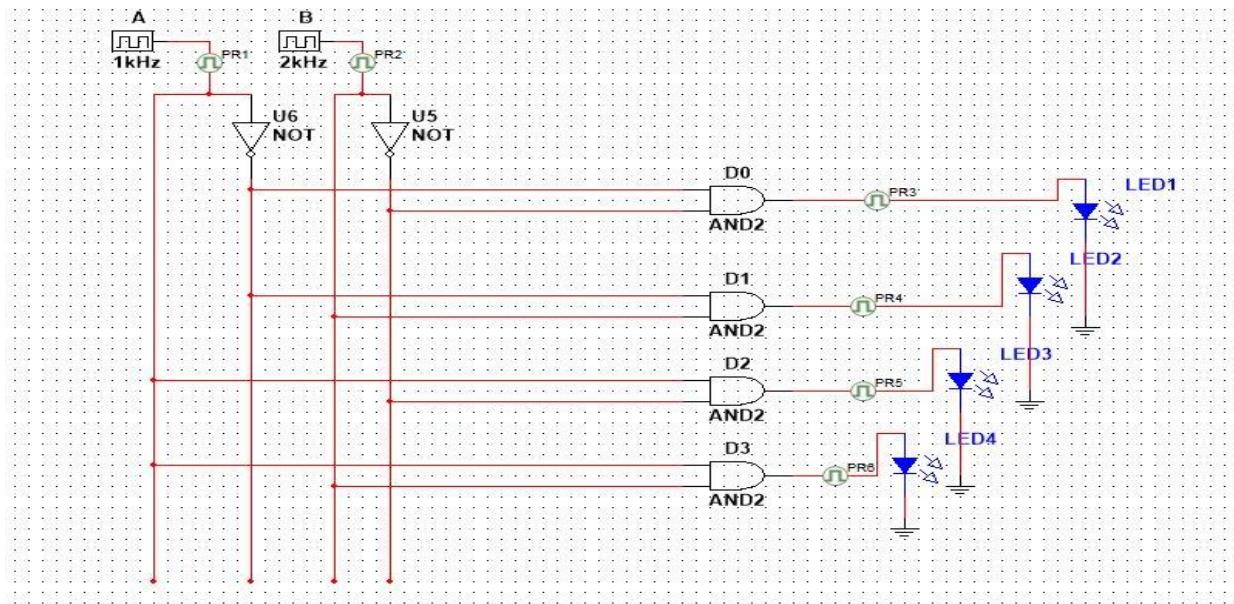
Apparatus:

- | | | |
|---------------|------------|--------|
| 1. NOT Gate - | IC 7404 | 1[pcs] |
| 2. AND Gate - | IC 7408 | 1[pcs] |
| 3. OR Gate - | 5 input OR | 1[pcs] |
| | 4 input OR | 2[pcs] |
| | 2 input OR | 1[pcs] |

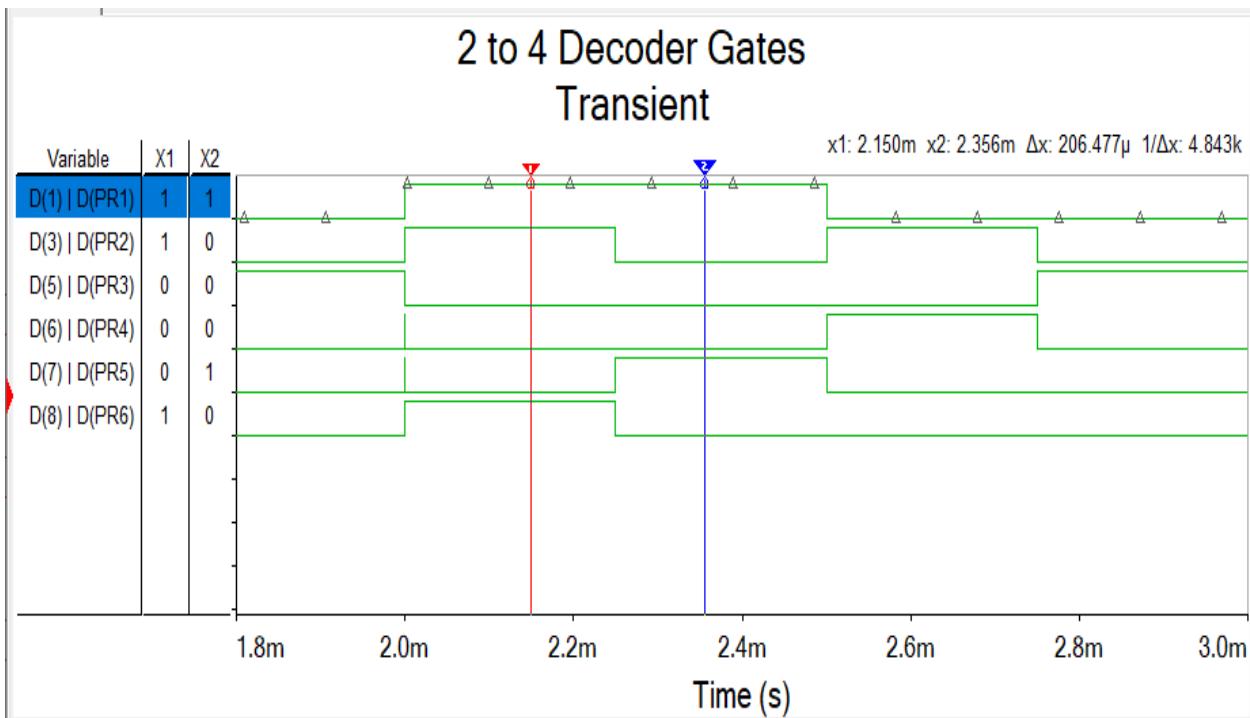
Simulation and Result:

1. 2 to 4 line Decoder by gates:

Simulation:

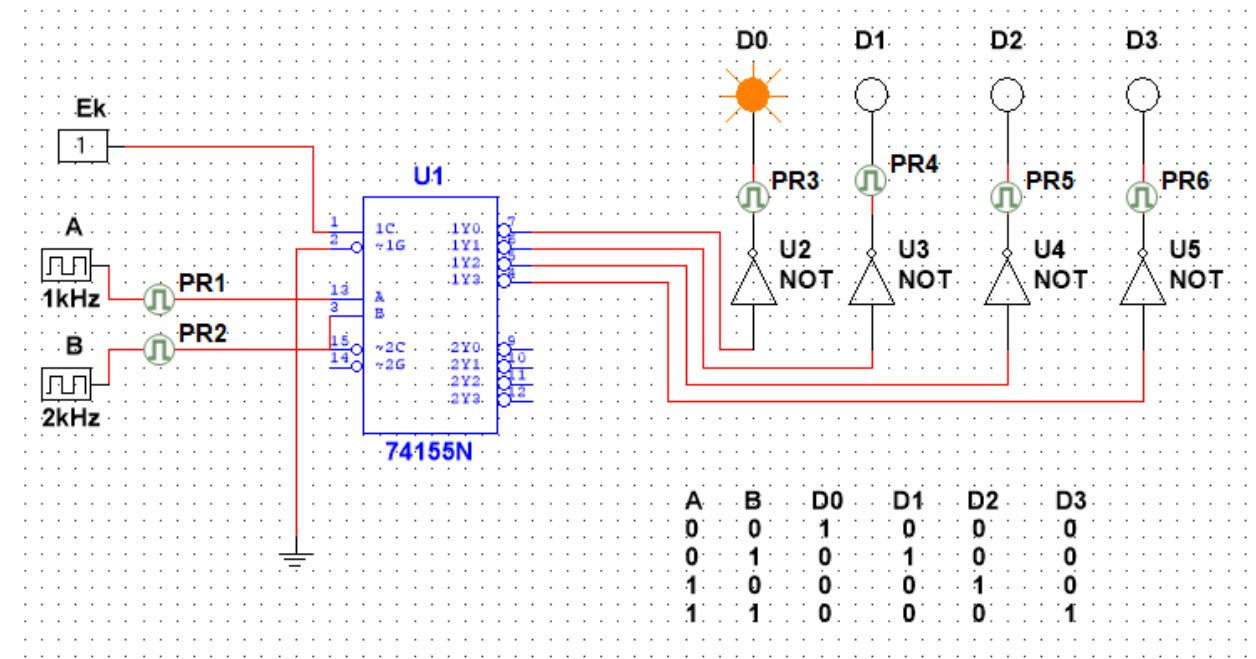


Graph:

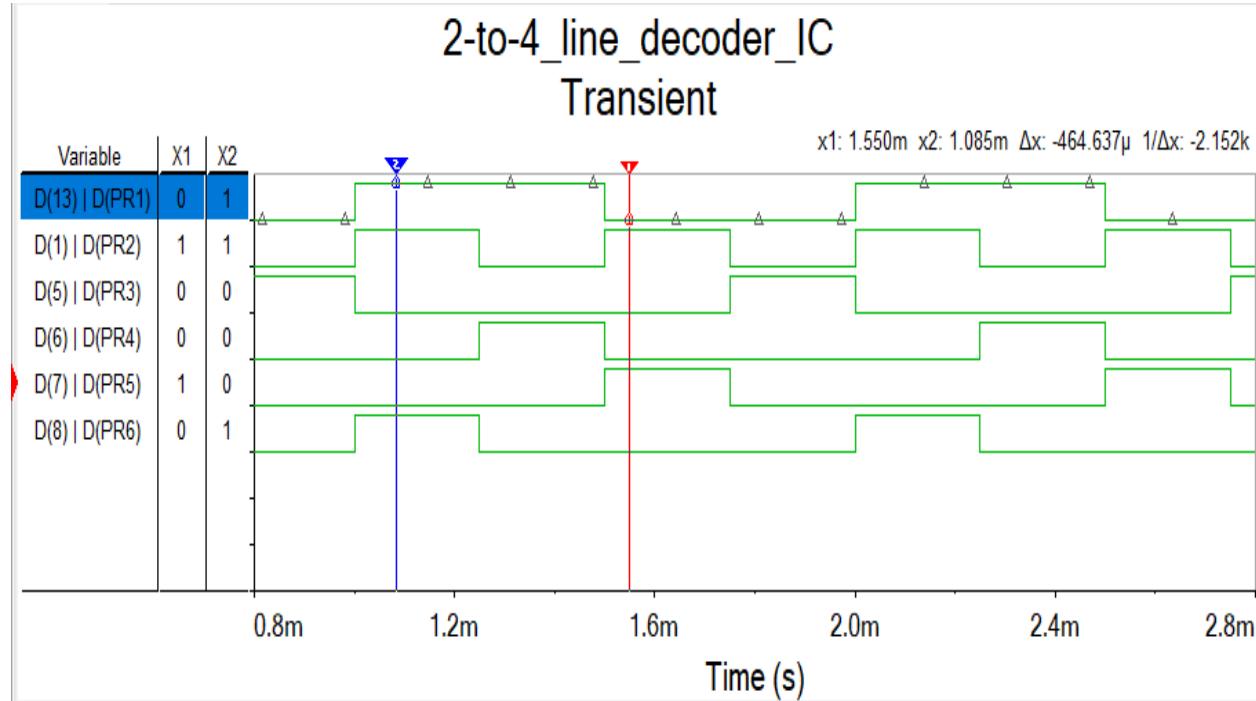


2. 2 to 4 line Decoder by IC:

Simulation:

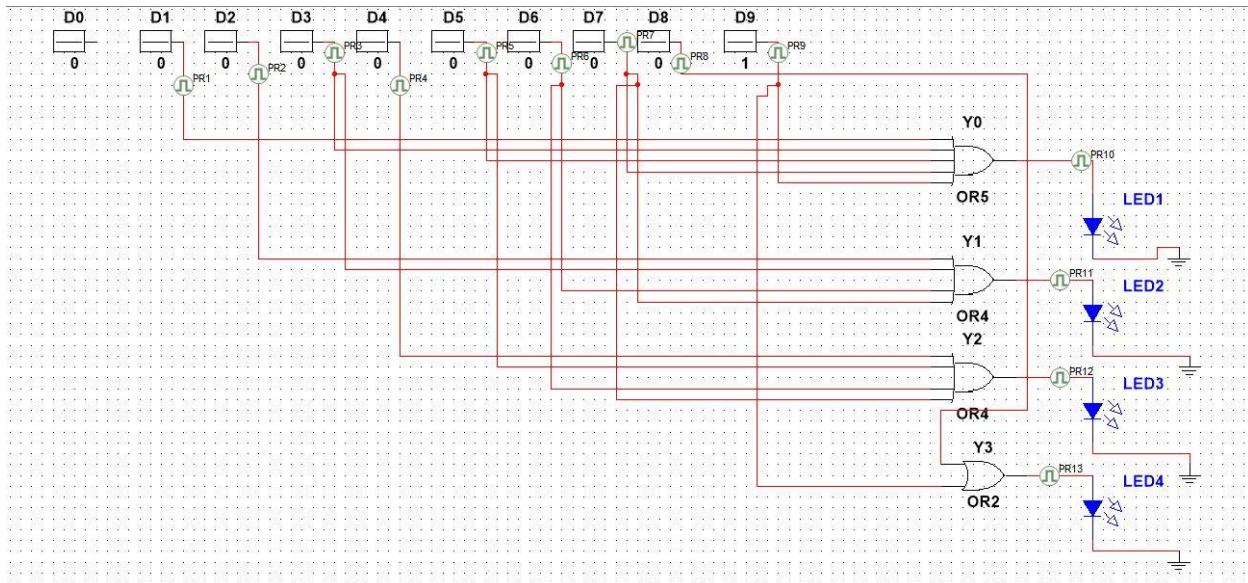


Graph:



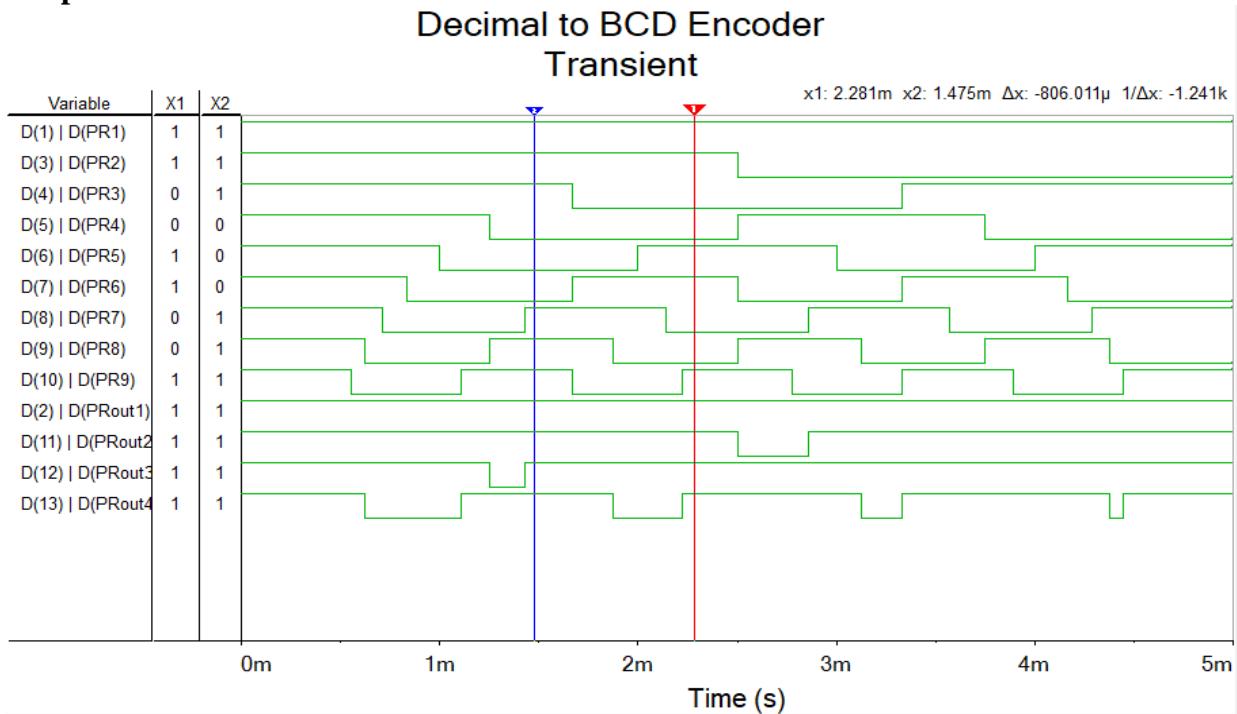
3. Decimal to BCD Encoder by gates:

Simulation:



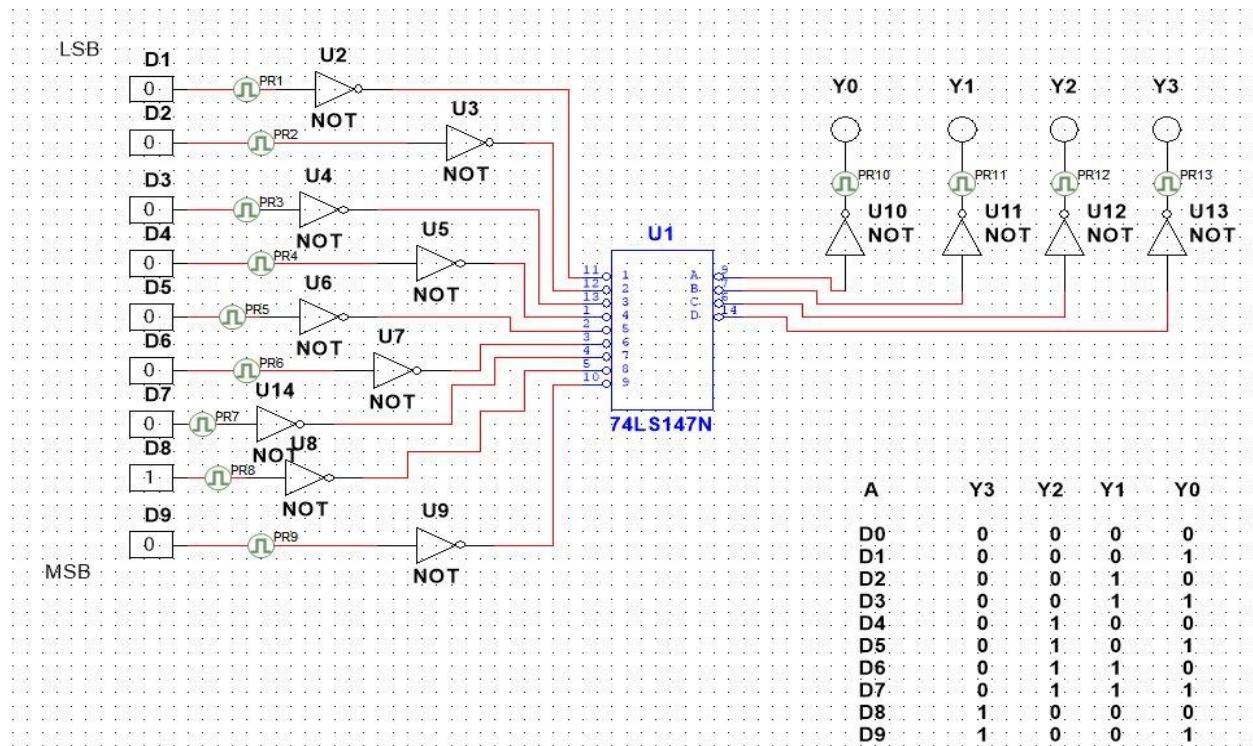
Graph:

Decimal to BCD Encoder
Transient

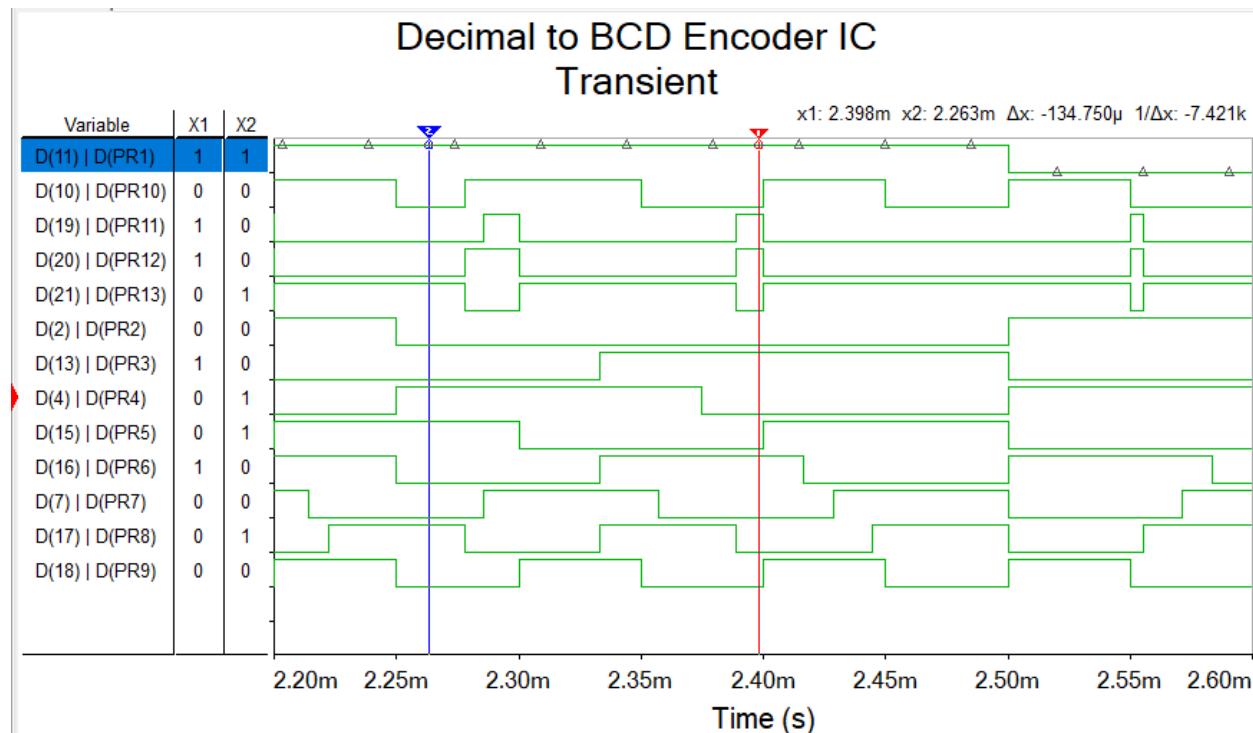


4. Decimal to BCD Encoder by IC:

Simulation:

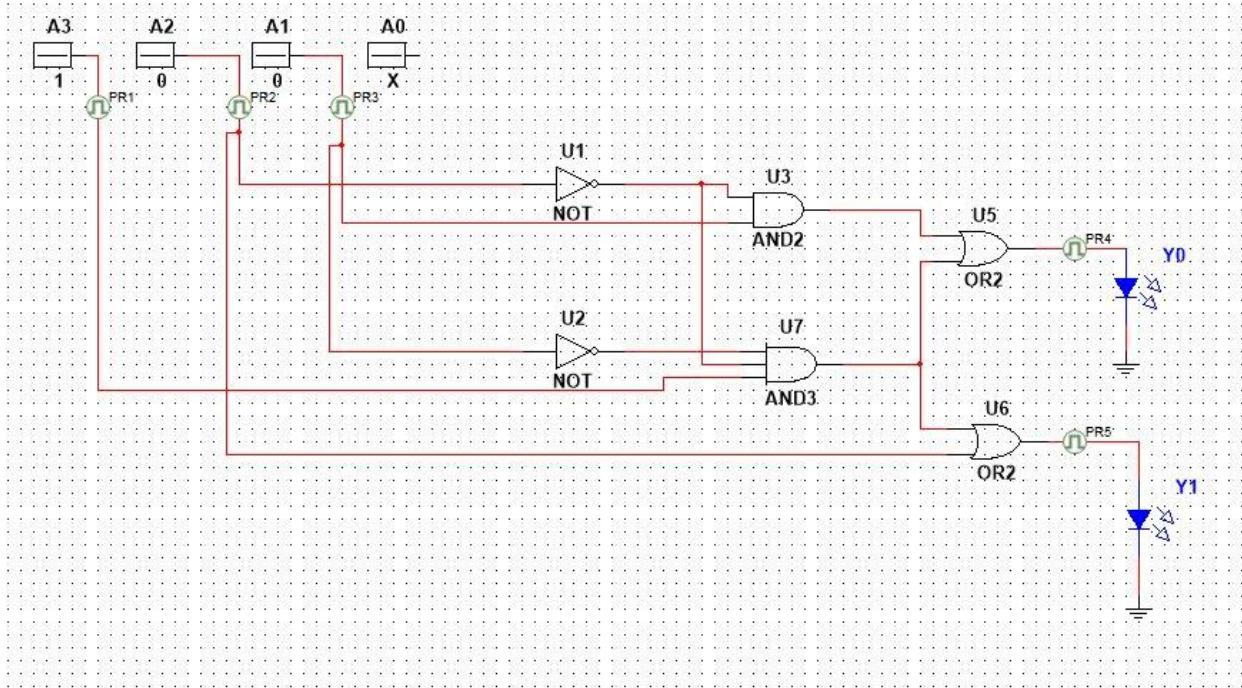


Graph:



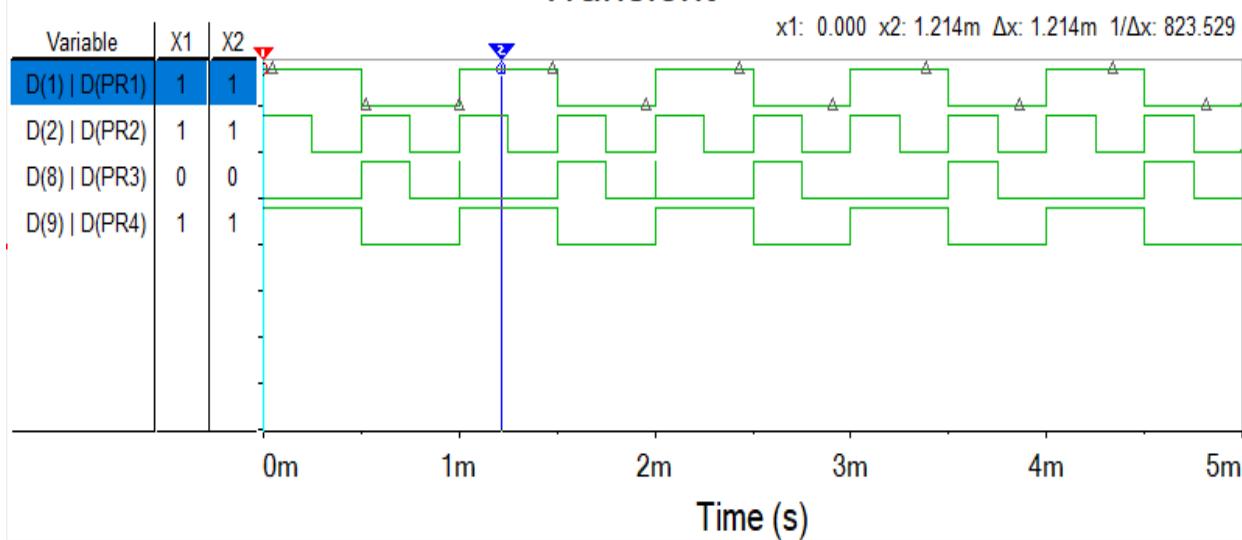
5. Priority Encoder by gates:

Simulation:



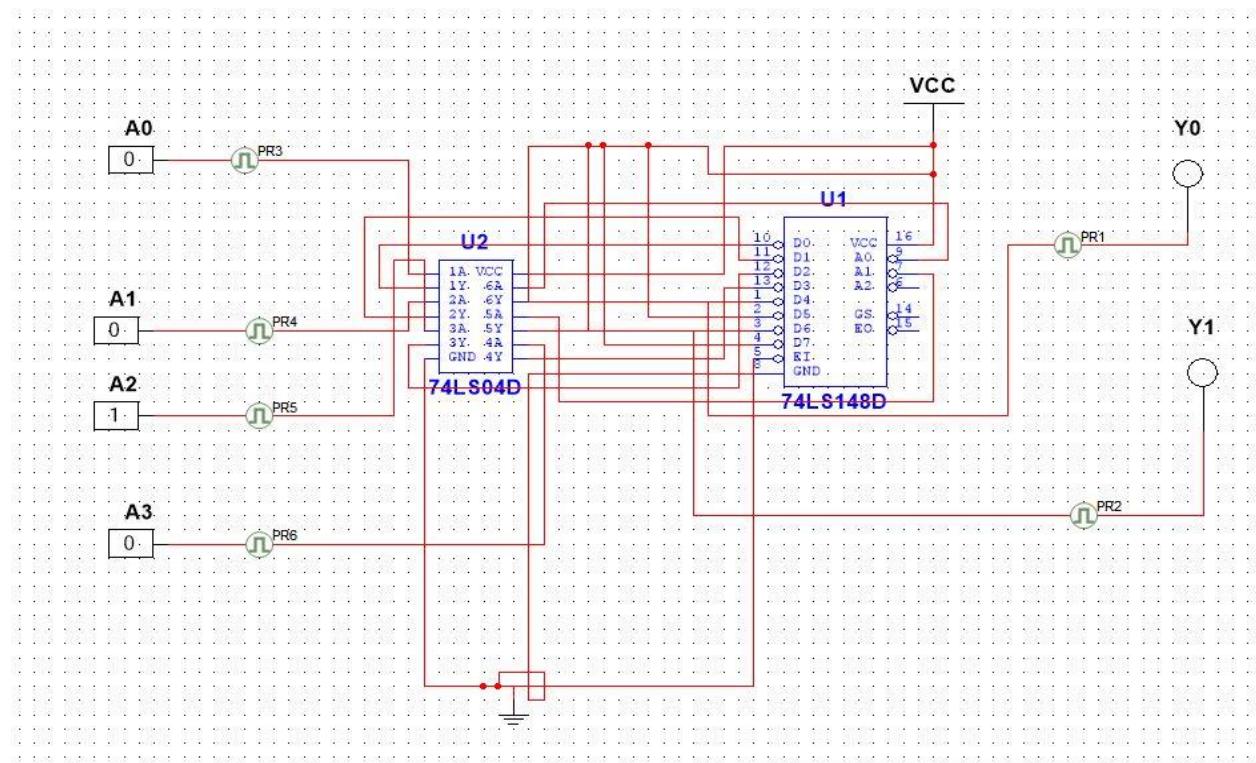
Graph:

4-to-2-priority-encoder-with-a-priority-sequence-of-2130
Transient



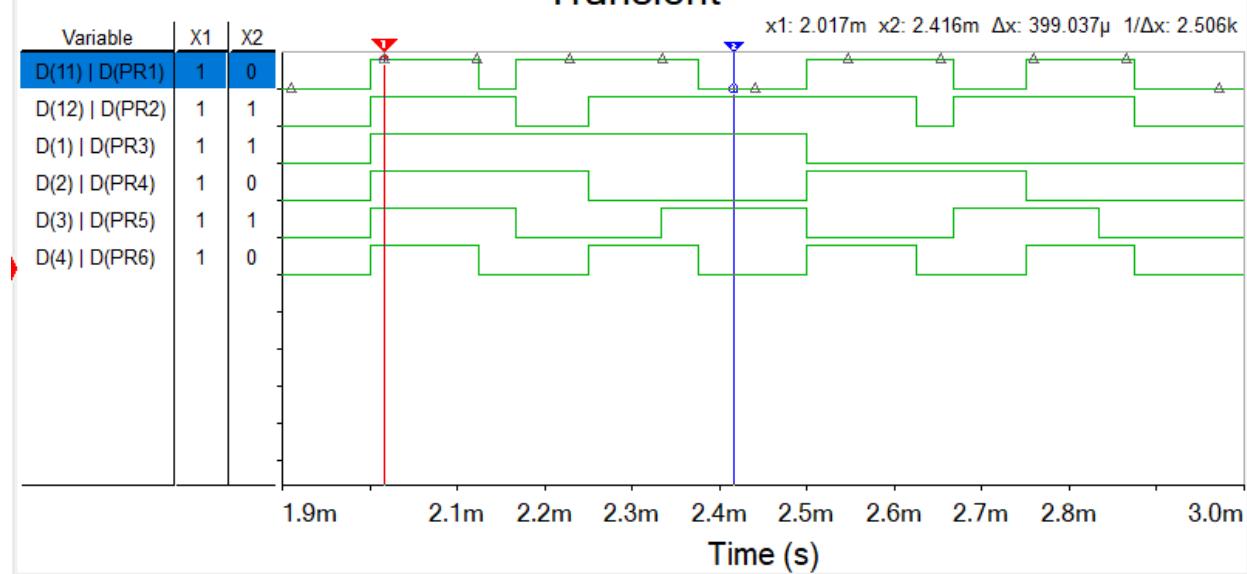
6. Priority Encoder by IC:

Simulation:



Graph:

4-to-2-priority encoder IC with priority sequence 2130
Transient



Discussion and Conclusion: In this experiment our main aim was to know about the Encoder, Decoder and Priority encoder. Mainly Encoder circuit basically converts the applied information signal into a coded digital bit stream. Decoder performs reverse operation and recovers the original information signal from the coded bits. After simulating the above 2 to 4 line Decoder circuit, we measure the output by using the digital probe. We get the correct output. Then we were simulating the Decimal to BCD encoder circuit and was able to get the correct output. We used both IC and Gates for Encoder, Decoder and Priority Encoder circuit for measure the output curve. Lastly, we simulated the above priority encoder circuit. Here the sequence of the priority encoder was 2, 1,3, 0. We got the correct output curve for this circuit also. We used some sources for collecting the circuit models. After doing all of this we have done this experiment properly.

Reference:

1. <https://electronics-course.com/priority-encoder>
2. Multisim
3. <https://www.geeksforgeeks.org/difference-between-encoder-and-decoder/>

AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh



Title: Design of Multiplexer (MUX) and Demultiplexer (DEMUX).

Experiment No: 07

Date of Submission: 12/08/2021

Course Title: Digital
Logic Design
Laboratory

Course Code: EEE2206

Section: J

Semester: Summer

2020-2021

Course Teacher: Sudipta Das

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* Student(s) must complete all details except the faculty use part.

** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.: 02

Submitted By: Shailee, Nowrin Muhaimin (18-37259-1)

No	Name	ID	Program	Signature
1	Shailee, Nowrin Muhaimin	18-37259-1		
2	Bari, SK Tasnim	18-37201-1		
3	Patwary, Tanjib	18-37230-1		
4	Hoque, Fardin	18-37245-1		
5	Fahim, Ashik Ahmed	18-37269-1		
6				
7				
8				
9				
10				

Faculty use only

FACULTY COMMENTS

Marks Obtained

Total Marks

Title: Design of Multiplexer (MUX) and Demultiplexer (DEMUX).

Introduction:

The multiplexer is a device that has multiple inputs and single line output. The select lines determine which input is connected to the output, and also increase the amount of data that can be sent over a network within a certain time. It is also called a data selector.

De-multiplexer is also a device with one input and multiple output lines. It is used to send a signal to one of the many devices. The main difference between a multiplexer and a de-multiplexer is that a multiplexer takes two or more signals and encodes them on a wire, whereas a de-multiplexer does reverse to what the multiplexer does. Both the multiplexer and demultiplexer are combinational circuits.

Objectives:

In this experiment, we will learn how to design and implement multiplexers (MUX) and demultiplexers (DEMUX) of different sizes using basic logic gates and also learn how to construct bigger multiplexer using smaller multiplexers.

Theory and Methodology:

Multiplexer:

A Multiplexer is a device that allows one of several analogs or digital input signals which are to be selected and transmits the input that is selected into a single medium. Multiplexer is also known as Data Selector. A multiplexer of 2^n inputs has n select lines that will be used to select input line to send to the output. Multiplexer is abbreviated as Mux. These devices are used to increase the amount of data that can be transmitted over a network. A network multiplexer is used with a variety of signal types like digital, video, analog, and optical. They also support different transports mechanisms, such as wireless and wired. Like other types of network equipment, network multiplexers support specific network protocols like Ethernet. These devices help businesses reduce operational costs and improve scalability and efficiency while overcoming distance limitations.

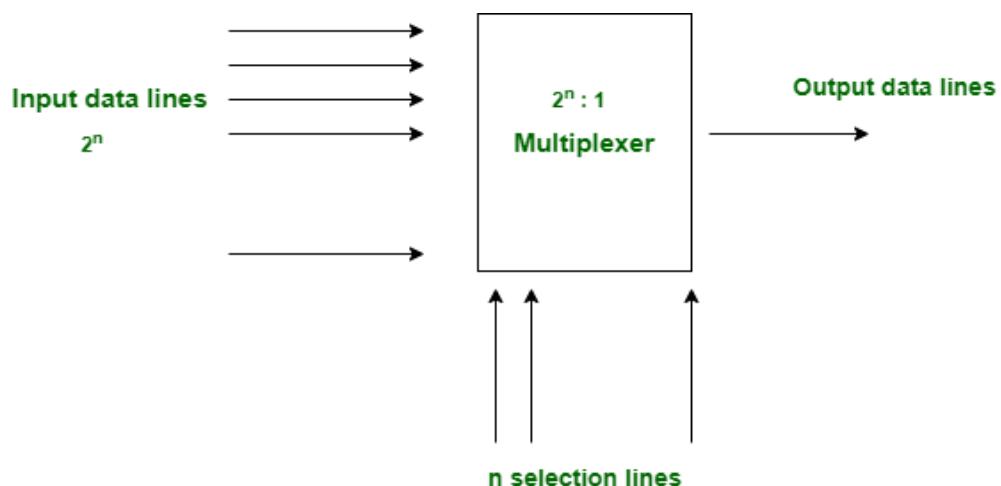


Figure 01: Multiplexer.

Demultiplexer:

A demultiplexer also known as a DeMux or data distributor is defined as a circuit that can distribute or deliver multiple outputs from a single input. A demultiplexer can perform as a single input with many output switches. The demultiplexer's output lines are 'm' in number, the select line number is 'n', and $m = 2^n$. The control signal or select input code decides the output line to which the input has to be transmitted. The DeMux can also perform as a binary to decimal decoder. The data input line should be at the logic 1 level, and the binary input is given to the select input lines. The corresponding line will give the output. In the designing of multiple combinational circuits, this circuit is advantageous.

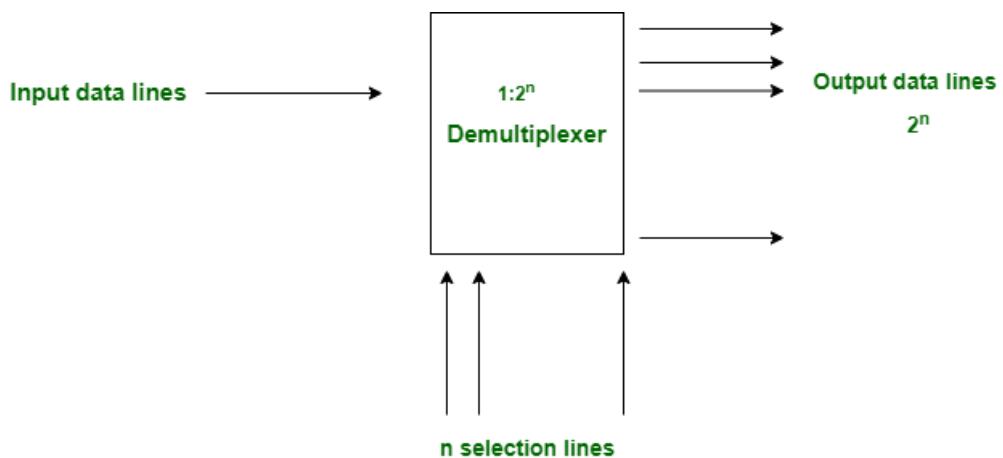


Figure 02: Demultiplexer.

Truth Table:

4 to 1 MUX:

S1	S0	F
0	0	D0
0	1	D1
1	0	D2
1	1	D3

1 to 4 DEMUX:

S1	S0	D0	D1	D2	D3
0	0	Din	0	0	0
0	1	0	Din	0	0
1	0	0	0	Din	0
1	1	0	0	0	Din

2 to 1 MUX:

S	F
0	D0
1	D1

1 to 2 DEMUX:

S	D0	D1
0	Din	0
1	0	Din

8 to 1 MUX:

S2	S1	S0	f
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	0	0	D4
1	0	1	D5
1	1	0	D6
1	1	1	D7

1 to 8 DEMUX:

S2	S1	S0	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	Din	0	0	0	0	0	0	0
0	0	1	0	Din	0	0	0	0	0	0
0	1	0	0	0	Din	0	0	0	0	0
0	1	1	0	0	0	Din	0	0	0	0
1	0	0	0	0	0	0	Din	0	0	0
1	0	1	0	0	0	0	0	Din	0	0
1	1	0	0	0	0	0	0	0	Din	0
1	1	1	0	0	0	0	0	0	0	Din

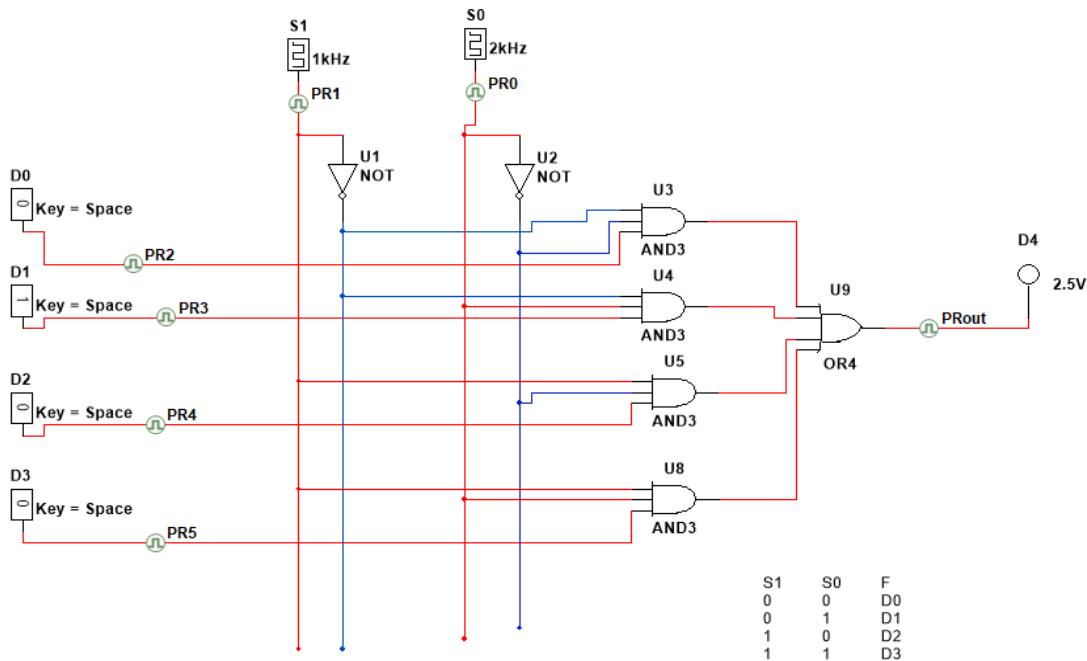
Apparatus:

1. Digital trainer board.
2. NOT Gate - IC 7404
3. AND Gate - IC 7408
4. OR Gate - IC 7408
5. Connecting wires.
6. Oscilloscope.

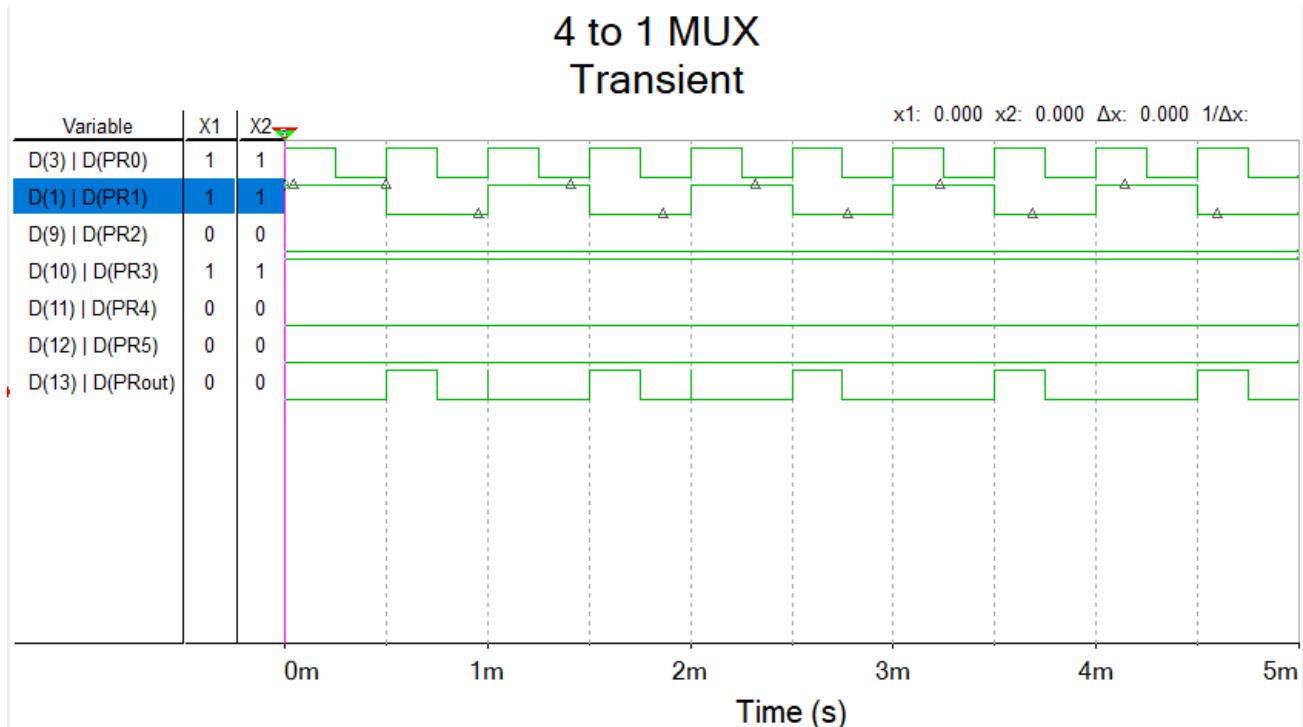
Simulation and Results:

4 to 1 Multiplexer using Logic Gates:

Simulation:

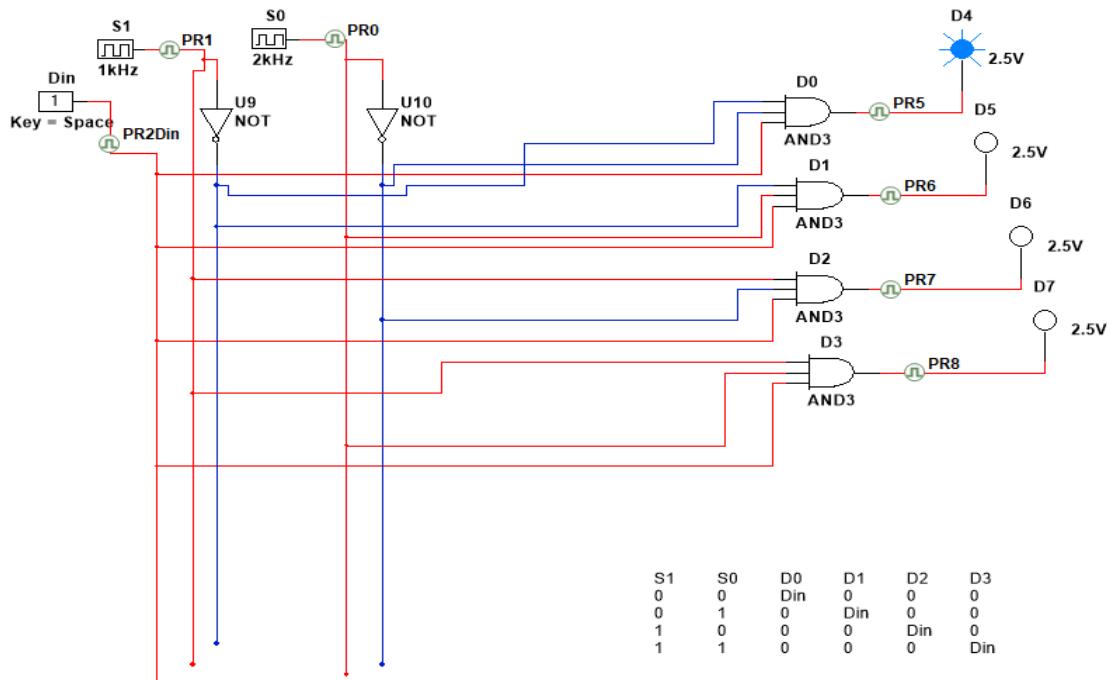


Graph:

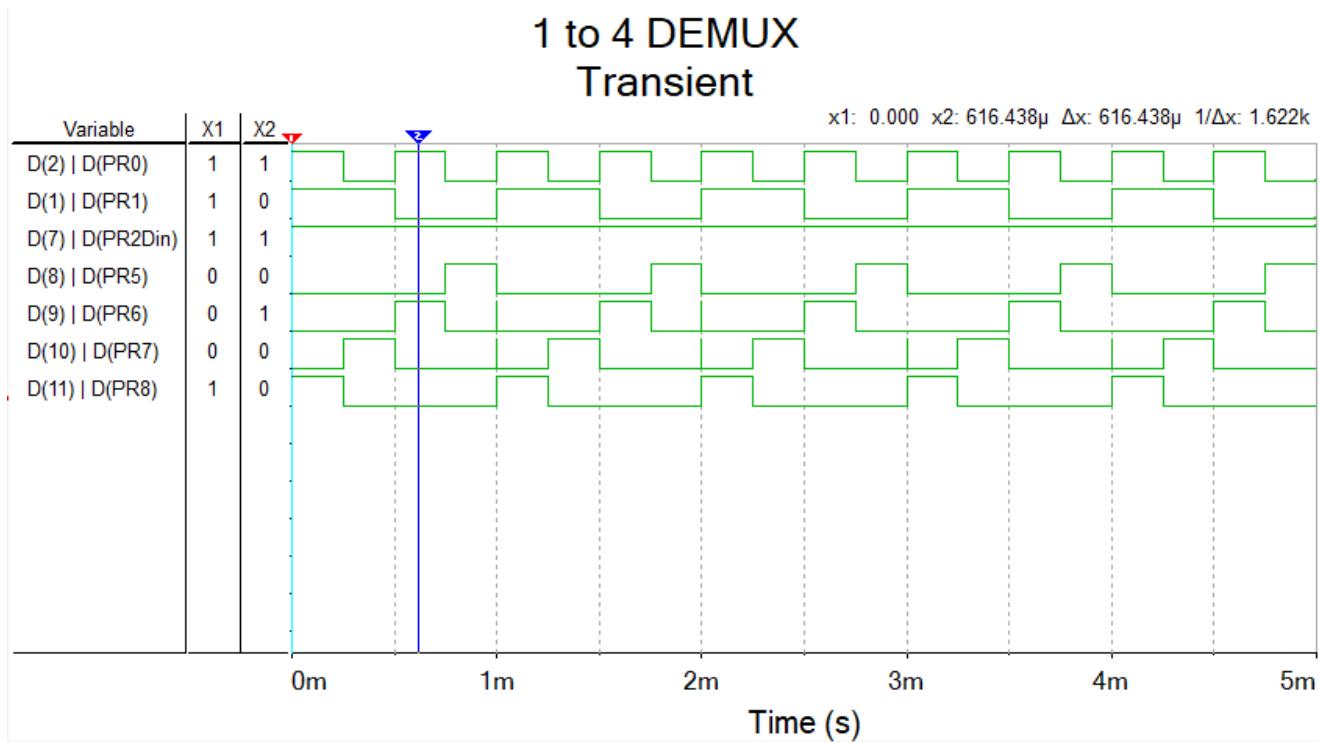


1 to 4 Demultiplexer using Logic Gates:

Simulation:



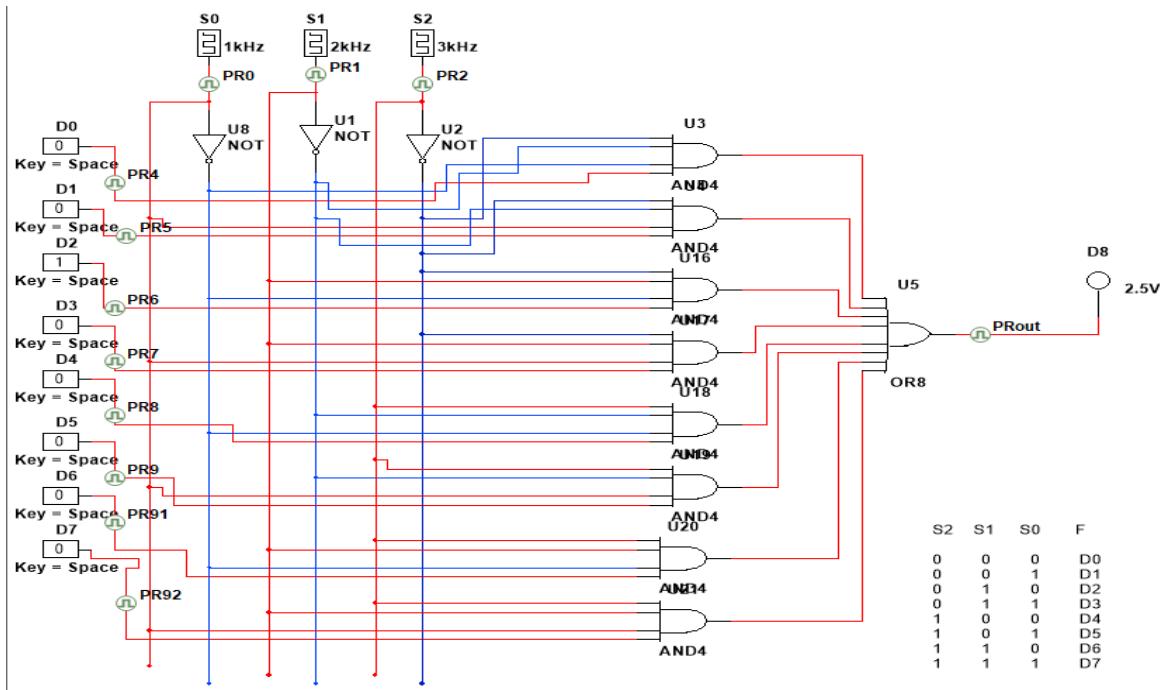
Graph:



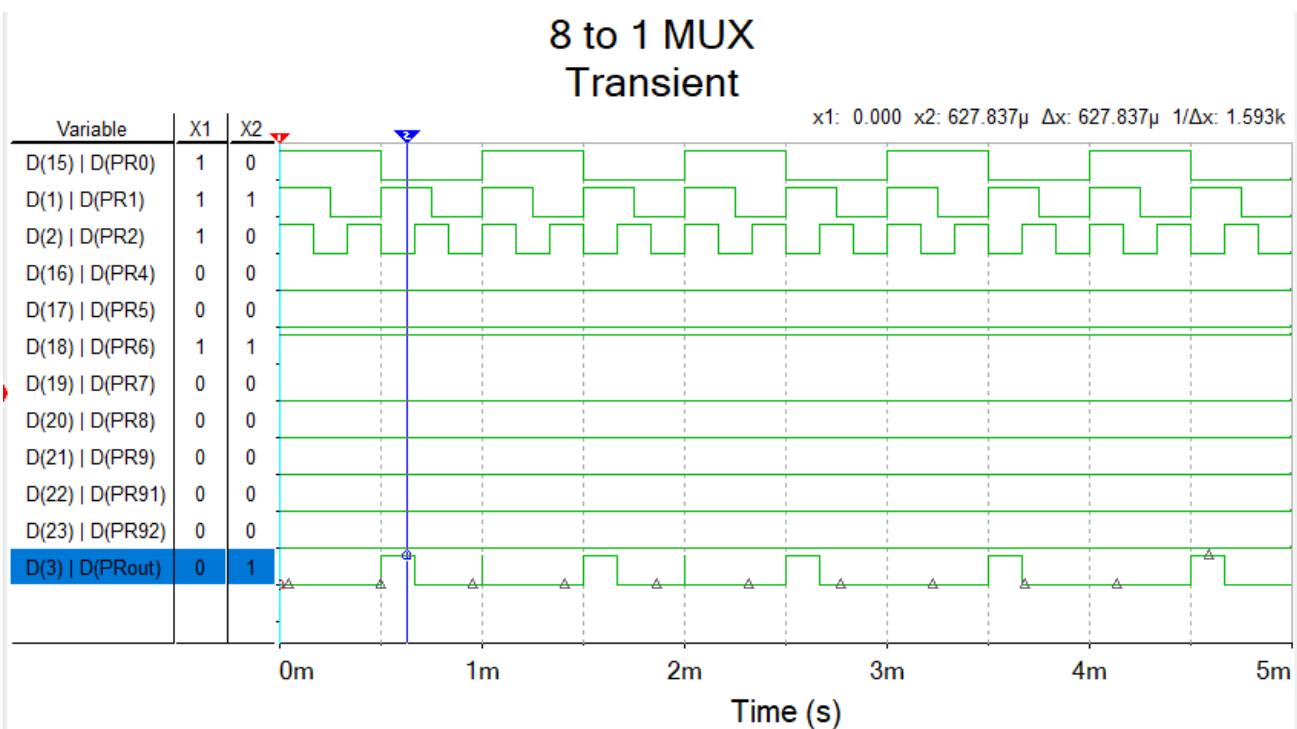
Questions with answers for report writing:

1. 8 to 1 multiplexer using basic logic gates:

Simulation:

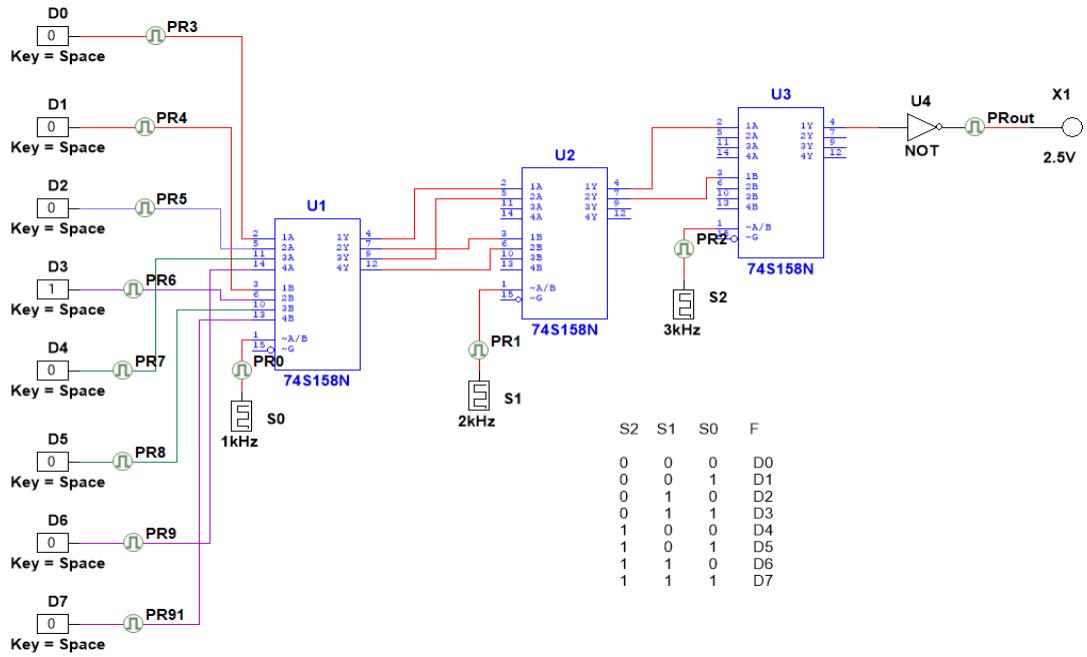


Graph:



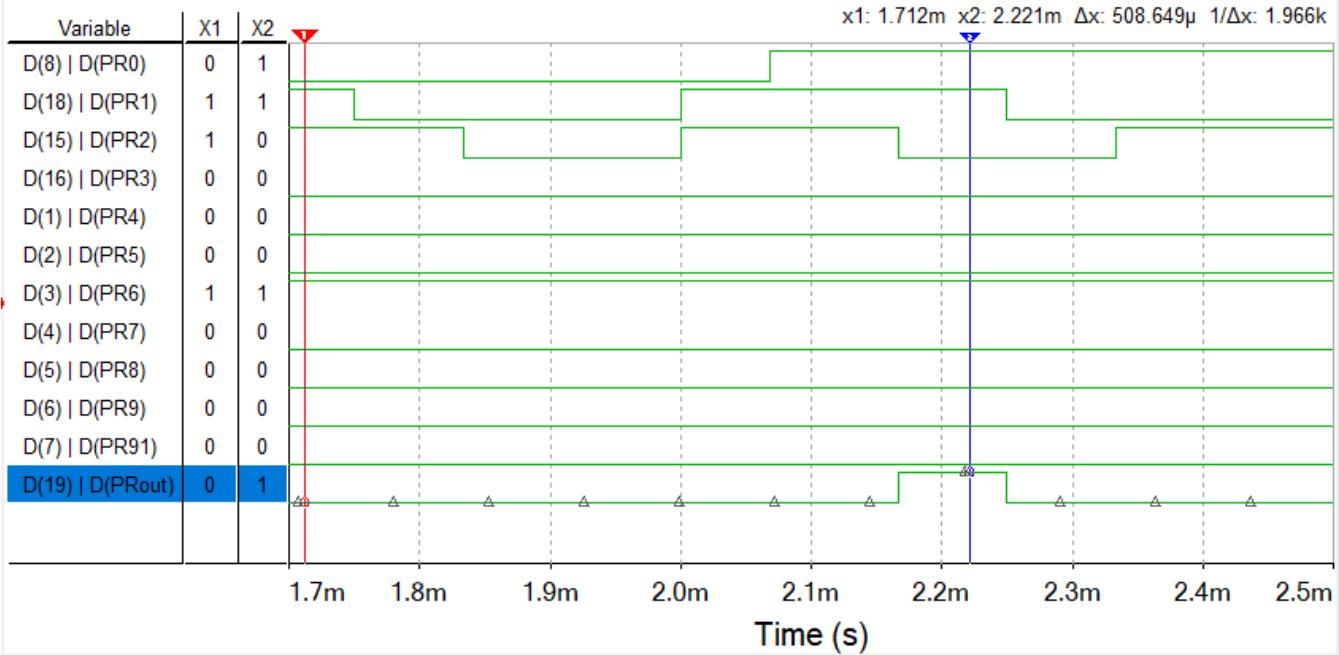
2. 8 to 1 multiplexer using 2 to 1 multiplexers:

Simulation:



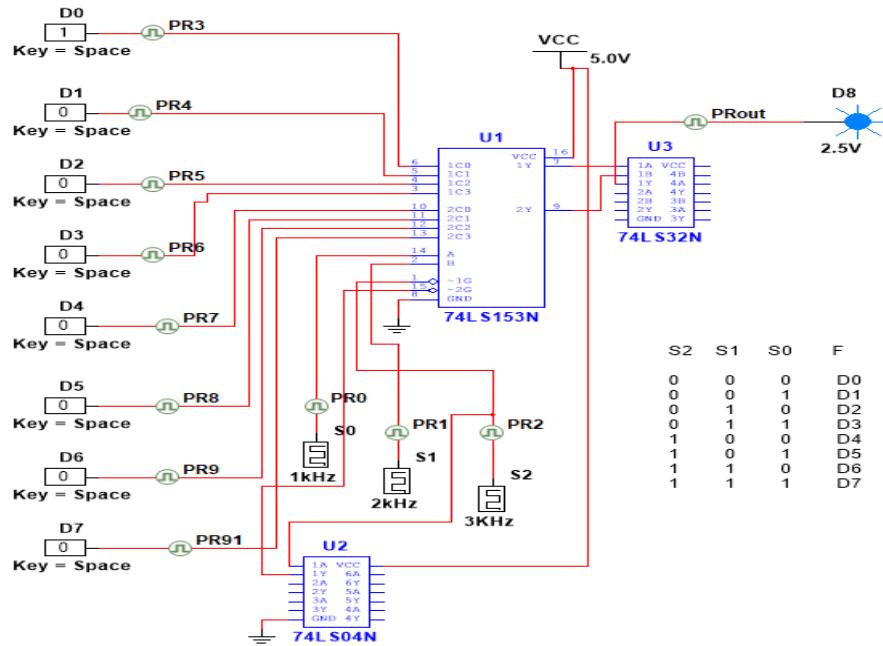
Graph:

8 to 1 MUX using 2 to 1 MUX
Transient



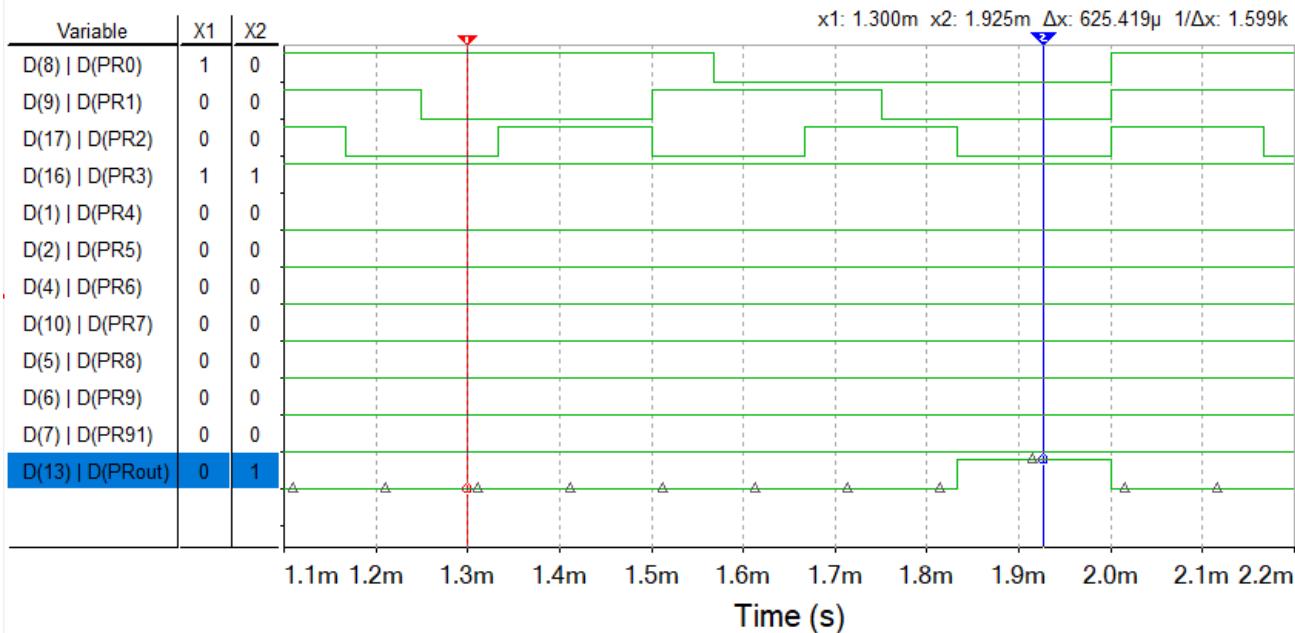
3. 8 to 1 multiplexer using 4 to 1 multiplexers:

Simulation:



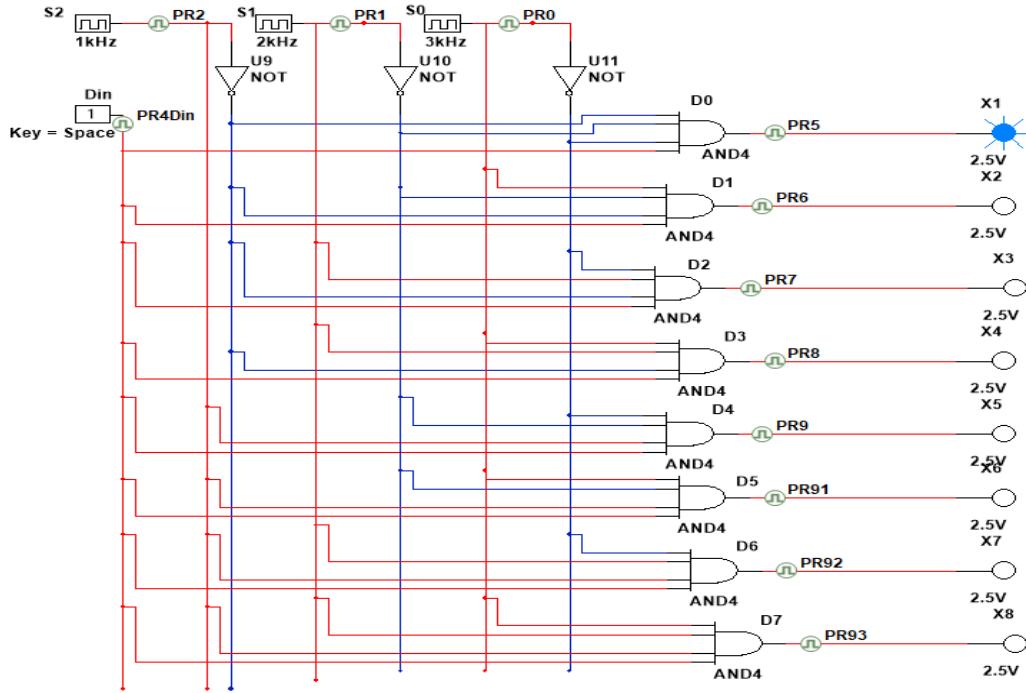
Graph:

8 to 1 MUX using 4 to 1 MUX
Transient



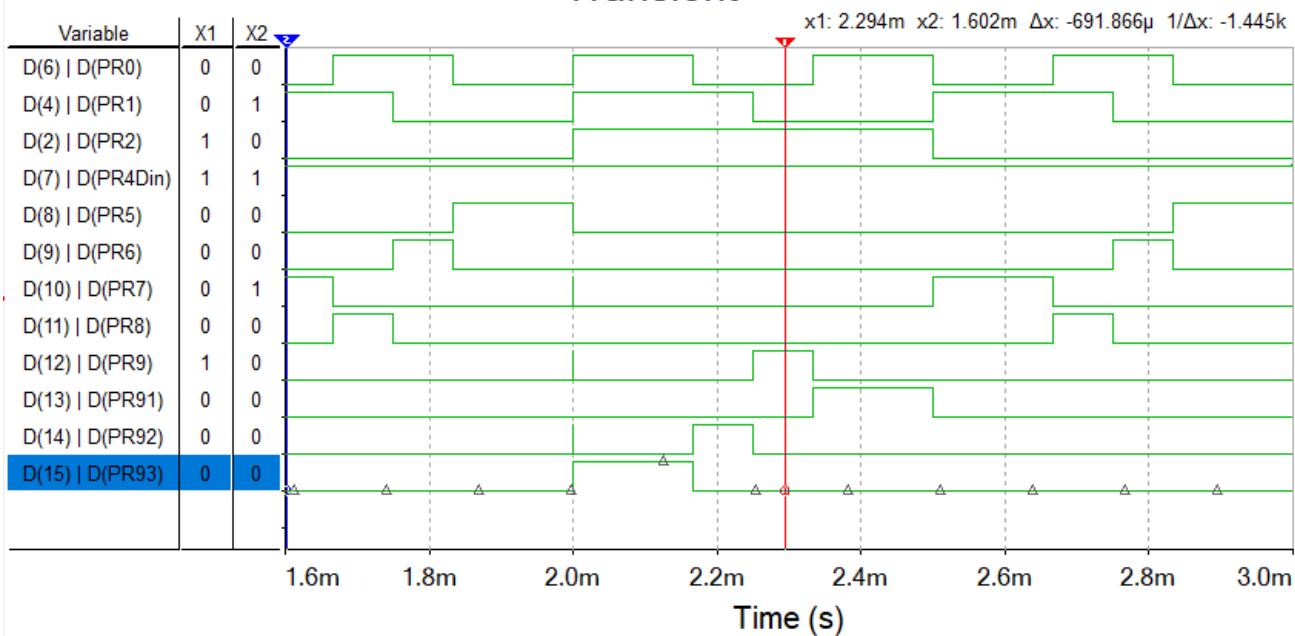
4. 1 to 8 demultiplexer using basic logic gates:

Simulation:



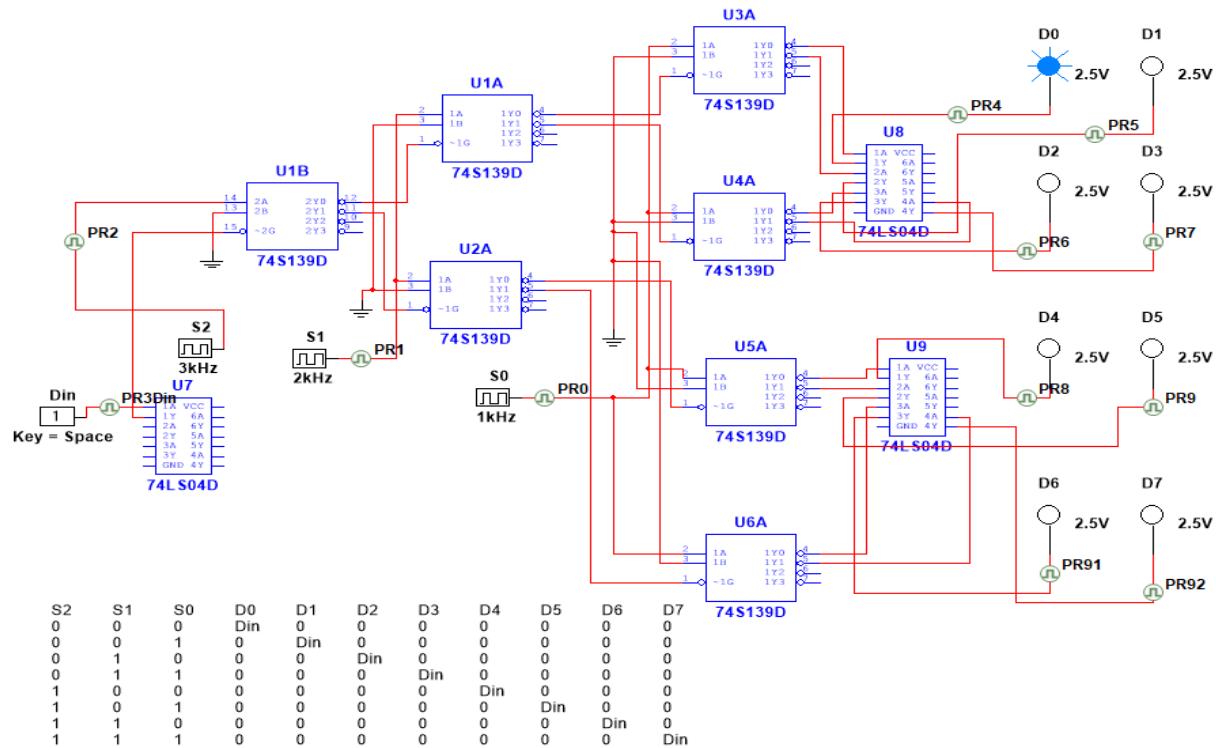
Graph:

1 to 8 DEMUX
Transient

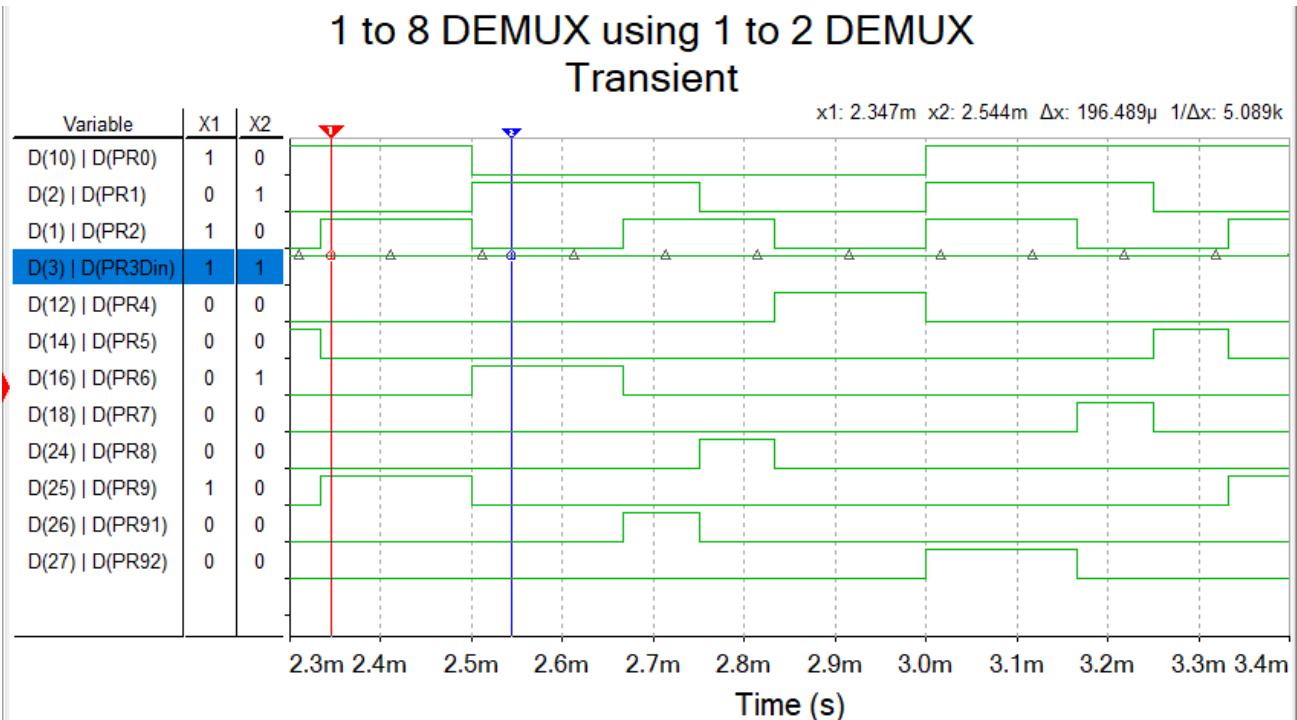


5. 1 to 8 demultiplexer using 1 to 2 demultiplexers:

Simulation:

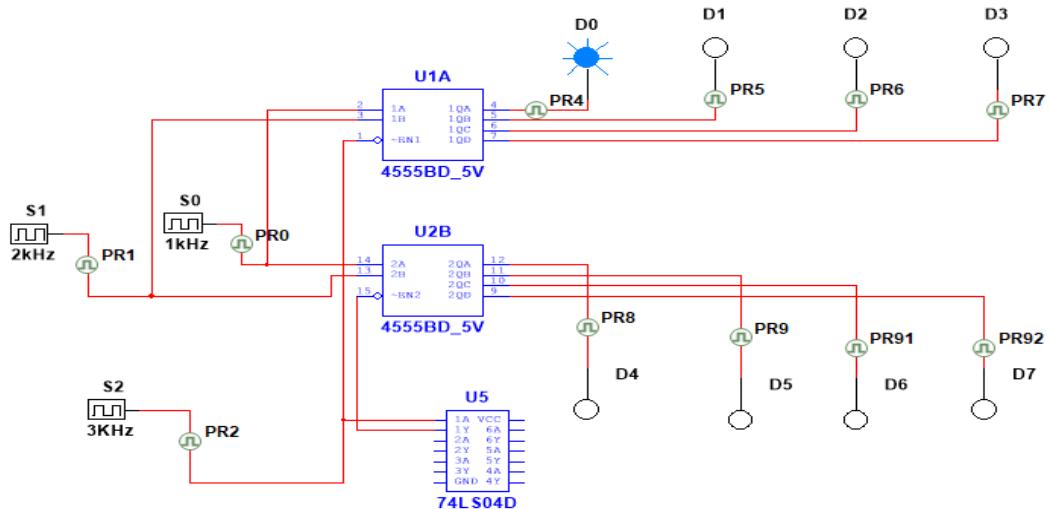


Graph:



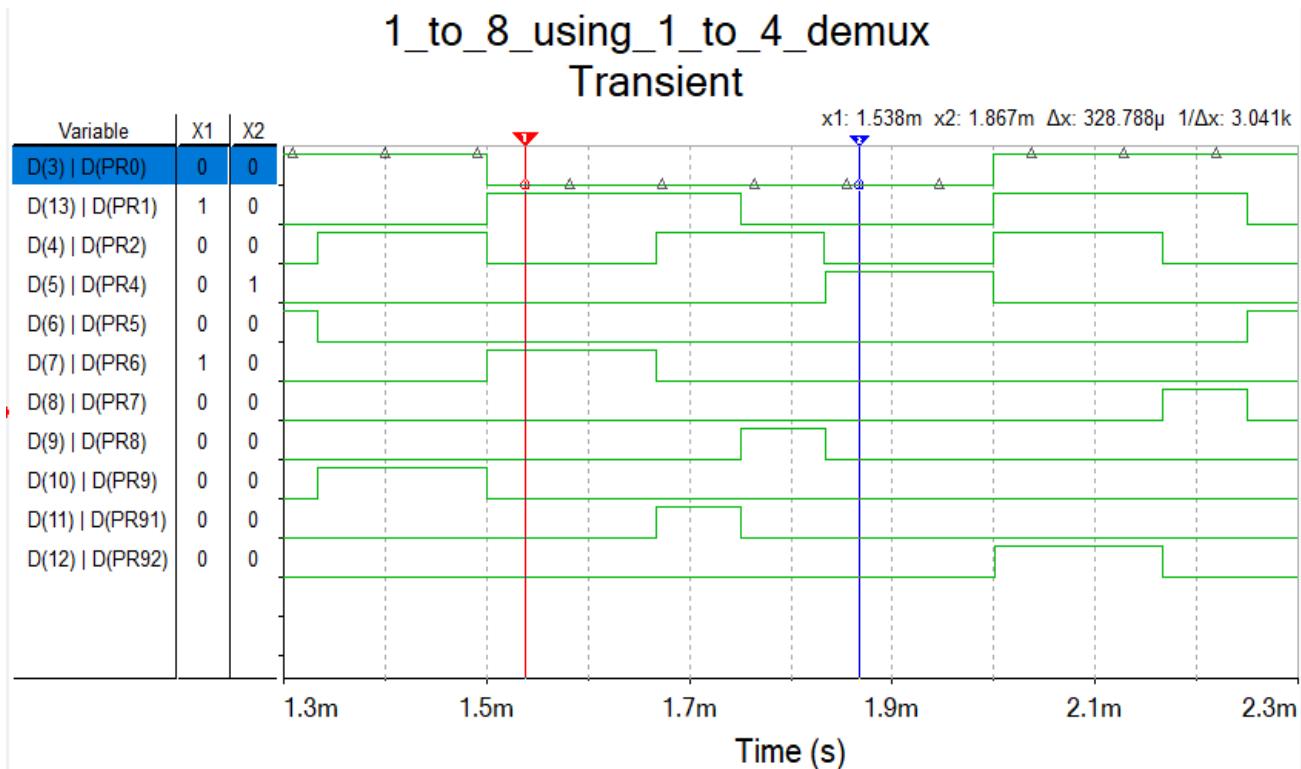
6. 1 to 8 demultiplexer using 1 to 4 demultiplexers:

Simulation:



S2	S1	S0	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	Din	0	0	0	0	0	0	0
0	0	1	0	Din	0	0	0	0	0	0
0	1	0	0	0	Din	0	0	0	0	0
0	1	1	0	0	0	Din	0	0	0	0
1	0	0	0	0	0	0	Din	0	0	0
1	0	1	0	0	0	0	0	Din	0	0
1	1	0	0	0	0	0	0	0	Din	0
1	1	1	0	0	0	0	0	0	0	Din

Graph:



Discussion and Conclusion:

In this experiment, we learn about the process of design and implement multiplexers (MUX) and demultiplexers (DEMUX) of different sizes using basic logic gates and also learn how to construct bigger multiplexer using smaller multiplexers with ICs.

The simulation results and graphs are matched with the truth table. We connect all the components and wires properly and done the simulation in Multisim software. Finally we can said that the experiment is successful in complying with the goal. During the simulation we faced some difficulties to find accurate IC of multiplexers and demultiplexers. So with future studies we can improve those mistakes.

Reference(s):

1. "Fundamentals of Digital Logic with verilog design" by – Brown & Vranesic.
2. <https://www.geeksforgeeks.org/difference-between-multiplexer-and-demultiplexer/>
3. <https://www.elprocus.com/what-is-multiplexer-and-demultiplexer-types-and-differences/>
4. <https://www.electrical4u.com/demultiplexer/>

AMERICAN INTERNATIONAL UNIVERSITY - BANGLADESH

408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh



Title: Study of Different Flip-Flops

Experiment No: 08

Date of Submission: 12/08/21

Course Title: Digital
Logic Design Lab

Course Code:

Section: J

Semester: Summer **2020-2021**

Course Teacher: Sudipta Das

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* Student(s) must complete all details except the faculty use part.

** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.: 02

Submitted by: Fahim, Ashik Ahmed. Shailee, Nowrin Muhammin

No	Name	ID	Program	Signature
1	Fahim, Ashik Ahmed	18-37269-1	CSE	
2	Shailee, Nowrin Muhammin	18-37259-1	CSE	
3	Bari, SK Tasnim	18-37201-1	CSE	
4	Patwary, Tanjib	18-37230-1	CSE	
5	Hoque, Fardin	18-37245-1	CSE	
6				
7				
8				
9				
10				

Faculty use only

FACULTY COMMENTS

Marks Obtained

Total Marks

Title: Study of Different Flip-Flops.

Introduction: Flip Flops are the basic building blocks of sequential logic circuit. A clock is used in the Flip Flop Circuit. One bit can be stored in each Flip Flop. Flip Flops have many variations. D Flip Flop, T Flip Flop, J-K Flip Flop are some of them.

Theory and Methodology: In this experiment different types of flip flops were built. Flip flops mainly have two or three pins. Their control pin is named as their model's name and the other pin is input clock.

There are two types of flip flop by their triggering type. One of them is positive edge triggered and another of them is negative edge triggered. But in this experiment, we will stick to the positive edge triggered flip flop. So, when using positive edge triggered flip flop if the value is not positive then the output value is held that means the values does not change then.

Flip flop has always two outputs Q and Q'. So, when the value of Q is zero then the value of Q' is one and vice versa. But other than this any output is not valid. Because these are complementing of one another.

D Flip Flop: In D flip flop, the single input "D" is referred to as the "Data" input. When the data input is set to 1, the flip flop would be set, and when it is set to 0, the flip flop would change and become reset. However, this would be pointless since the output of the flip flop would always change on every pulse applied to this data input. The "CLOCK" or "ENABLE" input is used to avoid this for isolating the data input from the flip flop's latching circuitry. When the clock input is set to true, the D input condition is only copied to the output Q. This forms the basis of another sequential device referred to as D Flip Flop. When the clock input is set to 1, the "set" and "reset" inputs of the flip-flop are both set to 1. So it will not change the state and store the data present on its output before the clock transition occurred. In simple words, the output is "latched" at either 0 or 1.

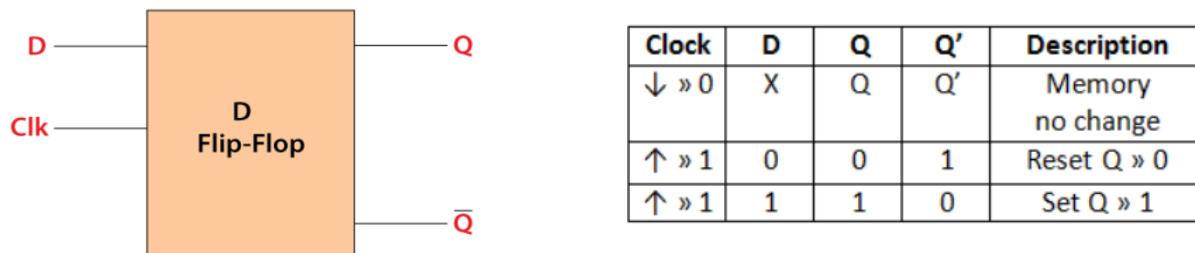


Fig 01: D Flip Flop and Truth table.

T Flip Flop: The name T flip-flop is termed from the nature of toggling operation. The major applications of T flip-flop are counters and control circuits. T flip flop is modified form of JK flip-flop making it to operate in toggling region. Whenever the clock signal is LOW, the input is never going to affect the output state. The clock has to be high for the inputs to get active. Thus,

T flip-flop is a controlled Bi-stable latch where the clock signal is the control signal. Thus, the output has two stable states based on the inputs which have been discussed below.

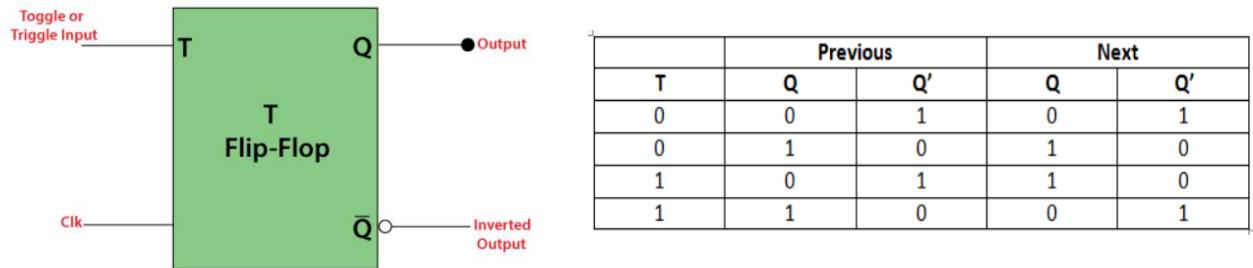


Fig 02: T Flip Flop and Truth table.

J-K Flip Flop: The JK flip flop is one of the most used flip flops in digital circuits. The JK flip flop is a universal flip flop having two inputs 'J' and 'K'. In SR flip flop, the 'S' and 'R' are the shortened abbreviated letters for Set and Reset, but J and K are not. The J and K are themselves autonomous letters which are chosen to distinguish the flip flop design from other types. The JK flip flop work in the same way as the SR flip flop work. The JK flip flop has 'J' and 'K' flip flop instead of 'S' and 'R'. The only difference between JK flip flop and SR flip flop is that when both inputs of SR flip flop is set to 1, the circuit produces the invalid states as outputs, but in case of JK flip flop, there are no invalid states even if both 'J' and 'K' flip flops are set to 1.

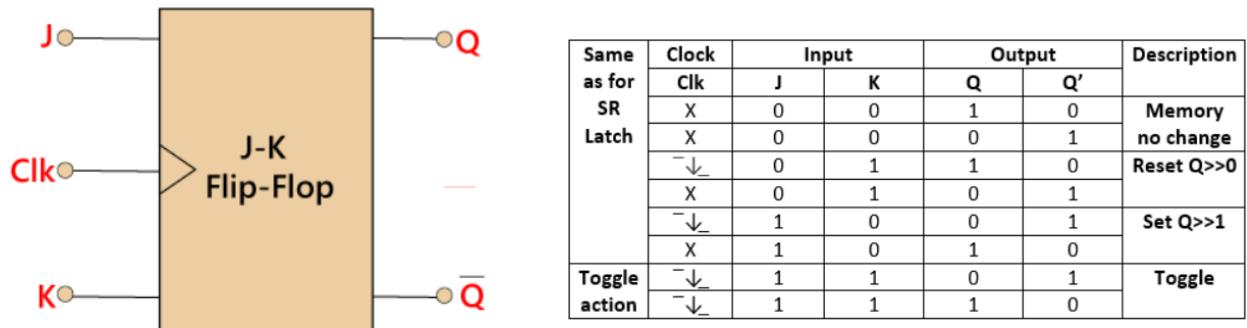
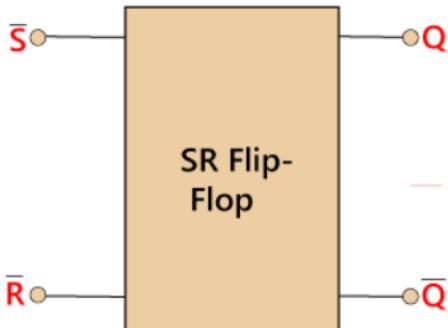


Fig 03: J-K Flip Flop and Truth table.

S-R Flip Flop: The SR flip flop is a 1-bit memory bistable device having two inputs, i.e., SET and RESET. The SET input 'S' set the device or produce the output 1, and the RESET input 'R' reset the device or produce the output 0. The SET and RESET inputs are labeled as S and R, respectively. The SR flip flop stands for "Set-Reset" flip flop. The reset input is used to get back the flip flop to its original state from the current state with an output 'Q'. This output depends on the set and reset conditions, which is either at the logic level "0" or "1". The NAND gate SR flip flop is a basic flip flop which provides feedback from both of its outputs back to its opposing input. This circuit is used to store the single data bit in the memory circuit. So, the SR flip flop has a total of three inputs, i.e., 'S' and 'R', and current output 'Q'. This output 'Q' is related to the current history or state. The term "flip-flop" relates to the actual operation of the device, as it can be "flipped" to a logic set state or "flopped" back to the opposing logic reset state.



State	S	R	Q	Q'	Description
Set	1	0	0	1	Set Q'>>1
	1	1	0	1	No change
Reset	0	1	1	0	Reset Q'>>0
	1	1	1	0	No change
Invalid	0	0	1	1	Invalid Condition

Fig 04: S-R Flip Flop and Truth table.

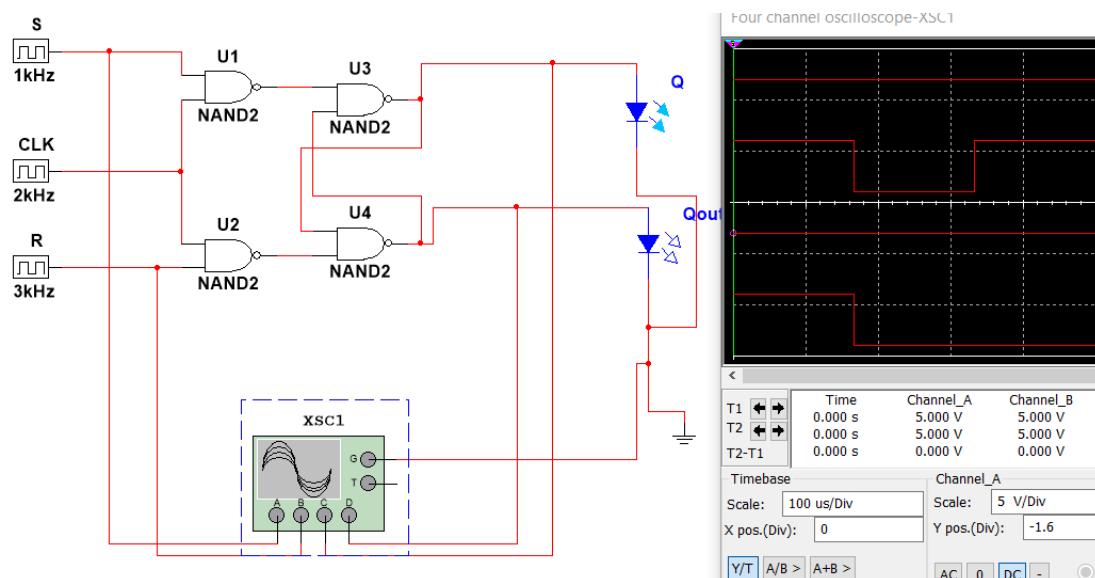
Apparatus:

IC: 7404 (NOT Gate), 7408 (AND Gate), 7432 (OR Gate), 7400 (NAND Gate), 7474 (D flip-flop), 7476 (J-K flip-flop).

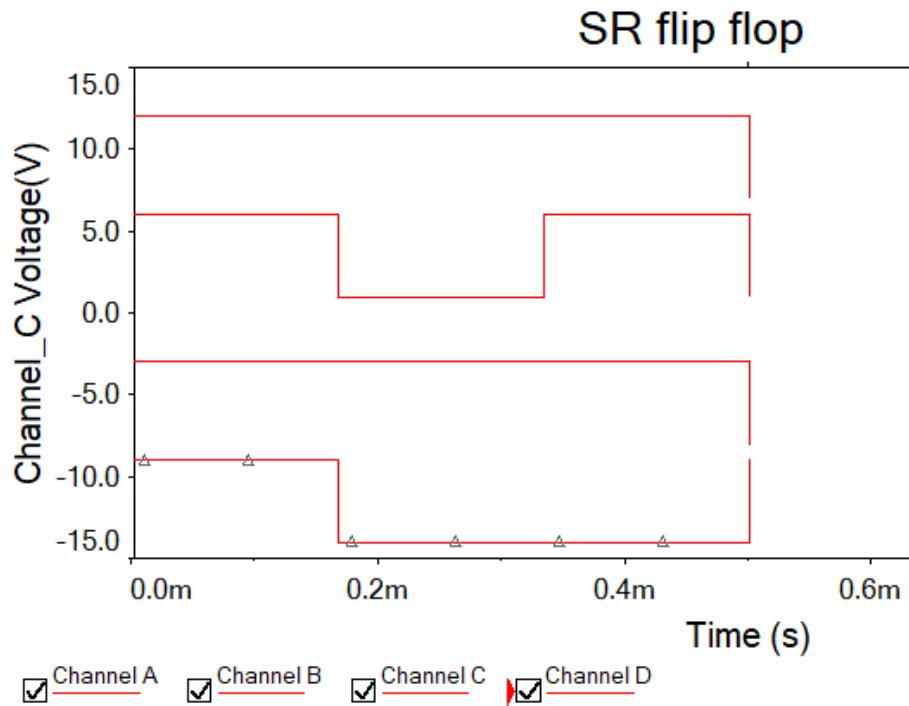
Simulation and Results:

S-R Flip Flop:

Simulation:



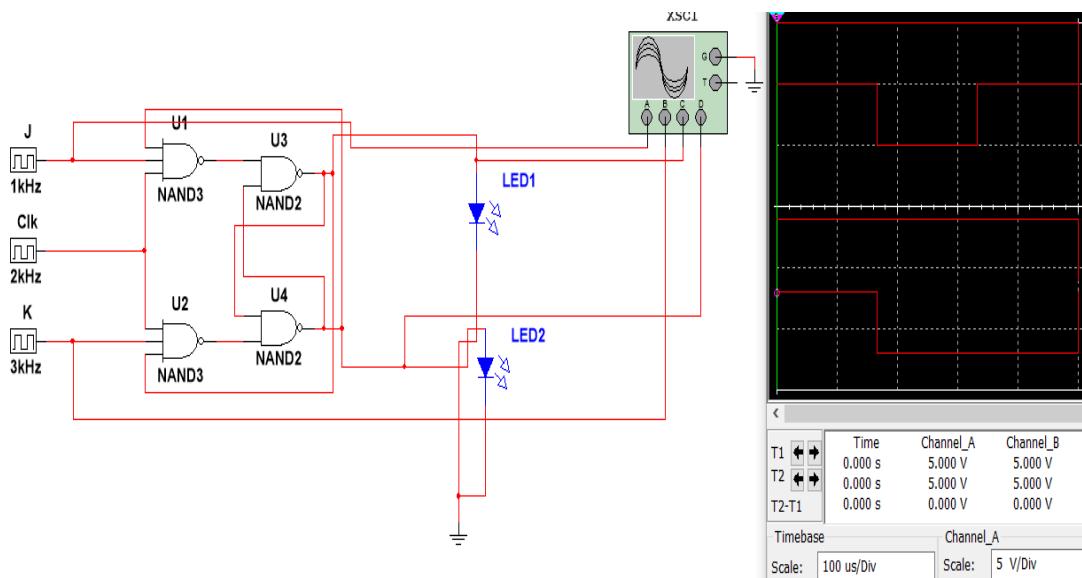
Graph:



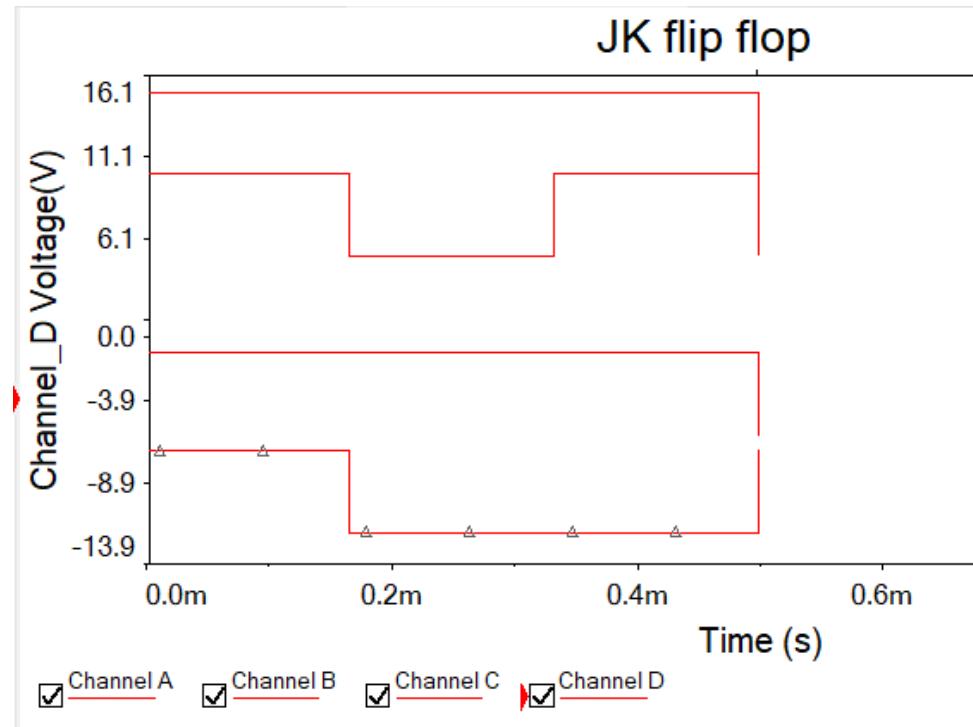
The simulation graph is not accurate because of the components configuration of Multisim.

J-K Flip Flop:

Simulation:



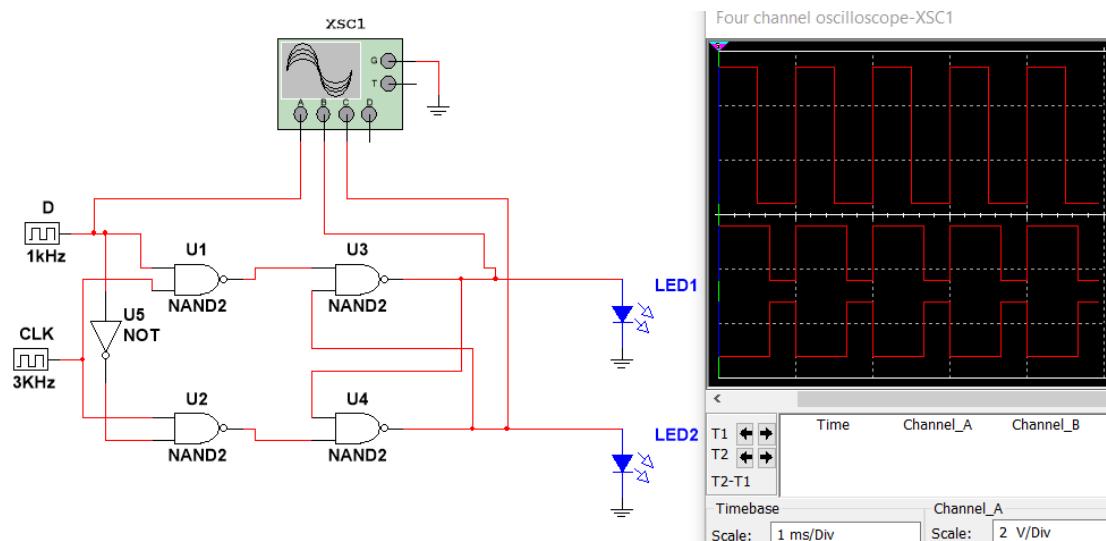
Graph:



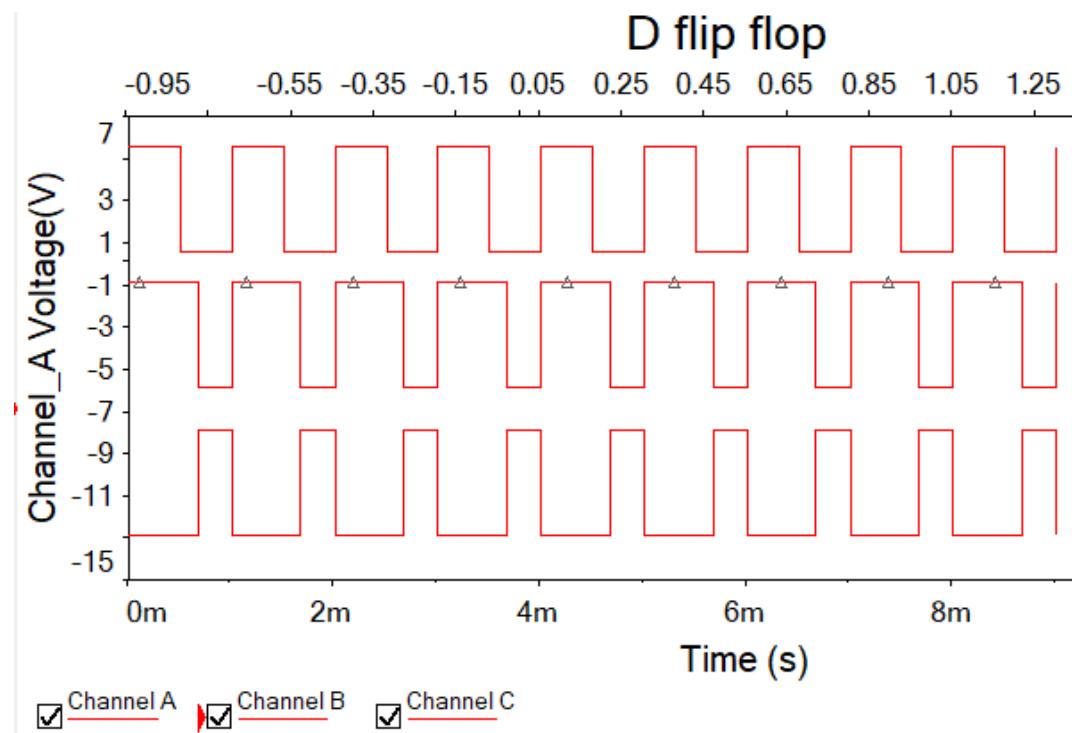
The simulation graph is not accurate because of the components configuration of Multisim.

D Flip Flop:

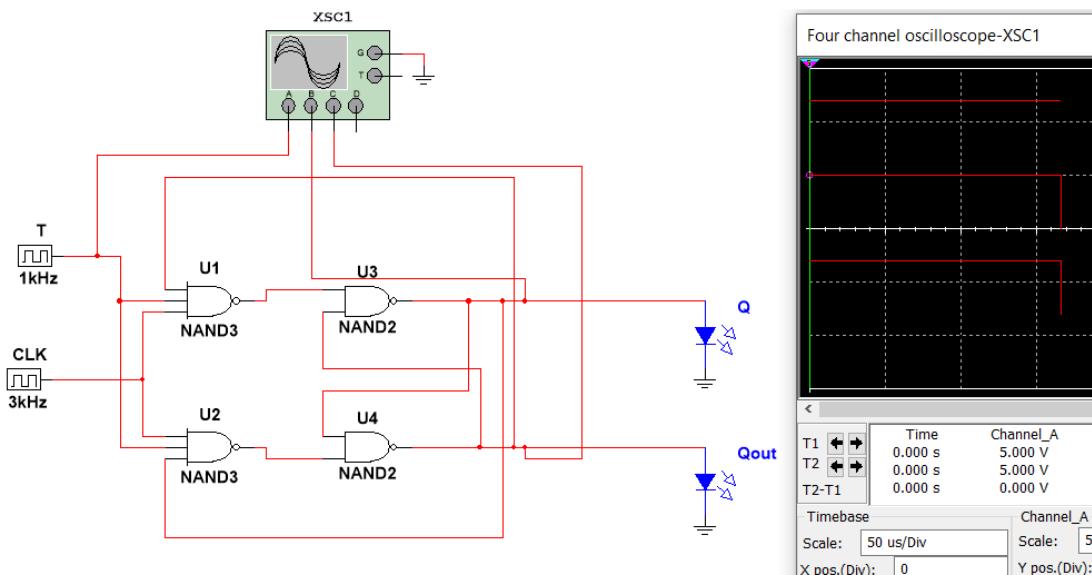
Simulation:



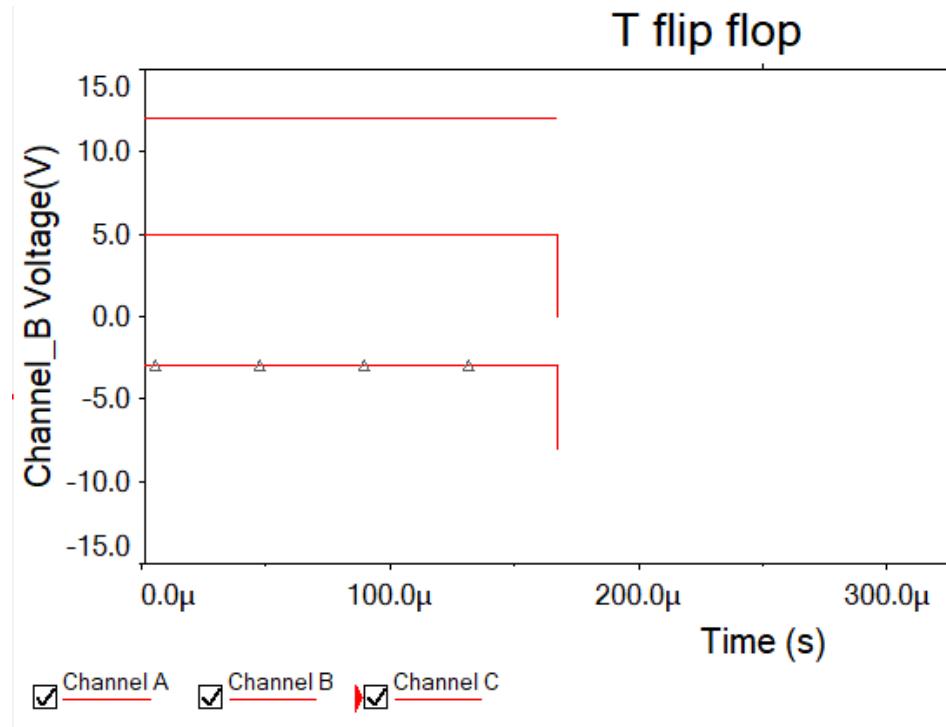
Graph:



T Flip Flop:
Simulation:



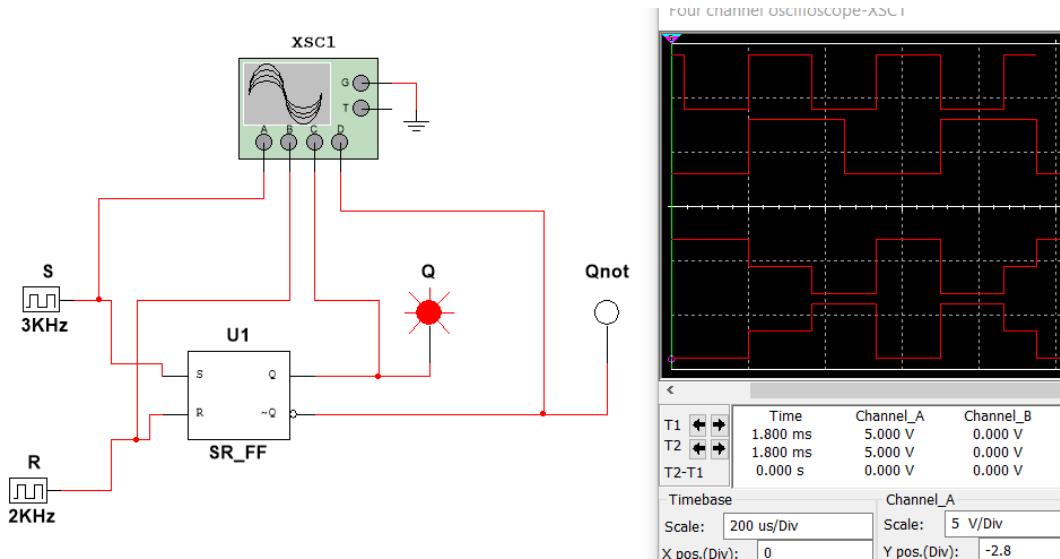
Graph:



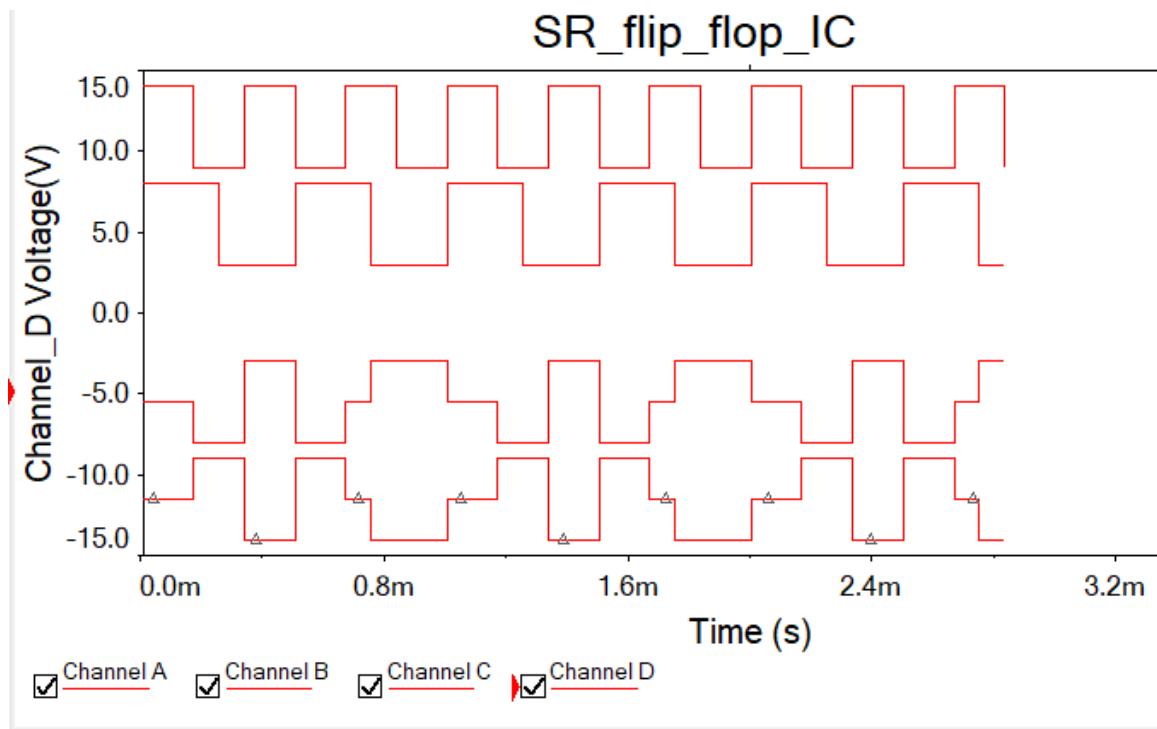
The simulation graph is not accurate because of the components configuration of Multisim.

S-R Flip Flop using IC:

Simulation:

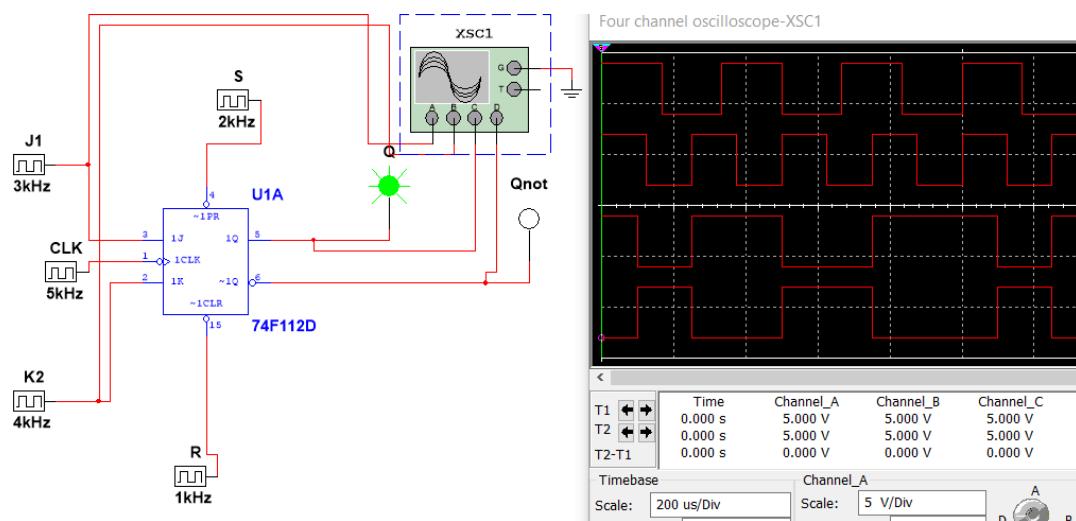


Graph:

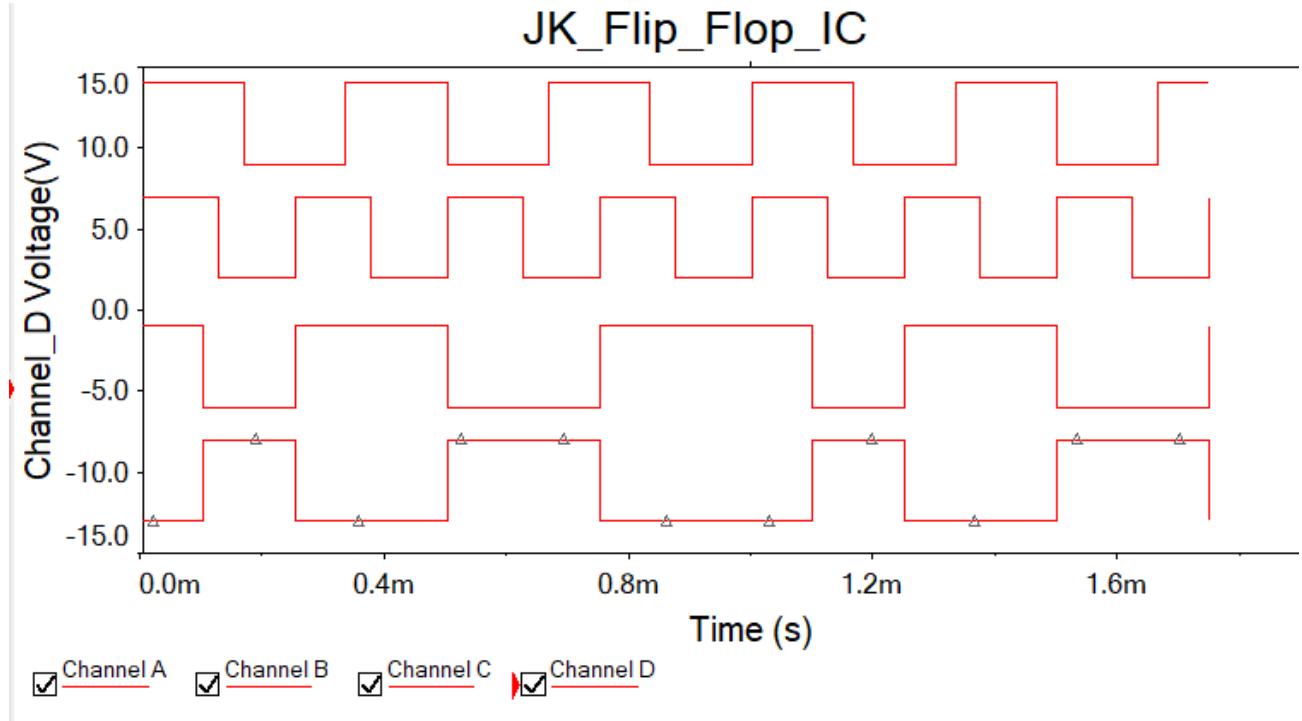


J-K Flip Flop using IC:

Simulation:

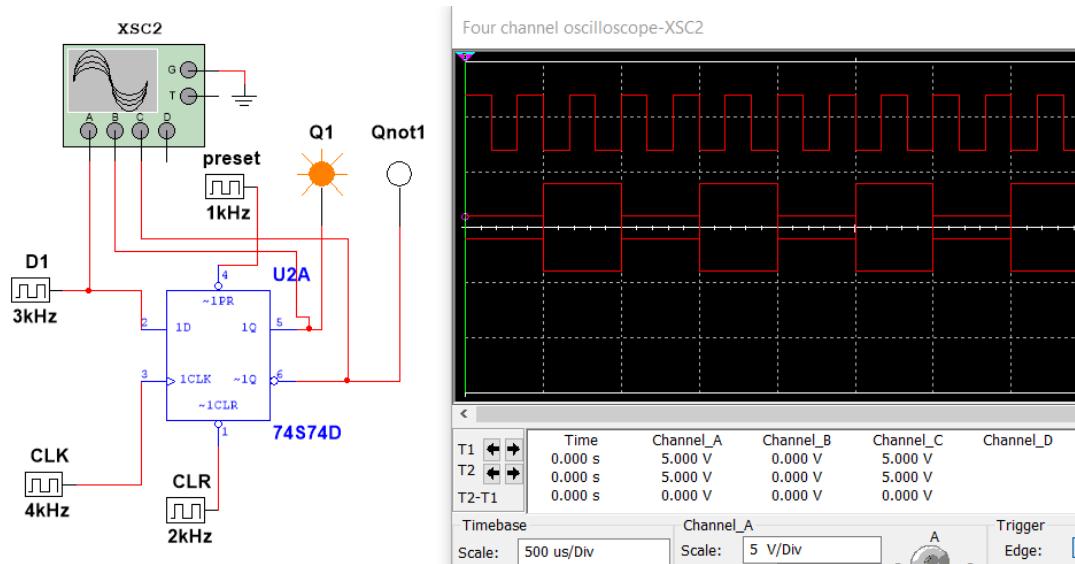


Graph:

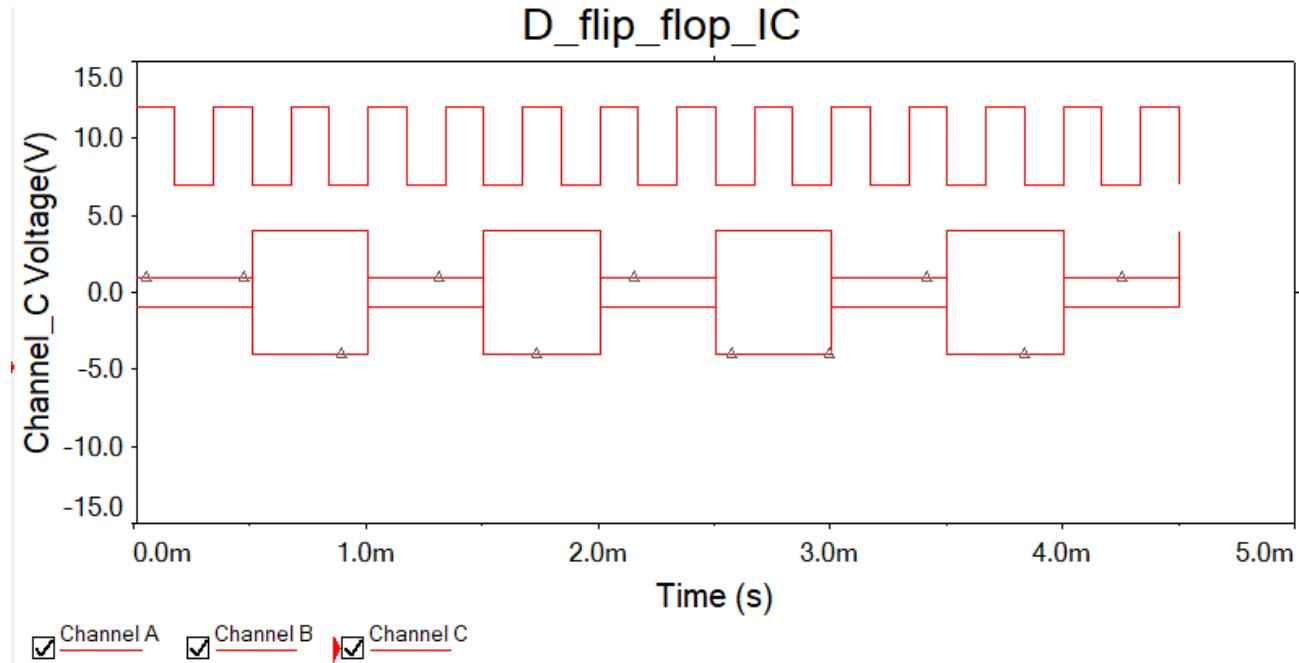


D Flip Flop using IC:

Simulation:

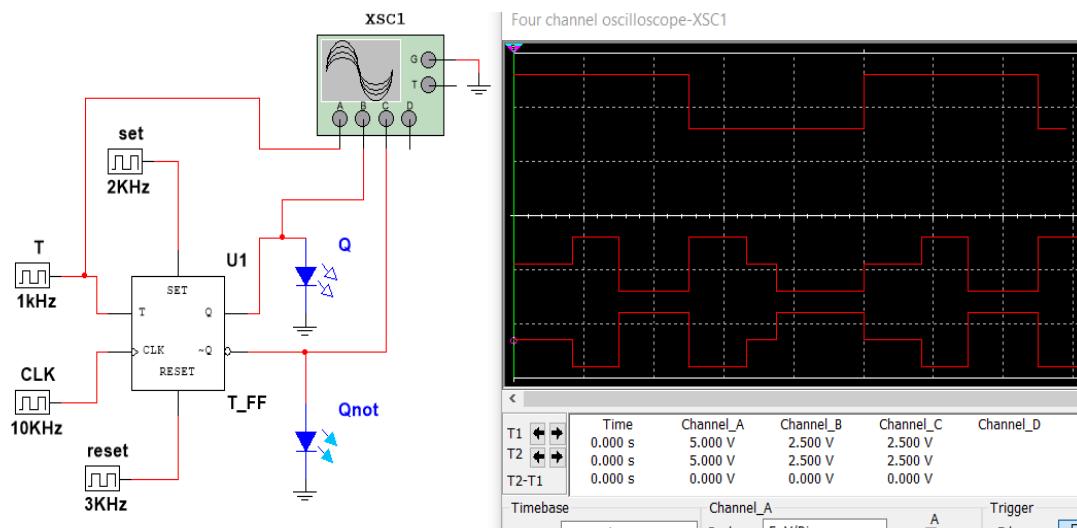


Graph:

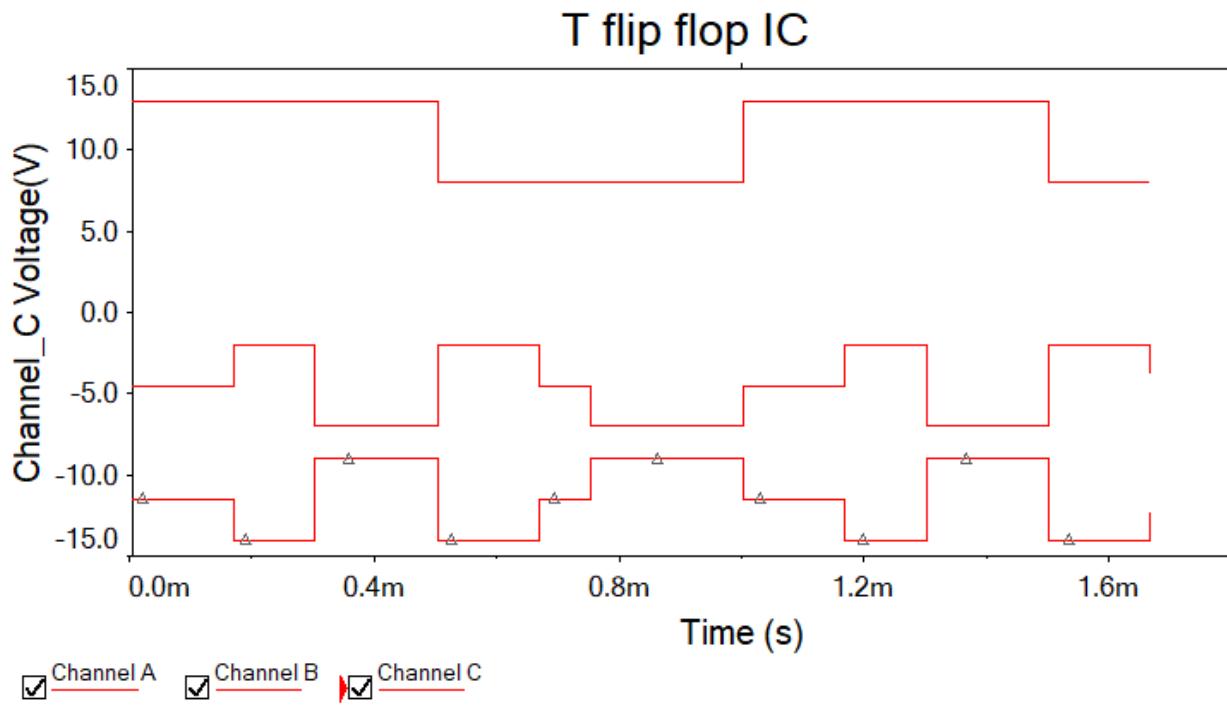


T Flip Flop using IC:

Simulation:

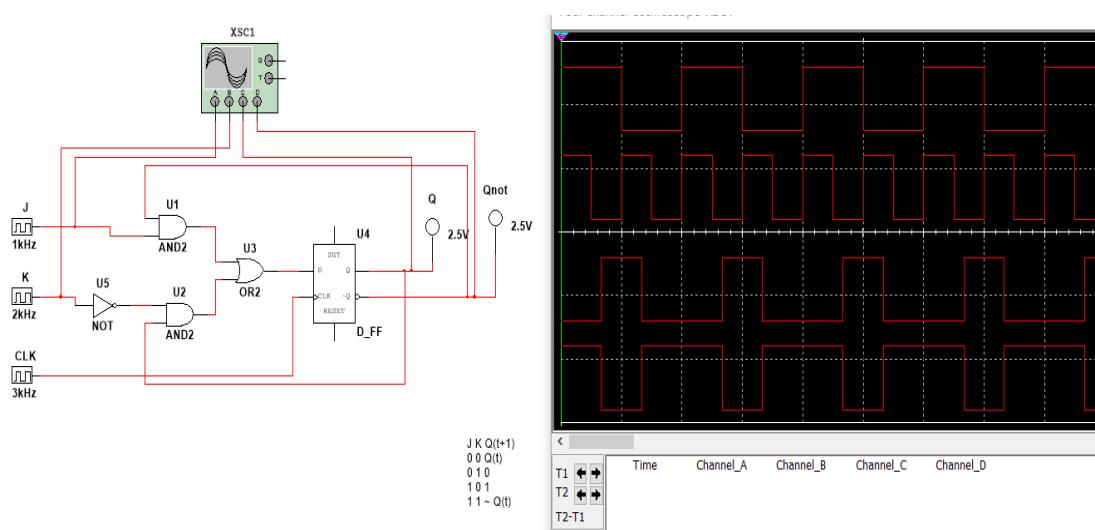


Graph:

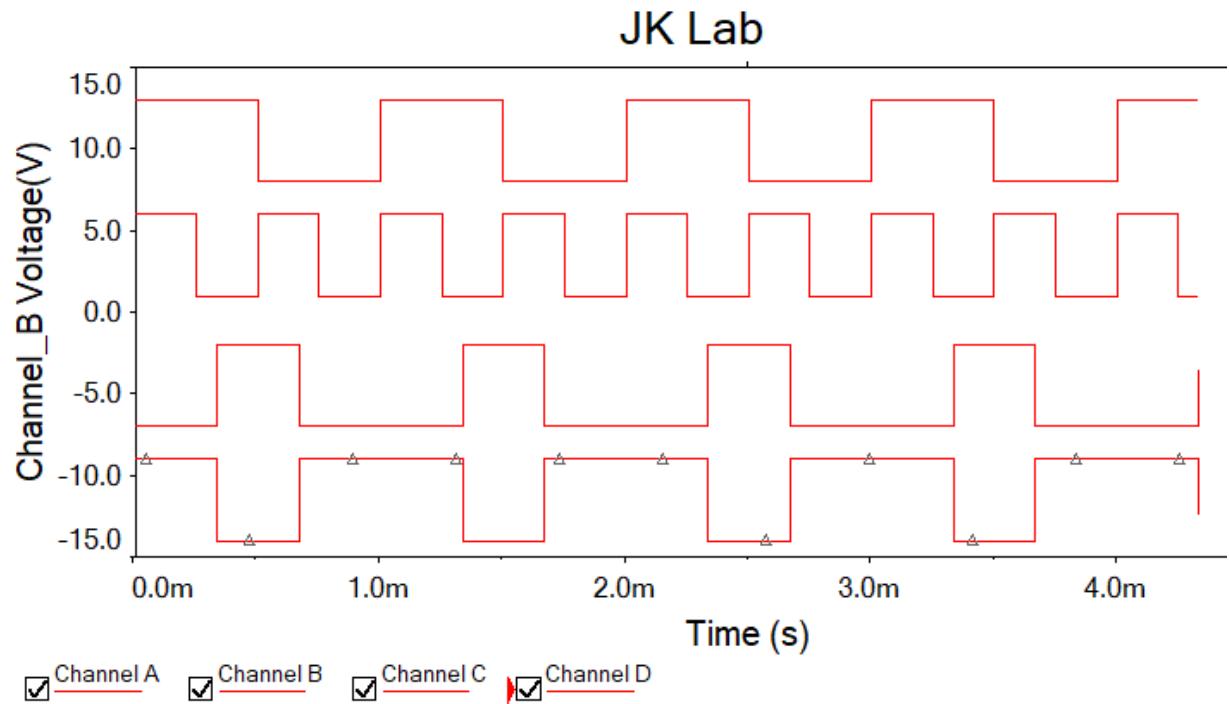


J-K Flip Flop (Lab manual):

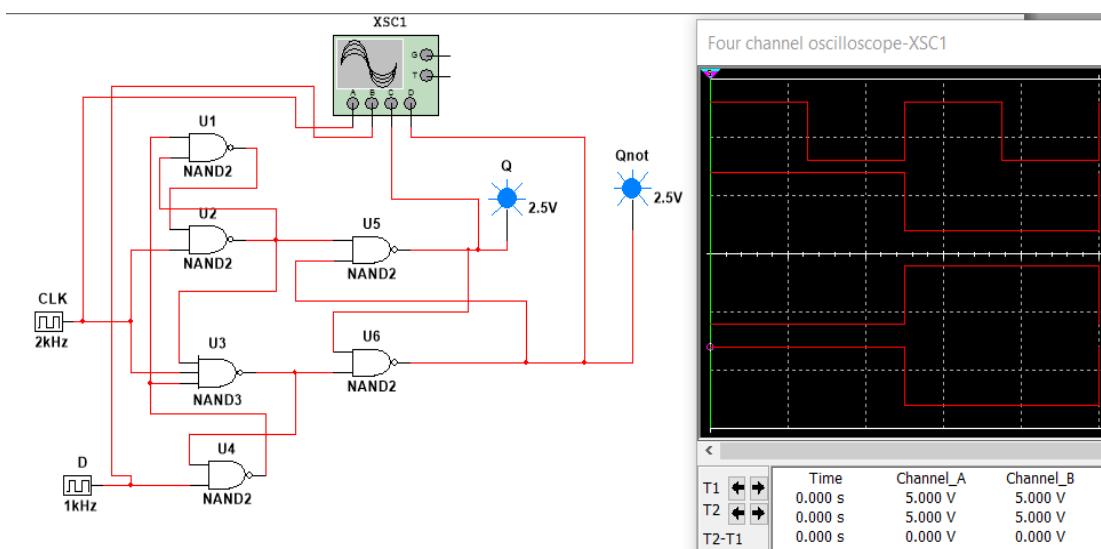
Simulation:



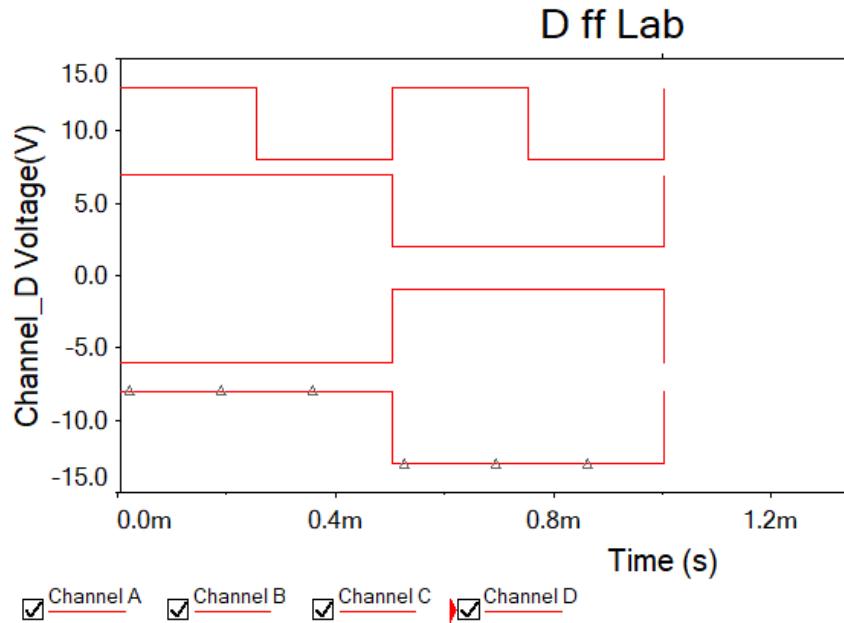
Graph:



D Flip Flop (Lab manual):
Simulation:



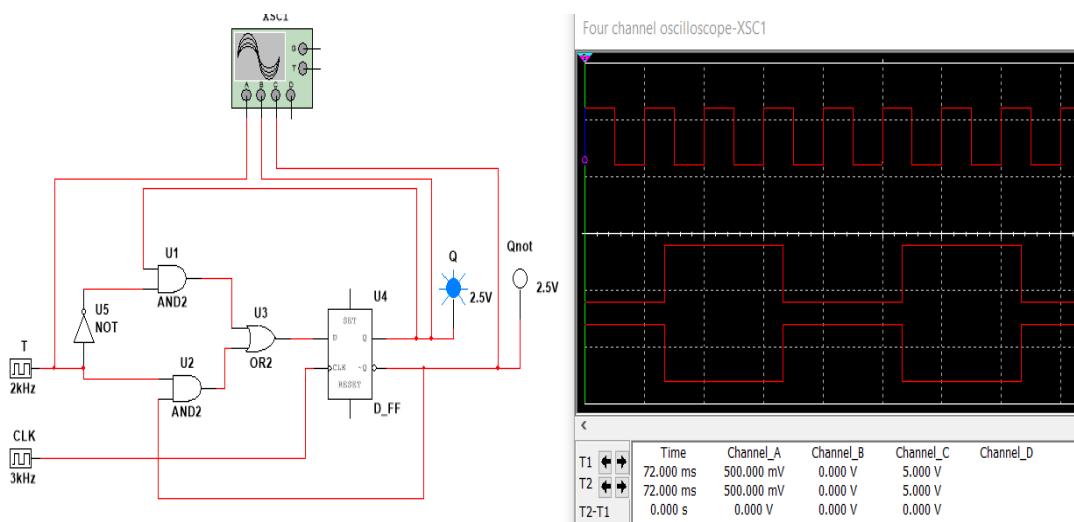
Graph:



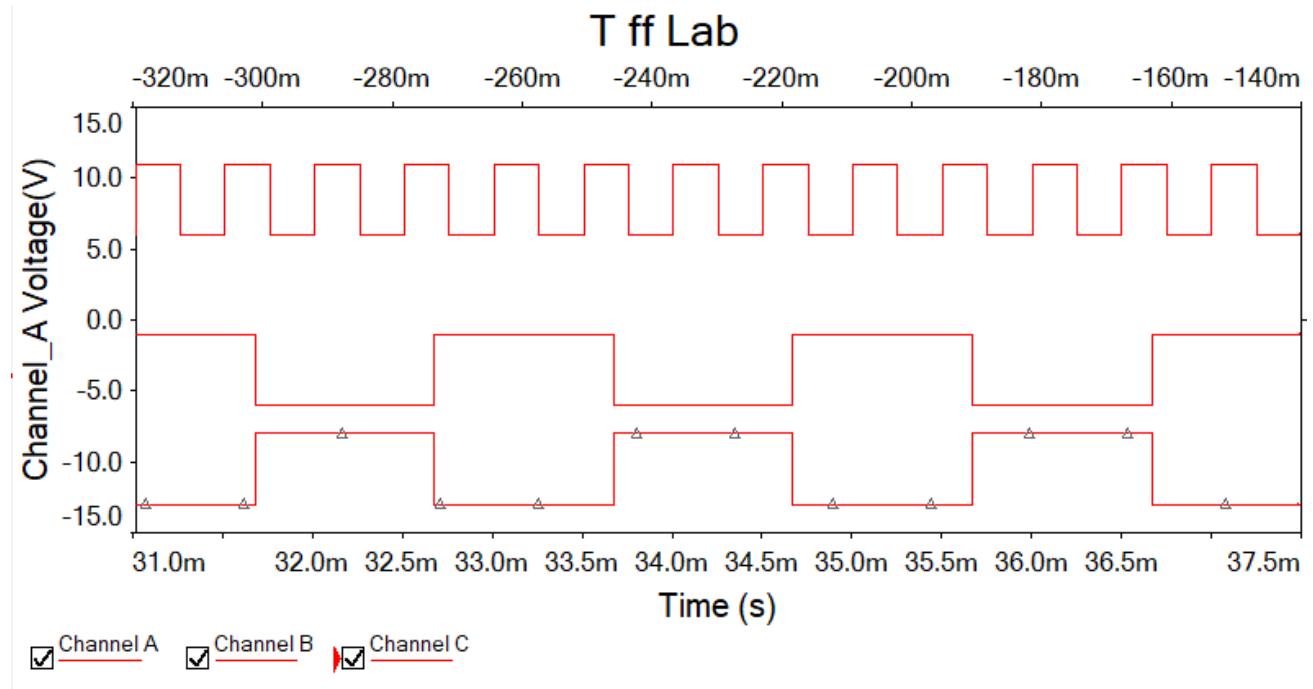
The simulation graph is not accurate because of the components configuration of Multisim.

T Flip Flop (Lab manual):

Simulation:



Graph:



Discussion: In this experiment our main aim was to know about all types of flip flop. Basically, a flip flop has two/three inputs. One input is a control input. For a D flip flop, the control input is labeled D. For a T flip flop, the control input is labelled T. For J-K flip flop the control inputs are J and K. The other input is the clock. All the simulations are done in NI Multisim. We used both IC and Gates for all type of flip flop circuit for measure the output curve. But some output is not as same as expected because of some tool error in Multisim. We used some sources for collecting the circuit models. After doing all of this we have done this experiment properly.

Reference:

- [1] "Fundamentals of Digital Logic with verilog design" by – Brown & Vranesic
- [2] <https://www.javatpoint.com/sr-flip-flop-in-digital-electronics>

AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh



Title: Implementation of Asynchronous and synchronous counters using flip-flops.

Experiment No: 09

Date of Submission: 12/08/2021

Course Title: Digital
Logic Design
Laboratory

Course Code: EEE2206

Section: J

Semester: Summer

2020-2021

Course Teacher: Sudipta Das

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* Student(s) must complete all details except the faculty use part.

** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.: 02

Submitted By: Patwary, Tanjib

No	Name	ID	Program	Signature
1	Shailee, Nowrin Muhammin	18-37259-1		
2	Bari, SK Tasnim	18-37201-1		
3	Patwary, Tanjib	18-37230-1		
4	Hoque, Fardin	18-37245-1		
5	Fahim, Ashik Ahmed	18-37269-1		
6				
7				
8				
9				
10				

Faculty use only

FACULTY COMMENTS

Marks Obtained

Total Marks

Title: Implementation of Asynchronous and synchronous counters using flip-flops.

Introduction:

A counter is a device that keeps track of (and occasionally displays) the number of times a specific event or process has occurred, usually in conjunction with a clock. The most common type is a sequential digital logic circuit, which has numerous output lines and a clock input line. We commonly refer to counters as "Timers." The best example of flip flop applications is counter circuits. Counters are created by combining flip flops together and sending them a single clock signal. The number of states possible in a counter when N flip-flops are utilized is 2^N . The counter is considered to be a modulo K (or MOD K) counter if it cycles through K of these states, where $K \leq 2^N$.

According to how they are clocked, counters are divided into two categories: asynchronous and synchronous. In Asynchronous Counter, also known as Ripple Counter, different flip flops are triggered with different clocks, not simultaneously. In synchronous counter, all flip flops are triggered with same clock simultaneously.

Objective:

To design the following counters using J -K Flip-Flops (IC 74LS76)

- a) n-bit Binary Asynchronous Counter
- b) n-bit Binary Synchronous Counter

Theory and Methodology:

Asynchronous Counter:

Asynchronous counters are those whose output is free from the clock signal. Because the flip flops in asynchronous counters are supplied with different clock signals, there may be delay in producing output.

3-bit Asynchronous Counter:

It contains three flip flops. A 3-bit ripple counter can count up to 8 states. It counts from 0 to 7 (up) or 7 to 0 (down).

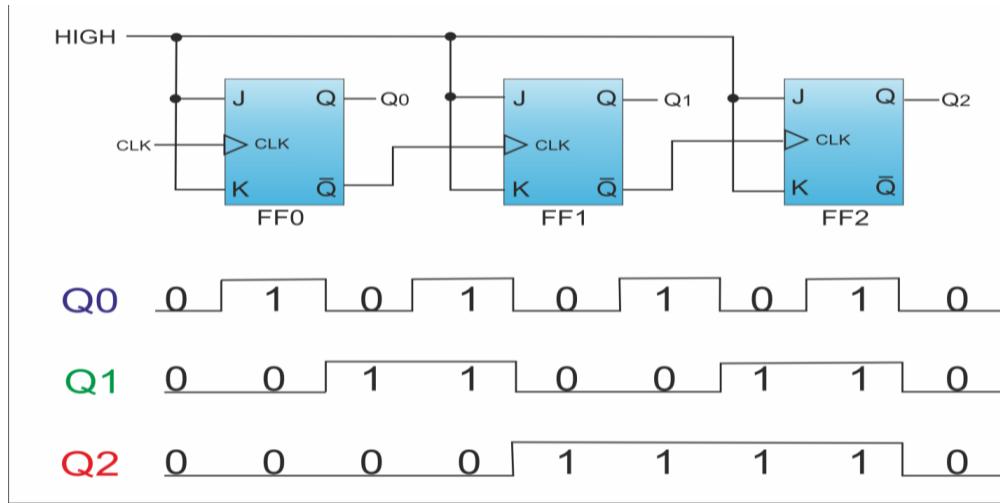


Figure 1: 3-bit Asynchronous Counter with timing diagram

4-bit Asynchronous Counter:

It counts from binary 0000 to 1111.

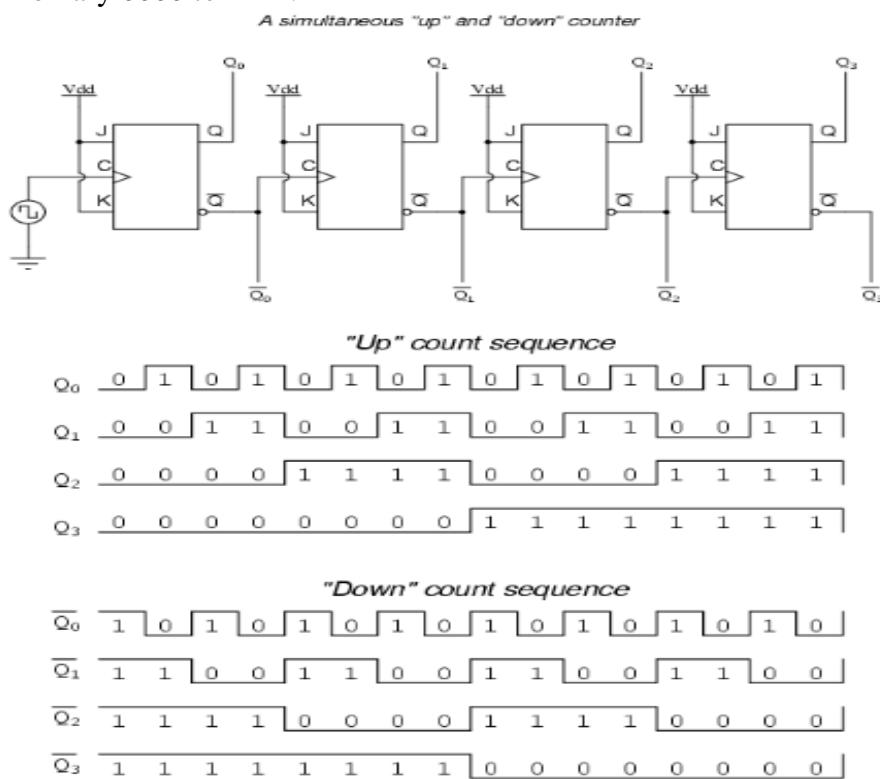


Figure 2: 4-bit Asynchronous Counter (Up & Down) with timing diagram

Synchronous Counter:

A counter consisting of an interconnected series of flip-flops in which all the flip-flop outputs change state at the same instant, normally on application of a pulse at the counter input.

3-bit Synchronous Counter:

In the 3-bit synchronous counter, three j-k flip-flops are used.

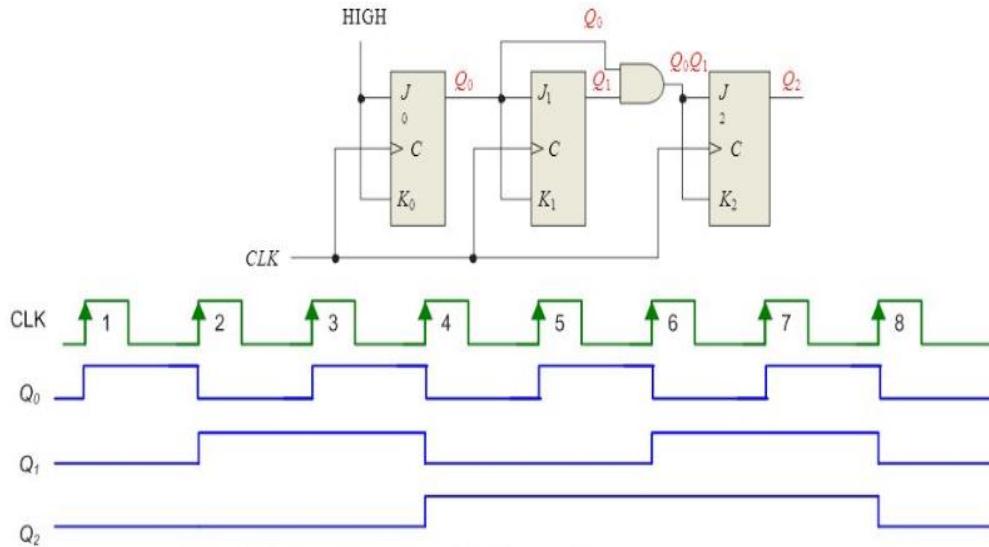


Figure 3: 3-bit Synchronous Counter with timing diagram

4-bit Synchronous Counter:

This counter has two modes of counting i.e. up counting and down counting. There is a mode switch which switches between the two modes of the counter. When the mode $M = 0$ it counts up & when mode $M = 1$ then it counts down.

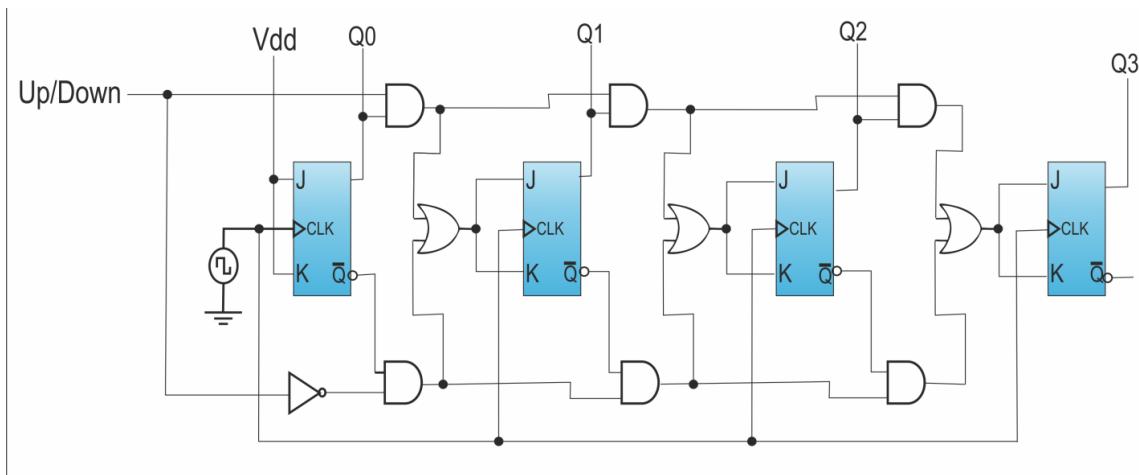


Figure 4: 4-bit Synchronous Counter

Pin Configuration of 74LS76 and 7408:

74LS76 is a negative edge-triggered J-K flip-flop.

There are 2 J-K Flip Flops in one IC. Here is the pin configuration of the IC 74LS76:

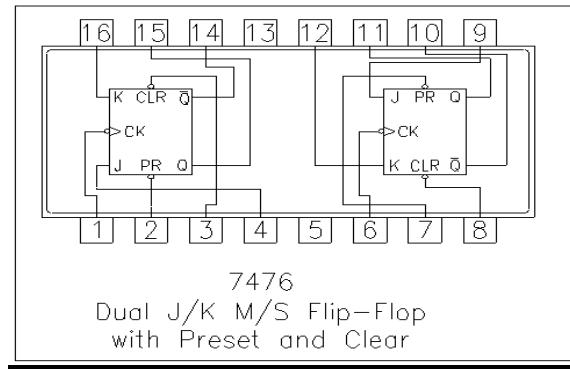


Figure 5: IC 74LS76

IC 7408 contains 4 AND gates in it. The pin configuration is shown below:

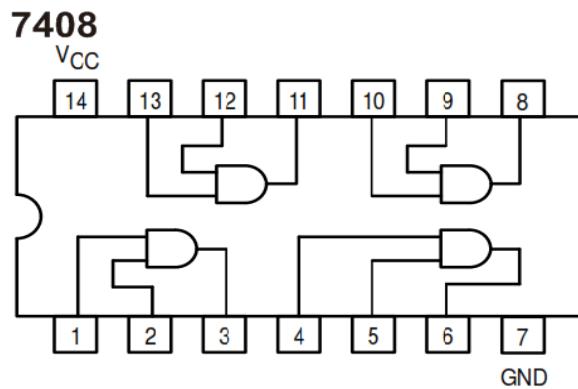


Figure 6: IC 7408

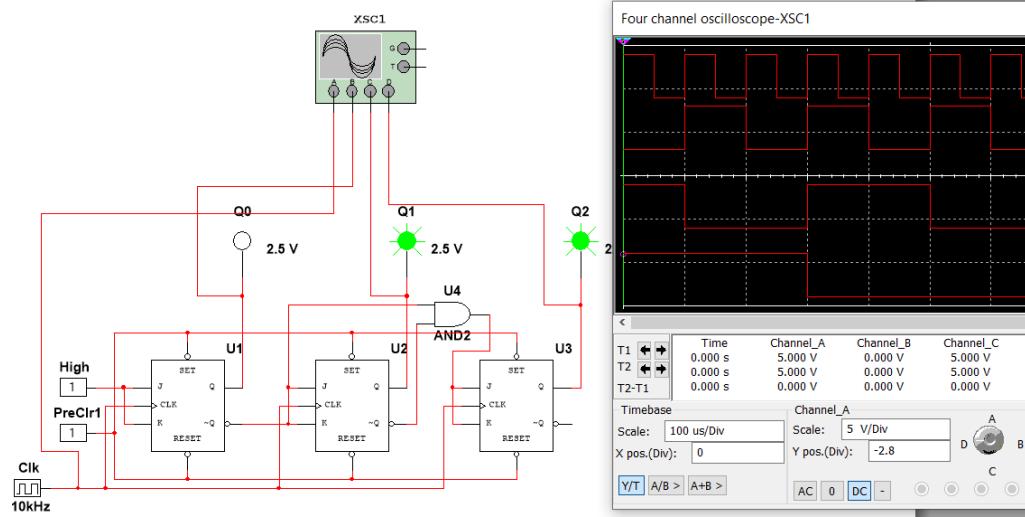
Apparatus:

- 1) IC 74LS76 (JK Flip Flop)
- 2) IC 7408 (AND Gate)
- 3) LED Lamps or Display
- 4) Trainer Board
- 5) Oscilloscope
- 6) Connecting Wires

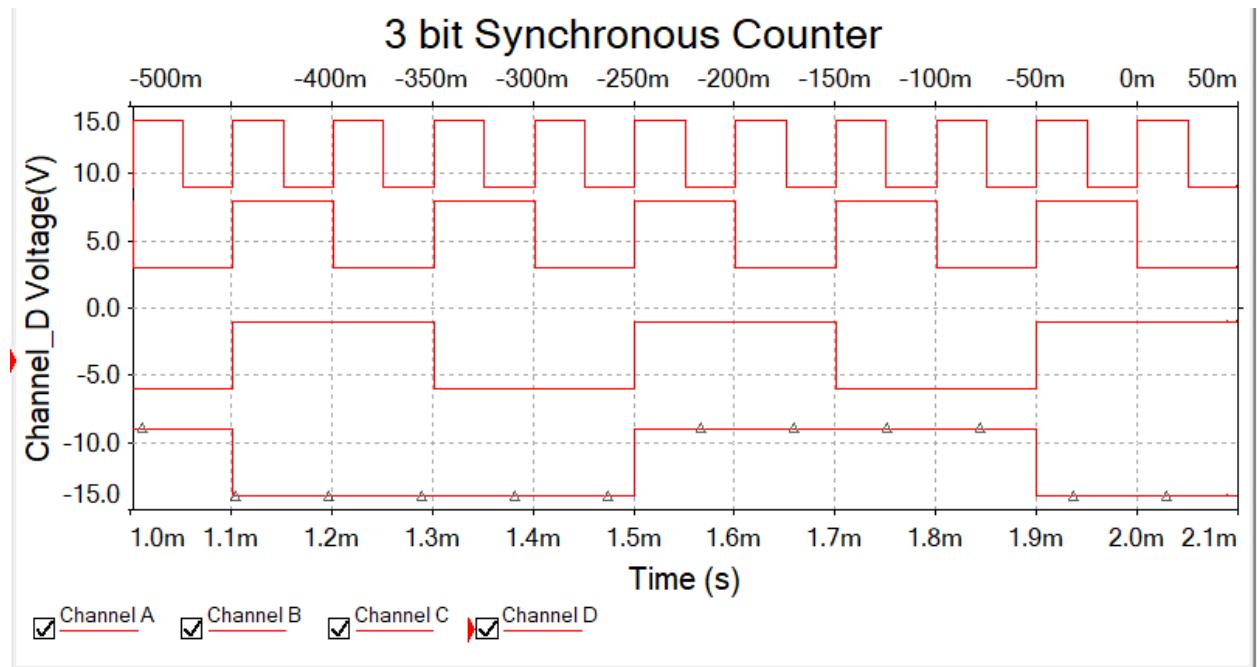
Simulation and Results:

1. 3 bit Synchronous Counter:

Simulation:

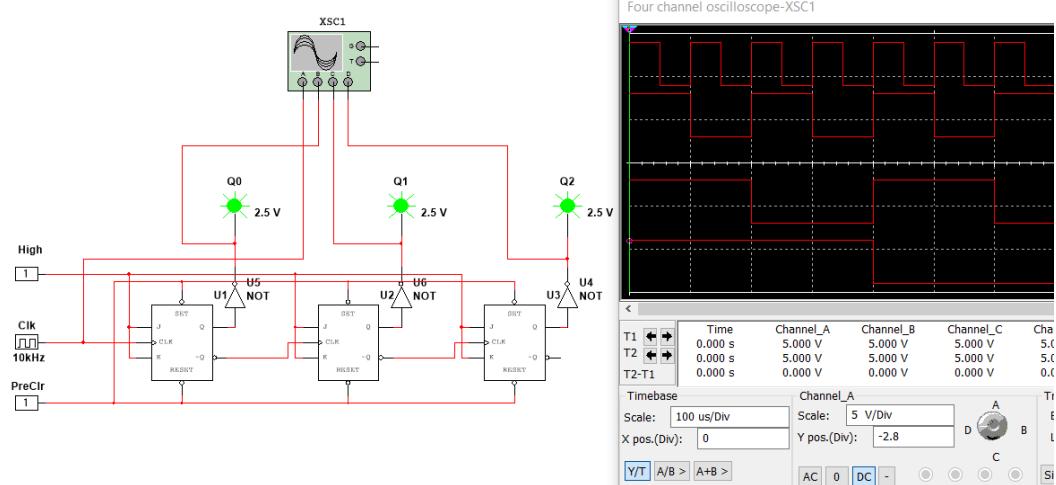


Graph:

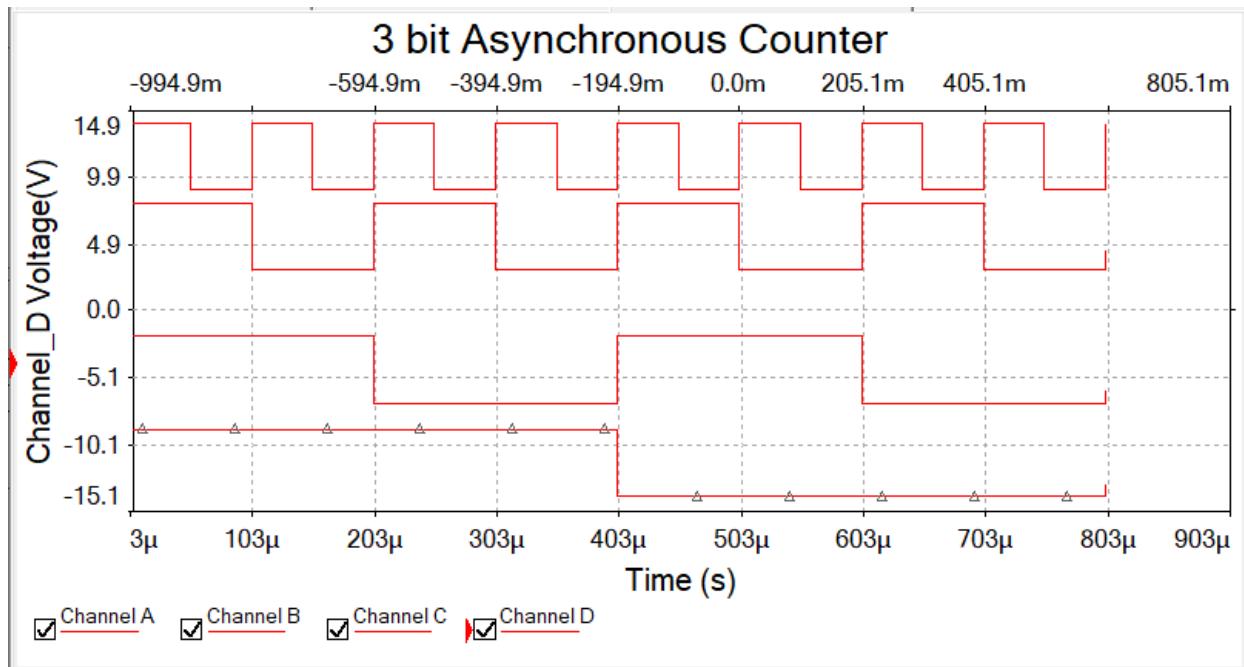


2. 3 bit Asynchronous Counter:

Simulation:



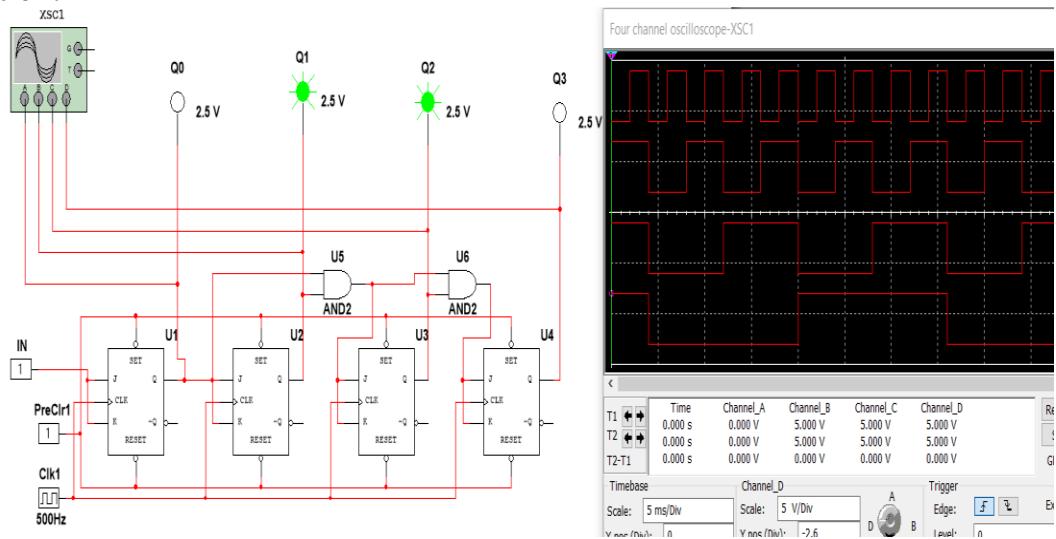
Graph:



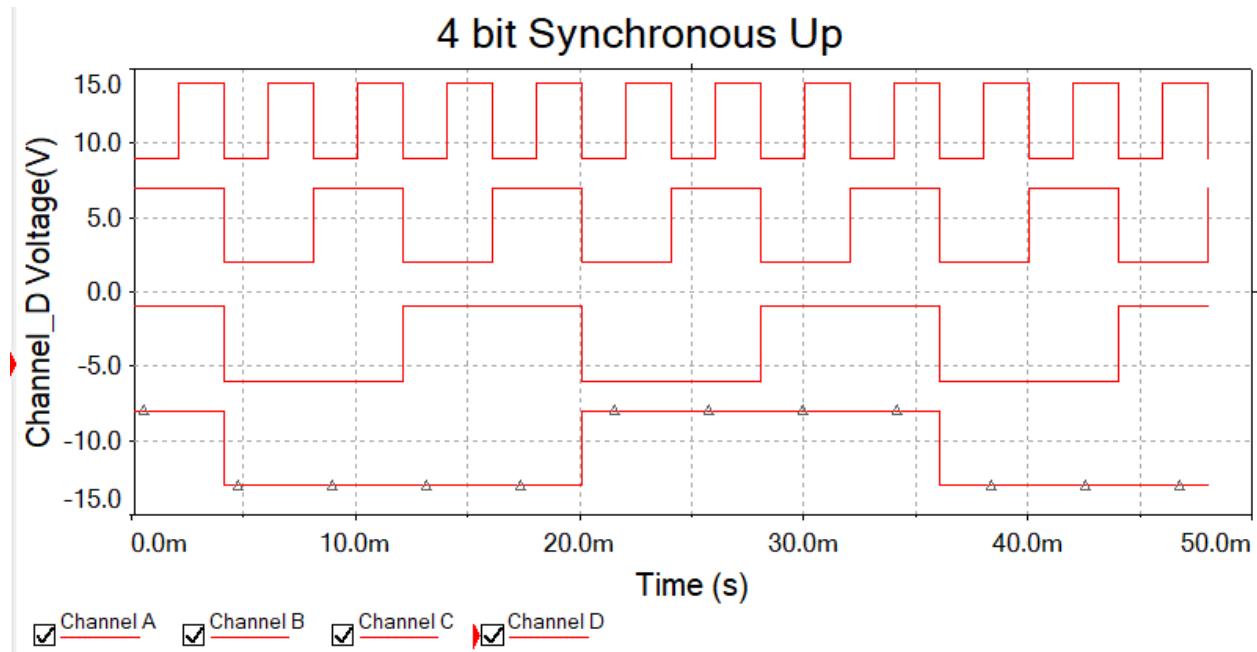
Questions with answers for report writing:

1. 4 bit Synchronous Counter:

Simulation:

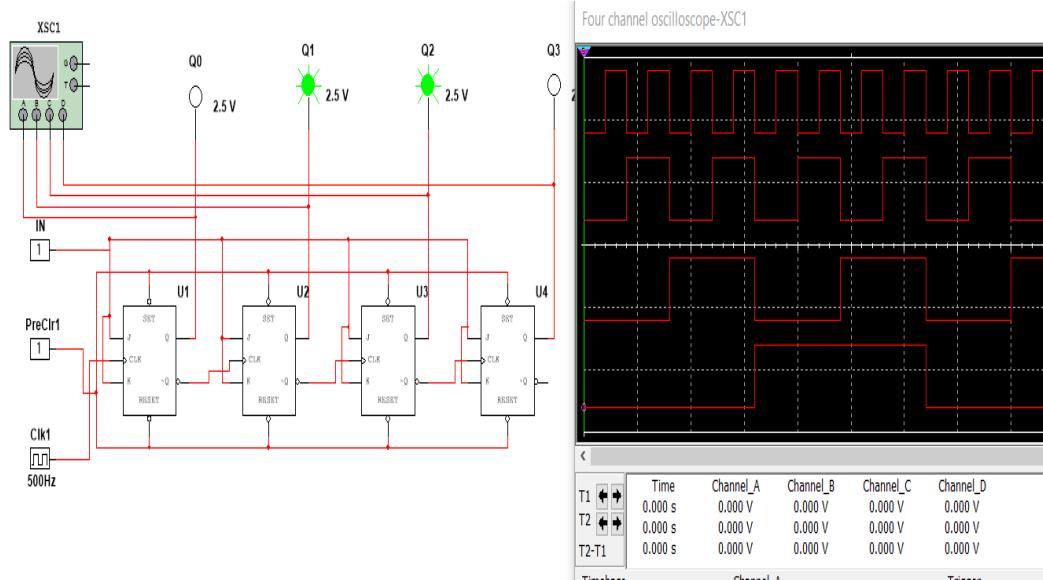


Graph:

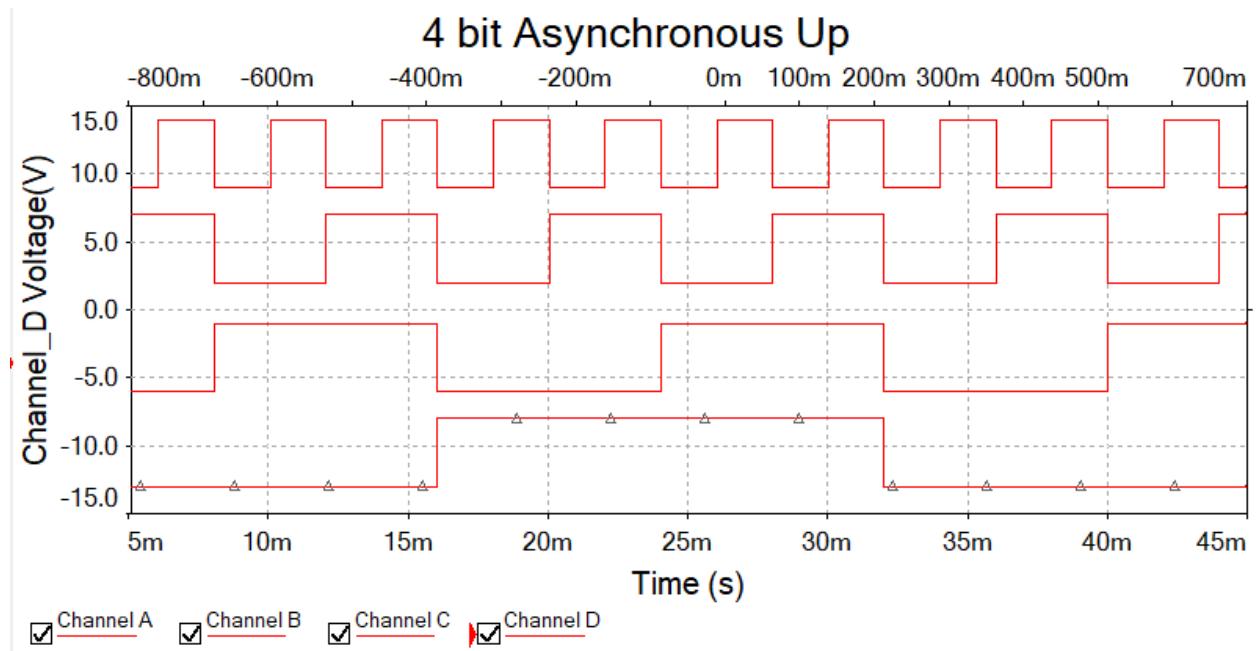


2. 4 bit Asynchronous Counter:

Simulation:



Graph:



Discussion and Conclusion:

In this experiment our main aim was to know about the synchronous and asynchronous counter. Where in synchronous counters, the clock inputs of all the flip-flops are connected together and are triggered by the input pulses. And in asynchronous counters, all the flip flop has different clock pulse. We done all the simulation in Multisim software. We mainly used both JK Flip Flop IC. And we got the correct output curve for all the circuits also. We used some sources for collecting the circuit models. After doing all of this we have done this experiment properly.

Reference:

- 1) Thomas L. Floyd, “Digital Fundamentals”, Ninth Edition
- 2) <https://learn.circuitverse.org/docs/seq-msi/counters.html>
- 3) <https://workforce.libretexts.org>

AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh



Title: Implementation of shift registers using flip-flops.

Experiment No: 10

Date of Submission: 12/08/2021

Course Title: Digital
Logic Design
Laboratory

Course Code: EEE2206

Section: J

Semester: Summer

2020-2021

Course Teacher: Sudipta Das

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* Student(s) must complete all details except the faculty use part.

** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.: 02

Submitted By: Bari, SK. Tasnim

No	Name	ID	Program	Signature
1	Shailee, Nowrin Muhaimin	18-37259-1		
2	Bari, SK Tasnim	18-37201-1		
3	Patwary, Tanjib	18-37230-1		
4	Hoque, Fardin	18-37245-1		
5	Fahim, Ashik Ahmed	18-37269-1		
6				
7				
8				
9				
10				

Faculty use only

FACULTY COMMENTS

Marks Obtained

Total Marks

Title: Implementation of shift registers using flip-flops.

Abstract:

The objective of this experiment was to learn how flip-flops can be implemented in different types of shift registers. The experiment was completely done on a simulation software. The software was Multisim.

Introduction:

A flip-flop stores one bit of information. When a set of n-flip-flops is used to store n-bits of information, such as an n-bit number, we refer to these flip-flops as a register. A common clock is used for each flip-flop in a register, and each flip-flop operates as described in the previous sections. The term register is merely a convenience for referring to n-bit structures consisting of flip-flops.

The objective of this experiment is designing of the following Shift Registers using J-K Flip-flops (IC 74LS76): Serial In/ Serial Out (SISO) Shift Register, Serial In / Parallel Out (SIPO) Shift Register, Parallel In / Serial Out (PISO) Shift Register, Parallel In / Parallel Out (PIPO) Shift Register , and Bidirectional Shift Registers.

Theory and Methodology:

Shift Registers

In digital circuits, a **shift register** is a cascade of flip flops, sharing the same clock, in which the output of each flip-flop is connected to the "data" input of the next flip-flop in the chain, resulting in a circuit that shifts by one position the "bit array" stored in it, *shifting in* the data present at its input and *shifting out* the last bit in the array, at each transition of the clock input. More generally, a **shift register** may be multidimensional, such that its "data in" and stage outputs are themselves bit arrays: this is implemented simply by running several shift registers of the same bit-length in parallel.

Shift registers can have both parallel and serial inputs and outputs. These are often configured as **serial-in, parallel-out** (SIPO) or as **parallel-in, serial-out** (PISO). There are also types that have both serial and parallel input and types with serial and parallel output. There are also **bi-directional** shift registers which allow shifting in both directions: L→R or R→L. The serial input and last output of a shift register can also be connected to create a **circular shift register**. The operations of these different types of shift registers are explained below.

Serial In/ Serial Out (SISO) Shift Register

Figure 10.1(a) shows a four-bit shift register that is used to shift its contents one bit-position to the right. The data bits are loaded into the shift register in a serial fashion using the In input. The contents of each flip-flop are transferred to the next flip-flop at each positive edge of the clock. An illustration of the transfer is given in Figure 10.1(b), which shows what happens when the signal values at *In* during eight consecutive clock cycles are 1, 0, 1, 1, 1, 0, 0, and 0, assuming that the initial state of all flip-flops is 0.

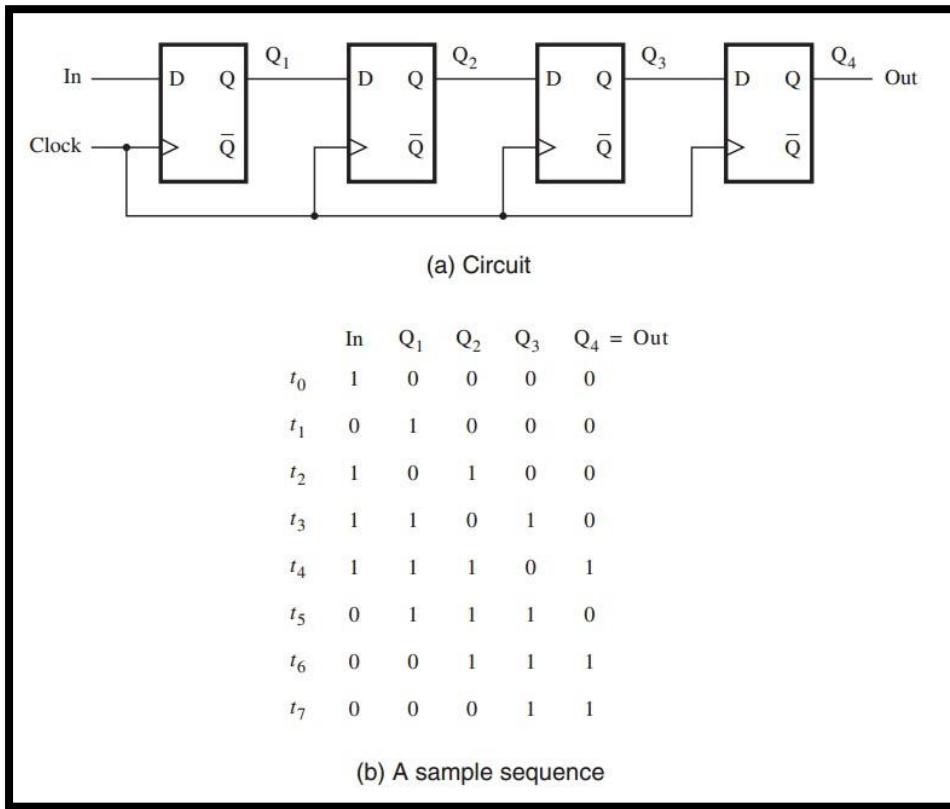


Figure 10.1: Serial In/ Serial Out (SISO) Shift Register

Serial In / Parallel Out (SIPO) Shift Register

In this type of Shift register, data bits are entered serially in the same manner as discussed in SISO but the output is taken from each stage as shown in the diagram below.

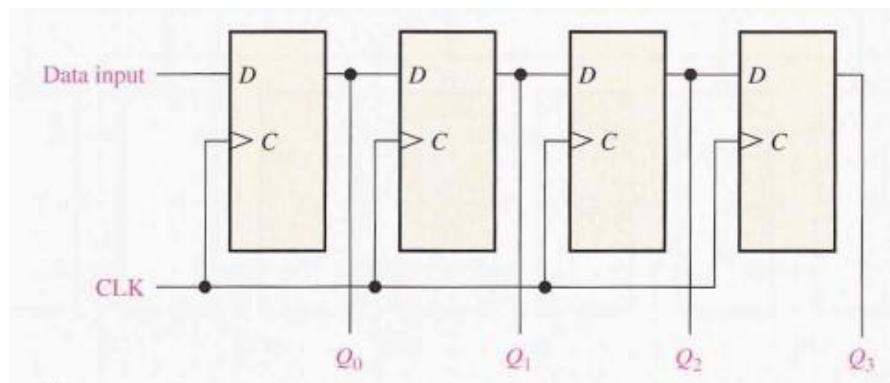
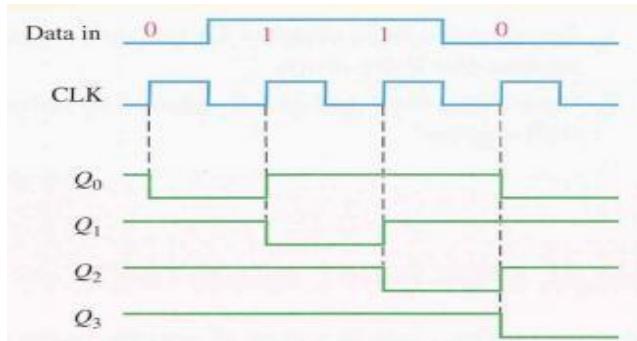


Figure 10.2(a): Serial In / Parallel Out (SIPO) Shift Register



**Figure 10.2(b): Timing Diagram of Serial In / Parallel Out (SIPO) Shift Register
Parallel In / Parallel Out (PIPO) Shift Register**

In this type of shift register, data bits are entered simultaneously in all the flip-flops and the bits appear on the parallel outputs as shown in the figure below.

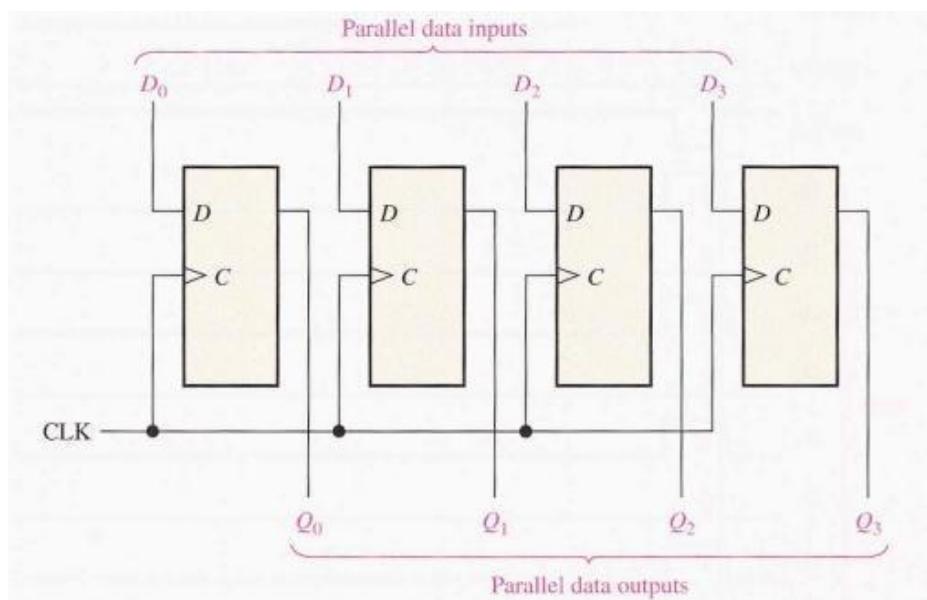


Figure 10.3: Parallel In / Parallel Out Shift Register

Parallel In / Serial Out (PISO) Shift Register

In computer systems it is often necessary to transfer n-bit data items. Transmitting all bits at once using n separate wires, in which case we say that the transfer is performed in parallel, may do this. But it is also possible to transfer all bits using a single wire, by performing the transfer one bit at a time, in n consecutive clock cycles. We refer to this scheme as serial transfer. To transfer an n-bit data item serially, we can use a shift register that can be loaded with all n bits in parallel (in one clock cycle). Then during the next n clock cycles, the contents of the register can be shifted out for serial transfer. The reverse operation is also needed. If bits are received serially, then after n clock cycles the contents of the register can be accessed in parallel as an n-bit item.

Figure 10.4 shows a four-bit shift register that allows the parallel access. Instead of using the normal shift register connection; the D input of each flip-flop is connected to two different sources. One source is the preceding flip-flop, which is needed for the shift-register operation. The other source is the external input that corresponds to the bit that is to be loaded into the flip-flop as a part of the parallel-load operation. The control signal Shift/Load is used to select the mode of operation. If Shift/Load=0, then the circuit operates as a shift register. If Shift/Load=1, then the parallel input data are loaded into the register. In both cases the action takes place on the positive edge of the clock.

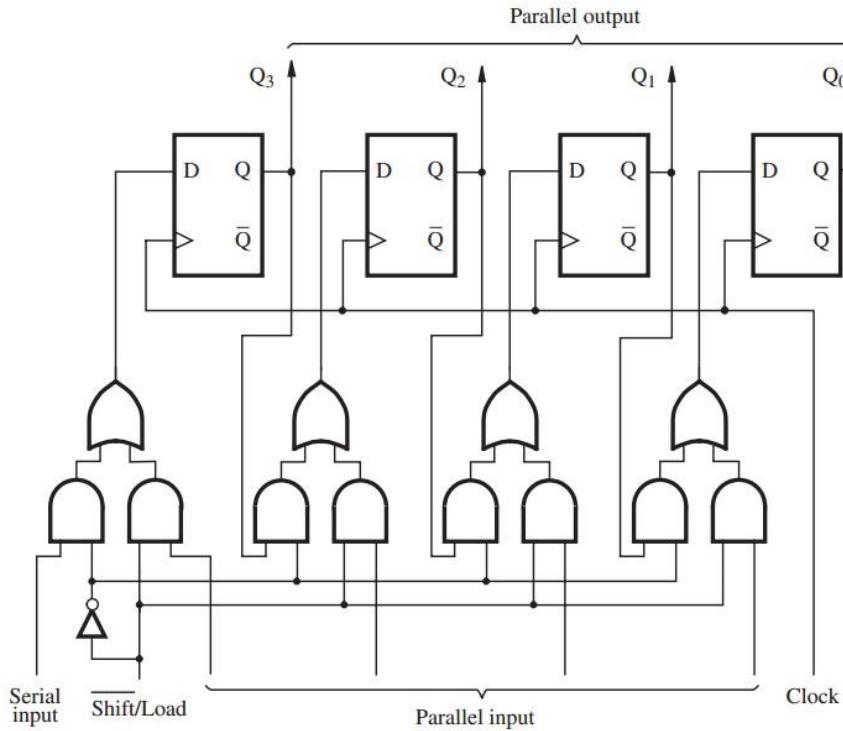


Figure 10.4: Parallel In / Serial Out (PISO) Shift Register

Bidirectional Shift Registers

A bidirectional shift register is one in which the data can be shifted either left or right as shown in Figure 10.5(a). A High on the *RIGHT/LEFT* control input allows data bits inside the register to be shifted to the right and a Low enables data bits inside the register to be shifted to the left. When the *RIGHT/LEFT* control input is HIGH, gates G1 through G4 are enabled and the state of the Q output of each flip-flop is passed through to the D input of the *following* flip flop. When a clock pulse occurs, the data bits are shifted one place to the *right*.

When the *RIGHT/LEFT* control input is LOW, gates G5 through G8 are enabled and the state of the Q output of each flip-flop is passed through to the D input of the *preceding* flip flop. When a clock pulse occurs, the data bits are shifted one place to the *left*.

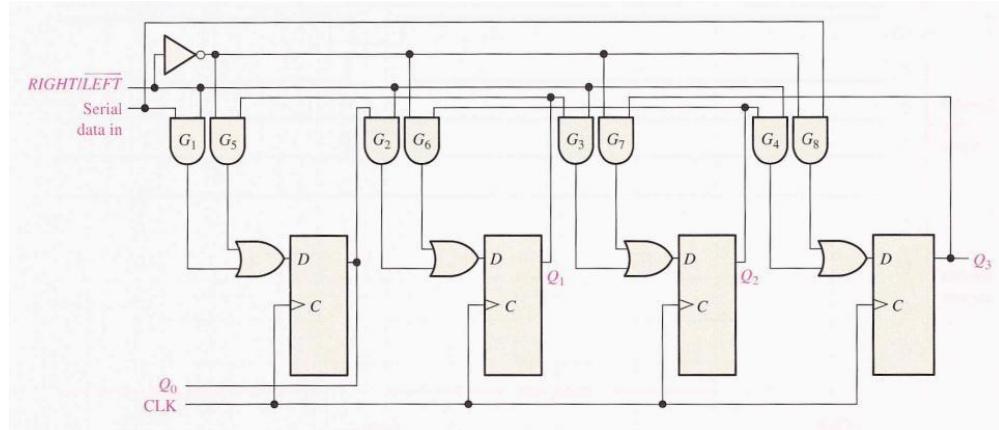


Figure 10.5(a): Bidirectional Shift Registers

Truth Tables:

- SISO :

Clock	Q3	Q2	Q1	Q0
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

- SIPO :

Clock	Q3	Q2	Q1	Q0
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	0	1	0
4	1	1	0	1

- PISO :

Clock	Q3	Q2	Q1	Q0
0	0	0	0	0
1	1	1	0	1
2	0	1	1	0
3	0	0	1	1

4	0	0	0	1
---	---	---	---	---

- **PIPO :**

Clock	Q3	Q2	Q1	Q0
0	0	0	0	0
1	1	1	0	1

The simulated circuits functioned exactly as the data shown in the above truth table. Therefore, it can be said that the simulated circuits are functioning properly.

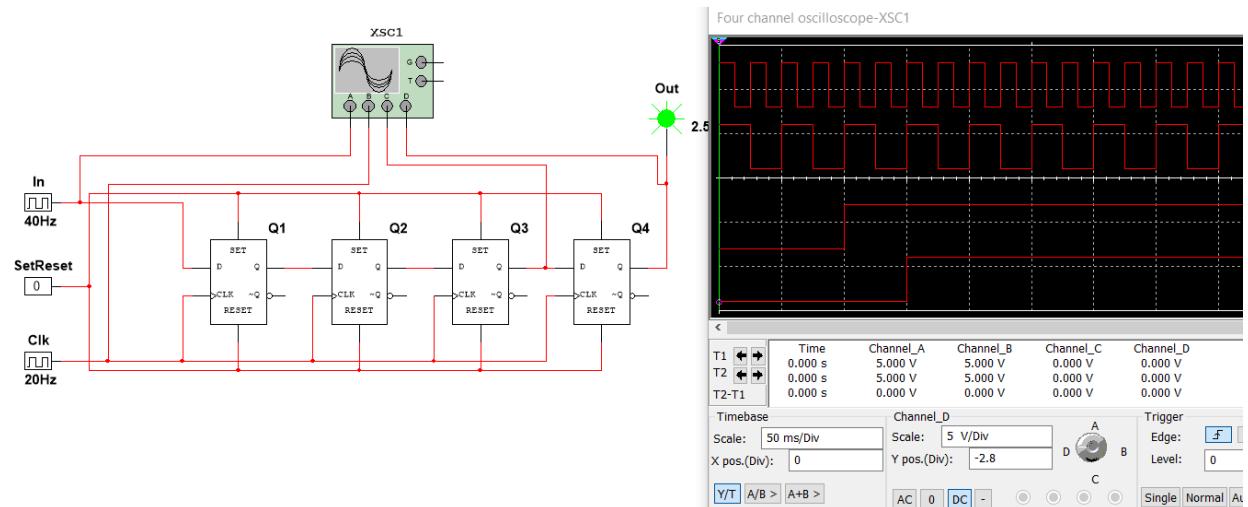
Apparatus:

- IC 74LS76 (JK Flip Flop)
- IC 7408 and 7432
- LED Lamps or Display
- Trainer Board
- Oscilloscope
- Connecting Wires

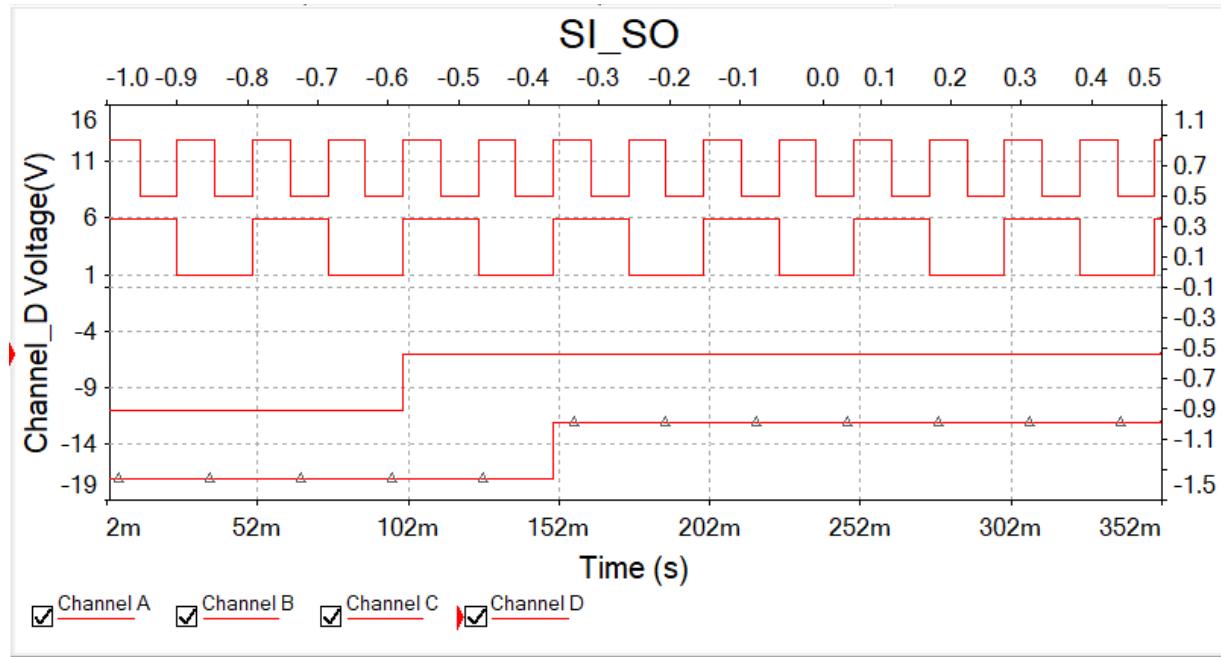
Simulation and Results:

1. Serial In/ Serial Out (SISO) Shift Register:

Simulation:

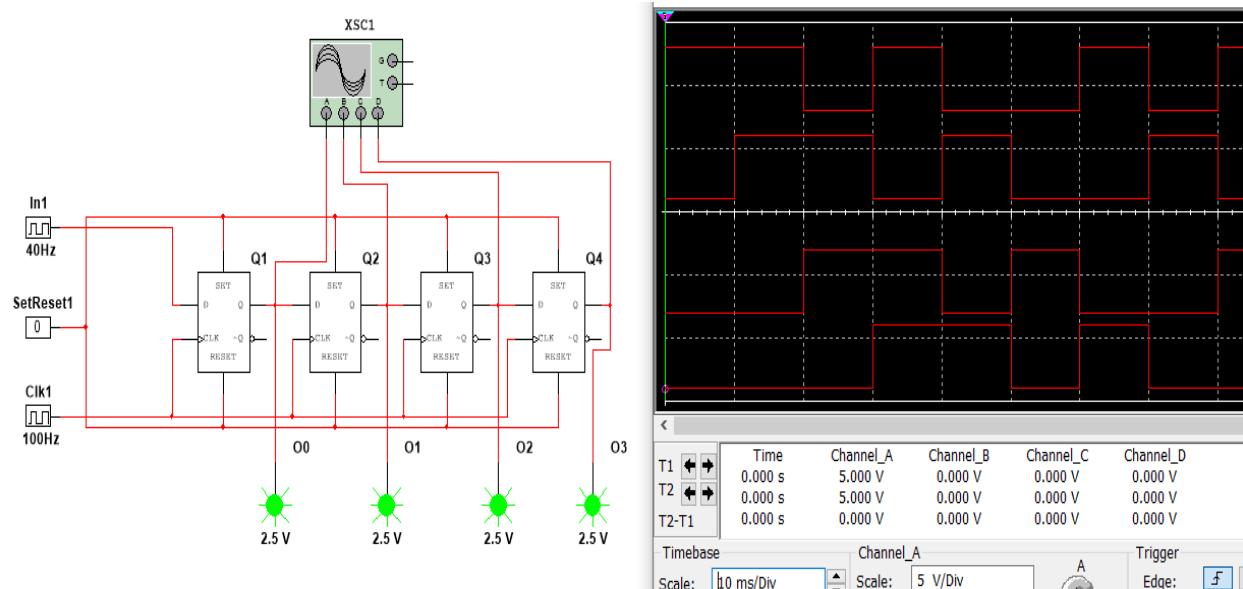


Graph:

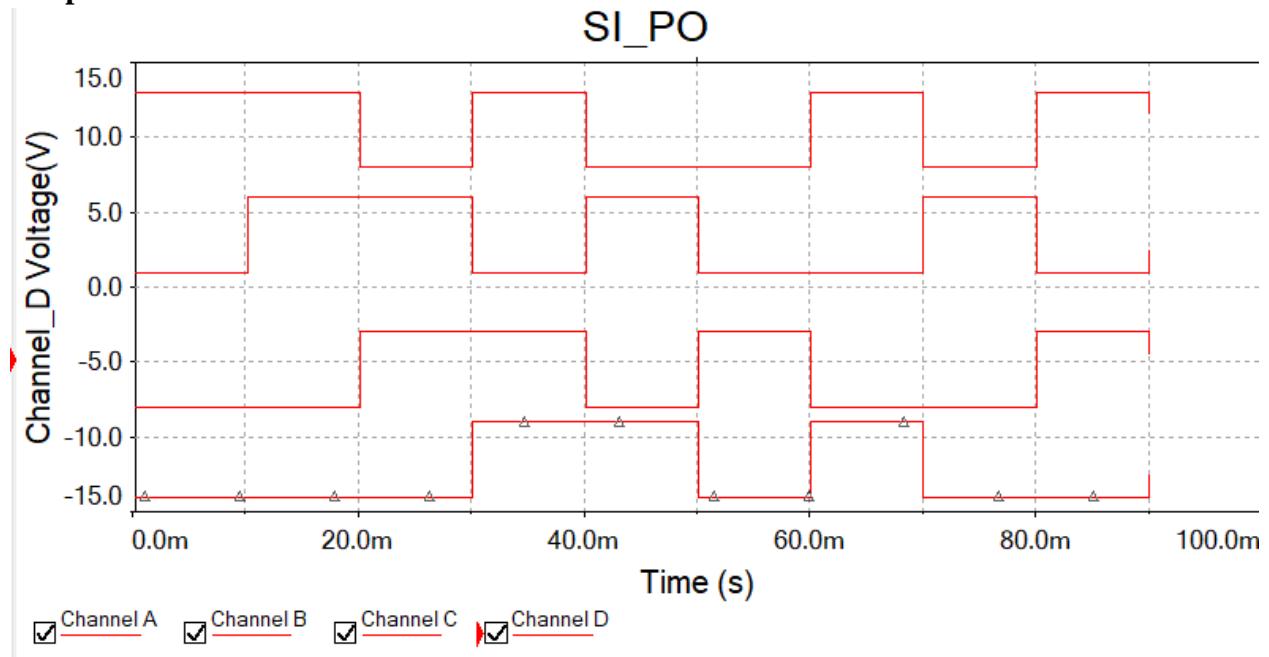


2. Serial In / Parallel Out (SIPO) Shift Register:

Simulation:

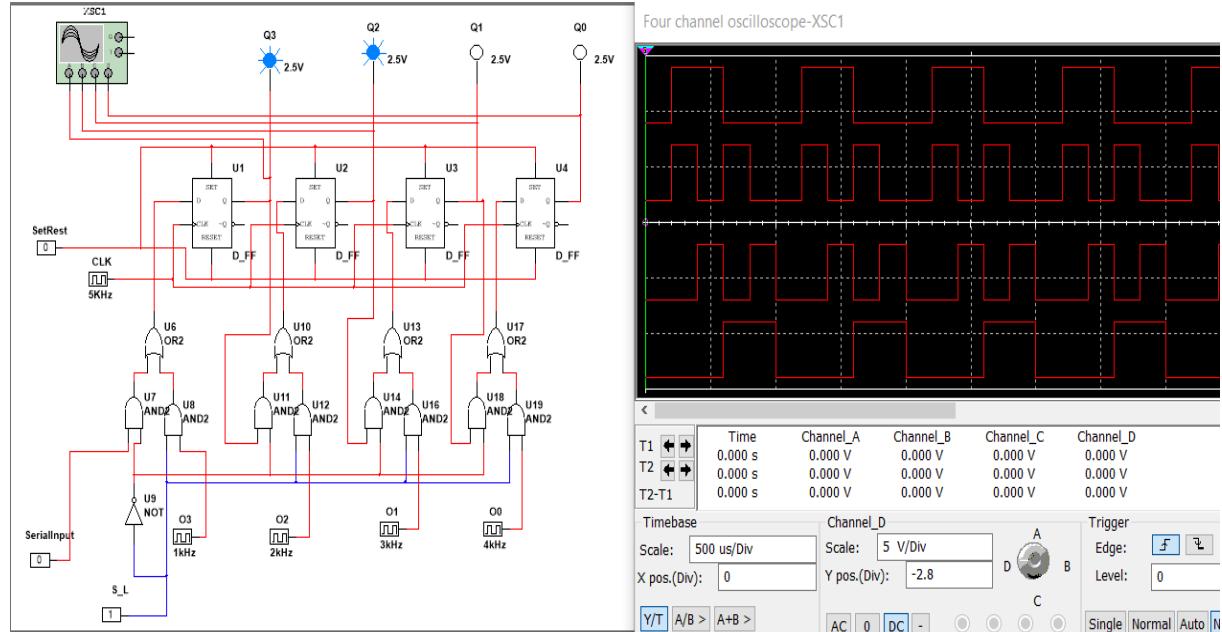


Graph:

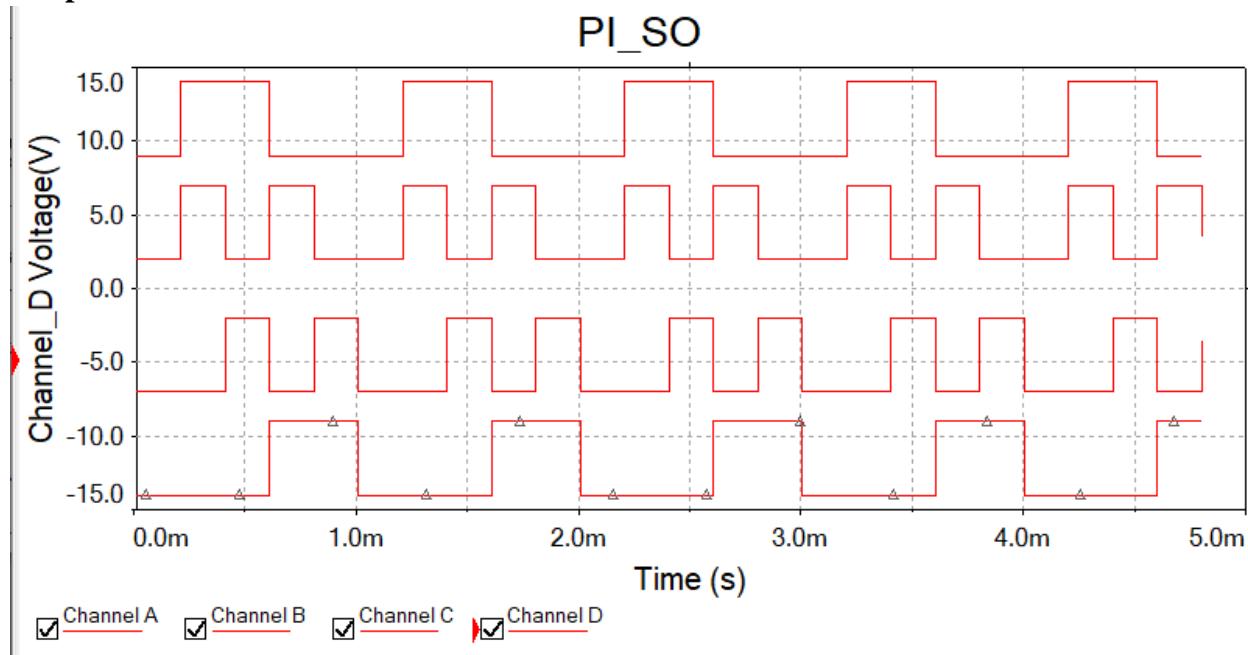


3. Parallel In / Serial Out (PISO) Shift Register:

Simulation:

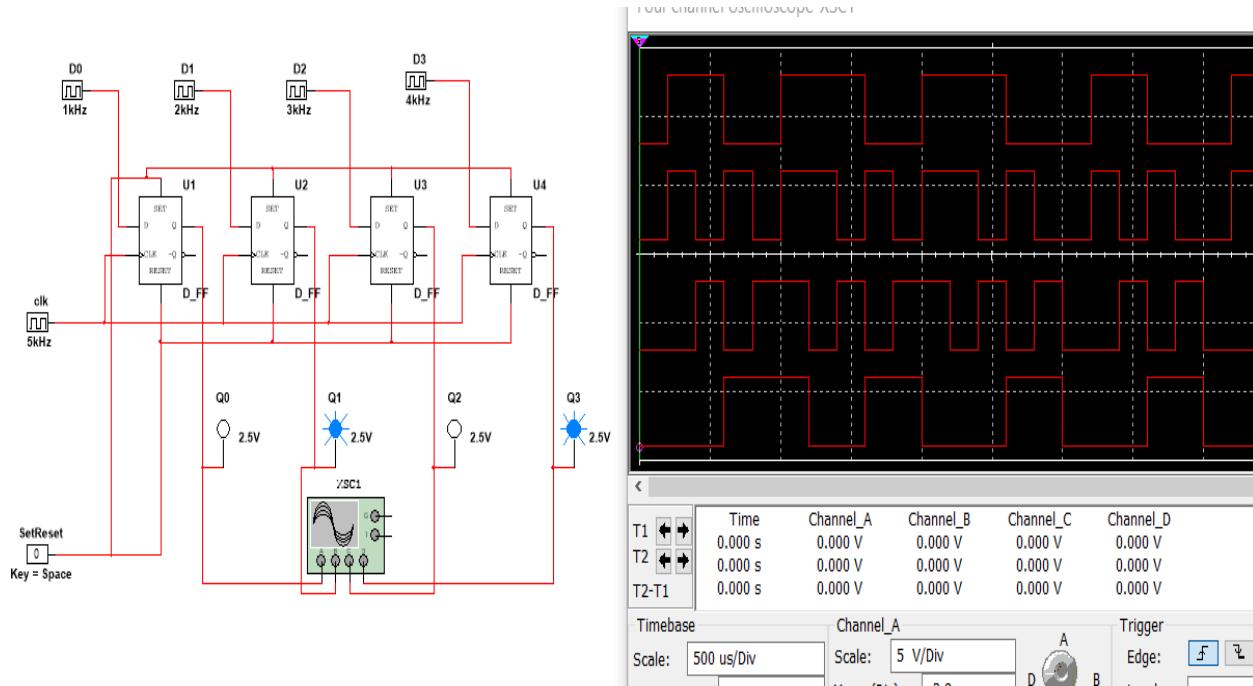


Graph:

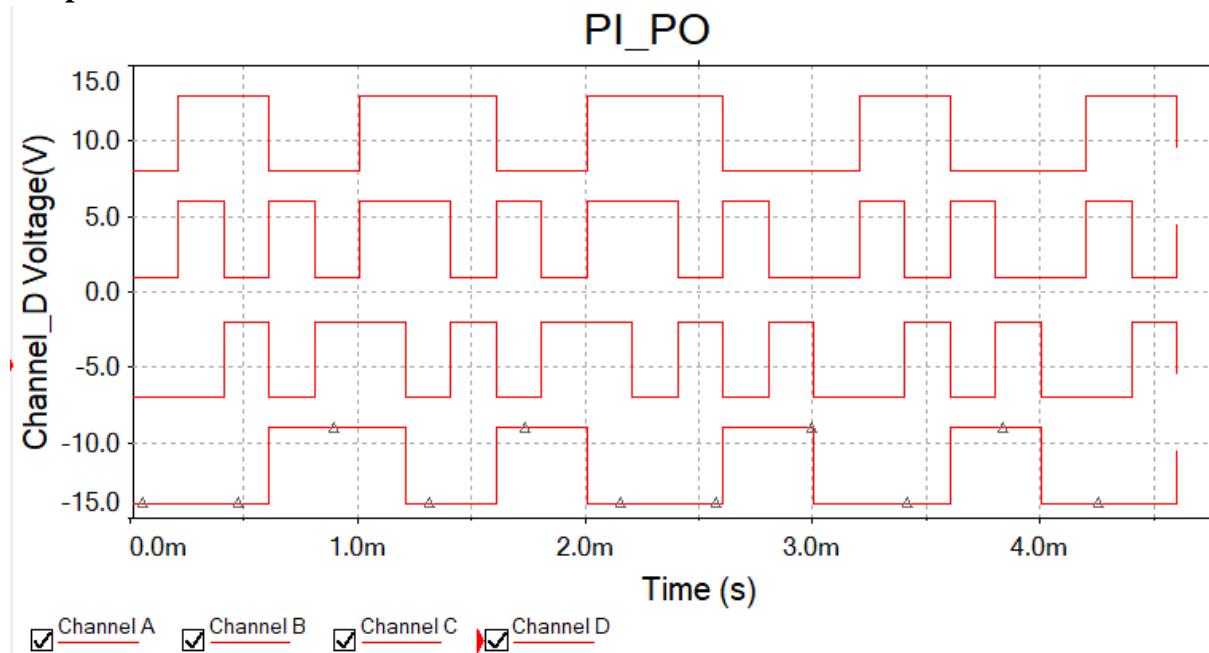


4. Parallel In / Parallel Out (PIPO) Shift Register:

Simulation:

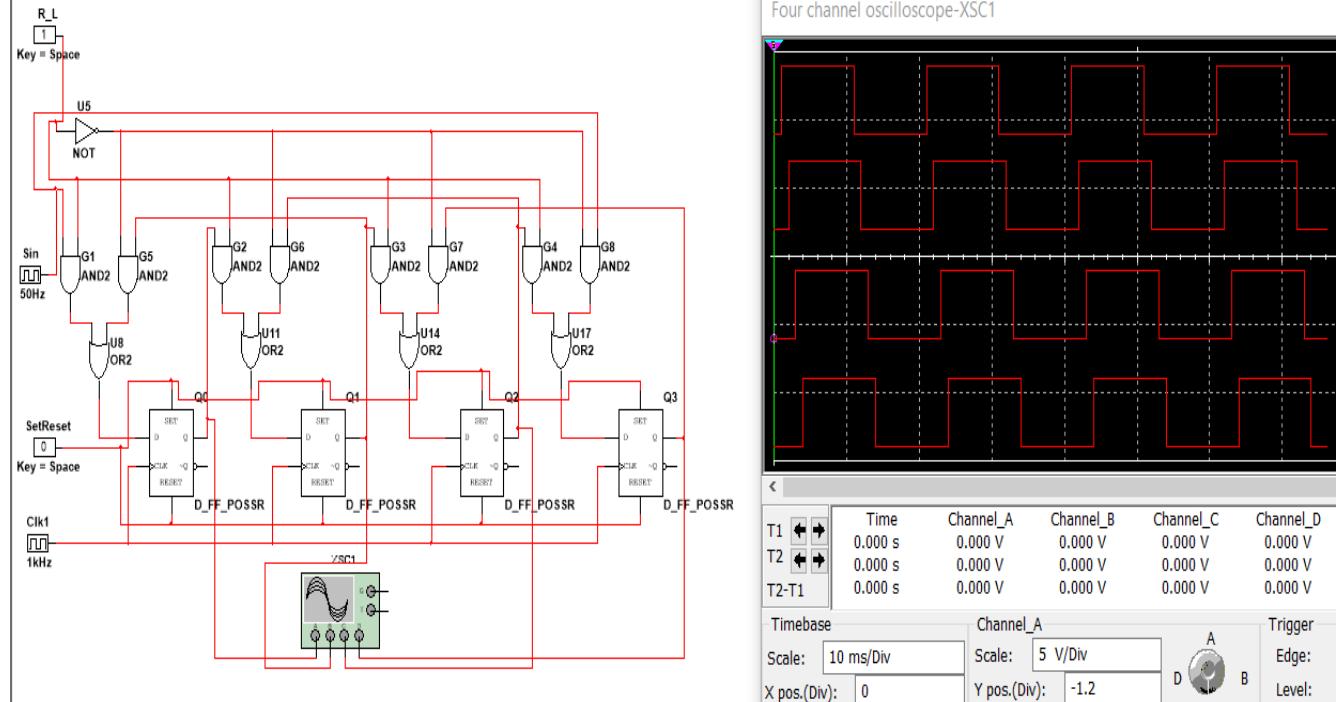


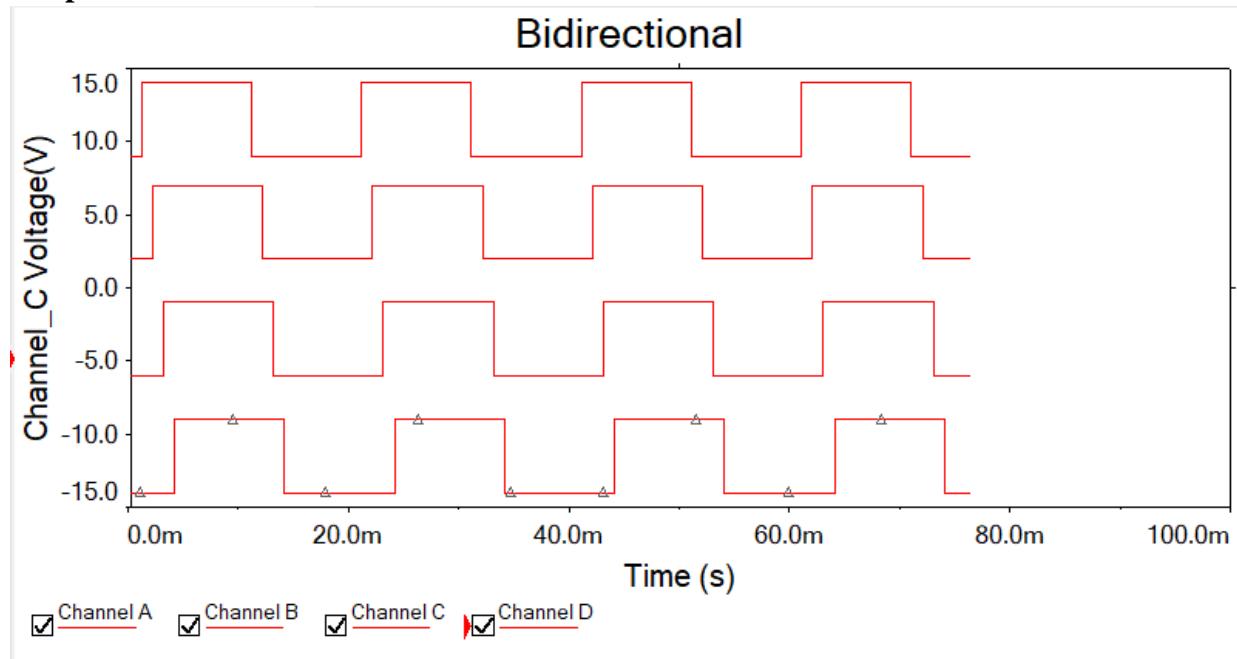
Graph:



5. Bidirectional Shift Registers:

Simulation:



Graph:**Discussion and conclusion:**

In this experiment our main aim was to know about all types of sifters. Mainly in digital circuits, a shift register is a cascade of flip flops, sharing the same clock, in which the output of each flip-flop is connected to the "data" input of the next flip-flop in the chain, resulting in a circuit that shifts by one position the "bit array" stored in it, shifting in the data present at its input and shifting out the last bit in the array, at each transition of the clock input. We mainly used D flip Flop ICs for measure the output curve. And we got the correct output curve for all the circuits also. We used some sources for collecting the circuit models. After doing all of this we have done this experiment properly.

Reference(s):

- i) Thomas L. Floyd, "Digital Fundamentals", Ninth Edition.