

**BCSC:0804 COMPUTER ORGANIZATION LAB**

**Semester- III L–T–P: 0–0–2**

**Credits: 01**

**EXPERIMENT LIST**

1. Bread Board implementation of Flip-Flops.
2. Experiments with clocked Flip-Flop.
3. Design of Counters.
4. Bread Board implementation of counters & shift register.
5. Implementation of Arithmetic algorithms.
6. Bread Board implementation of Adder/ Subtractor (Half, full)
7. Bread Board implementation of Binary Adder.
8. Bread Board implementation of seven segment display.

## INDEX

Name of student.....Class/ Batch..... Roll No. ....

Course/Branch..... Session..... Semester.....

Name of subject..... Subject Code.....

S. No.	Experiment	Date of Perform	Date of Submission	Grading scale(10)	Signature
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

**Overall Grading Average (10)=**

**Signature of Evaluator**

## EXPERIMENT NO. 1

**OBJECTIVE:** Bread board implementation of flip-flops (SR, JK, D, T) using IC 7400.

### PRE EXPERIMENT QUESTIONS:

Q1. What is the difference between latch and flip-flop

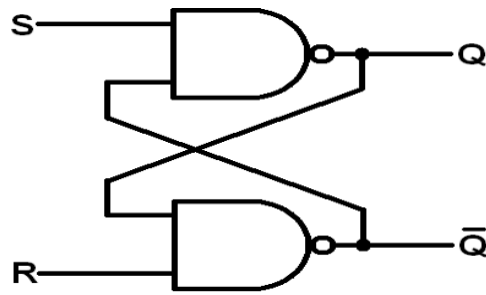
### APPARATUS AND MATERIAL REQUIRED:

S No.	COMPONENT	SPECIFICATION	QTY.
1.	NAND GATE	IC-7400	5
2.	POWER SUPPLY +5V	-	1
3.	BREAD BOARD	-	1
4.	LED		ADEQUATE
5.	CONNECTING WIRE		ADEQUATE

### THEORY:

**RS FLIP-FLOP:** The drawback of 1 bit latch that is there is no provision of entering the desired I/P, can be eliminated by RS Flip-Flop. The fig 1 shows the clocked RS Flip-Flop. It uses Four two input NAND Gate. The circuit has two stable states in one of the stable state  $Q=1$  which is referred to as the 1 state or the set state, whereas in the other stable state.  $Q=0$  which is referred to as the 0 state or reset state. These are given in the truth table. The circuit is designated as SR Flip-Flop because when S I/P bring the circuit in set state and R I/P bring it to reset or clear state.

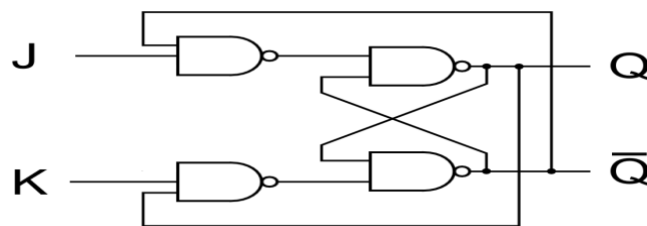
If  $S=R=1$  both O/Ps Q and  $\bar{Q}$  will try to become 1 which is not allowed, this I/P condition undefined, indeterminate or ambiguous and therefore indicated by question mark. SR Flip-Flop is an important circuit because all other Flip-Flop designed from it.



RS Flip Flop

**JK FLIP-FLOP:** A JK Flip-Flop is refinement of the RS Flip-Flop in that the indeterminate state of the RS type is defined in JK type. In JK inputs J&K are set and clear respectively.

When IP are applied to both J and K simultaneously, the Flip-Flop switches to its complement state, that is if  $Q=1$ , it switches to  $Q=0$  and vice versa.



JK Flip Flop

**D FLIP- FLOP:** The D Flip- Flop is a modification of the clocked JK Flip-Flop. The D I/P goes directly to input and its complement through NAND gate applied to the K I/P as shown in Fig.2. The Flip-Flop receives the designation from the ability to transfer data into a Flip-Flop. It is basically a JK Flip-Flop with an inverter in K I/P. The added inverter reduces the number of I/P from two to one. This type of the D Flip-Flop is called delay Flip-Flop because the I/P data appears at the end of the clock pulse and hence transfer of data from IP to O/P is delayed. The D Flip-Flop also eliminates the latch condition of the JK Flip-Flop.

**T FLIP-FLOP:** The T Flip-Flop is single I/P version of the JK Flip-Flop. As shown in Fig-3. The Flip-Flop is obtained from a JK type if both inputs are tied together. The designation T comes from the ability of the Flip-Flop to “TOGGLE” or change state for every clock pulse. It used as toggle switch.

## **TRUTH TABLE:**

### For SR Flip-Flop:

<b>R</b>	<b>S</b>	<b><math>Q_{n+1}</math></b>	<b>Action</b>
0	0	$Q_n$	No-Action
0	1	1	Set
1	0	0	Clear
1	1	?	Forbidden

### For JK Flip-Flop:

<b>J</b>	<b>K</b>	<b><math>Q_{n+1}</math></b>	<b>Action</b>
0	0	$Q_n$	No-Action
0	1	0	Clear
1	0	1	Set
1	1	$Q_n'$	Toggles

### For T Flip-Flop:

<b>INPUT(T)</b>	<b>OUTPUT(<math>Q_{n+1}</math>)</b>	<b>Action</b>
0	$Q_n$	No-Change
1	$Q_n'$	Toggles

### For D Flip-Flop:

<b>D</b>	<b><math>Q_{n+1}</math></b>	<b>Action</b>
0	0	Clear
1	1	Set

### PROCEDURE:

1. Make connections as per circuit diagram.
2. Logical inputs are given as per circuit diagram.
3. Observe the output and verify the truth table.

**RESULT:** various flip flop's has been implemented and verified.

**POST EXPERIMENTS QUESTIONS:**

- Q1. Describe the working of J-K flip-flop.
- Q2. What is the merit of J-K flip-flop versus S-R flip-Flop?
- Q3. What is meant by Race-Around condition?

**PRECAUTIONS:**

- 1. Switch off the power supply while making connections.
- 2. All the connections should be made properly as per the circuit diagram.
- 3. Get your circuit checked either by faculty or by lab assistant before you switch on the power supply.

## **EXPERIMENT NO. 2**

**OBJECTIVE:** To design and implement RS, JK, D and T flip-flop using logic gates.

### **PRE EXPERIMENT QUESTIONS:**

Q1. Define Latch.

Q2. Define Flip-flop.

Q3. What is the difference between Flip-Flop & latch?

### **APPARATUS REQUIRED:**

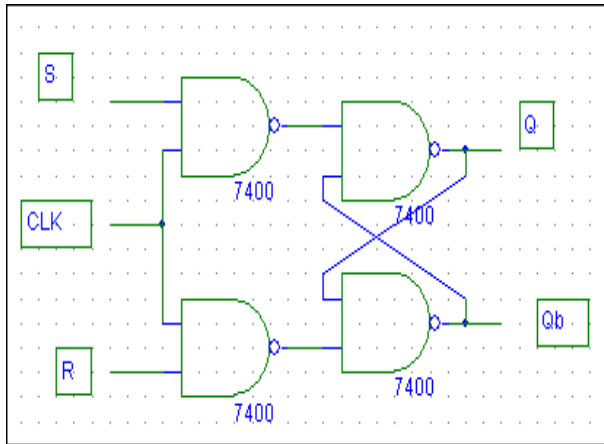
S No.	COMPONENT	SPECIFICATION	QTY.
1.	2 I/P NAND GATE	IC 7400	1
2.	3 I/P NAND GATE	IC 7410	1
3.	POWER SUPPLY +5V	-	1
4.	BREAD BOARD	-	1
5.	LED		ADEQUATE
6.	CONNECTING WIRE		ADEQUATE

### **THEORY:**

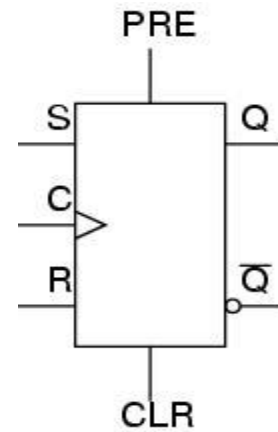
Logic circuits that incorporate memory cells are called *sequential logic circuits*; their output depends not only upon the present value of the input but also upon the previous values of output. Sequential logic circuits often require a timing generator (a clock) for their operation. The latch (flip-flop) is a basic bi-stable memory element widely used in sequential logic circuits. Usually there are two outputs, Q and its complementary value.

### **SR FLIP-FLOP:**

An S-R flip-flop can also be design using cross-coupled NAND gates as shown. A clocked S-R flip-flop has an additional clock input so that the S and R inputs are active only when the clock is high. When the clock goes low, the state of flip-flop is latched and cannot change until the clock goes high again. Therefore, the clocked S-R flip-flop is also called “enabled” S-R flip-flop.



SR flip-flop



Symbol

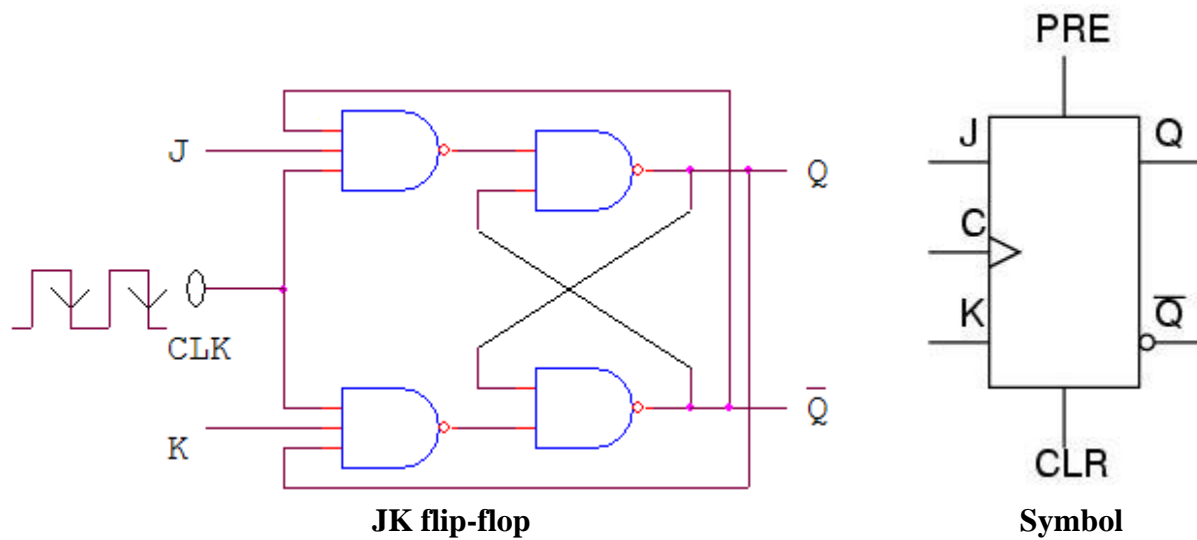
### TRUTH TABLE:

SR Flip-flop					
Clock	Q(t)	S	R	Q(t+1)	Action
1	0	0	0	Q(t)	No change
1	0	0	1	0	reset
1	0	1	0	1	set
1	0	1	1	x	indetermined

### JK FLIP-FLOP

JK flip-flop is the modification of SR flip-flop which has its own way of dealing with RS flip-flop's undesirable input combination (when S=1 and R=1). The JK flip-flop has 2 input J and K, each ANDed with clock pulse and its corresponding output



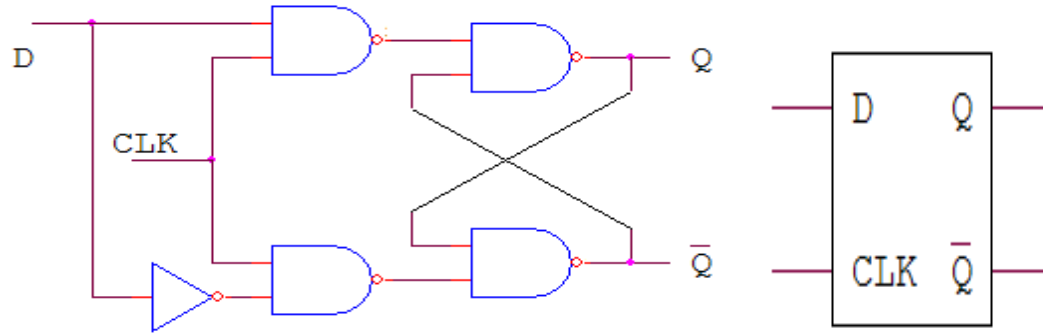


**TRUTH TABLE:**

JK Flip-flop					
Clock	Q(t)	J	K	Q(t+1)	Action
1	0	0	0	Q(t)	No change
1	0	0	1	0	reset
1	0	1	0	1	set
1	0	1	1	$\bar{Q}$	Race around condition

**D FLIP-FLOP**

A D latch combines the S and R inputs of an S-R latch into one input by adding an inverter. When the clock is high, the output follows the D input, and when the clock goes low, the state is latched.



**D flip-flop**

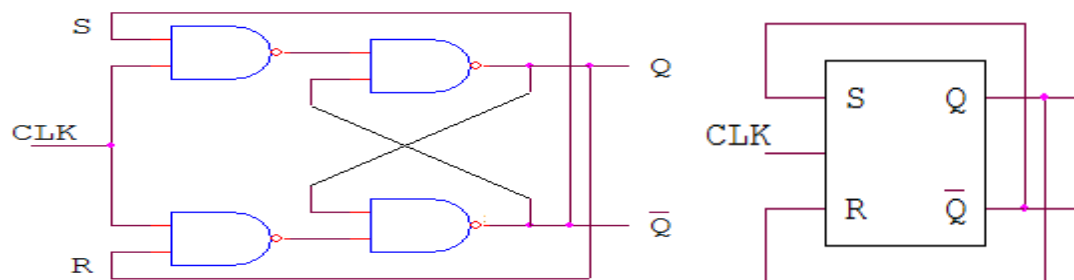
**Symbol**

**TRUTH TABLE:**

D flip-flop			
Clock	D	Q(t)	Q(t+1)
1	0	0	1
1	1	1	0

### T FLIP-FLOP (Toggle)

A S-R flip-flop can be converted to T-flip flop by connecting S input to Q' and R to Q.



**T flip-flop**

**Symbol**

**TRUTH TABLE:**

<b>T flip-flop</b>		
<b>Clock</b>	<b>T</b>	<b>Q(t + 1)</b>
<b>1</b>	<b>0</b>	<b>Q(t)</b>
<b>1</b>	<b>1</b>	<b>Q(t)'</b>

**PROCEDURE:**

1. Make connections as per circuit diagram.
2. Logical inputs are given as per circuit diagram.
3. Observe the output and verify the truth table.

**RESULT:** State table of SR, JK, D and T flip-flop has been verified.

**POST EXPERIMENTS QUESTIONS:**

- Q1. Which IC is used for SR flip-flop?
- Q2. What are the applications of different Flip-Flops?
- Q3. What is the advantage of Edge triggering over level triggering?
- Q4. What is the relation between propagation delay & clock frequency of flip-flop?
- Q5. What is race around in flip-flop & how to overcome it?

**PRECAUTIONS:**

1. Switch off the power supply while making connections.
2. All the connections should be made properly as per the circuit diagram.

Get your circuit checked, either by faculty or by lab assistant before you switch on the power supply.

### **EXPERIMENT NO. 3**

**OBJECTIVE:** Design of counter (Mod-10)

**PRE EXPERIMENT QUESTIONS:**

Q1. What mod-N counter means?

Q2. What is the function of counter?

**APPARATUS REQUIRED:**

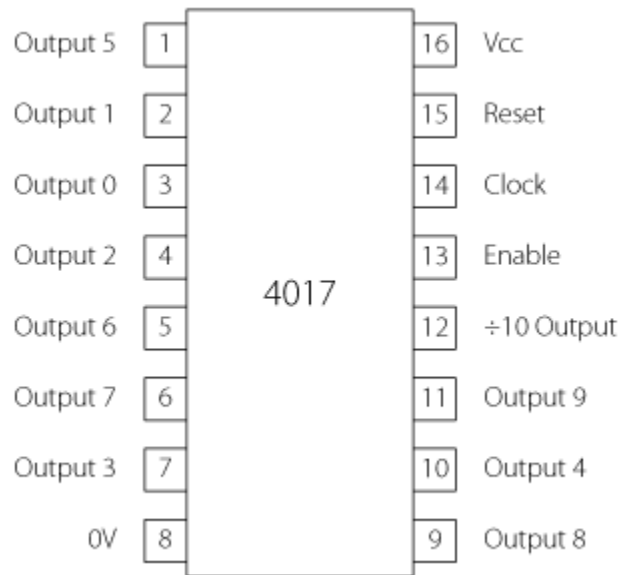
S No.	COMPONENT	SPECIFICATION	QTY.
1.	IC	IC 4017	1
2.	POWER SUPPLY +5V	-	1
3.	BREAD BOARD	-	1
4	LED		ADEQUATE
5	CONNECTING WIRE		ADEQUATE

**THEORY:**

A BCD Counter counts in binary coded decimal from 0000 to 1001 and back to 0000. Because of the return to 0 after a count of 9, a BCD counter does not have a regular pattern as in a straight binary count.

Synchronous BCD counter can be cascaded to form a counter for decimal number of any length.

**PIN DIAGRAM:**



### LOGIC DIAGRAM:

### TRUTH TABLE:

COUNT	Q <sub>9</sub>	Q <sub>8</sub>	Q <sub>7</sub>	Q <sub>6</sub>	Q <sub>5</sub>	Q <sub>4</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0

6	0	0	0	1	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0

**PROCEDURE:**

1. Make connections as per circuit diagram.
2. Logical inputs are given as per circuit diagram.
3. Observe the output and verify the truth table.

**RESULT:** The operation of BCD counter has been verified.

**POST EXPERIMENTS QUESTIONS:**

- Q1. What is the role of an encoder in communication?  
Q2. What is the advantage of using an encoder?

**PRECAUTIONS:**

1. Switch off the power supply while making connections.
2. All the connections should be made properly as per the circuit diagram.
3. Get your circuit checked, either by faculty or by lab assistant before you switch on the power supply.

### EXPERIMENT NO. 4(A)

**OBJECTIVE:** Bread board implementation of counter.

#### PRE EXPERIMENT QUESTIONS:

- Q1. Differentiate between combinational and sequential circuit.  
Q2. Define synchronous sequential circuits.  
Q3. Define Asynchronous sequential circuits.

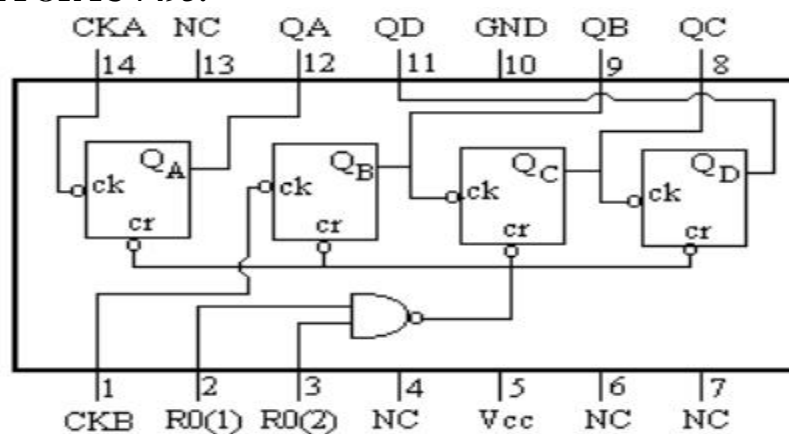
#### APPARATUS REQUIRED:

S No.	COMPONENT	SPECIFICATION	QTY.
1.	JK FLIP FLOP	IC 7493	2
2.	POWER SUPPLY +5V	-	1
3.	BREAD BOARD	-	1
4.	LED		ADEQUATE
5.	CONNECTING WIRE		ADEQUATE

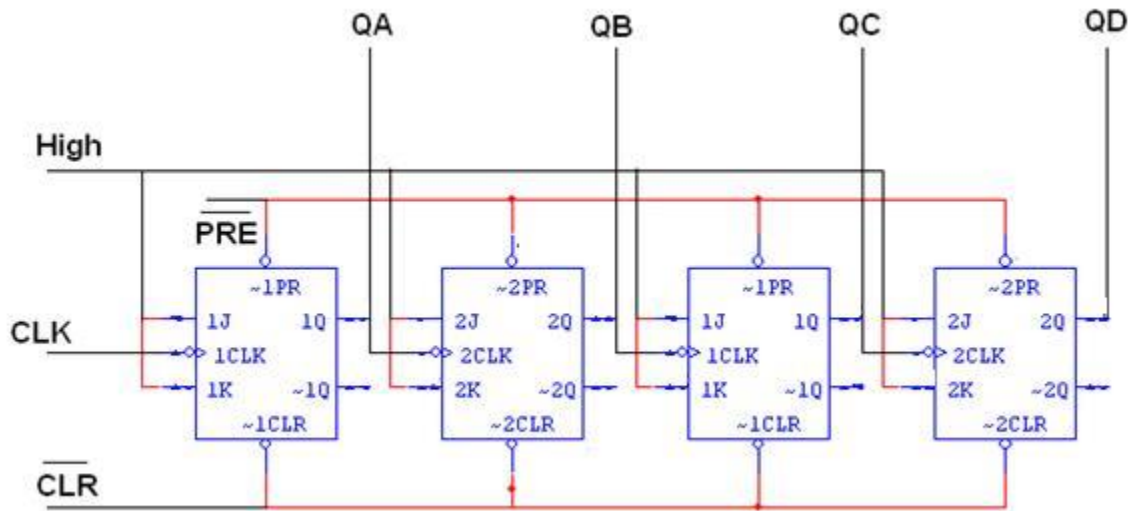
#### THEORY:

A counter is a register capable of counting number of clock pulse arriving at its clock input. Counter represents the number of clock pulses arrived. A specified sequence of states appears as counter output. This is the main difference between a register and a counter. There are two types of counter, synchronous and asynchronous. In synchronous common clock is given to all flip flop and in asynchronous first flip flop is clocked by external pulse and then each successive flip flop is clocked by Q or Q output of previous stage. As soon the clock of second stage is triggered by output of first stage. Because of inherent propagation delay time all flip flops are not activated at same time which results in asynchronous operation.

#### PIN DIAGRAM FOR IC 7493:



### LOGIC DIAGRAM FOR 4 BIT RIPPLE COUNTER:



### TRUTH TABLE:

CLK	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1
10	0	1	0	1
11	1	1	0	1
12	0	0	1	1
13	1	0	1	1
14	0	1	1	1
15	1	1	1	1



**PROCEDURE:**

1. Make connections as per circuit diagram.
2. Logical inputs are given as per circuit diagram.
3. Observe the output and verify the truth table.

**RESULT:** 4 bit ripple counter is been implemented and verified.

**POST EXPERIMENTS QUESTIONS:**

- Q1. Define 'module' (MOD) with respect to counter.  
Q2. Maximum number of states of a counter is?  
Q3. Design a MOD-3 ripple counter with 3 JK flip-flop.

**PRECAUTIONS:**

1. Switch off the power supply while making connections.
2. All the connections should be made properly as per the circuit diagram.
3. Get your circuit checked, either by faculty or by lab assistant before you switch on the power supply.

### **EXPERIMENT NO. 4(B)**

**OBJECTIVE:** To design and implement

- (i) Serial in serial out
- (ii) Parallel in parallel out
- (iii) Serial in parallel out
- (iv) Parallel in serial out

### **PRE EXPERIMENT QUESTIONS:**

Q1.What is PISO, SIPO, and SISO with respect to shift register?

Q2.Differentiate between serial data & parallel data

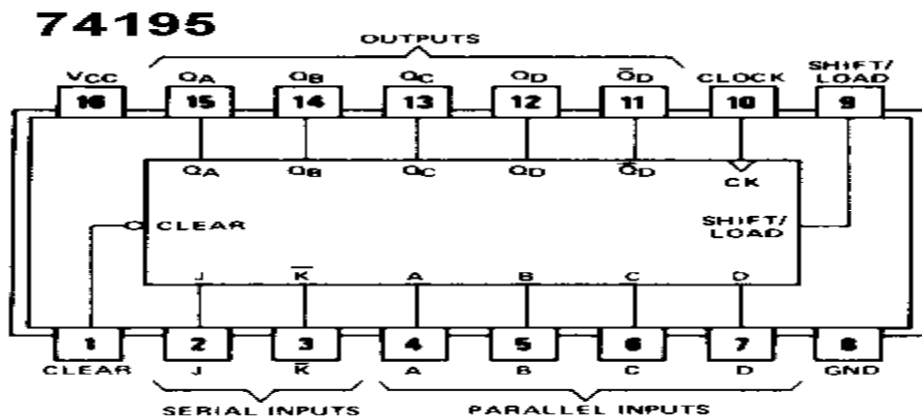
### **APPARATUS REQUIRED:**

S No.	COMPONENT	SPECIFICATION	QTY.
1.	UNIVERSAL SHIFT REGISTER	IC 74195	1
2.	POWER SUPPLY +5V	-	1
3.	BREAD BOARD	-	1
4.	LED		ADEQUATE
5.	CONNECTING WIRE		ADEQUATE

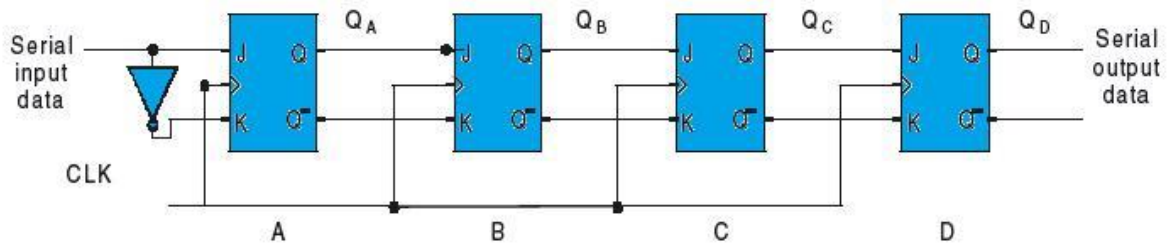
### **THEORY:**

A register is capable of shifting its binary information in one or both directions is known as shift register. The logical configuration of shift register consist of a D-Flip flop cascaded with output of one flip flop connected to input of next flip flop. All flip flops receive common clock pulses which causes the shift in the output of the flip flop. The simplest possible shift register is one that uses only flip flop. The output of a given flip flop is connected to the input of next flip flop of the register. Each clock pulse shifts the content of register one bit position to right.

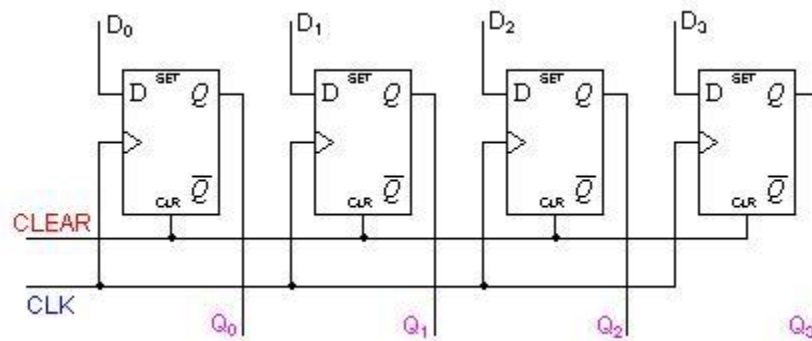
### **PIN DIAGRAM:**



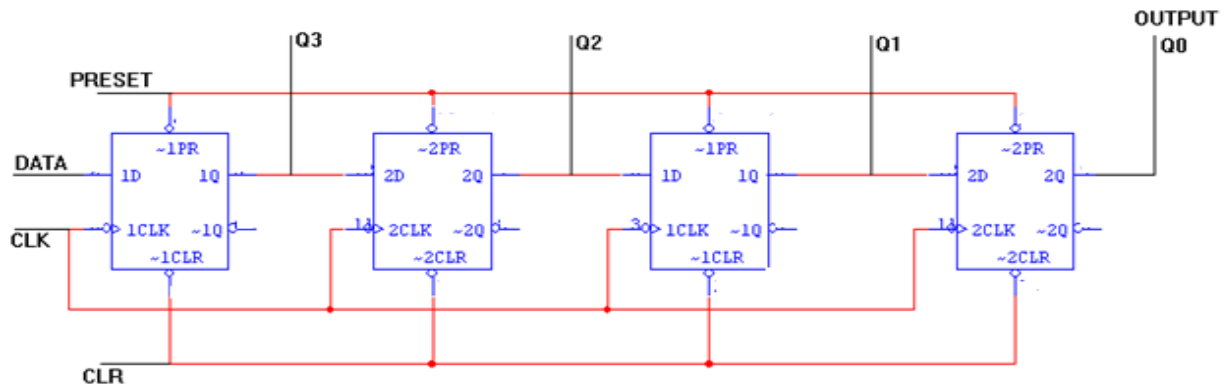
**LOGIC DIAGRAM:  
SERIAL IN SERIAL OUT:**



**PARALLEL IN SERIAL OUT:**



**SERIAL IN PARALLEL OUT:**

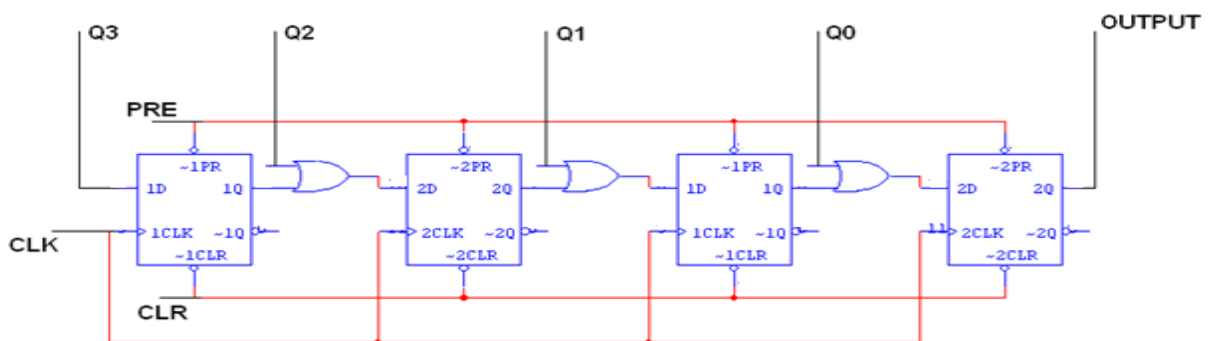


**TRUTH TABLE:**

CLK	DATA	OUTPUT			
		Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>C</sub>	Q <sub>D</sub>
INITIAL		0	0	0	0
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	0	0	1

**LOGIC DIAGRAM:**

**PARALLEL IN SERIAL OUT:**



**TRUTH TABLE:**

CLK	Q3	Q2	Q1	Q0	O/P
0	1	0	0	1	1

1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	1

**PROCEDURE:**

1. Make connections as per circuit diagram.
2. Logical inputs are given as per circuit diagram.
3. Observe the output and verify the truth table.

**RESULT:** Serial in parallel out and Parallel in serial out shift register has been implemented and verified

**POST EXPERIMENTS QUESTIONS:**

Q1. On the fifth clock pulse, a 4-bit Johnson sequence is  $Q_0 = 0$ ,  $Q_1 = 1$ ,  $Q_2 = 1$ , and  $Q_3 = 1$ . On the sixth clock pulse, the sequence is \_\_\_\_\_.

Q2. The bit sequence 0010 is serially entered (right-most bit first) into a 4-bit parallel out shift register that is initially clear. What are the Q outputs after two clock pulses?

**PRECAUTIONS:**

1. Switch off the power supply while making connections.
2. All the connections should be made properly as per the circuit diagram.
3. Get your circuit checked, either by faculty or by lab assistant before you switch on the power supply.

## EXPERIMENT NO. 5

**OBJECTIVE:** Implementation of Arithmetic Algorithms.

### PRE EXPERIMENT QUESTIONS:

Q1. Define ALU algorithms.

Q2. How many types of algorithms are there?

### APPARATUS AND MATERIAL REQUIRED:

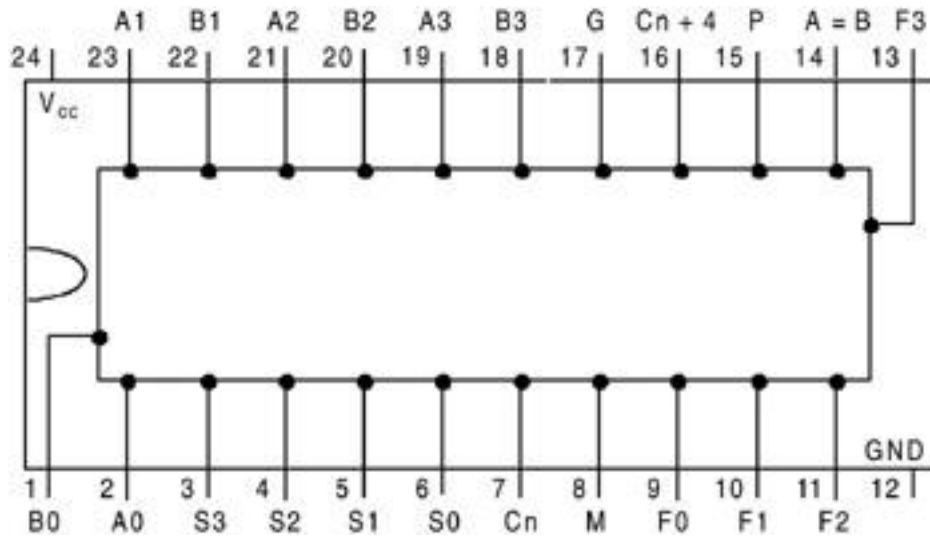
S No.	COMPONENT	SPECIFICATION	QTY.
1.		IC 74181	2
4.	POWER SUPPLY +5V	-	1
5.	BREAD BOARD	-	1
6	LED	-	ADEQUATE
7	CONNECTING WIRE	-	ADEQUATE

### THEORY:

Arithmetic Logic Unit is a multipurpose device capable of providing several different arithmetic and logic operations. The specific operation to be performed is selected by the user by placing a specific binary code on the mode select input. ALU's are available in large scale integrated Circuit packages.

Functional block diagram for 74181 ALU is shown in figure. It is a 4 bit ALU, which provides 16 arithmetic and 16 logic operations. The unit accepts two 4-bit words ( $A_3, A_2, A_1, A_0$  and  $B_3, B_2, B_1, B_0$ ) and a carry input  $C_n$  as inputs. The operations to be performed on those inputs are determined by logic levels on inputs.

### PIN DIAGRAM:



IC 74181

**FUNCTIONAL TABLE:**

MODE				DATA OUTPUTS		
				BA=H: LOG. FUNC	BA=L: ARITHMETIC FUNCTION	
S3	S2	S1	S0	Cyn=X	CYn=H	CYn=L
L	L	L	L	A'	A	A plus 1
L	L	L	H	(A+B)'	A+B	(A+B) plus 1
L	L	H	L	A'B	A+B'	(A+B') plus 1
L	L	H	H	L	Minus 1	Zero
L	H	L	L	(AB)'	A plus (AB)'	A plus (AB)' plus 1
L	H	L	H	B'	(A+B) plus (AB')	(A+B) plus (AB') plus 1
L	H	H	L	A XOR B	A minus B minus 1	A minus B
L	H	H	H	AB'	(AB') minus 1	AB'
H	L	L	L	A'+B	A plus (AB')	A plus (AB') plus 1
H	L	L	H	(A XOR B)'	A plus B	A plus B plus 1
H	L	H	L	B	(A+B') plus (AB)	(A+B') plus (AB) plus 1
H	L	H	H	AB	(AB) minus 1	AB
H	H	L	L	H	A plus A	A plus A plus 1
H	H	L	H	A+B'	(A+B) plus A	(A+B) plus A plus 1
H	H	H	L	A+B	(A + B') plus A	(A + B') plus A plus 1
H	H	H	H	A	A minus 1	A

**PROCEDURE:**

1. Make connections as per circuit diagram.
2. Logical inputs are given as per circuit diagram.
3. Observe the output and verify the truth table.

**RESULT:** functional table of IC 74181 is been verified

**POST EXPERIMENTS QUESTIONS**

Q1. Write all functions of ALU.

Q2. What is the difference between logic function and arithmetic operations?

**PRECAUTIONS**

1. Switch off the power supply while making connections.
2. All the connections should be made properly as per the circuit diagram.
3. Get your circuit checked, either by faculty or by lab assistant before you switch on the power supply.



## EXPERIMENT NO. 6

**OBJECTIVE:** To design and construct half adder, full adder, half subtractor and full subtractor circuits and verify the truth table using logic gates.

### PRE EXPERIMENT QUESTIONS:

- Q1. Define half adder.
- Q2. Define full adder.
- Q4. Define half subtractor.
- Q5. Define full subtractor.

### APPARATUS AND MATERIAL REQUIRED:

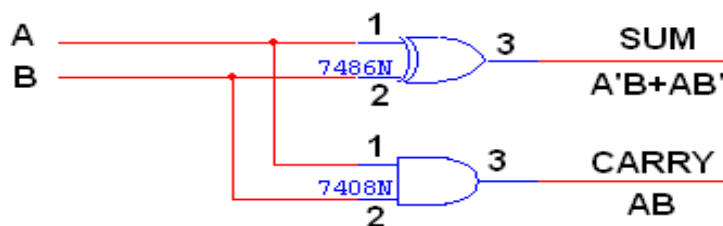
S No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	1
2.	X-OR GATE	IC 7486	1
3.	NOT GATE	IC 7404	1
4.	OR GATE	IC 7432	1
4.	POWER SUPPLY +5V	-	1
5.	BREAD BOARD	-	1
6	LED	-	ADEQUATE
7	CONNECTING WIRE	-	ADEQUATE

### 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.

#### LOGIC DIAGRAM:



**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:**

B	00	01
A 00		1
A 01	1	

$$\text{SUM} = A'B + AB'$$

**K-Map for CARRY:**

B	00	01
A 00		
A 01		1

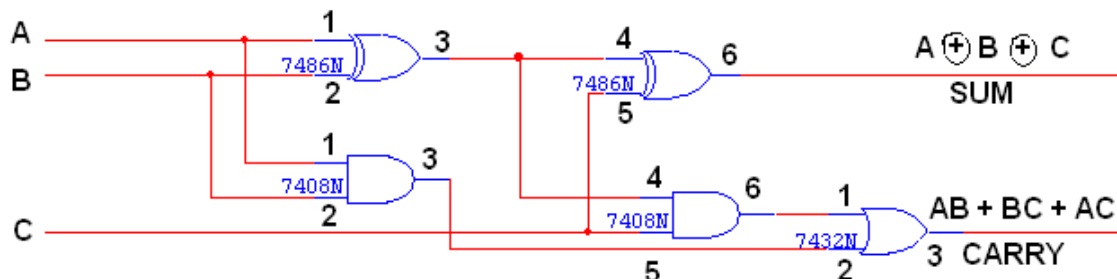
$$\text{CARRY} = AB$$

**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.

**LOGIC DIAGRAM:**

**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	

$$\text{SUM} = A'B'C + A'BC' + ABC' + ABC$$

**K-Map for CARRY:**

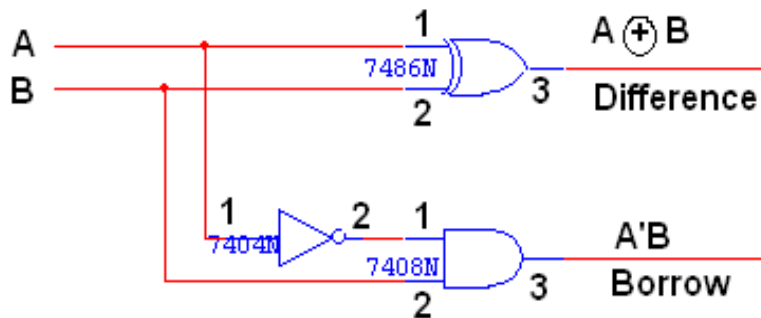
A \ BC	00	01	11	10
0			1	
1		1	1	1

$$\text{CARRY} = AB + BC + AC$$

**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.

**LOGIC DIAGRAM:**



### 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:

		B	
		00	01
A	00		1
	01	1	

$$\text{DIFFERENCE} = A'B + AB'$$

### K-Map for BORROW:

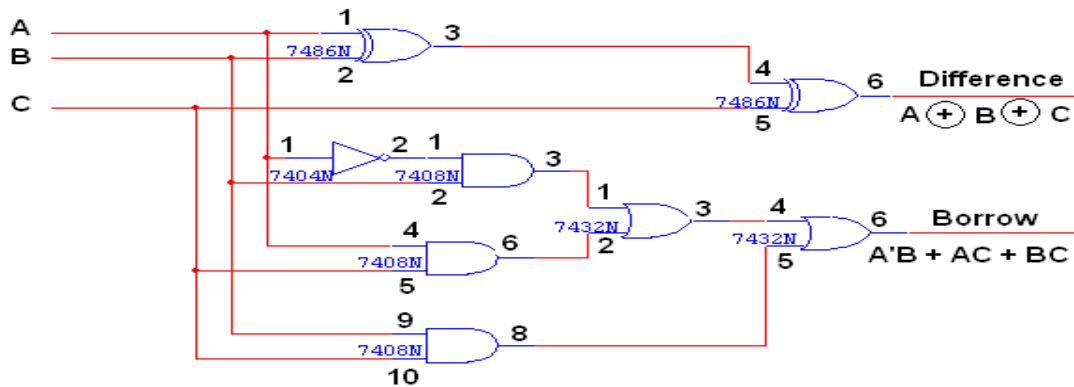
		B	
		00	01
A	00		1
	01		

$$\text{BORROW} = A'B$$

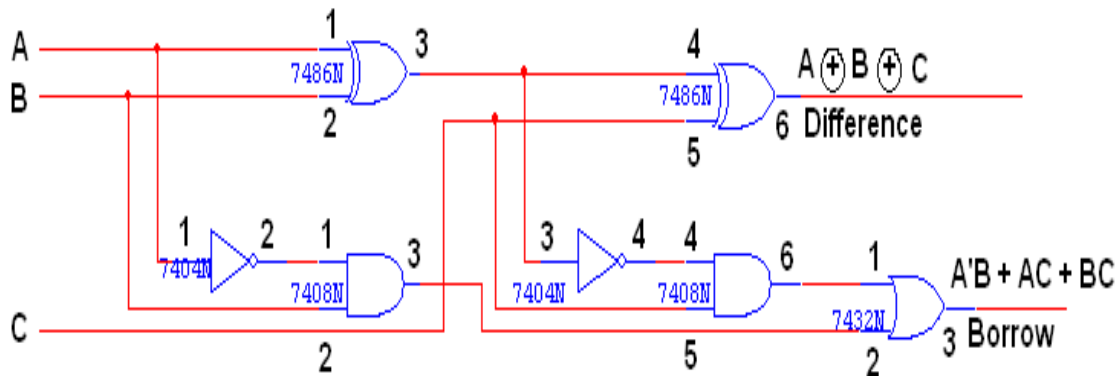
### FULL SUBTRACTOR:

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:**



**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	

$$\text{Difference} = A'B'C + A'BC' + AB'C' + ABC$$

**K-Map for Borrow:**

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

$$\text{Borrow} = A'B + BC + A'C$$

**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:** Half adder, Full adder, Halfsubtractor and Full subtractor circuits are implemented and verified.

**POST EXPERIMENTS QUESTIONS:**

- Q1. What are the applications of adders?
- Q2. What are the applications of subtractors?
- Q3. Realize a full adder using two half adders.
- Q4. Realize a full subtractors using two half subtractors.

**PRECAUTIONS:**

1. Switch off the power supply while making connections.
2. All the connections should be made properly as per the circuit diagram.
3. Get your circuit checked, either by faculty or by lab assistant before you switch on the power supply.

## EXPERIMENT NO. 7

**OBJECTIVE:** Bread board implementation of Binary Adder.

**PRE EXPERIMENT QUESTIONS:**

Q1. Describe the block diagram structure of IC 7483.

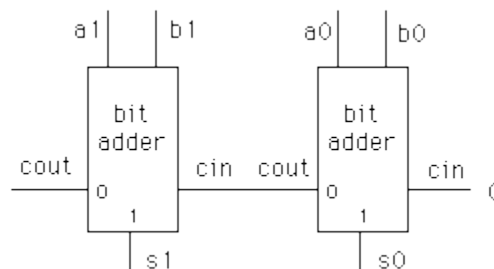
Q2. Design a full adder by using K- MAP.

**APPARATUS REQUIRED:**

S No.	COMPONENT	SPECIFICATION	QTY.
1.	4 BIT PARALLEL ADDER	IC 7483	1
2.	Ex- OR	IC 7486	1
3.	POWER SUPPLY +5V	-	1
4.	BREAD BOARD	-	1
5.	LED		ADEQUATE
6.	CONNECTING WIRE		ADEQUATE

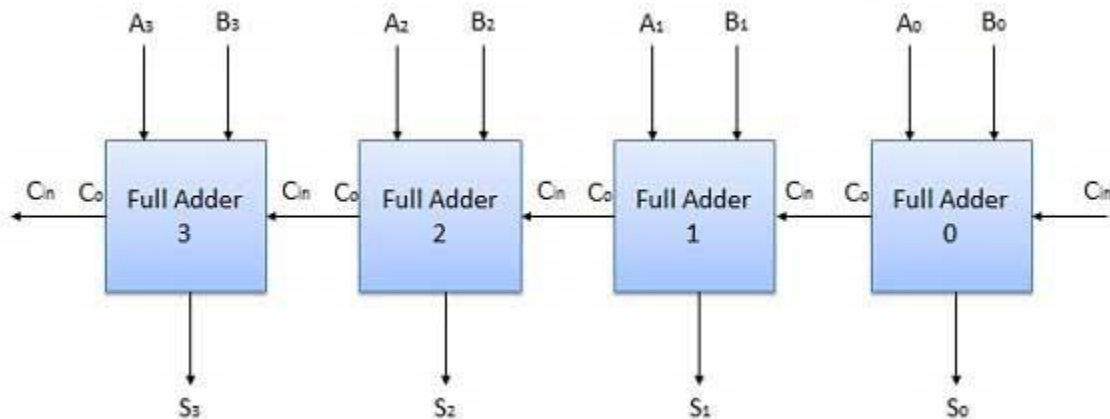
**THEORY:**

The Full adder can add single-digit binary numbers and a carry. The largest sum that can be obtained using a full adder is  $11_2$ . Parallel adders can add multiple-digit numbers. If full adders are placed in parallel, we can add two- or four-digit numbers or any other size desired.

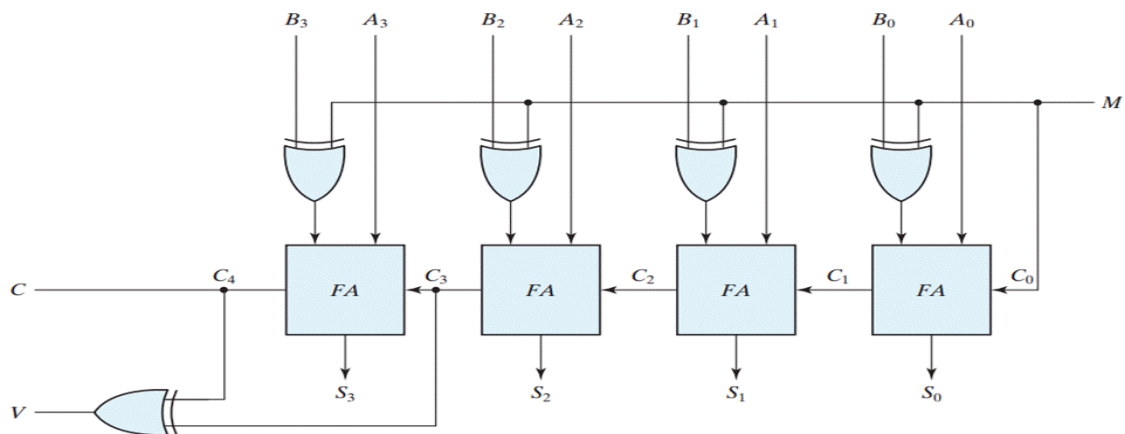


The addend would be on A inputs, and the augend on the B inputs. For this explanation we will assume there is no input to  $C_0$  (carry from a previous circuit). To add  $10_2$  (addend) and  $01_2$  (augend), the addend inputs will be 1 on  $A_2$  and 0 on  $A_1$ . The augend inputs will be 0 on  $B_2$  and 1 on  $B_1$ . Working from right to left, as we do in normal addition, let's calculate the outputs of each full adder. With  $A_1$  at 0 and  $B_1$  at 1, the output of adder1 will be a sum ( $S_1$ ) of 1 with no carry ( $C_1$ ).

Since  $A_2$  is 1 and  $B_2$  is 0, we have a sum ( $S_2$ ) of 1 with no carry ( $C_2$ ) from adder1. To determine the sum, read the outputs ( $C_2$ ,  $S_2$ , and  $S_1$ ) from left to right. In this case,  $C_2 = 0$ ,  $S_2 = 1$ , and  $S_1 = 1$ . The sum, then, of  $10_2$  and  $01_2$  is  $11_2$ . To add four bits we require four full adders arranged in parallel.

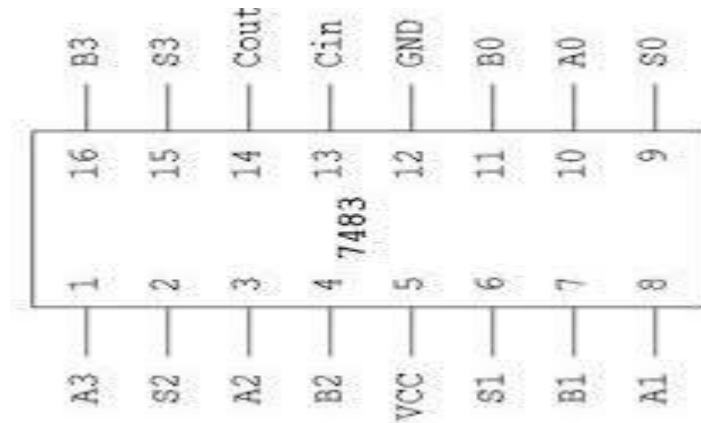


Subtraction is performed adding 2's complement of augend and addend i.e by simply taking 1's complement of augend and taking input carry 1. Common circuit can be used to perform addition as well as subtraction by simply passing each augend bit through an 2 input Ex-OR gate and connecting one input of each Ex-OR gate and input carry to a common terminal known as Mode input. This mode input decides whether circuit will perform addition or subtraction. If Mode =0, it perform addition, if Mode = 1, than circuit will perform subtraction.



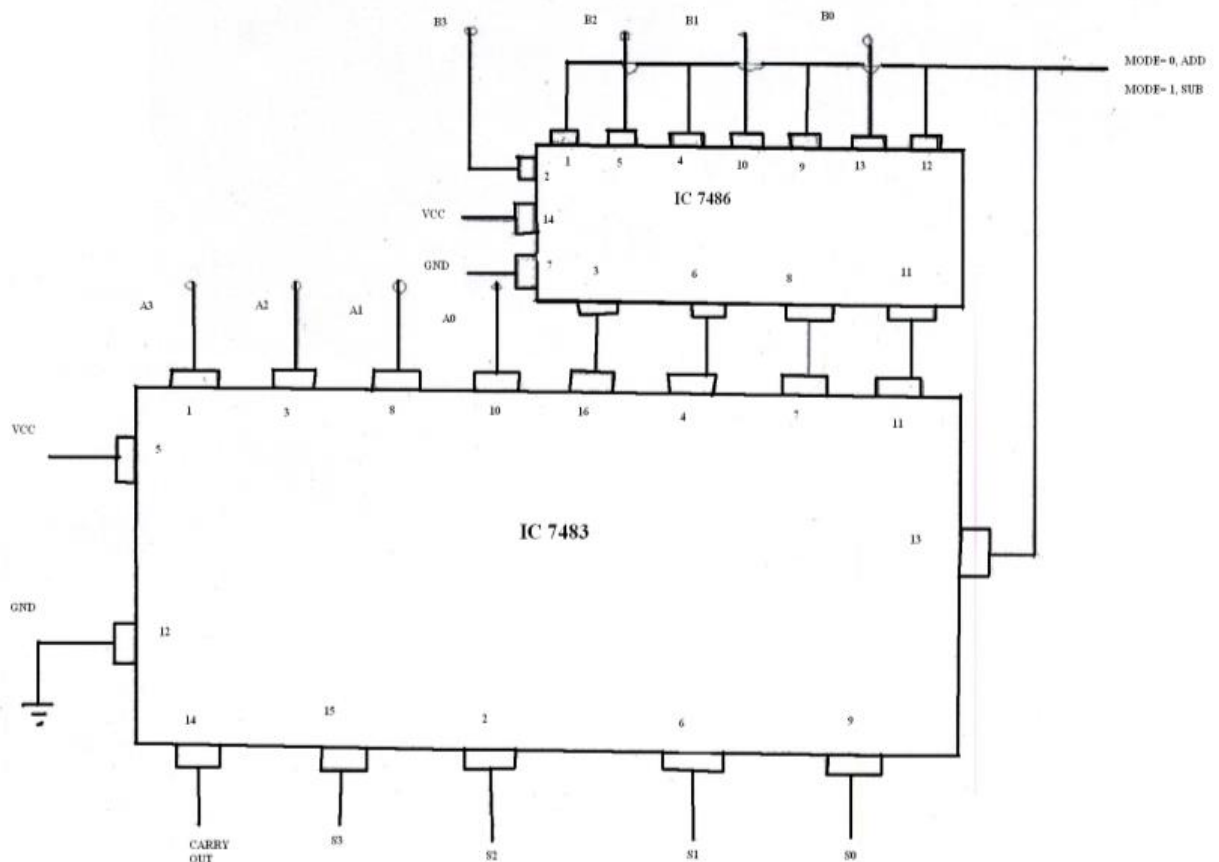


**PIN DIAGRAM:**



**IC 7483**

**CIRCUIT DIAGRAM**



**PROCEDURE:**

1. Connections are given as per circuit diagram.
2. Logical inputs are given as per circuit diagram.
3. Observe the output and verify the truth table.

**RESULT:** Adder/ Subtractor function of IC7483 has been performed and verified..

**POST EXPERIMENTS QUESTIONS:**

Q1. How we convert a 4-bit adder IC to 4-bit subtractor?

Q2. How we use X-OR for 1's complement?

**PRECAUTIONS:**

3. Switch off the power supply while making connections.
4. All the connections should be made properly as per the circuit diagram.
5. Get your circuit checked, either by faculty or by lab assistant before you switch on the power supply.