Laboratory Journal
of
Digital Electronics (DE)

For completion of term work of 4th *semester*
curriculum program

Bachelor of Technology

In

ELECTRONICS AND TELECOMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION

ENGINEERING

Dr. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY

Lonere-402 103, Tal. Mangaon, Dist. Raigad (MS)

INDIA

LIST OF EXPERIMENTS

1. Study of logic gates.
2. Design of basic gates using universal gates.
3. Design and implementation of multiplexer and demultiplexer using logic gates.
4. Design and implementation of adders using logic gates.
5. Design and implementation of subtractors using logic gates.
6. Implementation of BCD to 7 segment display using ic7447
7. Study of D Flip-Flops using gates.
8. Study of S R Flip-Flops using gates.
9. Study of J K Flip-Flops using gates.
10. Study of Encoder and Decoder.
11. Study of RAM.

EXPT NO:
DATE:

STUDY OF LOGIC GATES

AIM: To study about logic gates and verify their truth tables.

APPARATUS REQUIRED:

SL No.	COMPONENT	SPECIFICATION	QTY
1.	AND GATE	IC 7408	1
2.	OR GATE	IC 7432	1
3.	NOT GATE	IC 7404	1
4.	NAND GATE 2 I/P	IC 7400	1
5.	NOR GATE	IC 7402	1
6.	X-OR GATE	IC 7486	1
7.	NAND GATE 3 I/P	IC 7410	1
8.	PATCH CORD	-	14

THEORY:

Circuit that takes the logical decision and the process are called logic gates. Each gate has one or more input and only one output.

OR, AND and NOT are basic gates. NAND, NOR and X-OR are known as universal gates. Basic gates form these gates.

AND GATE:

The AND gate performs a logical multiplication commonly known as AND function. The output is high when both the inputs are high. The output is low level when any one of the inputs is low.

OR GATE:

The OR gate performs a logical addition commonly known as OR function. The output is high when any one of the inputs is high. The output is low level when both the inputs are low.

NOT GATE:

The NOT gate is called an inverter. The output is high when the input is low. The output is low when the input is high.

NAND GATE:

The NAND gate is a contraction of AND-NOT. The output is high when both inputs are low and any one of the input is low .The output is low level when both inputs are high.

NOR GATE:

The NOR gate is a contraction of OR-NOT. The output is high when both inputs are low. The output is low when one or both inputs are high.

X-OR GATE:

The output is high when any one of the inputs is high. The output is low when both the inputs are low and both the inputs are high.

PROCEDURE:

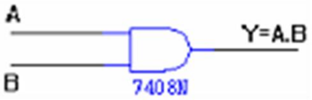
(i) Connections are given as per circuit diagram.

(ii) Logical inputs are given as per circuit diagram.

(iii) Observe the output and verify the truth table.

AND GATE

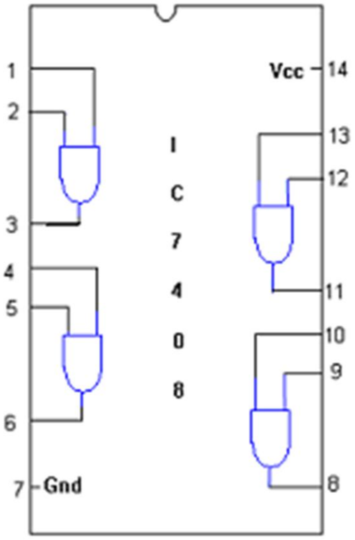
SYMBOL:



TRUTH TABLE

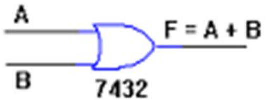
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

PIN DIAGRAM:



OR GATE

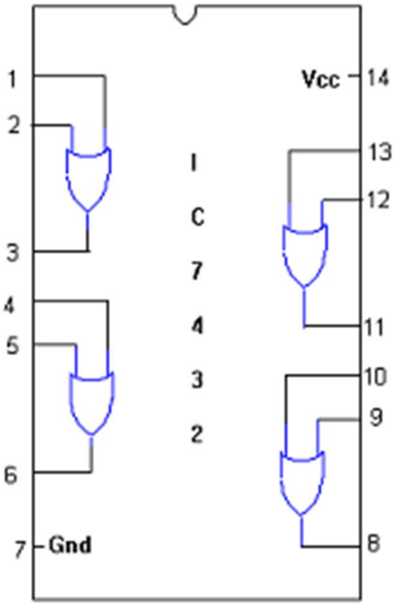
SYMBOL :



TRUTH TABLE

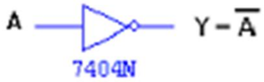
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

PIN DIAGRAM :

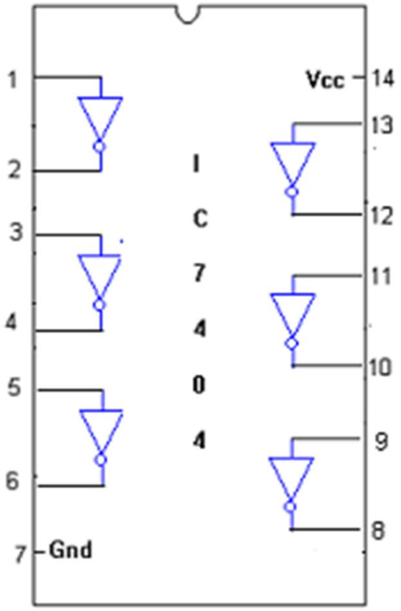


NOT GATE

SYMBOL:



PIN DIAGRAM:

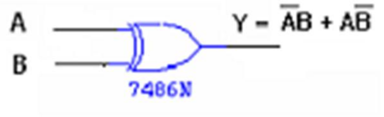


TRUTH TABLE :

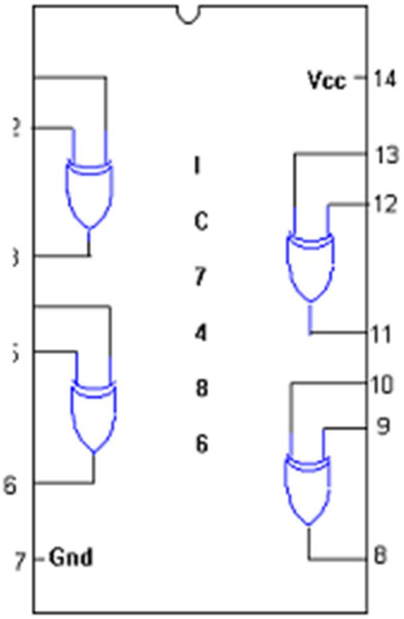
A	$\overline{A}$
0	1
1	0

XOR GATE

SYMBOL :



PIN DIAGRAM :

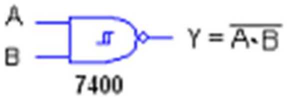


TRUTH TABLE :

A	B	$\overline{A}B + A\overline{B}$
0	0	0
0	1	1
1	0	1
1	1	0

2-INPUT NAND GATE

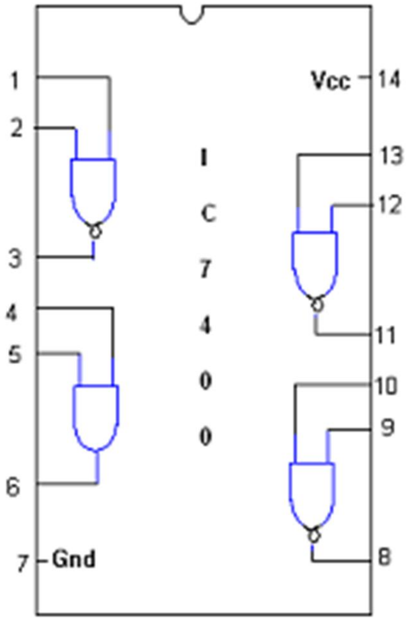
SYMBOL:



TRUTH TABLE

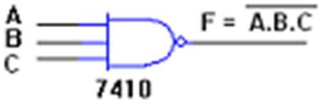
A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

PIN DIAGRAM:



3-INPUT NAND GATE

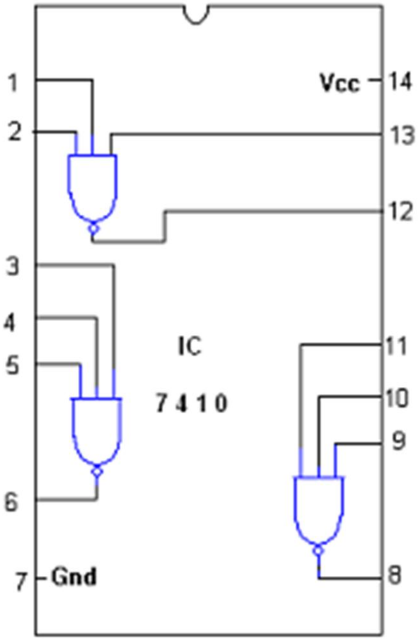
SYMBOL :



TRUTH TABLE

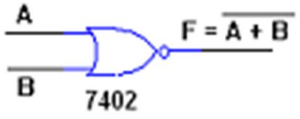
A	B	C	$\overline{A.B.C}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

PIN DIAGRAM :

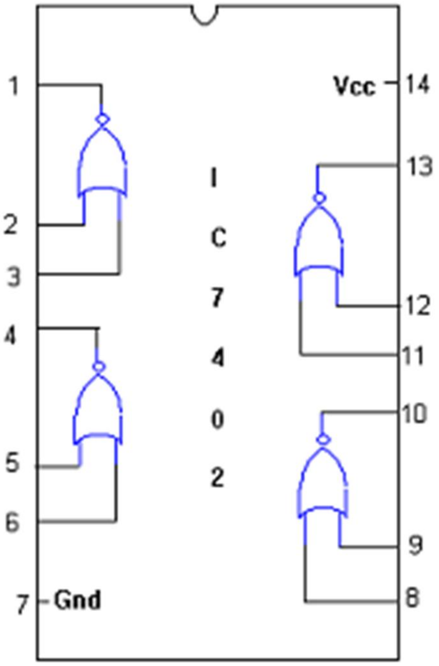


NOR GATE

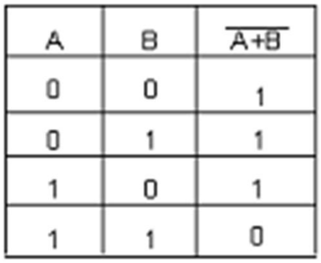
SYMBOL :



PIN DIAGRAM :



TRUTH TABLE



A	B	$\overline{A+B}$
0	0	1
0	1	1
1	0	1
1	1	0

RESULT:

CONCULSION:-

EXPT NO:
DATE:

DESIGN AND IMPLEMENTATION OF MULTIPLEXER AND DEMULTIPLEXER

AIM: To design and implement multiplexer and demultiplexer using logic gates and study of IC 74150 and IC 74154.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	3 I/P AND GATE	IC 7411	2
2.	OR GATE	IC 7432	1
3.	NOT GATE	IC 7404	1
2.	IC TRAINER KIT	-	1
3.	PATCH CORDS	-	32

THEORY:

MULTIPLEXER:

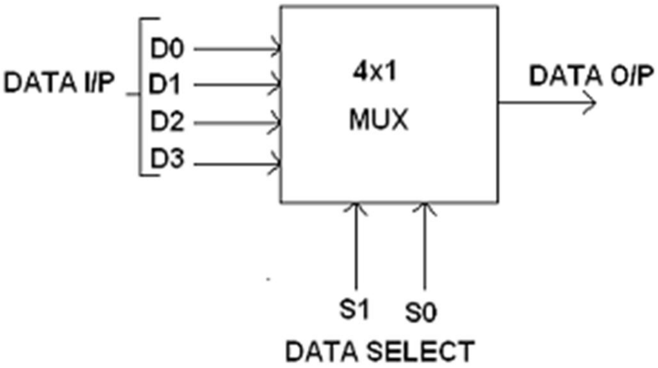
Multiplexer means transmitting a large number of information units over a smaller number of channels or lines. A digital multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally there are 2^n input line and n selection lines whose bit combination determine which input is selected.

DEMULTIPLEXER:

The function of Demultiplexer is in contrast to multiplexer function. It takes information from one line and distributes it to a given number of output lines. For this reason, the demultiplexer is also known as a data distributor. Decoder can also be used as demultiplexer.

In the 1: 4 demultiplexer circuit, the data input line goes to all of the AND gates. The data select lines enable only one gate at a time and the data on the data input line will pass through the selected gate to the associated data output line.

BLOCK DIAGRAM FOR 4:1 MULTIPLEXER:

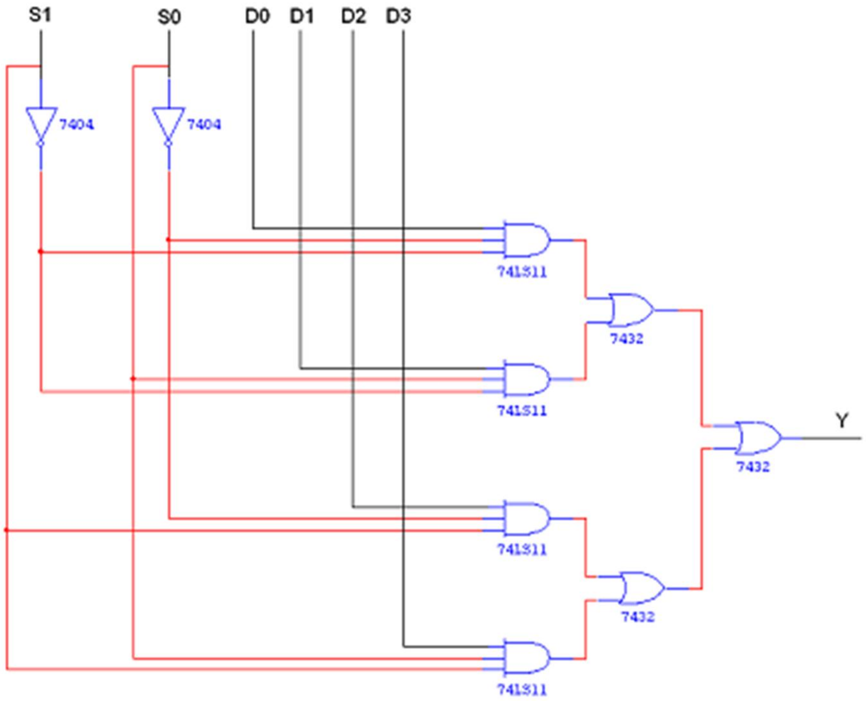


FUNCTION TABLE:

S1	S0	INPUTS Y
0	0	DO — DO S1' SO'
0	1	D1 — D1 S1' SO
1	0	D2 — D2 S1 SO'
1	1	D3 — D3 S1 SO

Y = DO S1' SO' + D1 S1' SO + D2 S1 SO' + D3 S1 SO

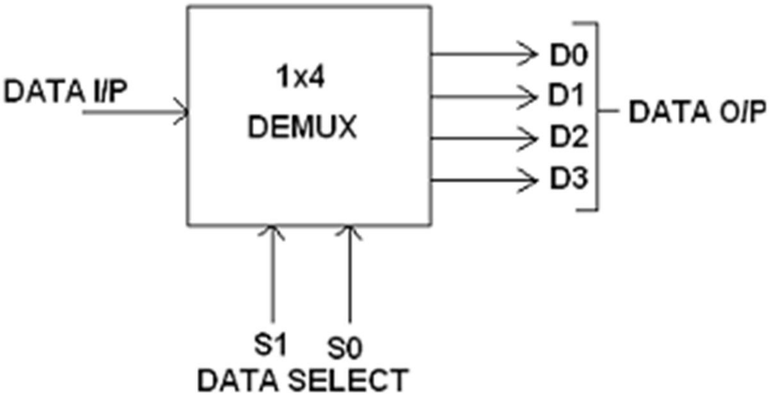
CIRCUIT DIAGRAM FOR MULTIPLEXER:



TRUTH TABLE:

S1	S0	Y = OUTPUT
0	0	D0
0	1	D1
1	0	D2
1	1	D3

BLOCK DIAGRAM FOR 1:4 DEMULTIPLEXER:

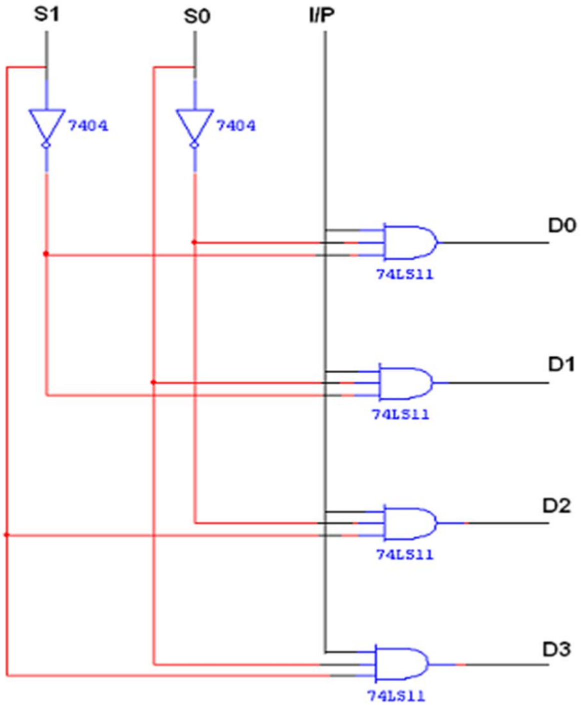


FUNCTION TABLE:

S1	S0	INPUT
0	0	X — DO = X S1' SO'
0	1	X — D1 = X S1' SO
1	0	X — D2 = X S1 SO'
1	1	X — D3 = X S1 S0

Y = X S1' SO' + X S1' SO + X S1 SO' + X S1 S0

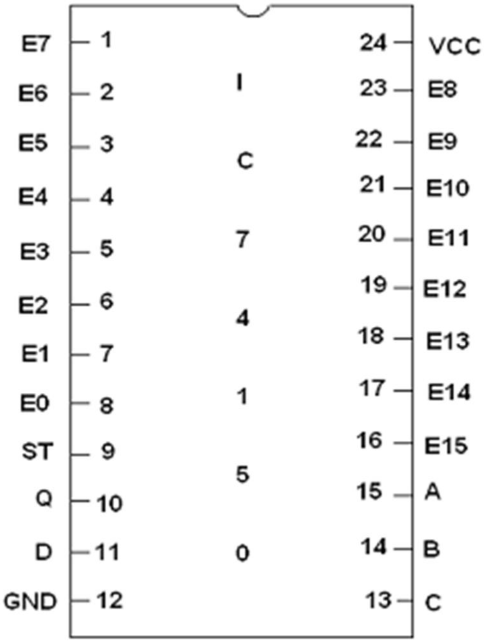
LOGIC DIAGRAM FOR DEMULTIPLEXER



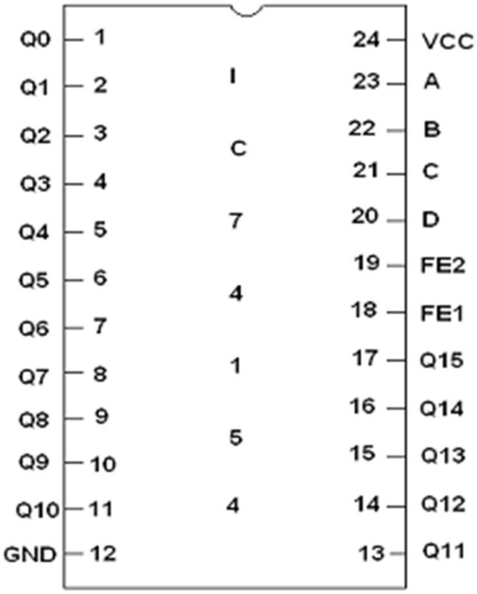
TRUTH TABLE:

INPUT			OUTPUT			
S1	**S0**	**I/P**	**D0**	**D1**	**D2**	**D3**
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	0
1	1	1	0	0	0	1

PIN DIAGRAM FOR IC 74150:



PIN DIAGRAM FOR IC 74154:



PROCEDURE:

(i) Connections are given as per circuit diagram.

(ii) Logical inputs are given as per circuit diagram.

(iii) Observe the output and verify the truth table.

RESULT:

CONCULSION:-

EXPT NO:
DATE:

DESIGN OF ADDER AND SUBTRACTOR

AIM: To design and construct half adder, full adder, half subtracter and full subtracter circuits and verify the truth table using logic gates.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	
2.	X-OR GATE	IC 7486	
3.	NOT GATE	IC 7404	
4.	OR GATE	IC 7432	
3.	IC TRAINER KIT	-	
4.	PATCH CORDS	-	23

THEORY:

HALF ADDER:

A half adder has two inputs for the two bits to be added and two outputs one from the sum ' S' and other from the carry ' c' into the higher adder position. Above circuit is called as a carry signal from the addition of the less significant bits sum from the X-OR Gate the carry out from the

AND gate.

FULL ADDER:

A full adder is a combinational circuit that forms the arithmetic sum of input; it consists of three inputs and two outputs. A full adder is useful to add three bits at a time but a half adder cannot do so. In full adder sum output will be taken from X-OR Gate, carry output will be taken from OR Gate.

HALF SUBTRACTOR:

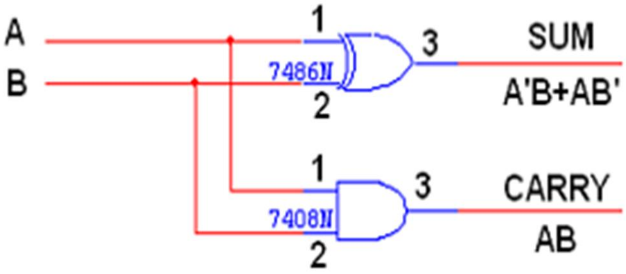
The half subtractor is constructed using X-OR and AND Gate. The half subtractor has two input and two outputs. The outputs are difference and borrow. The difference can be applied using X-OR Gate, borrow output can be implemented using an AND Gate and an inverter.

FULL SUBTRACTOR:

The full subtractor is a combination of X-OR, AND, OR, NOT Gates. In a full subtractor the logic circuit should have three inputs and two outputs. The two half subtractor put together gives a full subtractor .The first half subtractor will be C and A B. The output will be difference output of full subtractor. The expression AB assembles the borrow output of the half subtractor and the second term is the inverted difference output of first X-OR.

LOGIC DIAGRAM:

<u>HALF ADDER</u>



TRUTH TABLE:

A	B	CARRY	SUM
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

K-Map for SUM:

A \ B	00	01
00		1
01	1	

SUM = A'B + AB'

K-Map for CARRY:

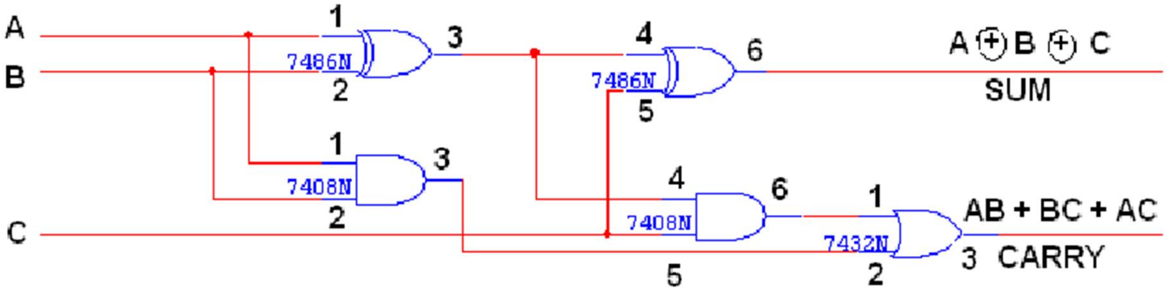
A \ B	00	01
00		
01		1

CARRY = AB

LOGIC DIAGRAM:

<u>FULL ADDER</u>

FULL ADDER USING TWO HALF ADDER



TRUTH TABLE

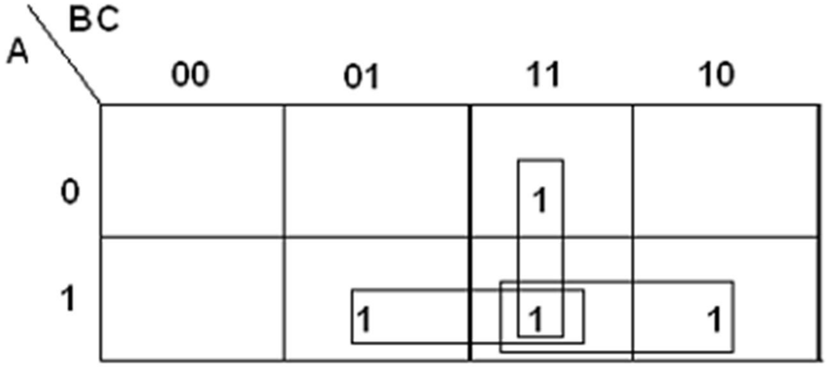
A	B	C	CARRY	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

K-Map for SUM

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

SUM = A'B'C + A'BC' + ABC' + ABC

K-Map for CARRY

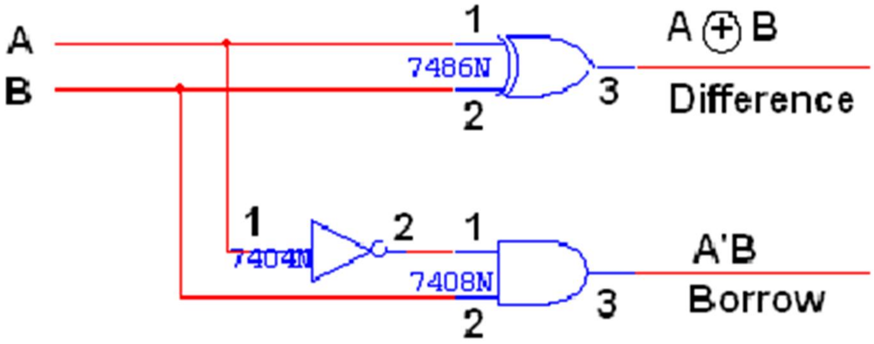


A \ BC	00	01	11	10
0			1	
1		1	1	1

CARRY = AB + BC + AC

LOGIC DIAGRAM:

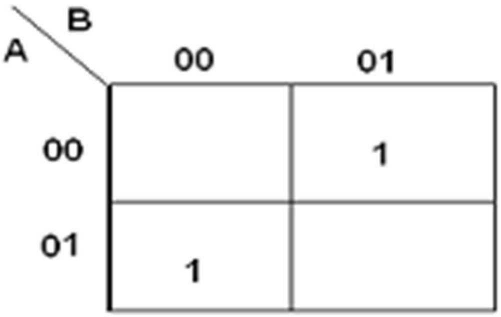
HALF SUBTRACTOR



TRUTH TABLE

A	B	BORROW	DIFFERENCE
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

K-Map for DIFFERENCE



DIFFERENCE = A'B + AB'

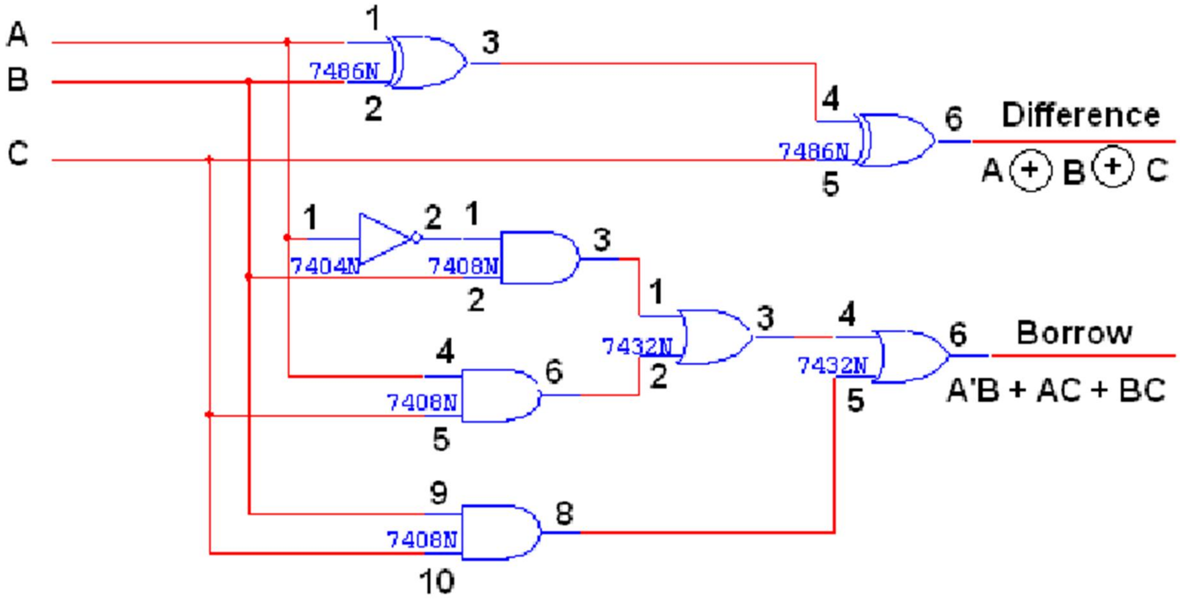
K-Map for BORROW

A \ B	00	01
00		1
01		

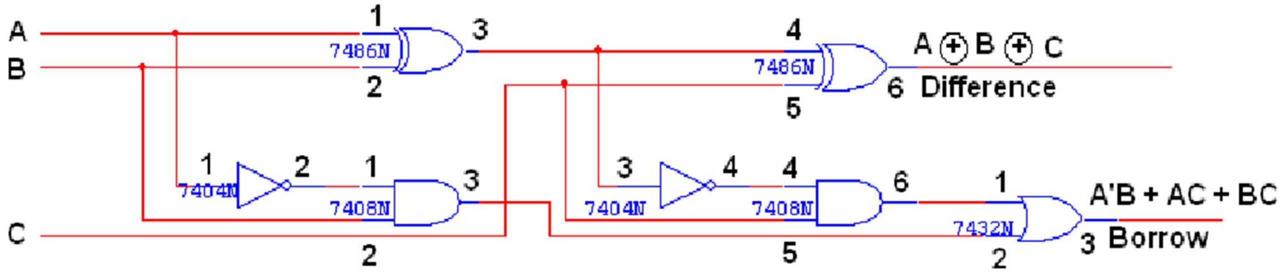
BORROW = A'B

LOGIC DIAGRAM:

<u>FULL SUBTRACTOR</u>



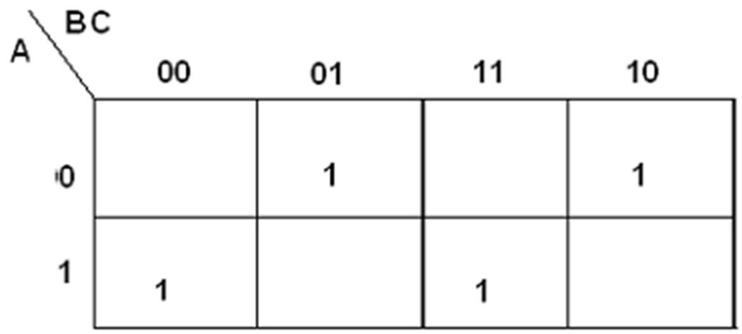
FULL SUBTRACTOR USING TWO HALF SUBTRACTOR:



TRUTH TABLE:

A	B	C	BORROW	DIFFERENCE
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

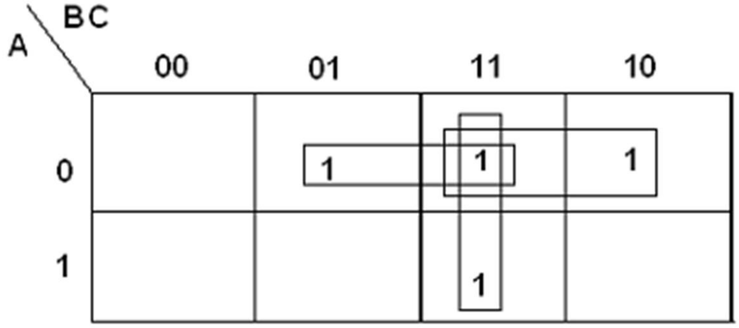
K-Map for DIFFERENCE



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

Difference = A'B'C + A'BC' + AB'C' + ABC

K-Map for BORROW



Borrow = A'B + BC + A'C

PROCEEDURE:

(i) Connections are given as per circuit diagram.

(ii) Logical inputs are given as per circuit diagram.

(iii) Observe the output and verify the truth table.

RESULT:

<u>CONCULSION:-</u>

EXPT NO:

DATE:

BCD to 7-seg Decoder

AIM: To study and implement BCD to 7-Segmant Decoder .

APPARATUS: Digital Logic Trainer kit, IC 7447, connecting wires.

THEORY:

The output information from digital systems is often needed to be displayed in numerical form. For example, in calculators, digital watches or in digital multimeters. By Forward biasing different LEDs, we can display the digital 0 through 9 (see **Fig.1)** which shows a seven segment indicator i.e. seven LEDS labeled a through g. For instance, to display a 0, we need to light up segments a, b, c, d, e and f.

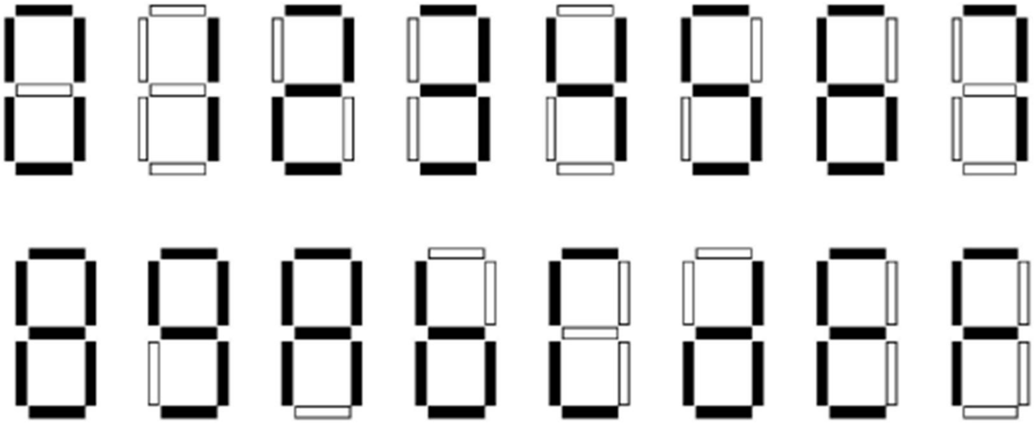


Fig 1. Display of 0 to 9 using 7-sagment LED Display

There are two types of LED 7-segment displays: common cathode (CC) and common anode (CA). The difference between the two displays is the common cathode has all the cathodes of the 7-segments connected directly together and the common anode has all the anodes of the **7-**segments connected together. Shown below (Fig-2) is a common anode and common cathode seven segment Display.

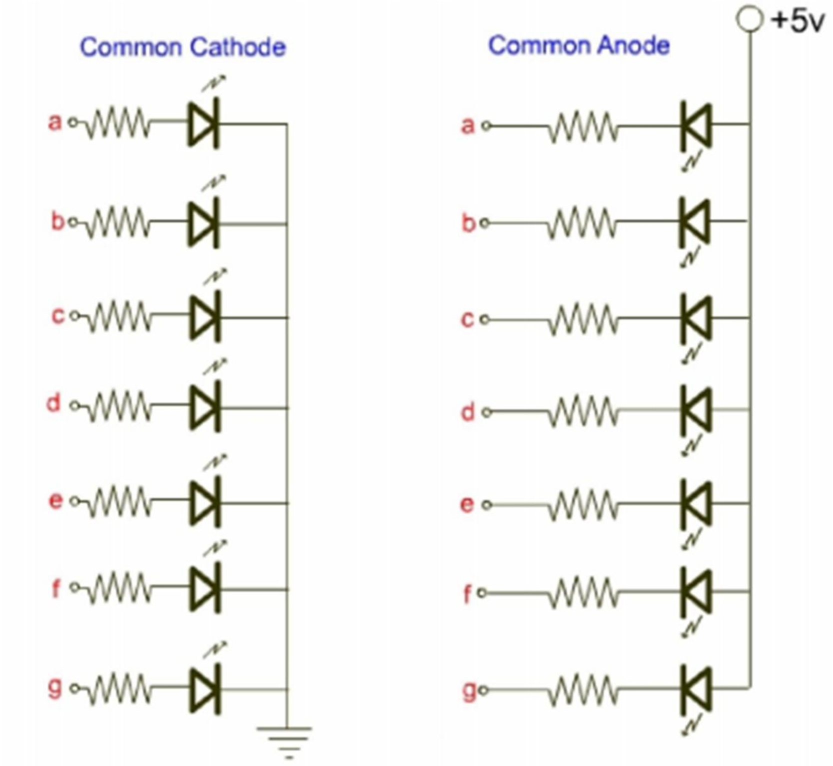
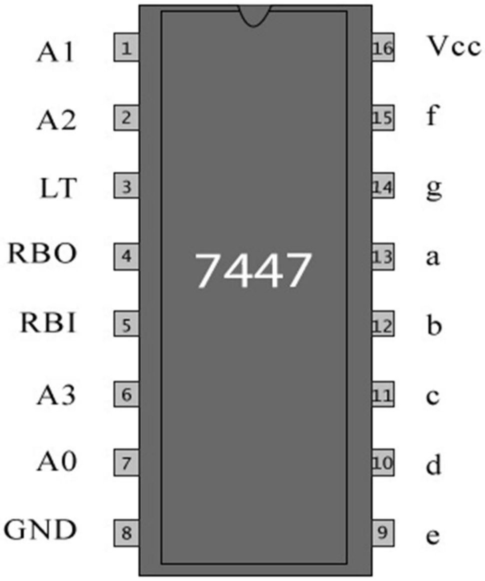


Fig 2. Types of 7-Sagment Display

In Common anode, all the anode segments are connected together. When working with a common anode seven segment display, power must be applied externally to the anode connection that is common to all the segments. Then by applying a ground to a particular segment connection(a-g), the appropriate segment will light up. An additional resistor must be added to the circuit to limit the amount of current flowing through each LED segment.

In common cathode, cathodes of all the LEDs are connected together. For the use of this seven segment the common cathode connection must be grounded and power must be applied to appropriate segment in order to illuminate that segment.

Fig.3 given below shows the pin diagram of IC 7447 which is a BCD to 7-sagment Decoder.



The Functions of LT, RB1, and RBO are explained below.
RBI: It is connected to logic-1 for normal decoding operation, If is connected to 0 level, the segment outputs with generate data for normal 7 segment decoding for all BCD inputs except zero. RBI is the Ripple Blanking input pin, and when held Low, it blanks the digit 0 on the connected 7-segment LED. The 0 blanking facility is required for blanking the leading zeros in a multi-digit display. For example, in an 8-digit calculator a number, say 5, is to be displayed as 5 only and not as 00000005. Hence, all the RBI lines corresponding to the leading 0s are held low.
RBO: This output, which is normally at logic-1 goes to logic 0 during zero blanking interval. This is used for cascading purpose.
LT: This is used to check the segments of LEDS. There is one dot LEDs in the segment display device, one on the right hand side of the digit. This is used to display the decimal point when necessary. Fig.3 shows the relationship of the 4 input and 7 output of the BCD-to-Seven segment code converter (IC - 7447)

The circuits accept 4-bit binary-coded-decimal (BCD) and, depending on the state of the inputs, decodes this data to drive a 7-segment display indicator. The relative positive-logic output levels, as well as conditions required at the inputs, are shown in the truth tables (fig.4)

Decimal No.	LT'	RBI’	D	C	B	A	BI’ /RBO’	a	b	c	d	e	f	g
0														
1														
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														

Fig.4 Truth Table of BCD to 7-sagment Decoder

PROCEDURE:

1. With the help of connecting wires, connect logic input switches to the A,B,C,D input of 7447 Also connect the output a to g of 7447 to the logic output indicator.
2. Change input with the help of switches.
3. Observe output on 7-sagment Display and verify the truth table of BCD to 7-Sagment Decoder.

RESULTS

CONCULSION:

EXPT NO:

DATE:

Study of D flip flop

Aim: Implementation of D flip flop

Theory:

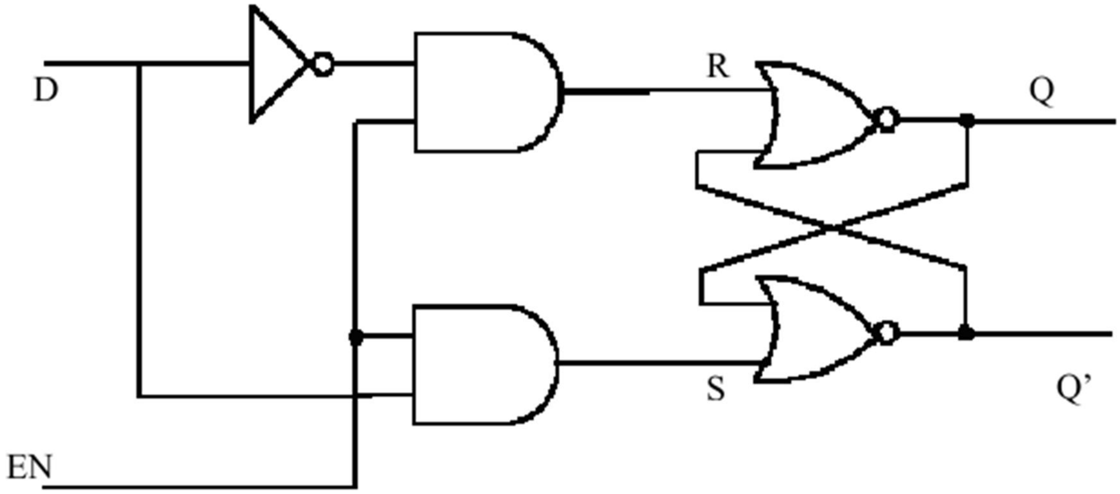
D FLIP-FLOP

An RS flip-flop is rarely used in actual sequential logic because of its undefined outputs for inputs R= S= 1. It can be modified to form a more useful circuit called D flip-flop, where D stands for data. The D flip-flop has only a single data input D as shown in the circuit diagram. That data input is connected to the S input of an RS flip-flop, while the inverse of D is connected to the R input. To allow the flip-flop to be in a holding state, a D-flip flop has a second input called Enable, EN. The Enable-input is AND-ed with the D-input.

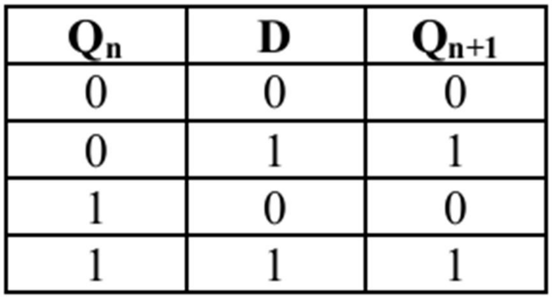
- When **EN=0,** irrespective of D-input, the R = S = 0 and the **state is held.**
- When **EN= 1,** the S input of the RS flip-flop equals the D input and R is the inverse of D. Hence, output **Q follows D,** when EN= 1.
- When **EN returns to 0,** the most recent input **D is 'remembered'.**

The circuit operation is summarized in the characteristic table for **EN=1.**

Circuit Diagram:



Characteristic Table:



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

RESULT:

CONCLUSION:

EXPT NO:

DATE:

Study of SR flip flop

Aim: Implementation of SR flip flop using gates

Overview:

So far you have encountered with *combinatorial logic,* i.e. circuits for which the output depends only on the inputs. In many instances it is desirable to have the next output depending on the current output. A simple example is a *counter,* where the next number to be output is determined by the current number stored. Circuits that remember their current output or state are often called *sequential logic* circuits. Clearly, sequential logic requires the ability to store the current state. In other words, *memory* is required by sequential logic circuits, which can be created with boolean gates. If you arrange the gates correctly, they will remember an input value. This simple concept is the basis of RAM (random access memory) in computers, and also makes it possible to create a wide variety of other useful circuits.

Memory relies on a concept called **feedback.** That is, the output of a gate is fed back into the input. The simplest possible feedback circuit using two inverters is shown below (Fig.1):

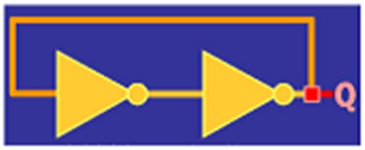


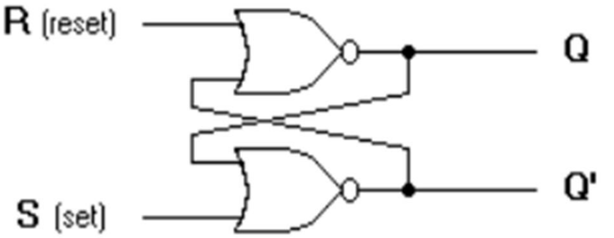
Fig.1: Simplest realization of feedback circuit

If you follow the feedback path, you can see that if Q happens to be 1 (or 0), it will always be 1 (or 0). Since it's nice to be able to control the circuits we create, this one doesn't have much use -- but it does let you see how feedback works. It turns out that in "real" sequential circuits, you can actually use this sort of simple inverter feedback approach. The memory elements in these circuits are called *flip-flops.* A flip-flop circuit has two outputs, one for the normal value and one for the complement value of the stored bit. Binary information can enter a flip-flop in a variety of ways and gives rise to different types of flip-flops.

RS Flip-Flop

RS flip-flop is the simplest possible memory element. It can be constructed from two NAND gates or two NOR gates. Let us understand the operation of the RS flip-flop using NOR gates as shown below using the truth table for 'A NOR B' gate. The inputs R and S are referred to as the Reset and Set inputs, respectively. The outputs Q and Q' are complements of each other and are referred to as the normal and complement outputs, respectively. The binary state of the flip-flop is taken to be the value of the normal output. When Q=1 and Q'=0, it is in the *set state* (or 1-state). When Q=0 and Q'=1, it is in the *reset/clear state* (or 0-state).

Circuit Diagram:

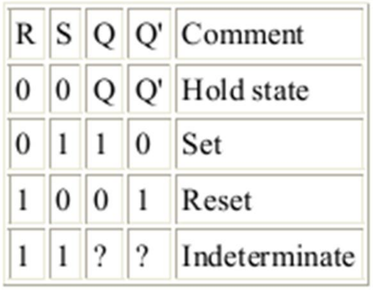


A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

- **S=1 and R=0:** The output of the bottom NOR gate is equal to zero, Q'=0. Hence both inputs to the top NOR gate are equal to 0, thus, Q=1. Hence, the input combination S=1 and R=0 leads to the flip-flop being **set** to Q=1.
- **S=0 and R=1:** Similar to the arguments above, the outputs become Q=0 and Q'=1. We say that the flip-flop is **reset.**
- **S=0 and R=0:** Assume the flip-flop was previously in set (S=1 and R=0) condition. Now changing S to 0 results Q' still at 0 and Q=1. Similarly, when the flip-flop was previously in a reset state (S=0 and R=1), the outputs do not change. Therefore, with inputs S=0 and R=0, the flip-flop holds its state.
- **S=1 and R=1:** This condition violates the fact that both outputs are complements of each other since each of them tries to go to 0, which is not a stable configuration. It is impossible to predict which output will go to 1 and which will stay at 0. In normal operation this condition must be avoided by making sure that 1's are not applied to both inputs simultaneously, thus making it one of the main disadvantages of RS flip-flop.

All the above conditions are summarized in the characteristic table below:

Characteristic Table:



R	S	Q	Q'	Comment
0	0	Q	Q'	Hold state
0	1	1	0	Set
1	0	0	1	Reset
1	1	?	?	Indeterminate

Debounce circuit

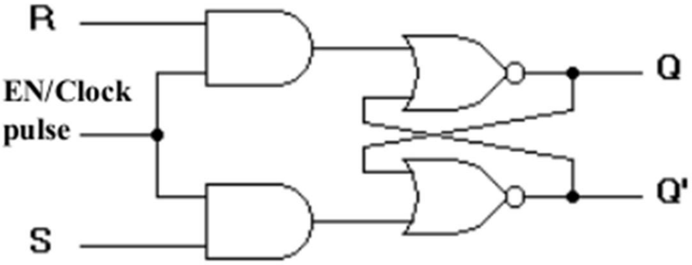
An elementary example using this flip-flop is the debounce circuit. Suppose a piece of electronics is to change state under the action of a mechanical switch. When this switch is moved from position S to R (S=0, R=1), the contacts make and break several times at R before settling to good contact. It is desirable that the electronics should respond to the first contact and then remain stable, rather than switching back and forth as the circuit makes and breaks. This is achieved by RS flip-flop which is reset to Q=0 by the first signal R=1 and remains in a fixed state until the switch is moved back to position S, when the signal S=1 sets the flip-flop to Q=1.

Gated or Clocked RS Flip-Flop

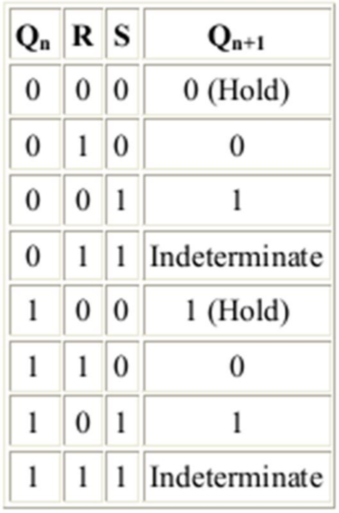
It is sometimes desirable in sequential logic circuits to have a bistable RS flip-flop that only changes state when certain conditions are met regardless of the condition of either the Set or the Reset inputs. By connecting a 2-input AND gate in series with each input terminal of the RS NOR Flip-flop a Gated RS Flip-flop can be created. This extra conditional input is called an "Enable" input and is given the prefix of "EN" as shown below. When the Enable input "EN" = 0, the outputs of the two AND gates are also at logic level 0, (AND Gate principles) regardless of the condition of the two inputs S and R, latching the two outputs Q and Q' into their last known state. When the enable input "EN" = 1, the circuit responds as a normal RS bistable flip-flop with the two AND gates becoming transparent to the Set and Reset signals. This Enable input can also be connected to a clock timing signal adding clock synchronisation to the flip-flop creating what is sometimes called a "Clocked SR Flip-flop".

So a **Gated/Clocked RS Flip-flop** operates as a standard bistable latch but the outputs are only activated when a logic "1" is applied to its EN input and deactivated by a logic "0". The property of this flip-flop is summarized in its characteristic table where Q_n is the logic state of the previous output and Q_{n+1} is that of the next output and the clock input being at logic 1 for all the R and S input combinations.

Circuit Diagram:



Characteristic Table:



Q_n	R	S	Q_{n+1}
0	0	0	0 (Hold)
0	1	0	0
0	0	1	1
0	1	1	Indeterminate
1	0	0	1 (Hold)
1	1	0	0
1	0	1	1
1	1	1	Indeterminate

RESULT:

CONCLUSION:

EXPT NO:

DATE:

Study of JK flip flop

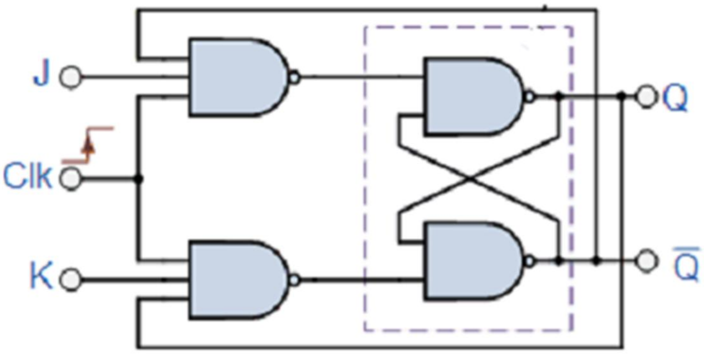
Aim: Implementation of JK flip flop

Theory:

JK FLIP-FLOP:

The JK flip flop (JK means Jack Kilby, a Texas instrument engineer, who invented it) is the most versatile flip-flop, and the most commonly used flip flop. Like the RS flip-flop, it has two data inputs, J and K, and an EN/clock pulse input (CP). Note that in the following circuit diagram NAND gates are used instead of NOR gates. It has no undefined states, however. The fundamental difference of this device is the feedback paths to the AND gates of the input, i.e. Q is AND-ed with K and CP and Q with J and CP.

Circuit Diagram:



CHARACTERISTIC TABLE:

Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1(Toggle,$\bar{Q}_n$)
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0(Toggle,$\bar{Q}_n$)

The JK flip-flop has the following characteristics:

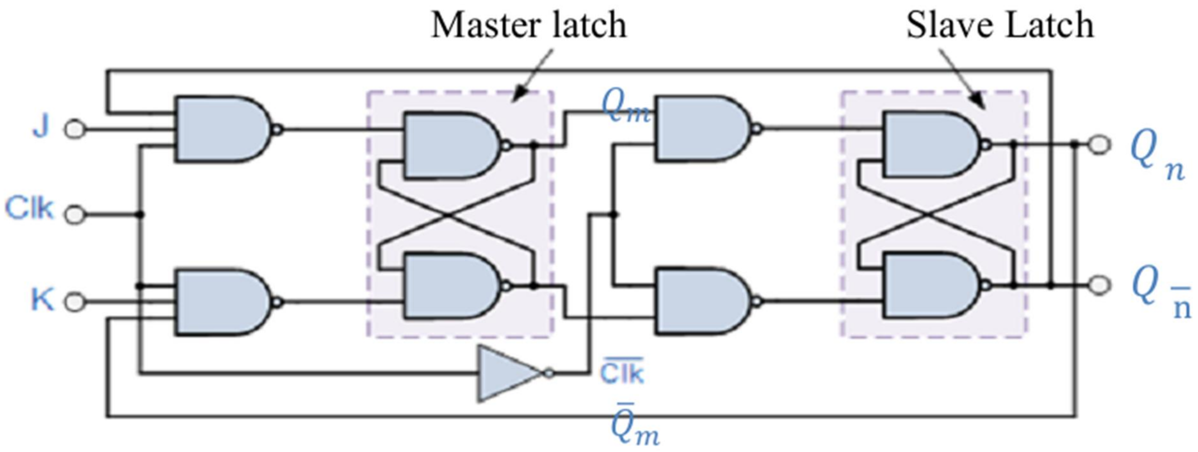
- If one input (J or K) is at logic 0, and the other is at logic 1, then the output is set or reset (by J and K respectively), just like the RS flip-flop.
- If both inputs are 0, then it remains in the same state as it was before the clock pulse occurred; again like the RS flip flop. CP has no effect on the output.
- If both inputs are high, however the flip-flop changes state whenever a clock pulse occurs; i.e., the clock pulse toggles the flip-flop again and again until the CP goes back to 0 as shown in the shaded rows of the characteristic table above. Since this condition is undesirable, it should be eliminated by an improvised form of this flip-flop as discussed in the next section.

MASTER-SLAVE JK FLIP-FLOP:

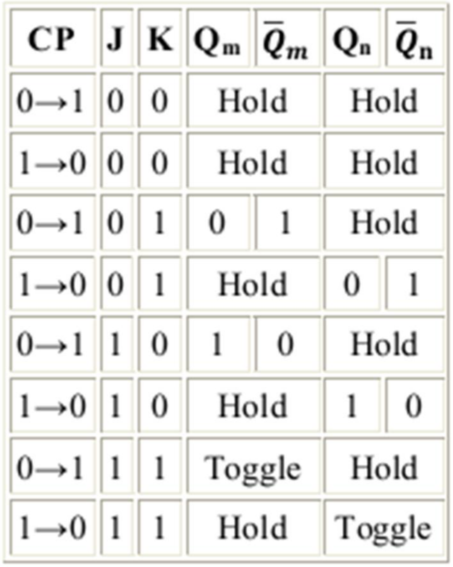
Although JK flip-flop is an improvement on the clocked SR flip-flop it still suffers from timing problems called "race" if the output Q changes state before the timing pulse of the clock input has time to go "OFF", so the timing pulse period (T) must be kept as short as possible (high frequency). As this is sometimes not possible with modern TTL IC's the much improved Master-Slave J-K Flip-Flop was developed. This eliminates all the timing problems by using two SR flip-flops connected together in series, one for the "Master" circuit, which triggers on the leading edge of the clock pulse and the other, the "Slave" circuit, which triggers on the falling edge of the clock pulse.

The master-slave JK flip flop consists of two flip flops arranged so that when the clock pulse enables the first, or master, it disables the second, or slave. When the clock changes state again (i.e., on its falling edge) the output of the master latch is transferred to the slave latch. Again, toggling is accomplished by the connection of the output with the input AND gates.

Circuit Diagram:



Characteristic Table



CP	J	K	Q_m	$\bar{Q}_m$	Q_n	$\bar{Q}_n$
0→1	0	0	Hold		Hold	
1→0	0	0	Hold		Hold	
0→1	0	1	0	1	Hold	
1→0	0	1	Hold		0	1
0→1	1	0	1	0	Hold	
1→0	1	0	Hold		1	0
0→1	1	1	Toggle		Hold	
1→0	1	1	Hold		Toggle	

RESULT:

CONCLUSION: