

VERIFICATION OF GATES

Aim: - To study and verify the truth table of logic gates

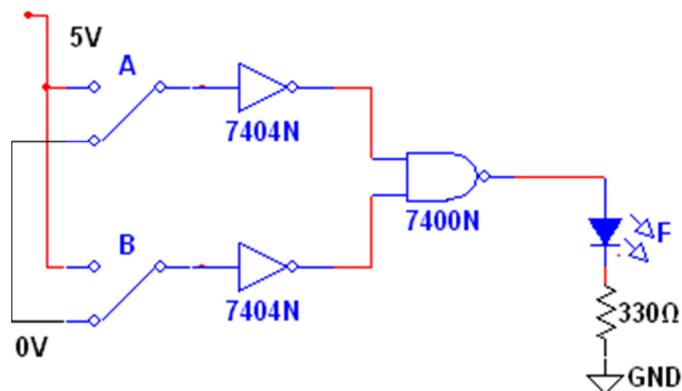
Apparatus Required: -

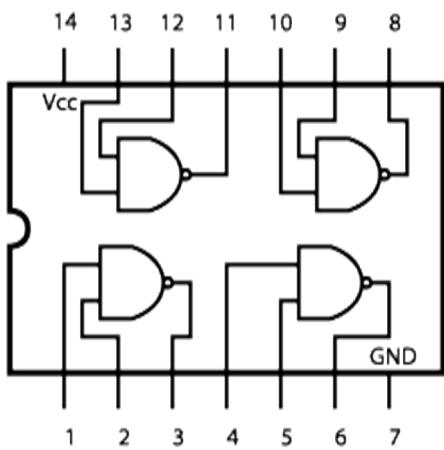
All the basic gates mention in the fig.

74LS00,74LS02,74LS04,74LS08,74LS32,74LS86

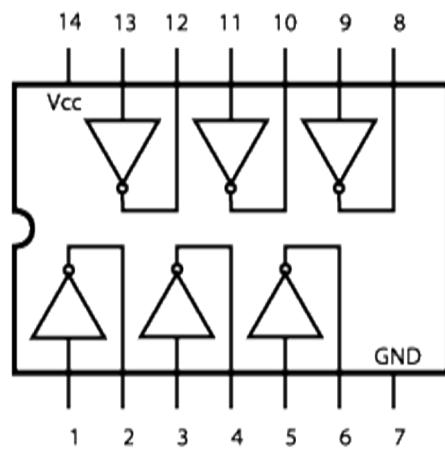
Procedure: -

1. Place the IC on IC Trainer Kit.
2. Connect V_{cc} and ground to respective pins of IC Trainer Kit.
3. Connect the inputs to the input switches provided in the IC Trainer Kit.
4. Connect the outputs to the switches of O/P LEDs,
5. Apply various combinations of inputs according to the truth table and observe condition of LEDs.
6. Disconnect output from the LEDs and note down the corresponding multimeter voltage readings for various combinations of inputs.

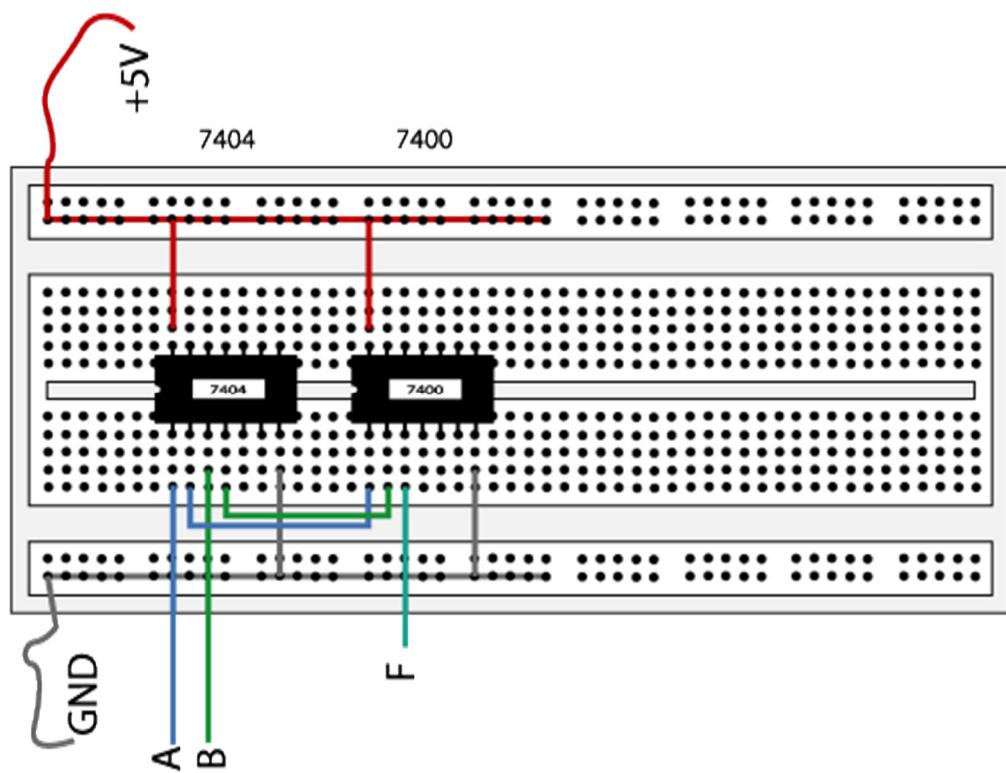




Quad 2 Input 7400



Hex 7404 Inverter



AND Gate Implementation

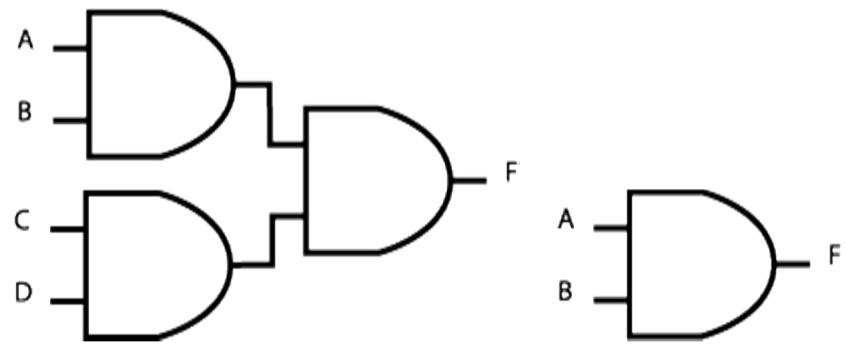


Figure 1. The 2-input AND gate

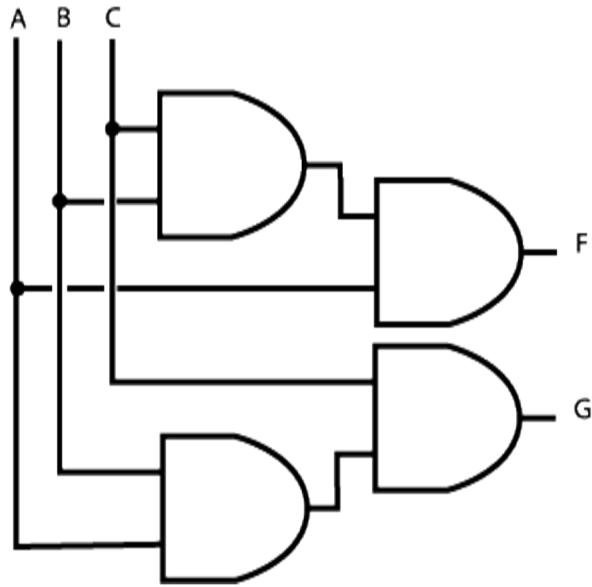
Figure 2. The 4-input AND gate, built using 2-input AND gates.

A	B	C	D	F
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	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	

.A	B	F	V (actual voltage)
0	0		
0	1		
1	0		
1	1		

Associative and Commutative Laws

A	B	C	F	G
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



Proving The Associative Law.

Laws of Boolean Algebra

- (a) To demonstrate the Distributive law, connect AND, OR gates as shown in Figure 4. Vary the inputs A, B and C to obtain all possible combinations and check that the outputs F and G are identical.

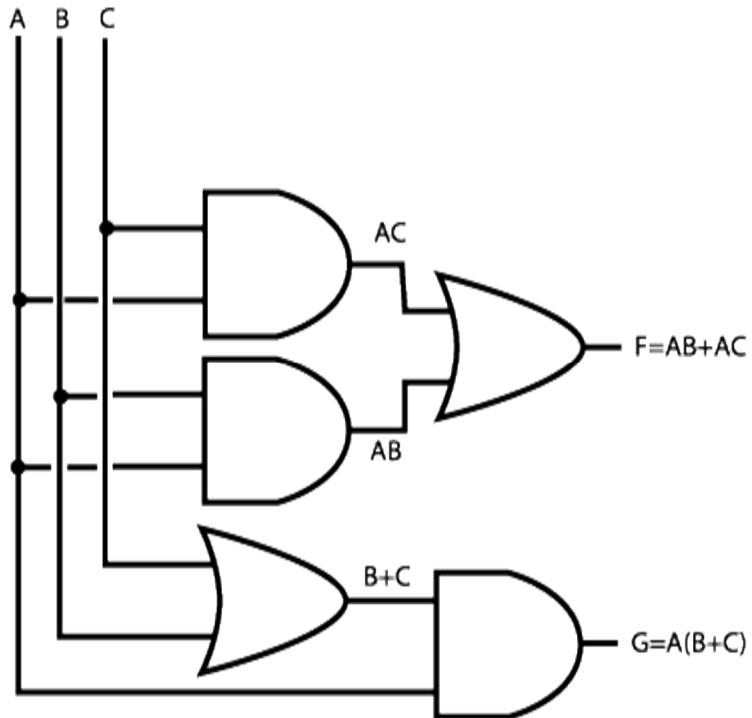


Figure 4. The Distributive Law.

Give the outputs in a truth table as shown below:

A	B	C	F	G
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Disconnected Inputs

- (a) Check the behaviour of an OR gate when one of its inputs is open (disconnected - not connected to ground). Check the behaviour when two of the inputs are open. Describe what you observed. Do the same for an AND gate.

HALF/FULL ADDER & HALF/FULL SUBTRACTOR

Aim: - To realize half/full adder and half/full subtractor.

- i. Using X-OR and basic gates
- ii. Using only nand gates.

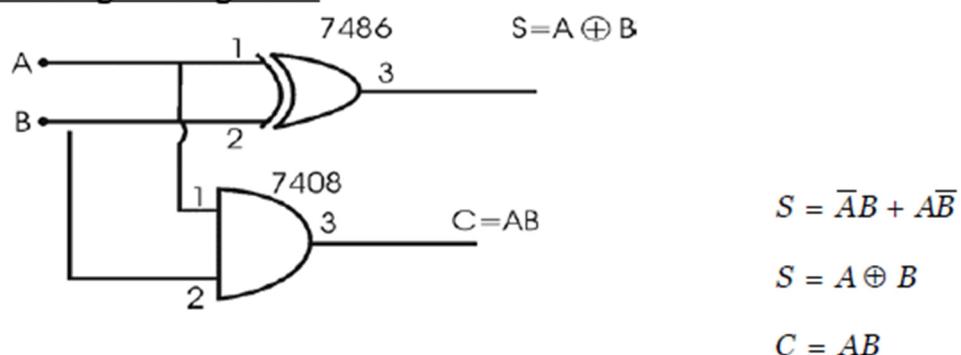
Apparatus Required: -

IC 7486, IC 7432, IC 7408, IC 7400, etc.

Procedure: -

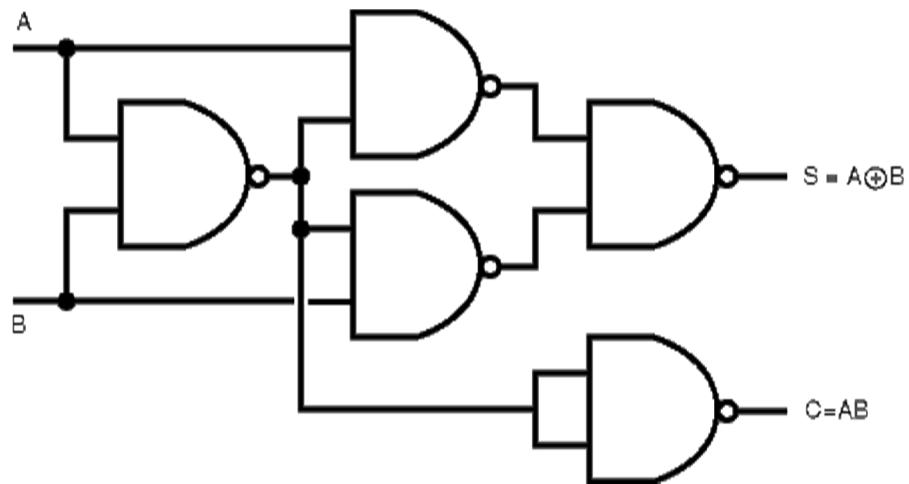
1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on V_{cc} and apply various combinations of input according to the truth table.
4. Note down the output readings for half/full adder and half/full subtractor sum/difference and the carry/borrow bit for different combinations of inputs.

Half Adder using basic gates:-

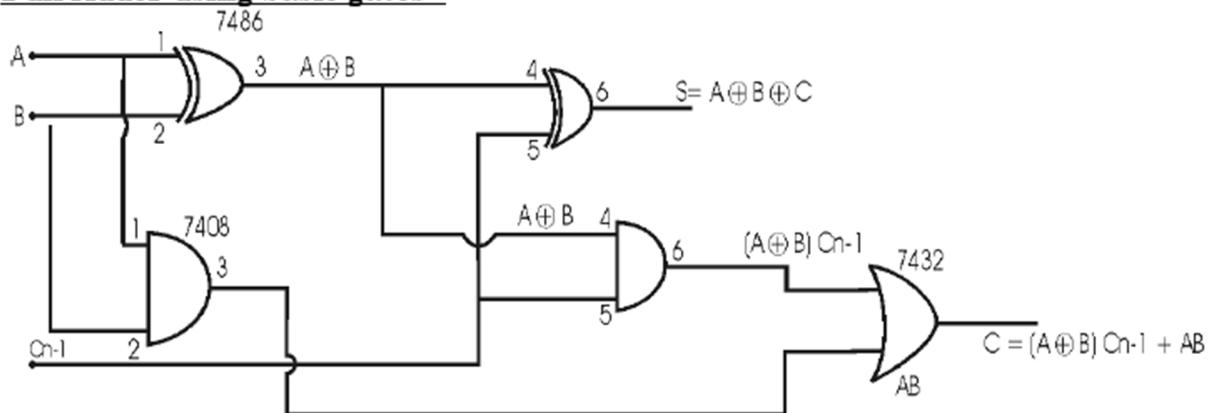


The Half Adder

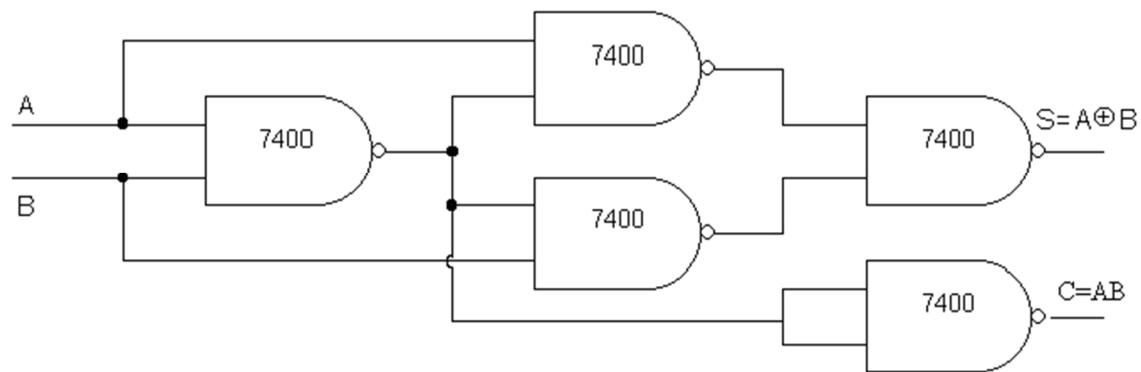
(a) Construct the following circuit as shown in Figure



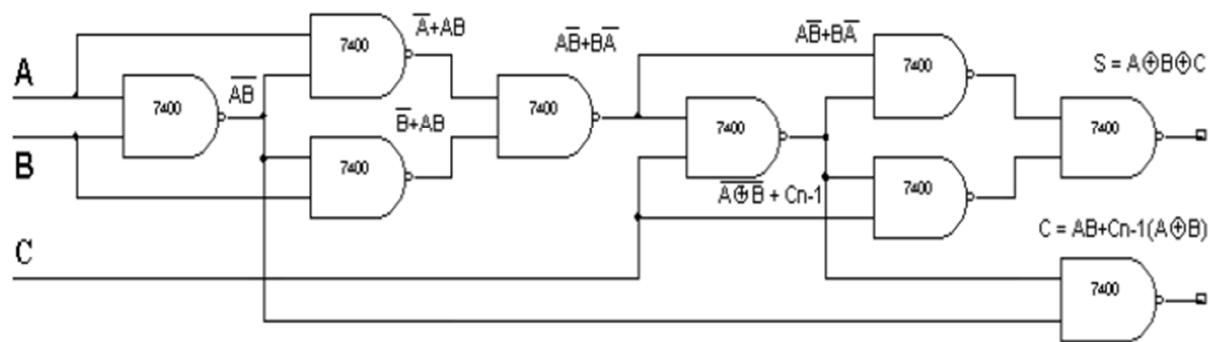
Full Adder using basic gates:-



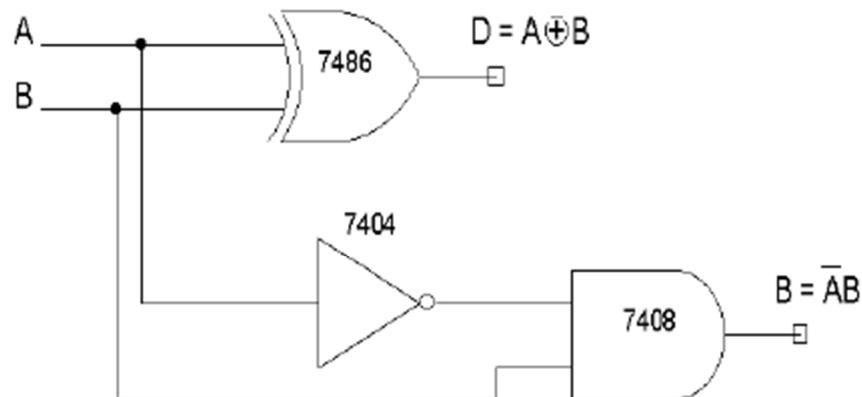
Half Adder using NAND gates only:-



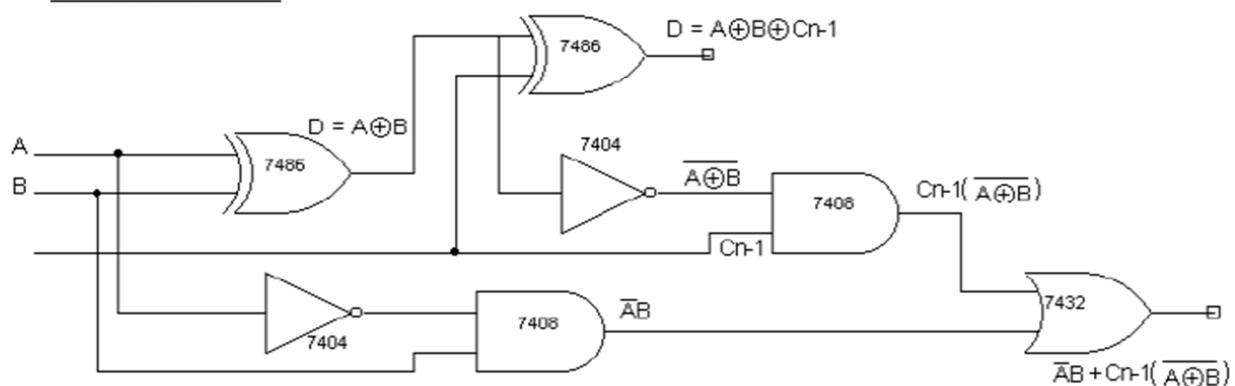
Full Adder using NAND gates only:-



Using X - OR and Basic Gates (a)Half Subtractor



Full Subtractor



Half Adder

A	B	S	C	S(V)	C(V)
0	0	0	0		
0	1	1	0		
1	0	1	0		
1	1	0	1		

Half Subtractor

A	B	D	B	D(V)	B(V)
0	0	0	0		
0	1	1	1		
1	0	1	0		
1	1	0	0		

Full Adder						
A	B	C _{n-1}	S	C	S(V)	C(V)
0	0	0	0	0		
0	0	1	1	0		
0	1	0	1	0		
0	1	1	0	1		
1	0	0	1	0		
1	0	1	0	1		
1	1	0	0	1		
1	1	1	1	1		

Full Subtractor							
A	B	C _{n-1}	D	B	D(v)	B(v)	
0	0	0	0	0	0		
0	0	1	1	1	1		
0	1	0	1	1	1		
0	1	1	0	1	1		
1	0	0	1	0	0		
1	0	1	0	0	0		
1	1	0	0	0	0		
1	1	1	1	1	1		

MUX/DEMUX USING 74153 & 74139

Aim: - To verify the truth table of multiplexer using 74153 & to verify a demultiplexer using 74139. To study the arithmetic circuits half adder half Subtractor, full adder and full Subtractor using multiplexer.

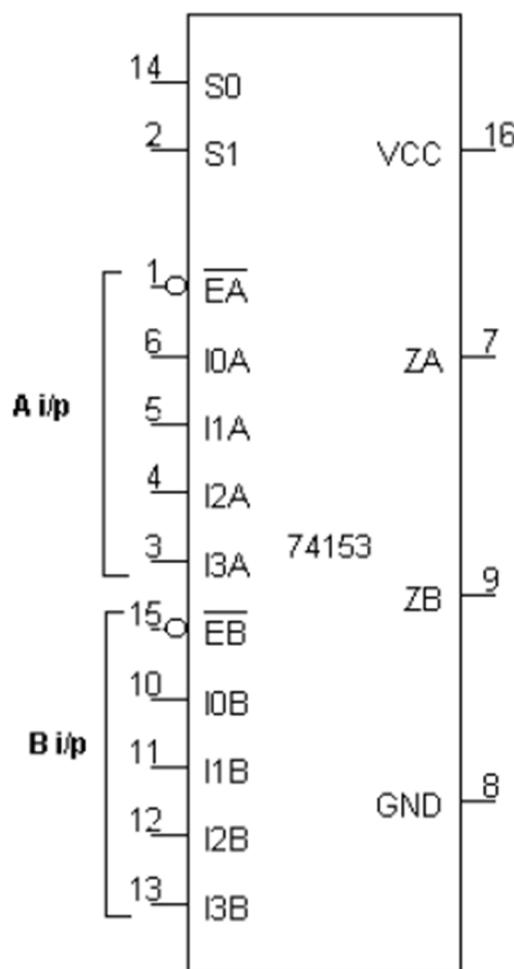
Apparatus Required: -

IC 74153, IC 74139, IC 7404, etc.

Procedure: - (IC 74153)

1. The Pin [16] is connected to + Vcc.
2. Pin [8] is connected to ground.
3. The inputs are applied either to 'A' input or 'B' input.
4. If MUX 'A' has to be initialized, E_a is made low and if MUX 'B' has to be initialized, E_b is made low.
5. Based on the selection lines one of the inputs will be selected at the output and thus the truth table is verified.
6. In case of half adder using MUX, sum and carry is obtained by applying a constant inputs at I_{0a} , I_{1a} , I_{2a} , I_{3a} and I_{0b} , I_{1b} , I_{2b} and I_{3b} and the corresponding values of select lines are changed as per table and the
7. In this case, the channels A and B are kept at constant inputs according to the table and the inputs A and B are varied. Making E_a and E_b zero and the output is taken at Z_a , and Z_b .
8. In full adder using MUX, the input is applied at C_{n-1} , A_n and B_n . According to the table corresponding outputs are taken at C_n and D_n .

MULTIPLEXERS



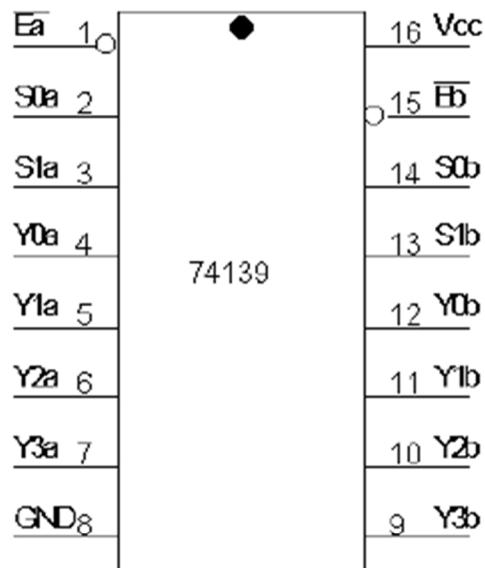
CHANNEL - A

INPUTS					SELECT LINES			O/P
E _a	I _{0a}	I _{1a}	I _{2a}	I _{3a}	S ₁	S ₂	Z _{a(v)}	
1	X	X	X	X	X	X	0	
0	0	X	X	X	0	0	0	
0	1	X	X	X	0	0	1	
0	X	0	X	X	0	1	0	
0	X	1	X	X	0	1	1	
0	X	X	0	X	1	0	0	
0	X	X	1	X	1	0	1	
0	X	X	X	0	1	1	0	
0	X	X	X	1	1	1	1	

CHANNEL - B

INPUTS					SELECT LINES			O/P
E _b	I _{0b}	I _{1b}	I _{2b}	I _{3b}	S ₁	S ₂	Z _{b(v)}	
1	X	X	X	X	X	X	0	
0	0	X	X	X	X	0	0	
0	1	X	X	X	X	0	1	
0	X	0	X	X	X	0	1	
0	X	1	X	X	X	0	1	
0	X	X	0	X	X	1	0	
0	X	X	1	X	X	1	0	
0	X	X	X	0	0	1	1	
0	X	X	X	X	1	1	1	

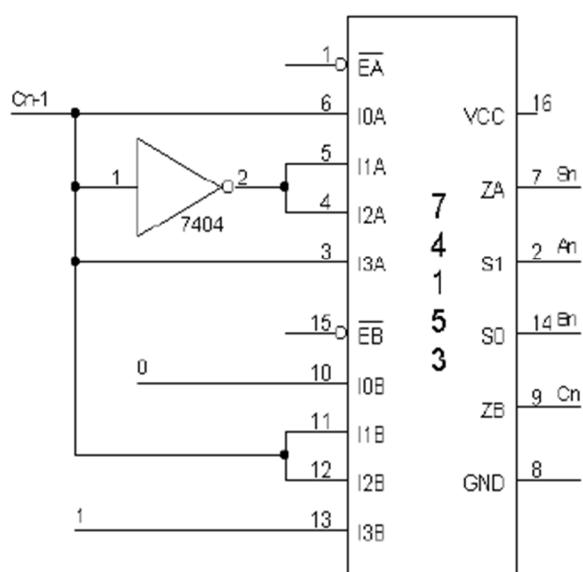
DUAL 2 TO 4 LINE DEMULTIPLEXERS



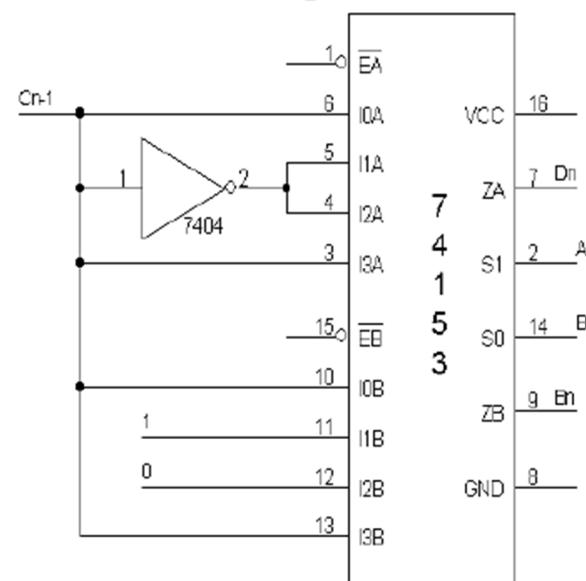
Truth Table For Demux: -

CHANNEL – A							CHANNEL – B						
Inputs			Outputs				Inputs			Outputs			
Ea	S1a	S0a	Y0a	Y1a	Y2a	Y3a	Eb	S1b	S0b	Y0b	Y1b	Y2b	Y3b
1	X	X	1	1	1	1	1	X	X	1	1	1	1
0	0	0	0	1	1	1	0	0	0	0	1	1	1
0	0	1	1	0	1	1	0	0	1	1	0	1	1
0	1	0	1	1	0	1	1	1	0	1	1	0	1
0	1	1	1	1	1	0	1	1	1	1	1	1	0

Full Adder Using 74153: -



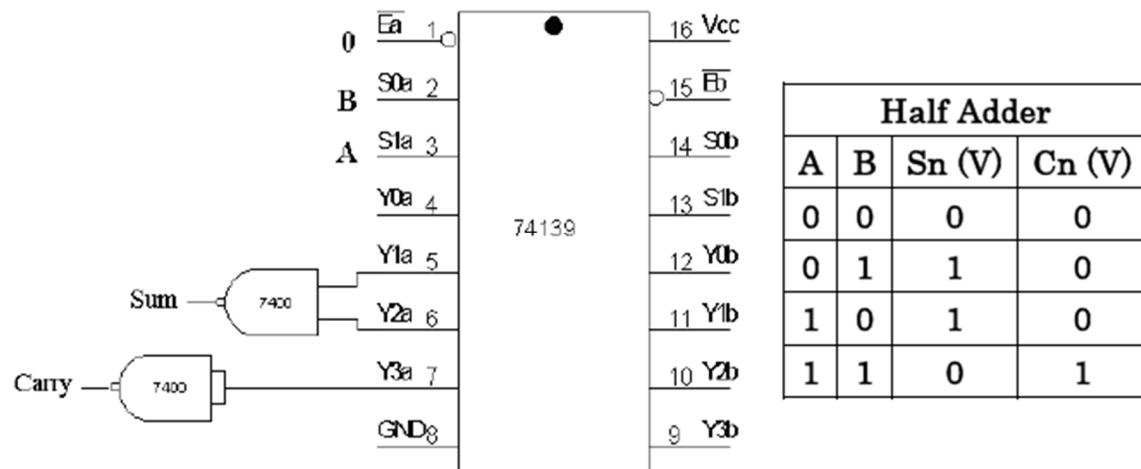
Full Subtractor Using 74153: -



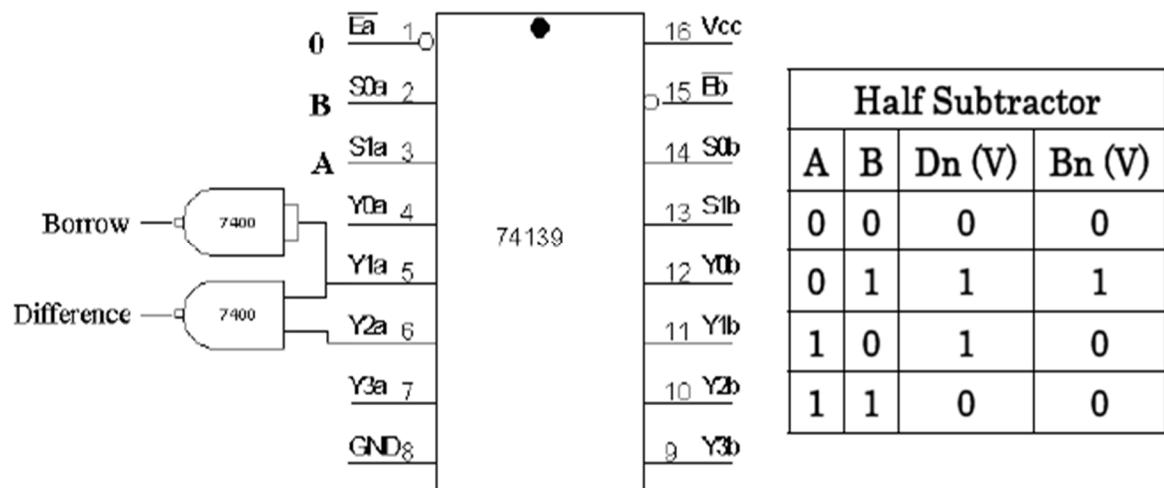
Truth Tables: - Same for both Subtractor and adder

Half adder/subtractor				Full Adder/subtractro				
An	Bn	Cn-1	Sn/Dn (V)	Cn/Bn (V)				
0	0	0						
0	1							
1	0							
1	1							

Half adder



Half subtractor:-



COMPARATORS

Aim: - To verify the truth table of one bit and two bit comparators using logic gates.

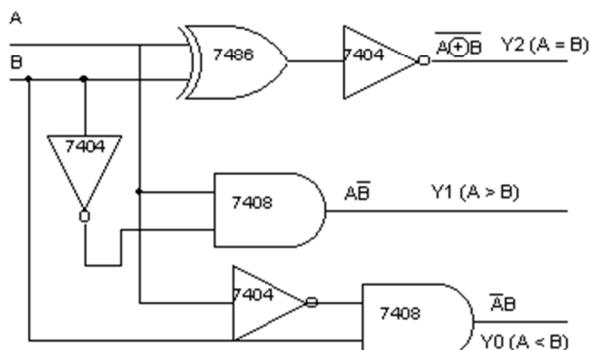
Apparatus Required: -

IC 7486, IC 7404, IC 7408, etc.

Procedure: -

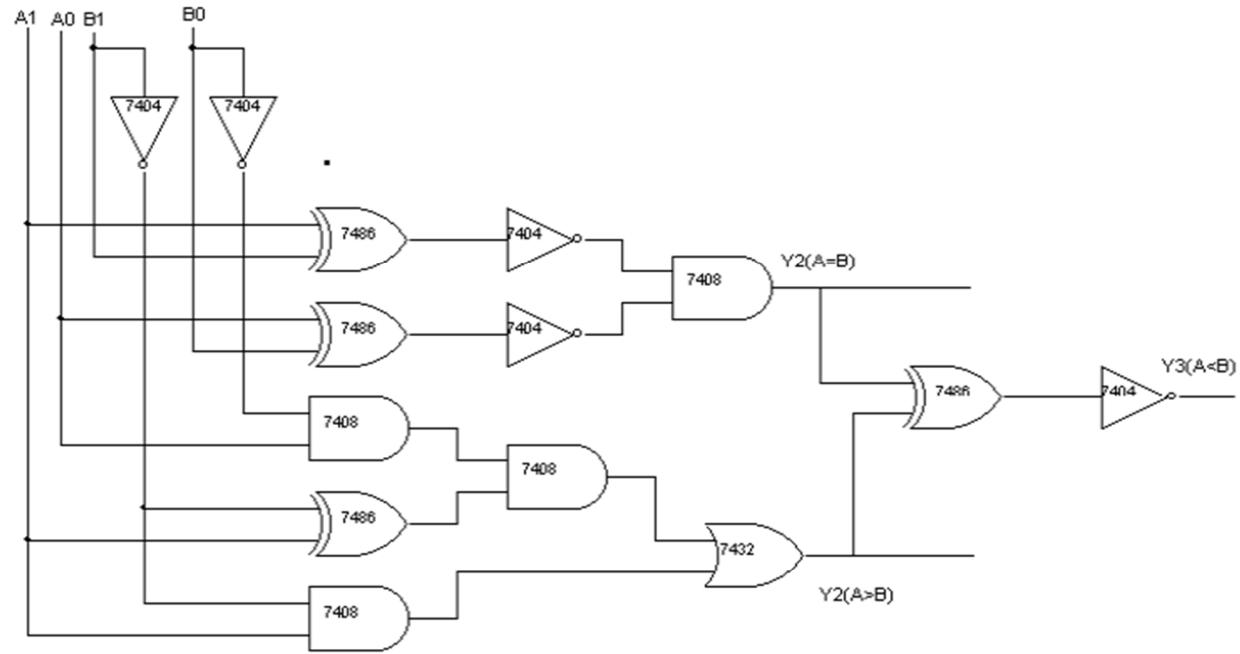
1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on Vcc.
4. Applying i/p and Check for the outputs.
5. The voltameter readings of outputs are taken and tabulated in tabular column.
6. The o/p are verified.

One Bit Comparator: -



A	B	Y_1 ($A > B$)	Y_2 ($A = B$)	Y_3 ($A < B$)
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

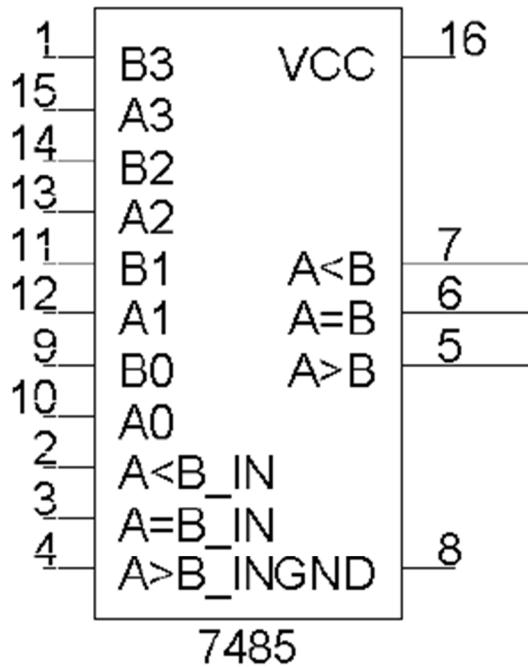
Two Bit Comparator:-



Two-Bit Comparator:-

A1	A0	B1	B0	Y1 (A > B)	Y2 (A = B)	Y3 (A < B)
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

4-bit Comparator



Tabular Column For⁴ Bit Comparator: -

$A_3 \ B_3$	$A_2 \ B_2$	$A_1 \ B_1$	$A_0 \ B_0$	$A > B$	$A = B$	$A < B$	$A > B$	$A = B$	$A < B$
$A_3 > B_3$	X	X	X	X	X	X			
$A_3 < B_3$	X	X	X	X	X	X			
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X			
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X			
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X			
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	X	X	X			
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X	X	X			
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X	X	X			
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	0			
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	0			
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	1			

BCD to Excess 3 AND Excess 3 to BCD

Aim: - To verify BCD to excess –3 code conversion using NAND gates. To study and verify the truth table of excess-3 to BCD code converter

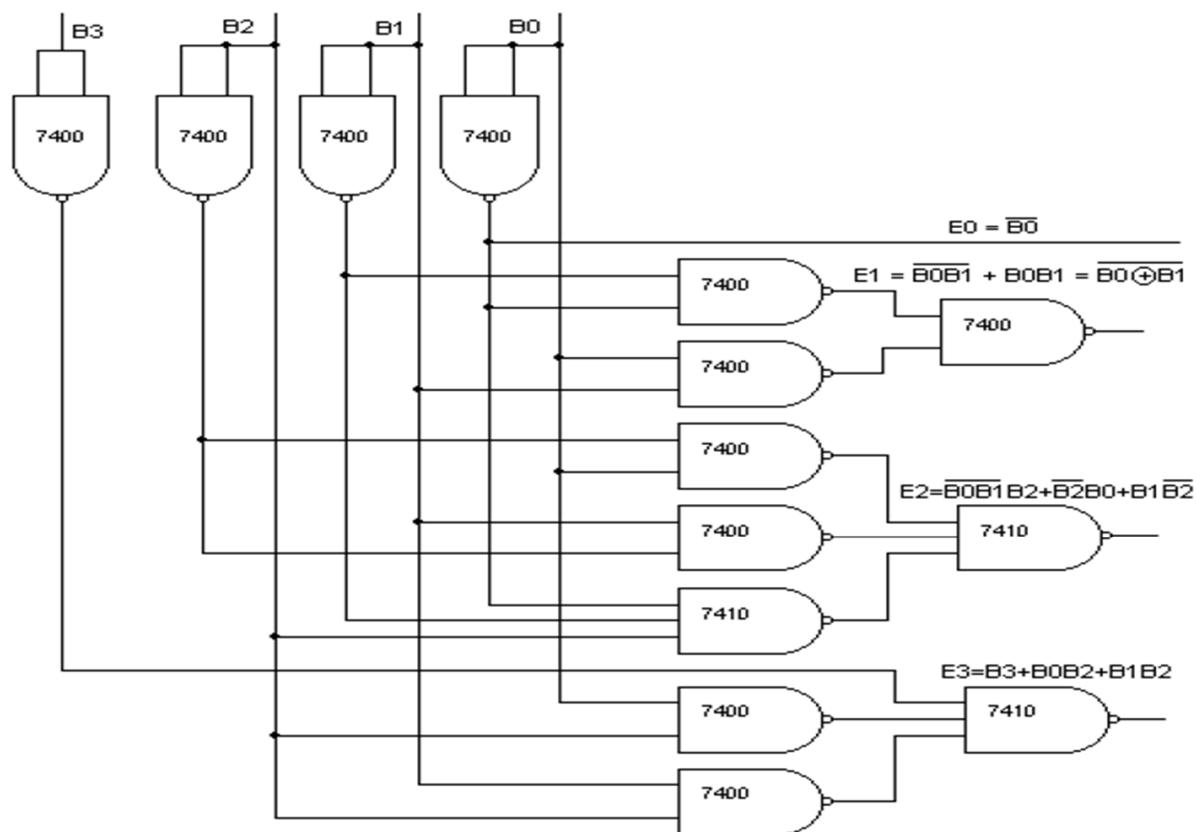
Apparatus Required: -

IC 7400, IC 7404, etc.

Procedure: - (BCD Excess 3 and Vice Versa)

1. Make the connections as shown in the fig.
2. Pin [14] of all IC'S are connected to +5V and pin [7] to the ground.
3. The inputs are applied at E3, E2, E1, and E0 and the corresponding outputs at B3, B2, B1, and B0 are taken for excess – 3 to BCD.
4. B3, B2, B1, and B0 are the inputs and the corresponding outputs are E3, E2, E1 and E0 for BCD to excess – 3.
5. Repeat the same procedure for other combinations of inputs.
6. Truth table is written.

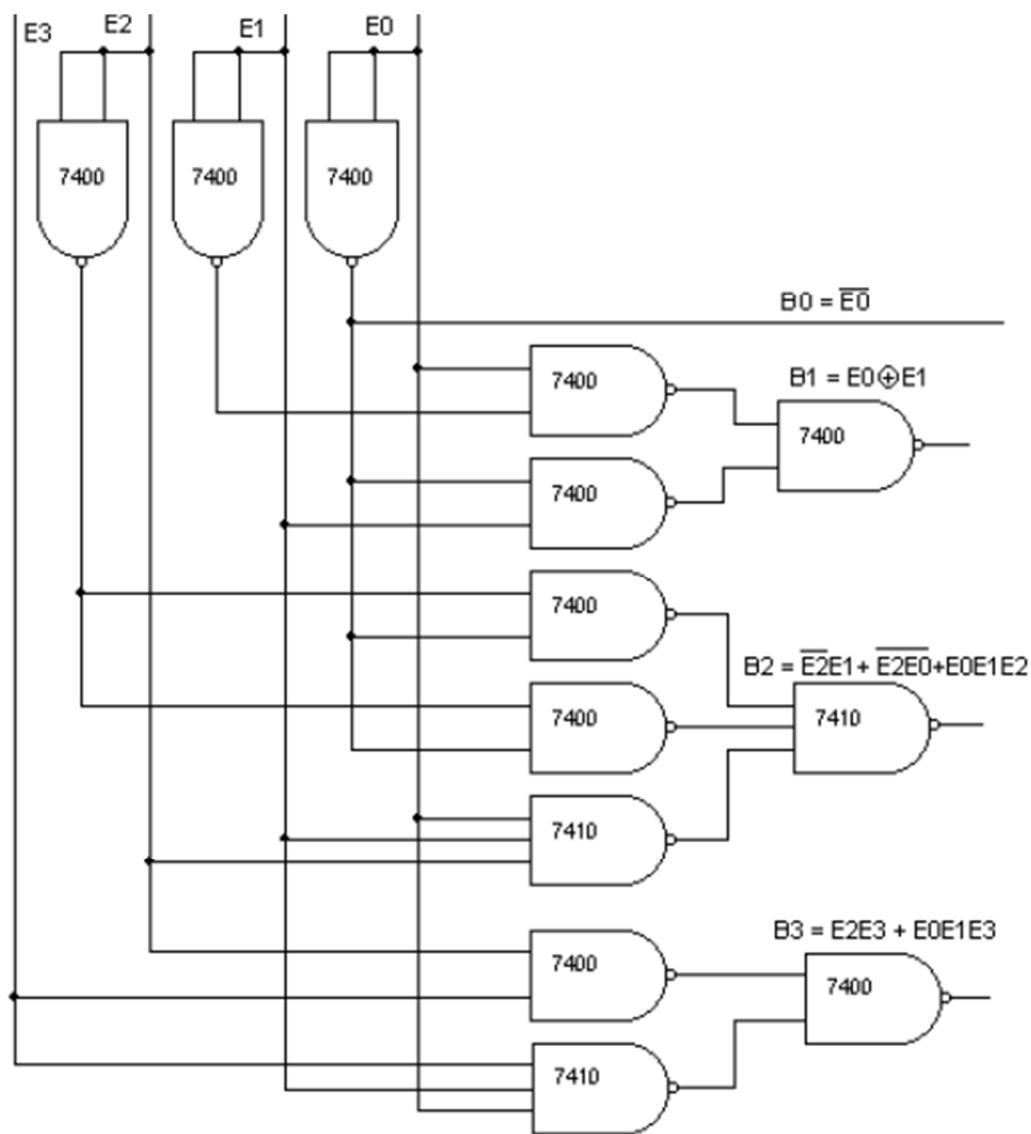
BCD To Excess-3



Truth Table For Code Conversion:-

Inputs				Outputs			
B3	B2	B1	B0	E3 (v)	E2 (v)	E1 (v)	E0 (v)
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

Excess-3 To BCD :-



Truth Table For Code Conversion:-

Inputs				Outputs			
E3	E2	E1	E0	B3 (v)	B2 (v)	B1 (v)	B0(v)
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1

FLIP-FLOP

Aim:- Truth table verification of Flip-Flops:

RS FLIP-FLOP

- (i) JK Master Slave
- (ii) D- Type
- (iii) T- Type.

Apparatus Required:-

IC 7410, IC 7400, etc.

Procedure:-

1. Connections are made as per circuit diagram.
2. The truth table is verified for various combinations of inputs.

Truth Table:- (Master Slave JK Flip-Flop)

Preset	Clear	J	K	Clock	Q_{n+1}	\bar{Q}_{n+1}	
0	1	X	X	X	1	0	Set
1	0	X	X	X	0	1	Reset
1	1	0	0	⊓	Q_n	\bar{Q}_n	No Change
1	1	0	1	⊓	0	1	Reset
1	1	1	0	⊓	1	0	Set
1	1	1	1	⊓	\bar{Q}_n	Q_n	Toggle

D Flip-Flop:-

Preset	Clear	D	Clock	Q_{n+1}	$\overline{Q_{n+1}}$
1	1	0	⊓	0	1
1	1	1	⊓	1	0

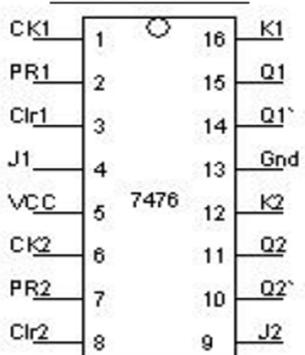
T Flip-Flop:-

Preset	Clear	T	Clock	Q_{n+1}	$\overline{Q_{n+1}}$
1	1	0	⊓	Q_n	$\overline{Q_n}$
1	1	1	⊓	$\overline{Q_n}$	Q_n

Exercise:-

- Write the timing diagrams for all the above Flip-Flops

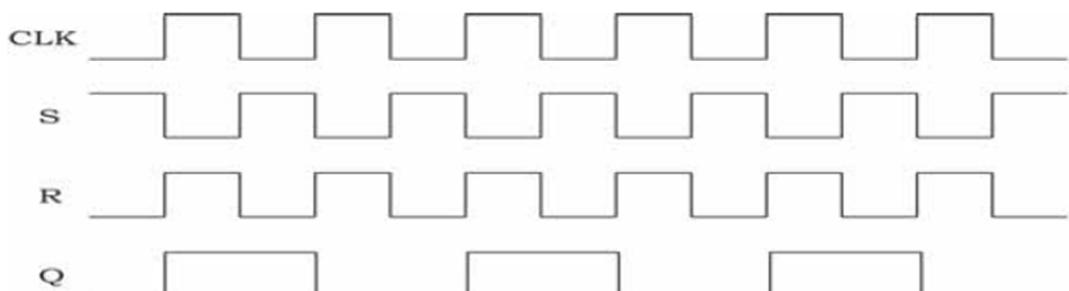
Pin Details:-



Truth Table:-

Clock	QC	QB	QA
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Timing Diagram:-



Preset	Clear	J	K	Clock	Q_{n+1}	\bar{Q}_{n+1}	
0	1	X	X	X	1	0	Set
1	0	X	X	X	0	1	Reset
1	1	0	0	—	Q_n	\bar{Q}_n	No Change
1	1	0	1	—	0	1	Reset
1	1	1	0	—	1	0	Set
1	1	1	1	—	\bar{Q}_n	Q_n	Toggle

(a) Connect the two NOR gates as shown in Figure 1.

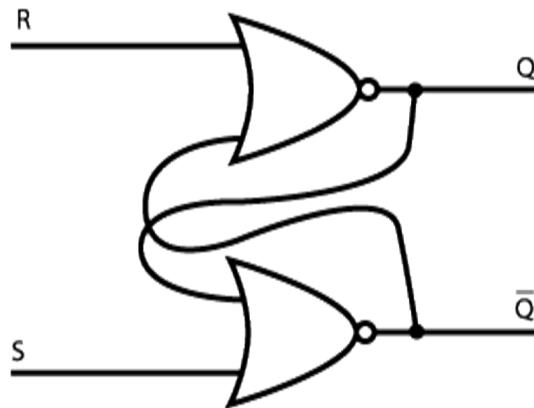


Figure 1. An RS Flip-Flop created using NOR gates.

(b) Vary the inputs R and S (i.e. 0 and +5V) to obtain all the possible combinations for Q and /Q.

(c) In your opinion why is there a Q and /Q output? In your opinion how does this circuit work?

Section 2. The Synchronous Flip-Flop

(a) Synchronous means that this flip-flop is concerned with time! Digital circuits can have a concept of time using a clock signal. The clock signal simply goes from low-to-high and high-to-low in a short period of time.

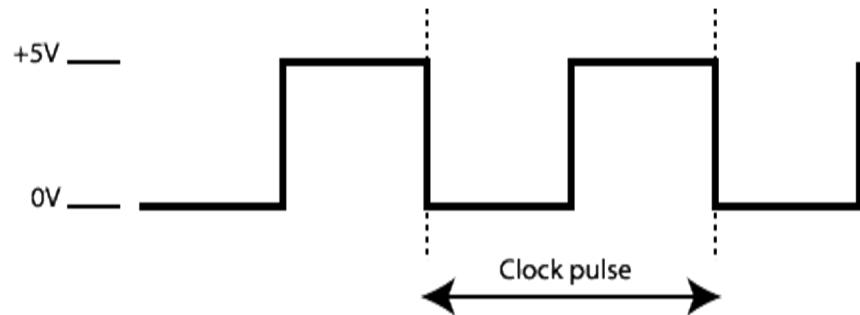


Figure 2. A typical clock signal.

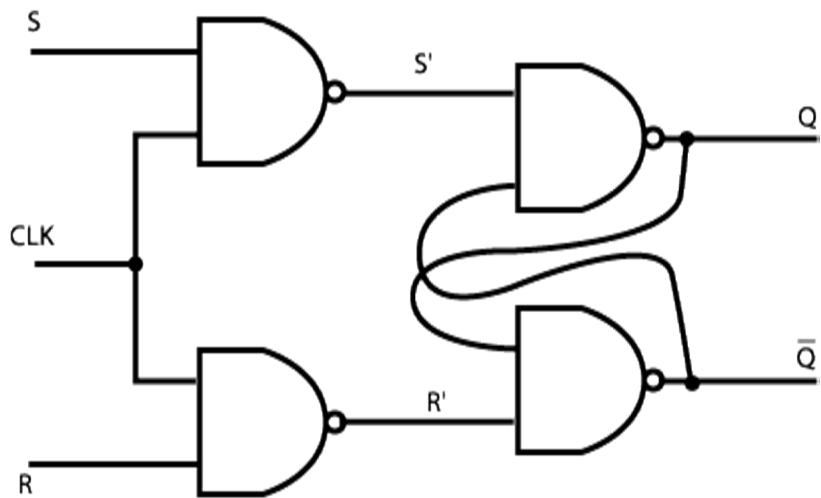


Figure 3. The Synchronous Flip Flop.

Implement the circuit in Figure 3. You can simulate a clock signal by moving the clock line from low to high and back again to low.

(b) Vary inputs R and S and apply the clock pulse. Write the output states into a table as below:

Q_n/Q_{n+1}	R	S	Q_{n+1}/Q_{n+1}
0	1	0	0
1	0	0	0
0	1	0	1
1	0	0	1
0	1	1	0
1	0	1	0
0	1	1	1
1	0	1	1

(c) Convert the circuit into a D-type flip flop (as shown in Figure 4.)

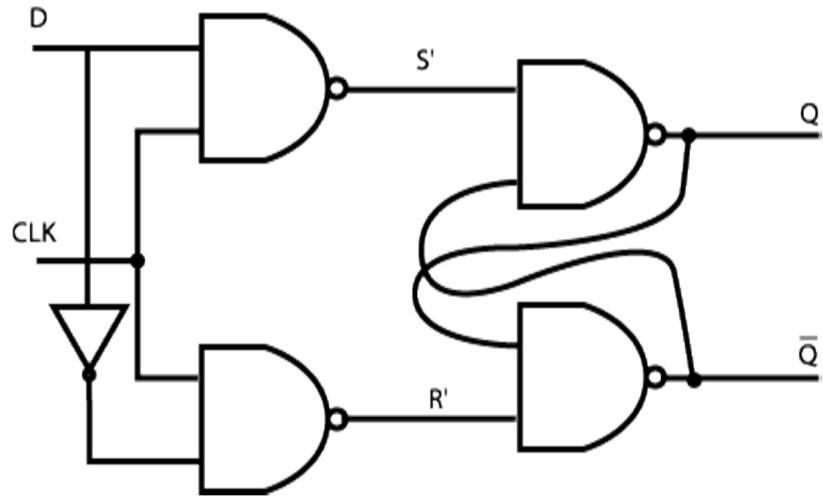


Figure 4. The D-type flip-flop

Draw up the truth table for a D-type flip flop. In your opinion how does it work? what could this circuit be useful for?

COUNTERS

Aim:- Realization of 3-bit counters as a sequential circuit and Mod-N counter design (7476, 7490, 74192, 74193).

Apparatus Required: -

IC 7408, IC 7476, IC 7490, IC 74192, IC 74193, IC 7400, IC 7416, IC 7432

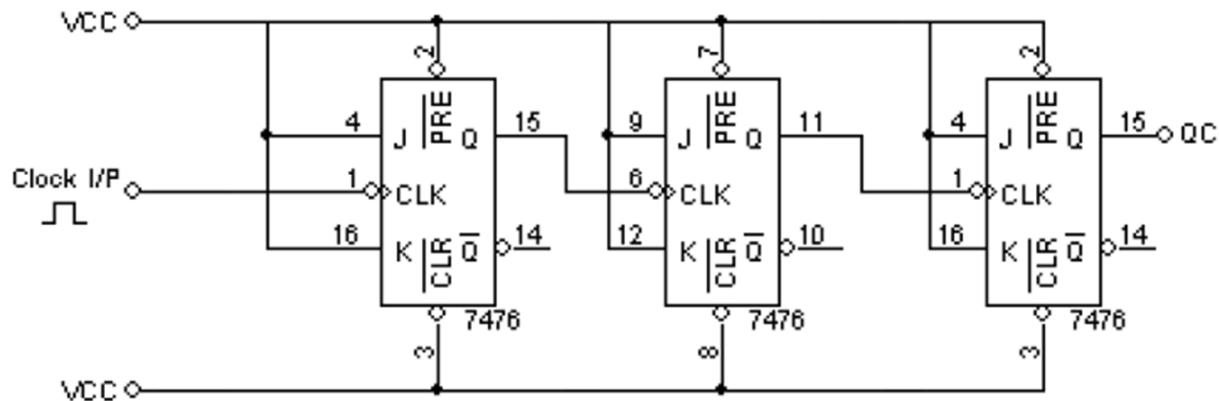
Procedure: -

1. Connections are made as per circuit diagram.
2. Clock pulses are applied one by one at the clock I/P and the O/P is observed at QA, QB & QC for IC 7476.
3. Truth table is verified.

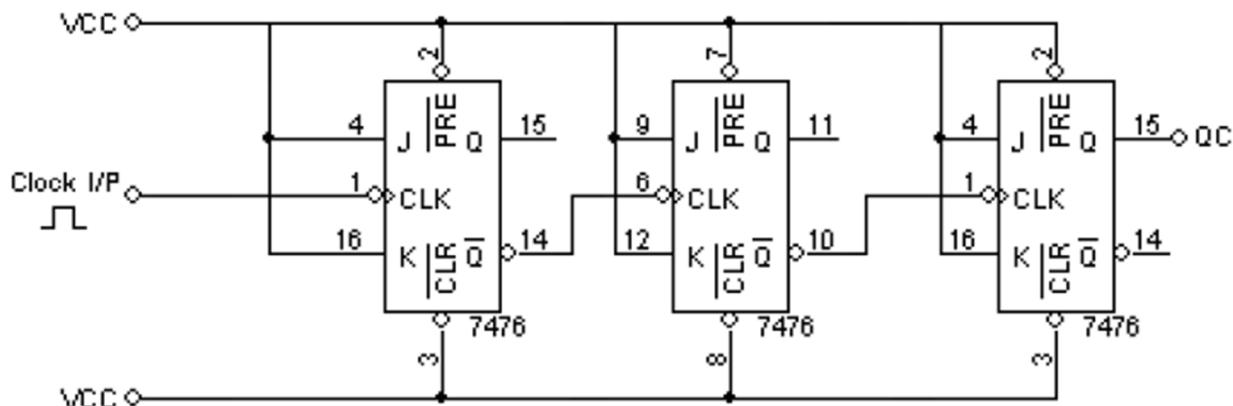
Procedure (IC 74192, IC 74193):-

1. Connections are made as per the circuit diagram except the connection from output of NAND gate to the load input.
2. The data $(0011) = 3$ is made available at the data i/p's A, B, C & D respectively.
3. The load pin made low so that the data 0011 appears at QD, QC, QB & QA respectively.
4. Now connect the output of the NAND gate to the load input.
5. Clock pulses are applied to "count up" pin and the truth table is verified.
6. Now apply $(1100) = 12$ for 12 to 5 counter and remaining is same as for 3 to 8 counter.
7. The pin diagram of IC 74192 is same as that of 74193. 74192 can be configured to count between 0 and 9 in either direction. The starting value can be any number between 0 and 9.

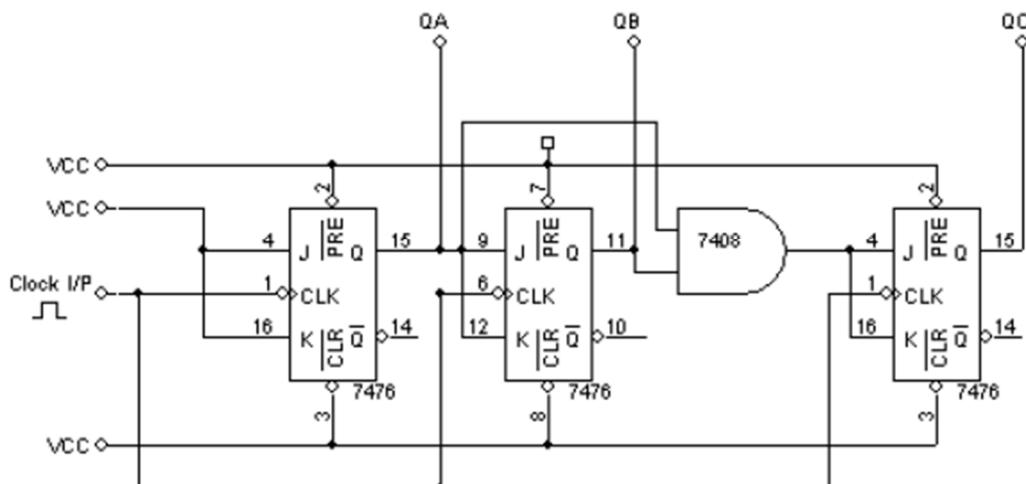
Circuit Diagram: - 3-Bit Asynchronous Up Counter



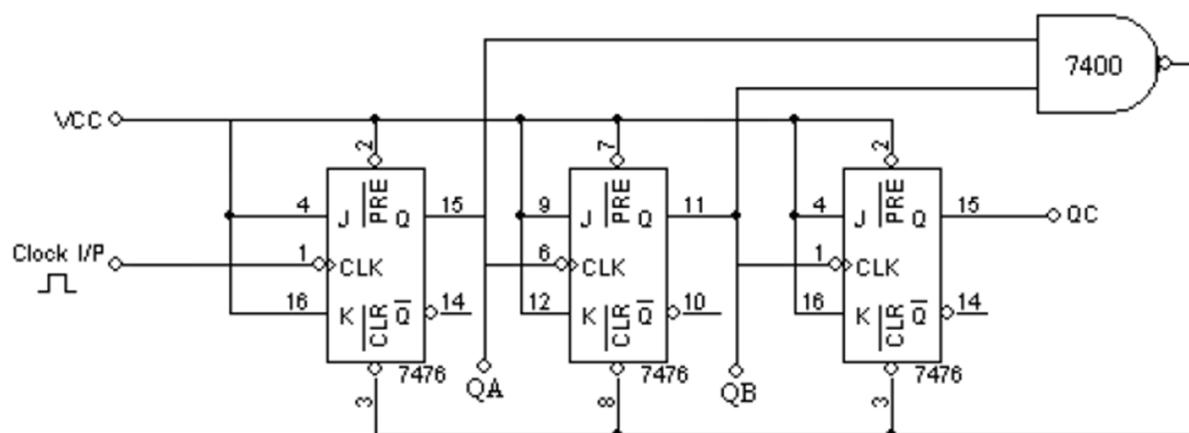
Circuit Diagram: - 3-Bit Asynchronous Down Counter



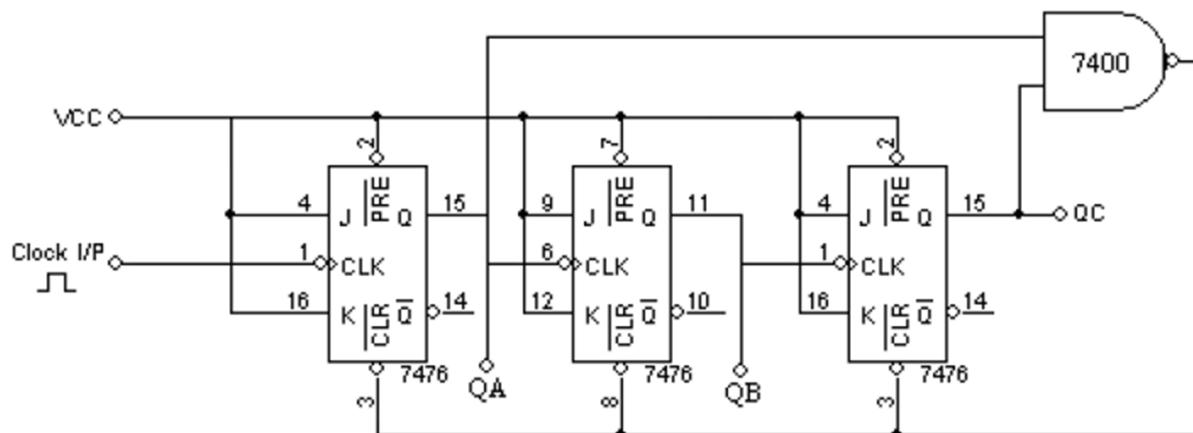
3-bit Synchronous Counter:-



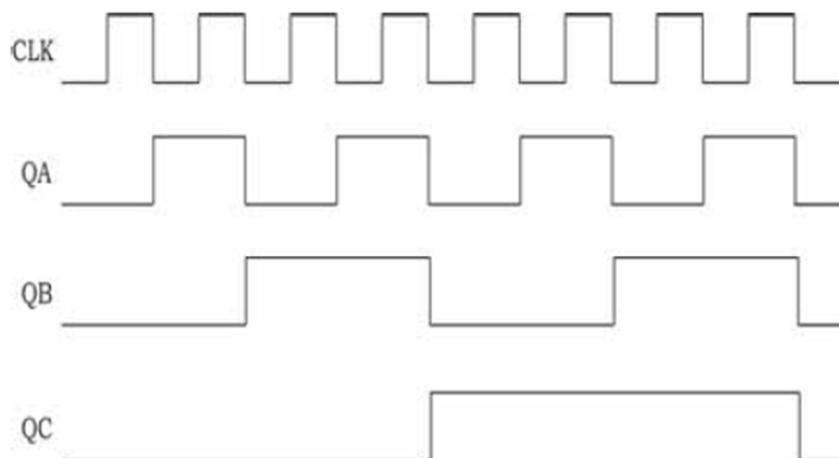
Mod 3 Asynchronous Counter:-



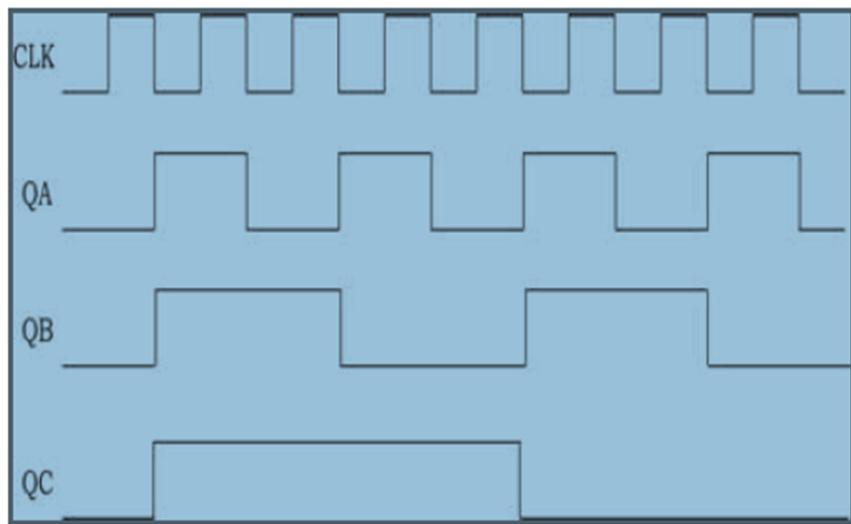
Mod 5 Asynchronous Counter:-



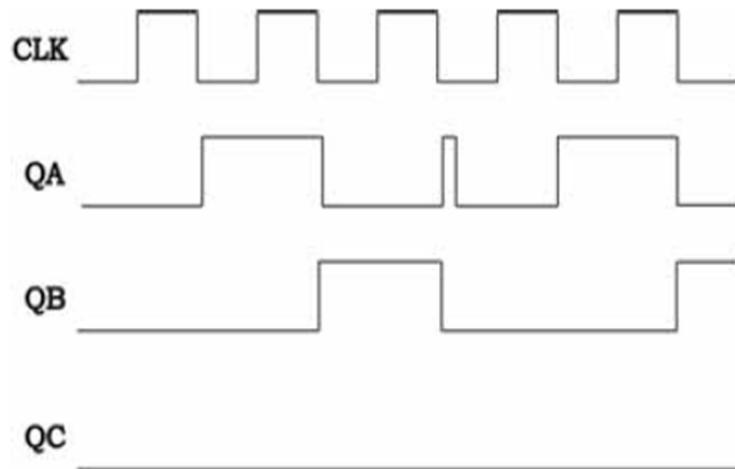
3-bit Asynchronous up counter			
Clock	QC	QB	QA
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0



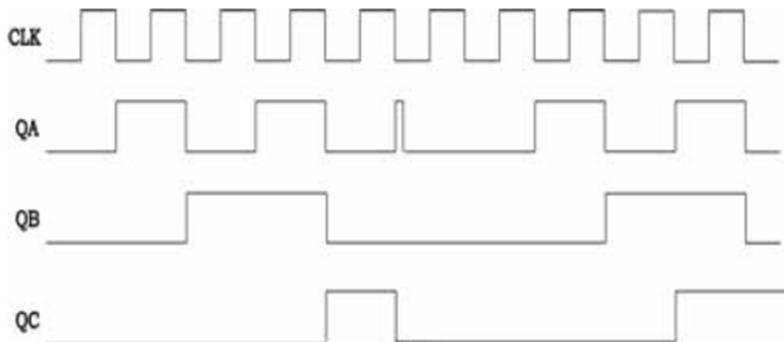
3-bit Asynchronous down counter			
Clock	QC	QB	QA
0	1	1	1
1	1	1	0
2	1	0	1
3	1	0	0
4	0	1	1
5	0	1	0
6	0	0	1
7	0	0	0
8	1	1	1
9	1	1	0



Mod 3 Asynchronous counter			
Clock	QC	QB	QA
0	0	0	0
1	0	0	1
2	0	1	0
3	0	0	0
4	0	0	1
5	0	1	0

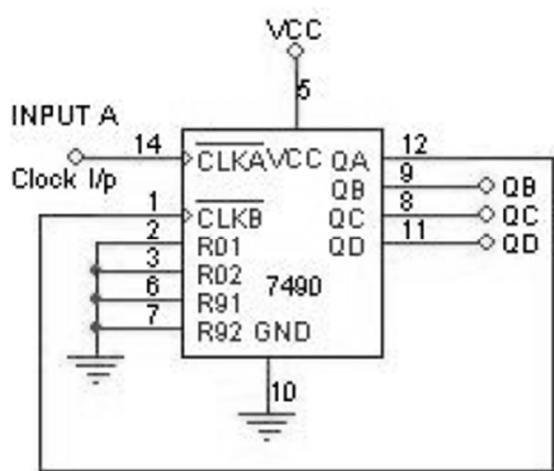


Mod 5 Asynchronous counter			
Clock	QC	QB	QA
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	0	0	0



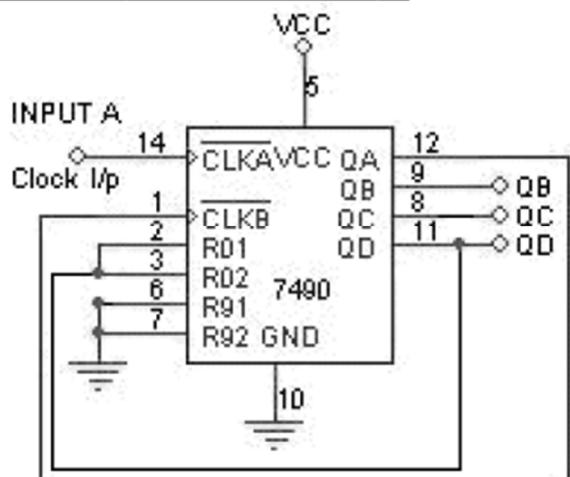
Mod 3 Asynchronous Counter:-

IC 7490 (Decade Counter):-



Clock	QD	QC	QB	QA
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

IC 7490 (MOD-8 Counter):-



Clock	QD	QC	QB	QA
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	0	0	0	0
9	0	0	0	1

SHIFT REGISTERS

Aim:-

Realization of 3-bit counters as a sequential circuit and Mod-N counter design (7476, 7490, 74192, 74193).

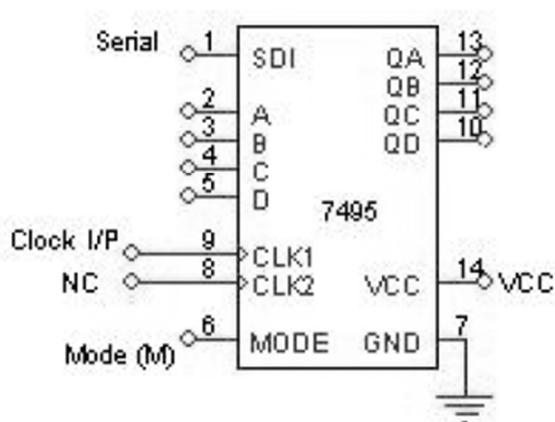
Apparatus Required: -

IC 7495, etc.

Serial In Parallel Out:-

1. Connections are made as per circuit diagram.
2. Apply the data at serial i/p
3. Apply one clock pulse at clock 1 (Right Shift) observe this data at QA.
4. Apply the next data at serial i/p.
5. Apply one clock pulse at clock 2, observe that the data on QA will shift to QB and the new data applied will appear at QA.
6. Repeat steps 2 and 3 till all the 4 bits data are entered one by one into the shift register.

SIFO (Right Shift):-

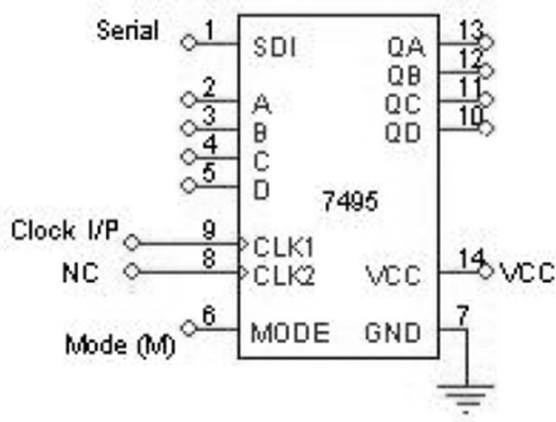


Clock	Serial i/p	QA	QB	QC	QD
1	0	0	X	X	X
2	1	1	0	X	X
3	1	1	1	0	X
4	1	1	1	1	0

Serial In Serial Out:-

1. Connections are made as per circuit diagram.
2. Load the shift register with 4 bits of data one by one serially.
3. At the end of 4th clock pulse the first data 'd0' appears at QD.
4. Apply another clock pulse; the second data 'd1' appears at QD.
5. Apply another clock pulse; the third data appears at QD.
6. Application of next clock pulse will enable the 4th data 'd3' to appear at QD. Thus the data applied serially at the input comes out serially at QD

SISO:-

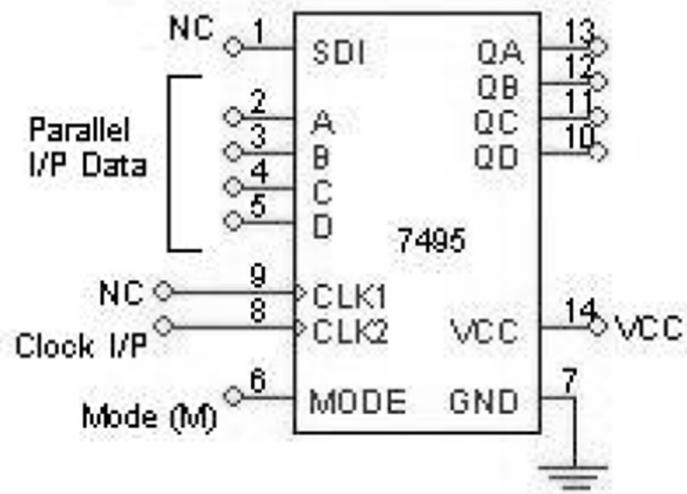


Clock	Serial i/p	QA	QB	QC	QD
1	d ₀ =0	0	X	X	X
2	d ₁ =1	1	0	X	X
3	d ₂ =1	1	1	0	X
4	d ₃ =1	1	1	1	0=d ₀
5	X	X	1	1	1=d ₁
6	X	X	X	1	1=d ₂
7	X	X	X	X	1=d ₃

Parallel In Parallel Out:-

1. Connections are made as per circuit diagram.
2. Apply the 4 bit data at A, B, C and D.
3. Apply one clock pulse at Clock 2 (Note: Mode control M=1).
4. The 4 bit data at A, B, C and D appears at QA, QB, QC and QD respectively.

PIPO:-

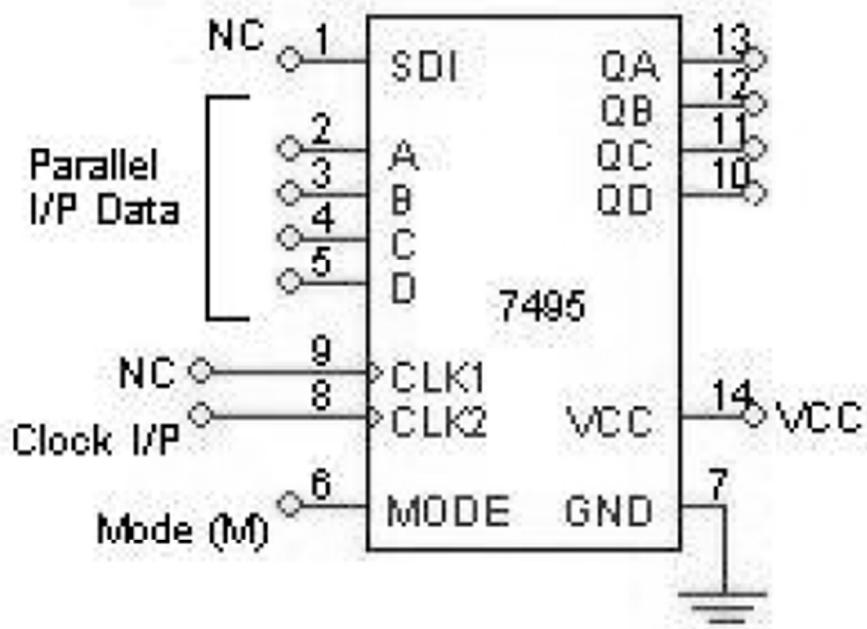


Clock	Parallel i/p				Parallel o/p			
	A	B	C	D	QA	QB	QD	
1	1	0	1	1	1	0	1	1

Parallel In Serial Out:-

1. Connections are made as per circuit diagram.
2. Apply the desired 4 bit data at A, B, C and D.
3. Keeping the mode control M=1 apply one clock pulse. The data applied at A, B, C and D will appear at QA, QB, QC and QD respectively.
4. Now mode control M=0. Apply clock pulses one by one and observe the data coming out serially at QD.

PISO:-



Mode	Clock	Parallel i/p				Parallel o/p			
		A	B	C	D	QA	QB	QC	QD
1	1	1	0	1	1	1	0	1	1
0	2	X	X	X	X	X	1	0	1
0	3	X	X	X	X	X	X	1	0
0	4	X	X	X	X	X	X	X	1

Left Shift:-

1. Connections are made as per circuit diagram.
2. Apply the first data at D and apply one clock pulse. This data appears at QD.
3. Now the second data is made available at D and one clock pulse applied. The data appears at QC to QD and the new data appears at QD.
4. Step 3 is repeated until all the 4 bits are entered one by one.
5. At the end 4th clock pulse the 4 bits are available at QA, QB, QC and QD.