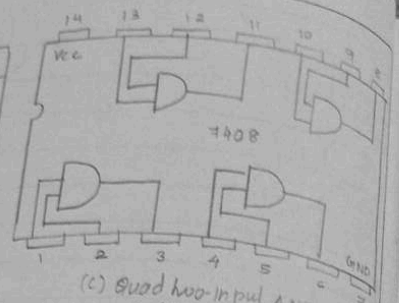
(a) Logic symbol

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

(b) truth table

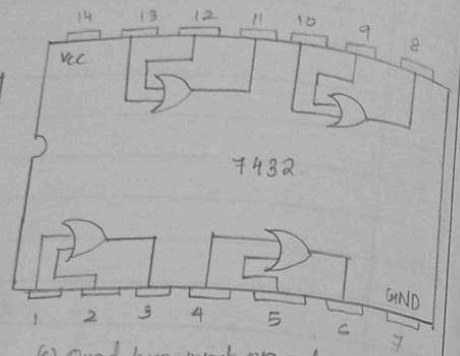


(c) Quad two-input AND gate IC 7408

(a) Logic symbol

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

(b) truth table

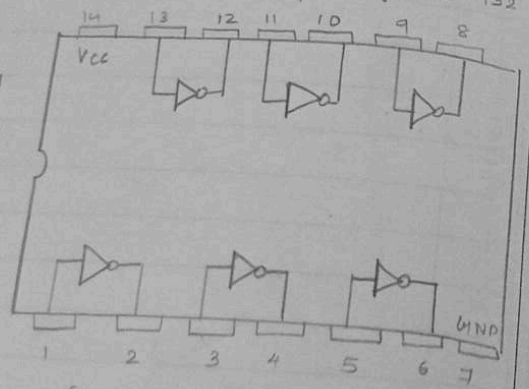


(c) Quad two-input OR gate IC 7432

(a) Logic symbol

A	Y
0	1
1	0

(b) truth table



(c) Hex NOT gate IC 7404

Symbol, truth table and pin out of IC 7404

/ /

1

Familiarisation of digital ICs and trainer kit
Experiment 1

Aim

To study a few basic logic gate IC packages and to verify truth tables of the logic gates, and familiarize with the digital IC trainer kit.

Components and equipment required

ICs 7408, 7432, 7404, 7400, 7402, 7486 and digital IC trainer kit

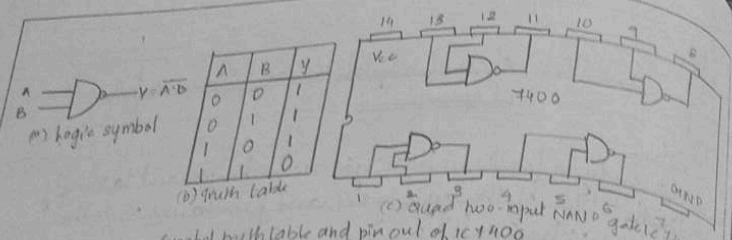
Theory

Digital IC trainer kit

The digital IC trainer kit is an equipment used to test and set up digital circuits for doing experiments. Integrated circuits can be fitted in sockets or breadboards provided in it. There are built-in voltage sources and clock signals in it. The frequency of clock can be selected by turning the knob to different positions. In order to feed one pulses manually, a debounce switch is also provided. A set of slide-switches are provided to select low and high states for digital inputs. LED's are provided to visualize the digital outputs.

Logic gates

In digital electronics, a gate is a logic circuit with one output and one more inputs. Logic gates are available as integrated circuits. Some of the logic gates are described

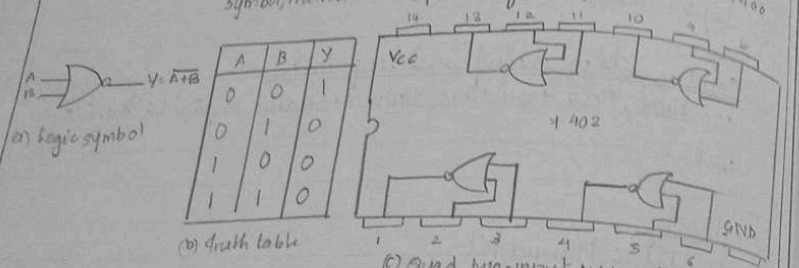


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

(b) truth table

(c) Quad two-input NAND gate IC 7400

Symbol, truth table and pin out of IC 7400

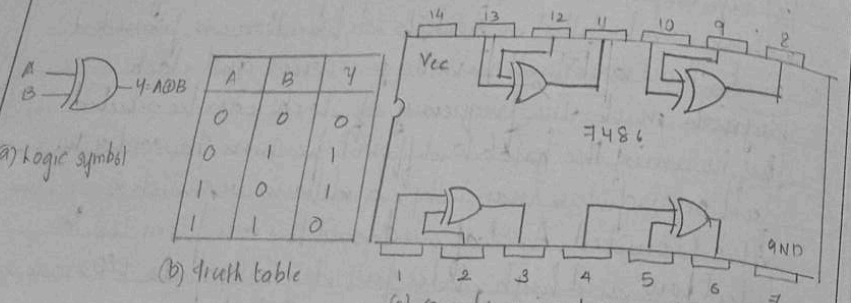


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

(b) truth table

(c) Quad two-input NOR gate IC 7402

Symbol, truth table and pin out IC 7402



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

(b) truth table

(c) Quad two-input XOR gate IC 7486

Below:

AND gate: The AND gate performs logical multiplication. The AND gate output is now high state only when all the inputs are high state. 7408 is a Dual in-line Package (DIP) IC in the TTL family and contains four two-input AND gates. For this reason it is called dual two-input AND gate IC.

OR gate: The OR gate performs logical addition. Its output is high when any of the inputs is in logic high state. 7432 is a quad two-input OR gate IC. The OR operation of more than two inputs can be realized using 7432 IC.

NOT gate: It performs a basic logic function called complementation. The purpose of the inverter is to change one logic level to opposite level. 7404 is a hex inverter gate IC.

NAND gate: A NOT gate following an AND gate is called NAND. Its output is in low if all the inputs are in high state. 7400 is a quad two-input NAND gate IC.

NOR gate: A NOT gate following an OR gate is called NOR gate. Its output is in low state when any of its inputs is in high state. 7402 is a quad two-input NOR gate IC.

XOR gate: Its output is high if and only if one of the inputs is in high state. 7486 is a quad two-input XOR gate IC.

Procedure

1. Test all the components and IC packages using a digital IC tester. Also assure whether all the connections wires are in good condition by testing for the continuity using a multimeter or a digital IC trainer kit.

2. Verify the pin out of IC before feeding the inputs

3. Verify that the ICs have been perfectly plugged into the breadboard or sockets before feeding inputs

4. Set up the circuits and observe the outputs. Enter the input and output state in truth table corresponding to the input combinations

Result:

Familiarized with digital trainer kit and verified truth table of NAND, NOR etc.

18/8/22

Procedure:

1. Test all the components and IC packages using a digital IC tester. Also assure whether all the connections wires are in good condition by testing for the continuity using a multimeter or a digital IC trainer kit.

2. Verify the pin out of IC before feeding the input.

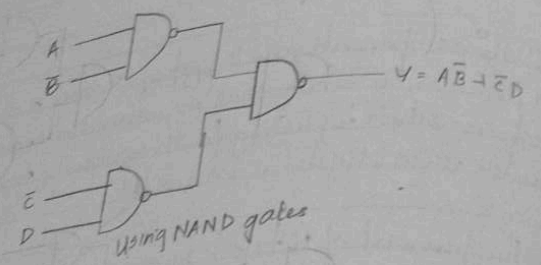
3. Verify that the ICs have been perfectly plugged into the breadboard or sockets before feeding inputs.

4. Set up the circuits and observe the outputs. Enter the input and output state in truth table corresponding to the input combinations.

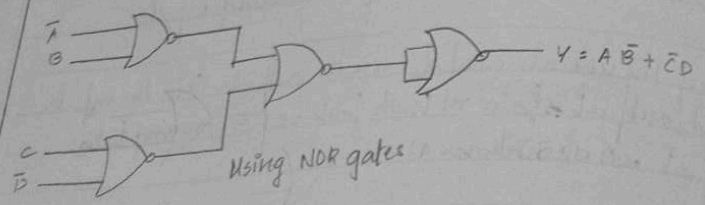
Result:

Familiarized with digital trainer kit and verified truth table of NAND, NOR etc.

K. 1/9/22



Using NAND gates



Using NOR gates

Realization of Y = AB̄ + C̄D with universal gates

/ /

Realization of logic functions using universal gates
Experiment-2

Aim

To verify the sum of products (SOP) and product of sums (POS) expressions using universal gates

Components and equipment required

ICs 7400, 7402 and digital IC trainer kit

Theory

The NAND and NOR gates are known as universal gates because many basic logic gates or Boolean expressions can be realized using either NAND gates or NOR gates alone.

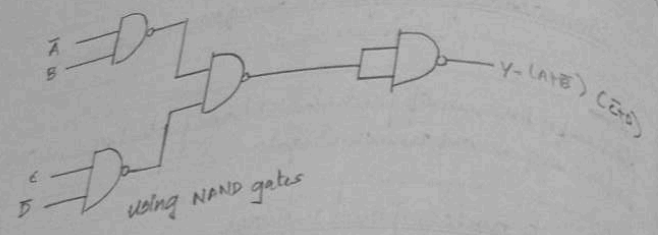
Realization of SOP expressions using NAND gates

Each product term in the SOP expression is called a minterm. The SOP expression can be economically realized using NAND gates if the following steps are adopted.

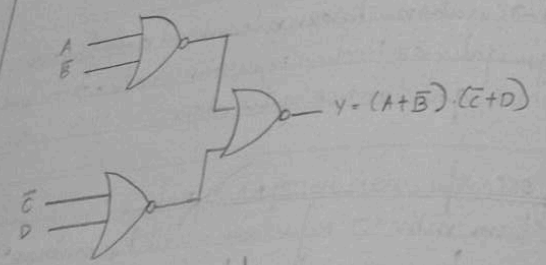
1. There will be two stages of NAND gates in the circuit. In the first stage, number of gates is equal to the number of minterms. Each NAND gate represent a minterm

2. Each literal in the minterm should be fed to corresponding NAND gate in the first stage.

3. Outputs of NAND gates in the first stage should be fed to the input of the NAND gate in second stage. The output



using NAND gates



Using NOR gate

of second stage is the required output.

The boolean expression $Y = A\bar{B} + \bar{C}D$ is realized using NAND gates.

Realization of SOP expression using NOR gates

The SOP expressions can be realized using NOR gates. The following steps are required for the realization

1. There will be three stages of NOR gates in the circuit. In the first stage, number of gates is equal to the number of minterms.

2. Complement of each literal in the minterm should be fed to corresponding NOR gates in the first stage.

3. Outputs of NOR gates in the first stage should be fed to the input of the NOR gate in second stage.

4. Third stage consists of one NOR gate as an inverter. Output of third stage is the required output.

The boolean expression $Y = A\bar{B} + \bar{C}D$ is realized using NOR gates as shown in fig 5

Realization of POS expression using NAND gates.

Each sum term in the POS expression is called a maxterm. The POS expressions can be realized using NAND gates using following steps.

1. There will be three stages of NAND gates in the circuit. In the first stage, number of gates is equal to the number of maxterms. Each of NAND gate represents a maxterm

2. Complement of each literal in the maxterm should be fed to corresponding NAND gate in the first stage

3. Outputs of NAND gates in the first stage should be fed to the input of the NAND gate in second stage

4. Third stage consists of a NAND gate as an inverter. Output of third stage is the required output

Realization of POS expressions using NOR gates

The POS expressions can be realized using NOR gates more economically if following steps are followed.

1. There will be two stages of NOR gates in the circuit. In the first stage, number of gates is equal to the number of maxterms. Each NOR gate represents a maxterms

2. Each literal in the maxterm should be fed to

corresponding NOR gate in the first stage.

3. Output of NOR gates in the first steps should be fed to the input of the NOR gate is second stage.

Procedure

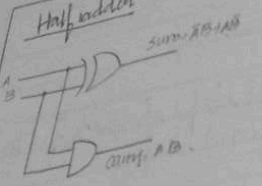
1. Test all components using digital IC tester.
2. Set up the SOP expressions using NAND and NOR gates as shown figure. Verify that the output are identical
3. Repeat the above step for POS expression as shown in fig

Result

Verified sum of product and product of sum expression using universal gates.

21/7/22

Half adder



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

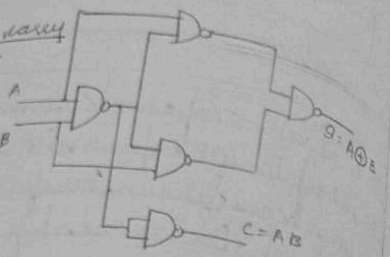
	B	$\bar{B}$
A	0	1
$\bar{A}$	1	1

SUM = $\bar{A}B + A\bar{B}$

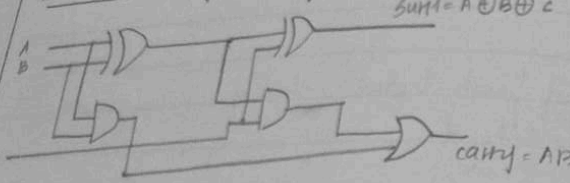
K-map for carry

	B	$\bar{B}$
A	1	0
$\bar{A}$	0	1

C = AB

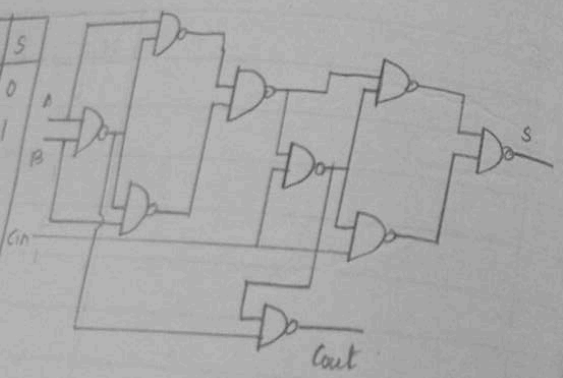


Full adder using two half adders



Truth table

A	B	C	C	S
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



Design of Adder and subtractor
Experiment - 3

Aim

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

Apparatus required

AND gate - IC 7408, X-OR gate - IC 7486, NOT gate - IC 7404, OR gate - IC 7432 and digital IC trainer kit.

Theory

Half adder

A half adder has 2 inputs for the 2 bits to be added and two outputs one from the sum 's' and other from the carry 'c' into the higher adder position. A bone circuit and is called carry adder digital form. The addition of the less significant bits 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 3 inputs and 2 outputs. A full adder is useful in add 3 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.

K-map for sum

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

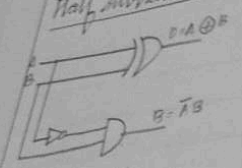
Sum = $\bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC$

K-map for carry

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

Carry = $AB + BC + AC$

Half subtractor



A	B	Borrow	Difference
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

K-map for difference

A \ B	0	1
0		1
1	1	

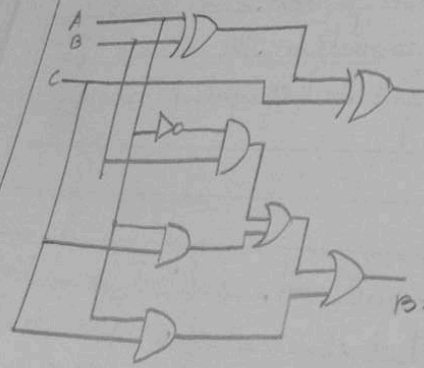
Difference = $\bar{A}B + A\bar{B}$

K-map for borrow

A \ B	0	1
0		1
1		

Borrow = $\bar{A}B$

Full subtractor



/ /

Half subtractor:

The half subtractor is constructed using X-OR and AND gate. The half subtractor has 2 input and 2 outputs. The difference can be realised using X-OR gate, borrow output can be implemented using an AND gate and a inverter.

Full subtractor

The full subtractor is combination of X-OR, AND, OR, NOT gates. In a full subtractor the logic circuit should have 3 input and 2 output. The 2 half subtractor put together forms a full subtractor. The first half subtractor will be C and AB. The output will be the difference output of full subtractor.

Procedure

1. Connections are given as per the circuit diagram.

2. Logic inputs are given as per a circuit diagram.

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

Result:

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

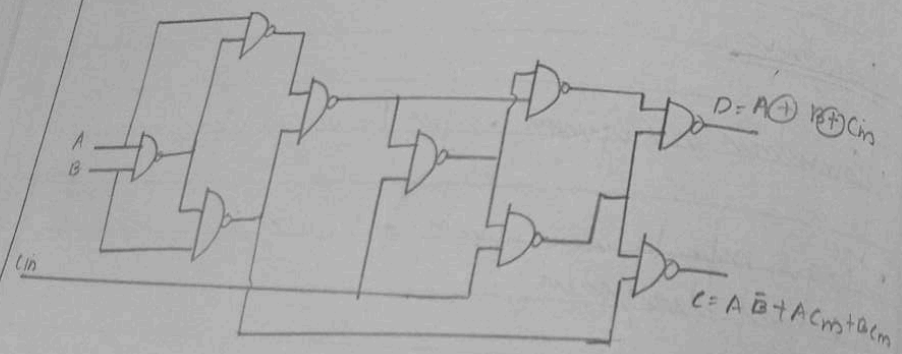
A \ BC	00	01	11	10
0		1		1
1	1		1	

$D = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

K-map for borrow

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

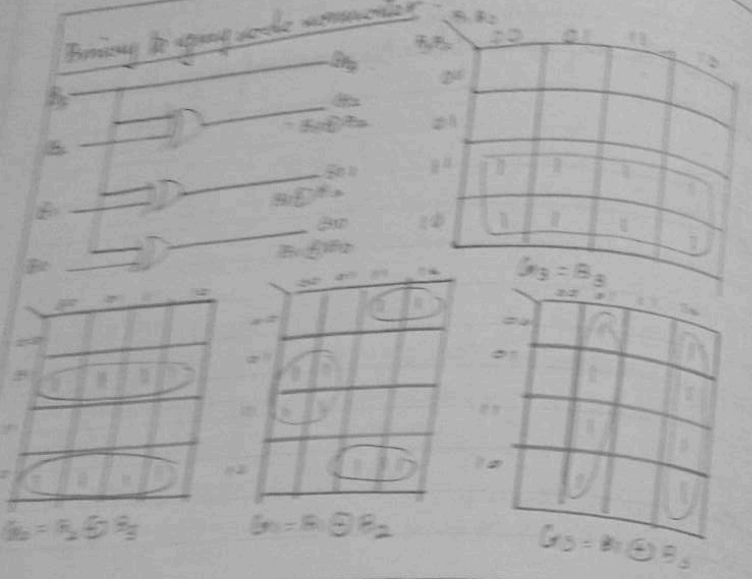
$B = \bar{A}B + BC + \bar{A}C$



/ /

Result of the experiment has been successfully obtained.

Binary to gray code converter



B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Design and implementation of code converter

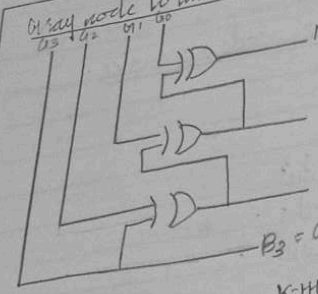
Experiment-6

Aim:
To design and implement a ckt
(i) Binary to gray code converter.
(ii) Gray to binary code converter.

Apparatus required
NOR gate - IC #402
AND gate - IC #408, OR gate - IC #432
NOT gate - IC #404 and digital trainer kit.

Theory
The availability of large varieties of codes for the same discrete elements of information results in the use of different codes by different systems. A converter circuit must be inserted between the 2 systems if each uses different codes for same information. Thus code converter is a circuit that makes the 2 systems compatible even though each uses different binary code.
The ckt combination assigned to turn any code to gray code. Gray code uses four bits to represent a decimal digit.
There are 4 inputs and 4 outputs. Gray code is a non weighted code.
The input variable are designated as B_3,

Gray code to binary converter



$B_0 = a_3 \oplus a_2 \oplus a_1 \oplus a_0$

$B_1 = a_3 \oplus a_2 \oplus a_1$

$B_2 = a_3 \oplus a_2$

$B_3 = a_3$

K-map for B_3

a_3a_2 \ a_1a_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$B_3 = a_3$

K-map for B_2

a_3a_2 \ a_1a_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

$B_2 = a_3 \oplus a_2$

K-map for B_1

a_3a_2 \ a_1a_0	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	1	1
10	1	1	0	0

$B_1 = a_3 \oplus a_2 \oplus a_1$

K-Map for B_0

a_3a_2 \ a_1a_0	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$B_0 = a_3 \oplus a_2 \oplus a_1 \oplus a_0$

/ /

B_2, B_1, B_0 and the output variables are designated as C_3, C_2, C_1, C_0 from the truth table, combinational circuit is designed. The boolean functions are obtained from K-map for each output variable.

A code converter is a circuit that makes the 2 systems compatable even though each uses a different binary code. Its convert from binary code to excess 3 code, The input lines must supply the bit combination of elements as specified by code and the output lines generated the corresponding bit combination of code. Each one of the 4 outputs of the circuit as a function of the 4 input variables

A 2 level logic diagram may be obtained directly from the boolean expression derived by the maps. There are various other possibilities for a logic diagram that implement. This circuit. Now the OR gate whose output is C+D has been used to implement partially each of 3 outputs.

Procedure:

1. Connections were given as per circuit diagram

2. Logical inputs was given as per truth table.

Truth table

G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	1	1	0	1	1	0
0	1	0	1	0	1	1	1
0	1	0	0	0	0	0	0
1	1	0	0	1	0	0	1
1	1	0	1	1	0	0	0
1	1	1	1	1	0	1	1
1	1	1	0	1	0	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

/ /

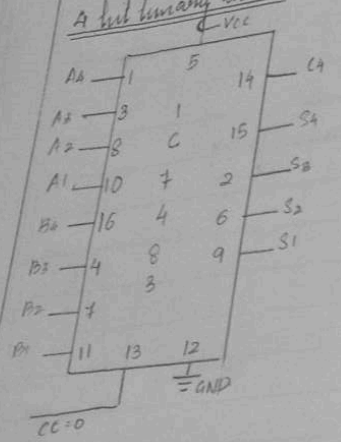
3. Assume the logical output and verify with the truth tables.

Result

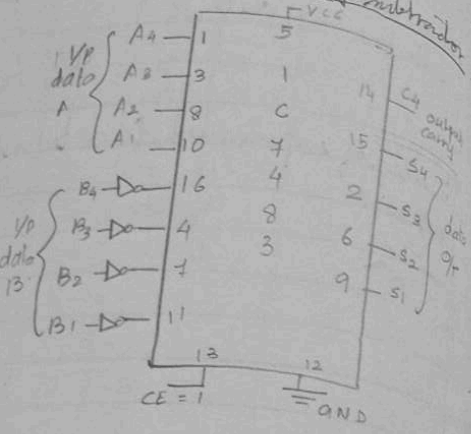
Result of experiment has been successfully ~~obtained~~

Rajan 12/2

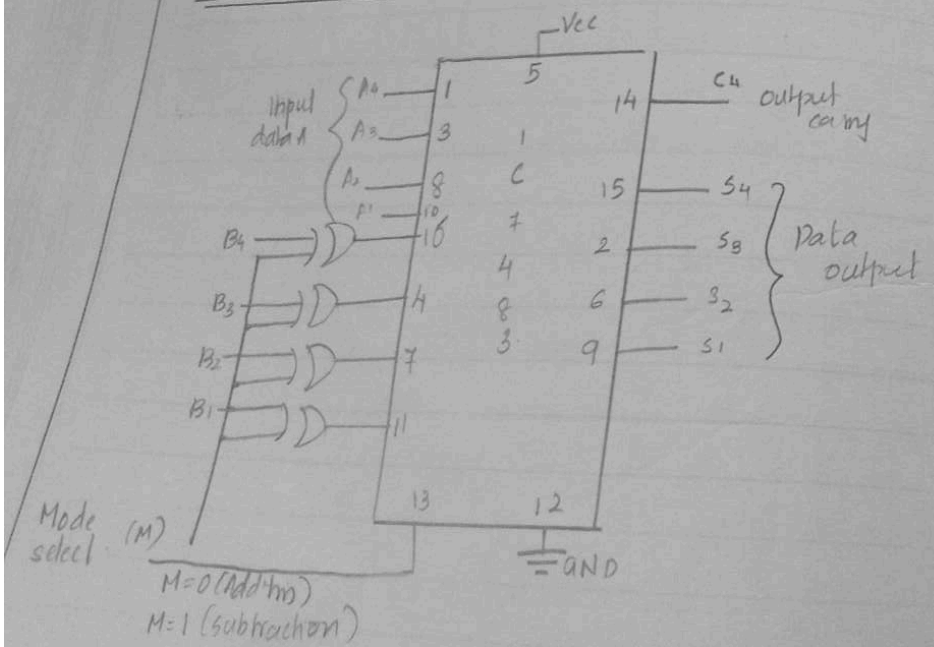
4 bit binary adder



4 bit binary subtractor



4 bit binary adder/subtracter



/ /

P4

Design of 4-bit adder and subtractor using IC 7483

Experiment-5

Aim

To design and implement 4-bit adder and subtractor using IC 7483

Apparatus Required

IC 7483, X-OR gate - IC 7486, NOT gate - IC 7404 and digital ICs trainer kit.

Theory

4 bit binary adder

A binary adder is a digital circuit that produces the arithmetic sum of 2 binary numbers. It can be constructed with full adders connected in cascade, with the output carry from each full adder connected to the input carry of next full adder in chain. The augends bits of 'A' and the addend bits of 'B' are designated by subscript numbers from right to left, with subscript 0 denoting the least significant bits. The carries are connected in chain through the full adder.

The input carry to the adder is C_0 and it ripples through the full adder to the output carry C_4.

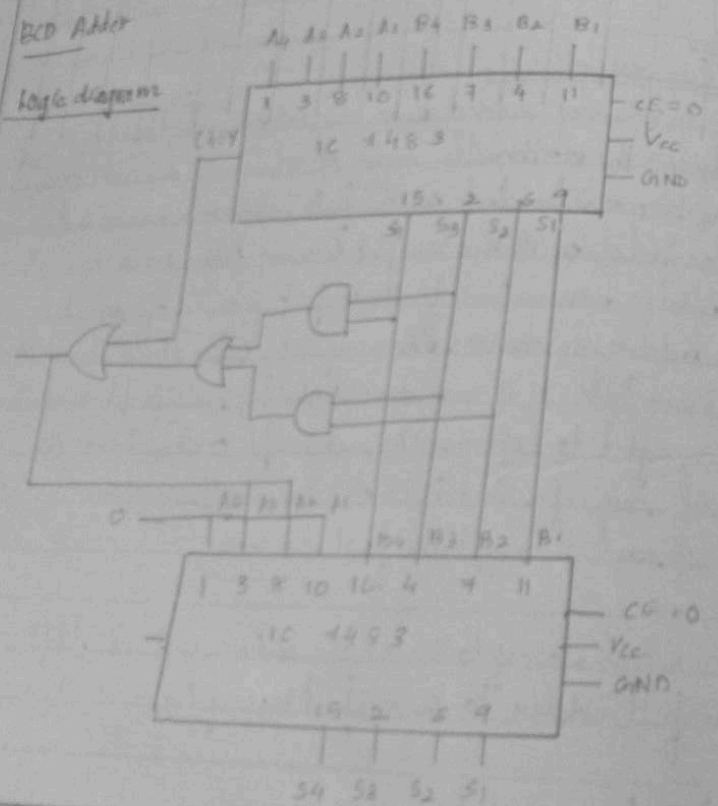
4 bit binary subtractor

Truth table

Input data A				Input data B				Addition					subtraction				
A_4	A_3	A_2	A_1	B_4	B_3	B_2	B_1	C	B_4	B_3	B_2	B_1	B	D_4	D_3	D_2	D_1
1	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0
1	0	1	0	1	0	0	0	0	1	0	1	0	0	1	0	1	0
0	0	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1	0
0	0	1	1	0	1	1	1	0	0	1	0	0	0	1	1	1	1
1	0	1	0	1	0	1	1	1	0	1	0	0	1	1	1	1	1
1	1	1	0	1	1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1

BCD Adder

Logic diagram



/ /

10

The circuit for subtracting A-B consists of an adder with inverters placed between each data input B and the corresponding input of full adder. The input carry Co must be equal to 1 when performing subtraction.

4-bit binary adder/subtractor:

The addition and subtraction operation can be combined into one circuit with one common binary adder. The mode input M controls the operation. When M=0, the circuit is adder circuit. When M=1, it becomes subtractor.

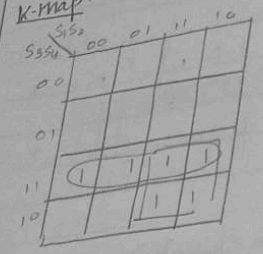
4 bit BCD adder

Consider the arithmetic addition of 2 decimal digits in BCD together with an input carry from a previous stage. Since each input digit does not exceed 9 the output sum cannot be greater than 19. The 1 in the sum being an input carry.

The output of 2 decimal digits must be represented in BCD and should appear in the form listed in the columns.

ABCD adder that adds 2 BCD digits and produces a sum digit in BCD. The 2 decimal digits together with the input carry are first added in the 4 bit adder to produce the binary sum.

K-map

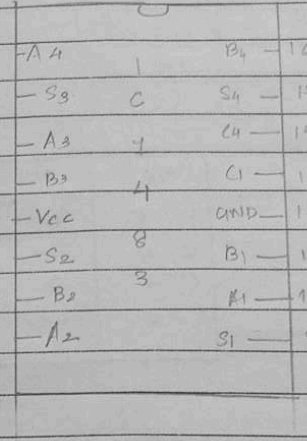


$Y = S_4(S_3 + S_2)$

Truth table

BCD SUM				carry
S_4	S_3	S_2	S_1	C_2
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

PIN diagram for IC 7483



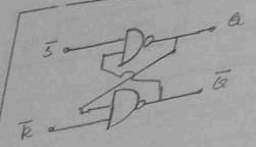
Procedure.

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

Result.

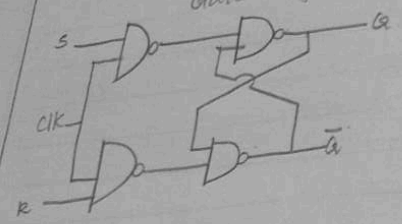
Result of experiment has been successfully obtained.

11/7/22



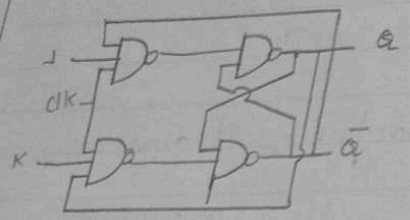
I/P		O/P
$\bar{S}$	$\bar{R}$	Q_{n+1}
0	0	X
0	1	1
1	0	0
1	1	Q_n

Gated SR flip flop.

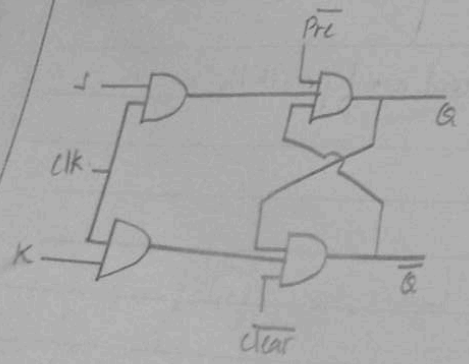


I/P		O/P
S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	X

JK flip flop



I/P		O/P
J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$



/ /

17

Study of flipflops (S,R,D,JK,T, master slave JK)
Experiment - 6

Aim

To setup the following using gates and verify the truth table. Also familiarize the flipflops IC's

1. RS flipflop
2. D flipflop
3. T flipflop
4. JK flipflop
5. Master slave flipflop

Components required

Digital IC trainer kit, IC 7400, IC 7410, IC 7473, IC 7474, IC 7476

Theory

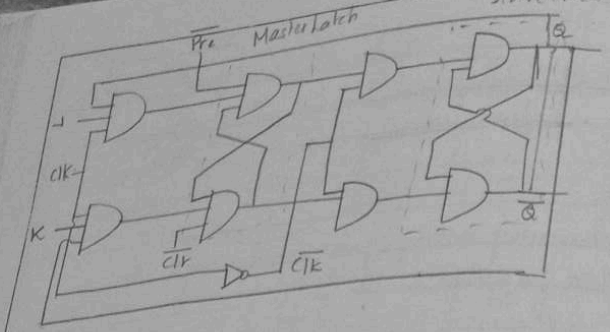
Flipflops are the basic building blocks in any memory system since its output will remain in its state until it is forced to change it by some means

SR flipflop

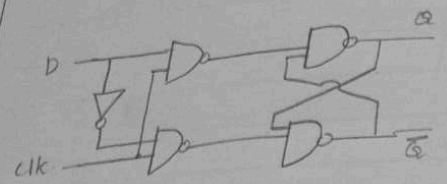
$\bar{S}$-$\bar{R}$ flipflop stands for set and reset. There are four input combination possible at the inputs But $\bar{S}=\bar{R}=1$ is forbidden since the output will be indeterminate

Gated SR flipflop

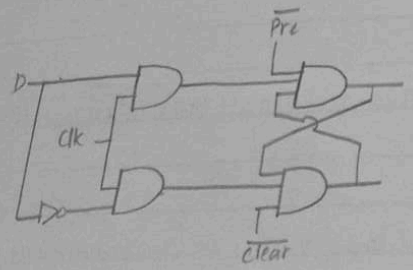
S and R stands for set and reset. There are four input combination possible at the inputs But S=R=1 is forbidden



I/P		O/P
J	K	Qn+1
0	0	Qn
0	1	0
1	0	1
1	1	Q̄n



I/P	O/P
D	Qn+1
0	0
1	1



since the output will be indeterminate

JK flipflop

The indeterminate output state of SR FF when S=R=1 is avoided by converting it to a JK FF. When flipflop is switched on its output state is certain. When an initial state is to be assigned two separate inputs called preset and clear are used

Master slave JK flipflop

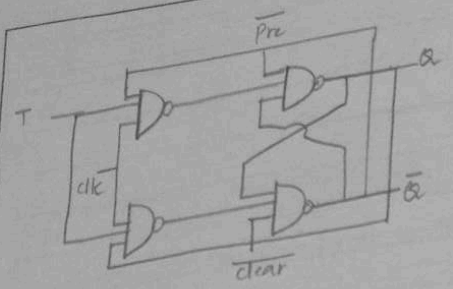
The race around condition of JK FF is rectified in master slave JK FF. Racing is toggling of output more than once during the positive clock edge. MS JK FF is created by cascading two JK FF. The clock is fed to the first stage and is inverted and fed to the second stage. This ensures that the master slave is always followed by the slave and eliminate the chance of racing.

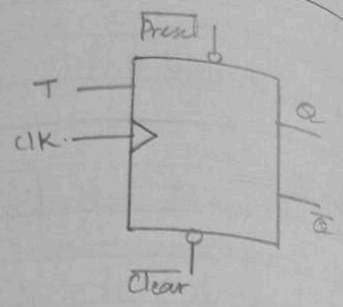
D FF

It has only one input called as D input or Data input. The input data is transferred to the output after a clock is applied. D-FF can be derived from JK FF by using J input as D input and J is inverted and fed to K input

T-FF

T stands for toggle. The output toggles when a clock pulse is applied. T-FF can be derived from JK-FF





T	Qn+1
0	Qn
1	$\overline{Qn}$

by observing J and K input

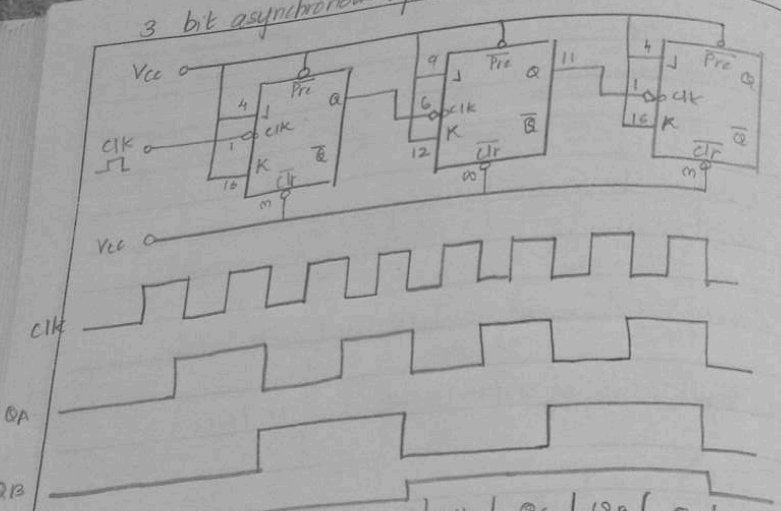
Procedure:

1. Test all components and IC packages using digital IC tester and multimeter
2. Set up FF using gates and verify their truth table
3. Verify the truth tables of 7473, 7474 and 74761 ICs

Result:

Result of the experiment has been successfully obtained

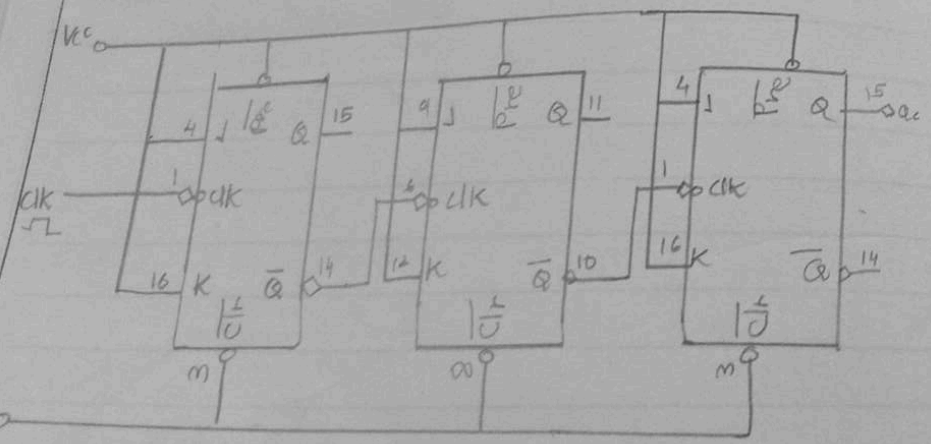
3 bit asynchronous up counter.



CLK	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

CLK	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

3 bit asynchronous down counter



/ /

20

Asynchronous counter design and mod-N counter

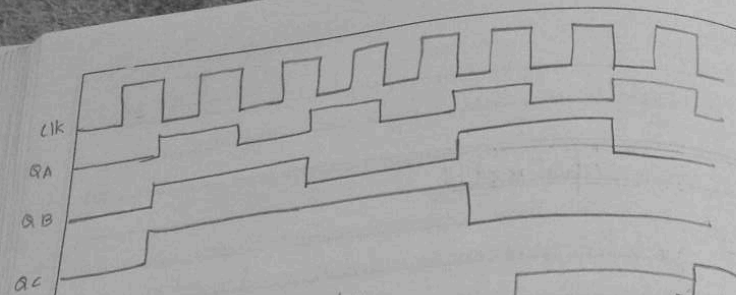
Experiment-7

Aim:
Realization of 3-bit asynchronous counter and mod-N counter design

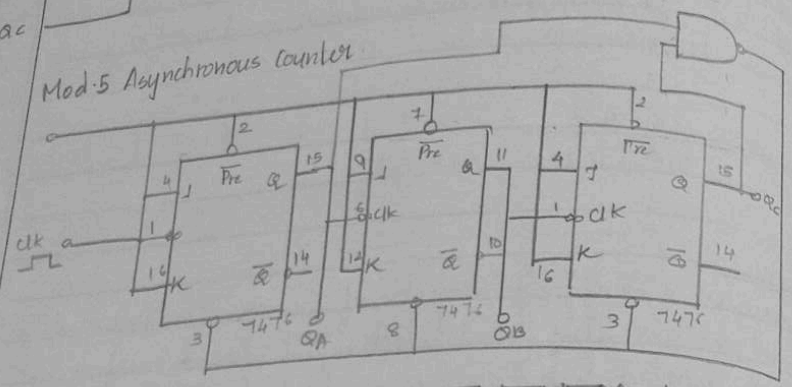
Apparatus required
IC 7408, IC 7476, IC 7400, IC 7432 etc.

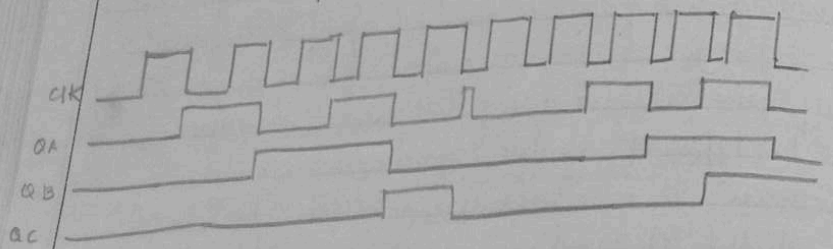
Theory
A counter is a register capable of counting number of clock pulse arriving at its clock input. Counter represents the number of clock pulses arrived. A specified sequence of states appears as counter output. This is the main difference between a register and a counter. There are two types of counter,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

As soon the clock of second stage is triggered by output of first stage.



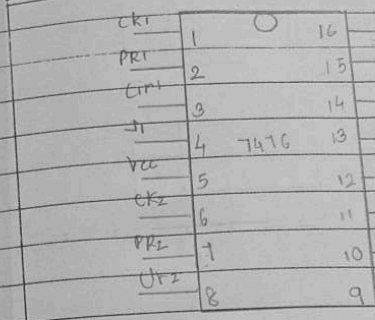
Mod-5 Asynchronous counter





CIK	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

PIN details



Procedure

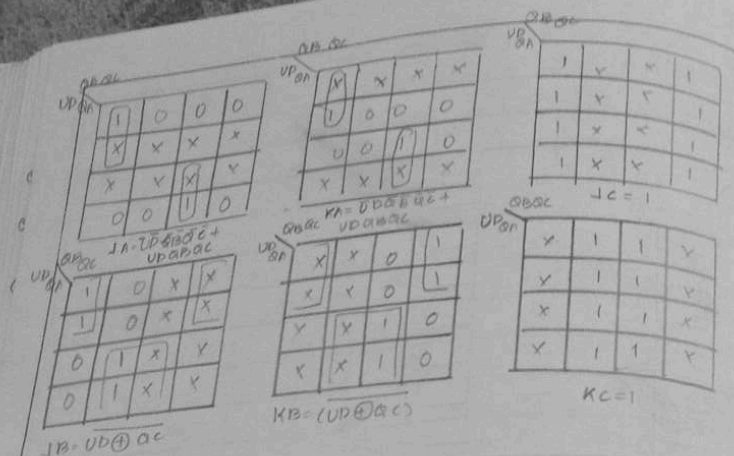
(i) Connections are given as per the circuit diagram

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

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

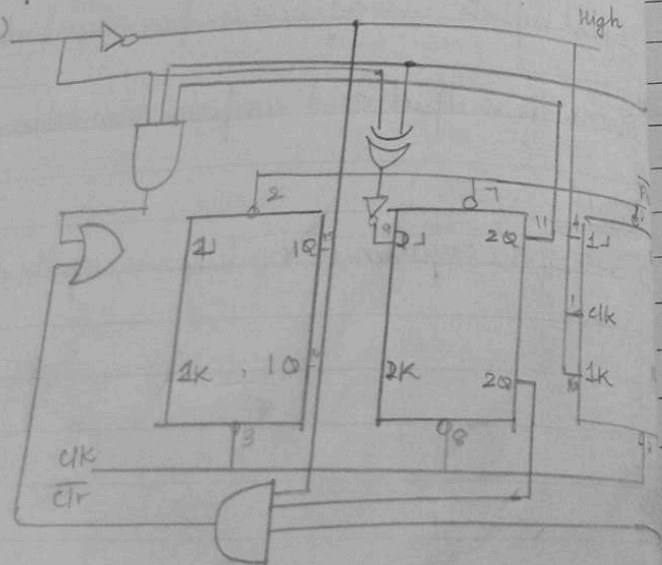
Result:

Result of the experiment has been successfully obtained



Characteristics table

Q	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



/ /

Design and implementation of 3 bit synchronous updown counter

Experiment 8

Aim

To design and implement 3 bit synchronous updown counter

Apparatus required

Sl.No	Component	Specification	Qty
1	JK FF	IC 7476	2
2	3 I/P AND gate	IC 7411	1
3	OR gate	IC 7432	1
4	XOR gate	IC 7486	1
5	NOT gate	IC 7404	1
6	IC trainer kit	-	1

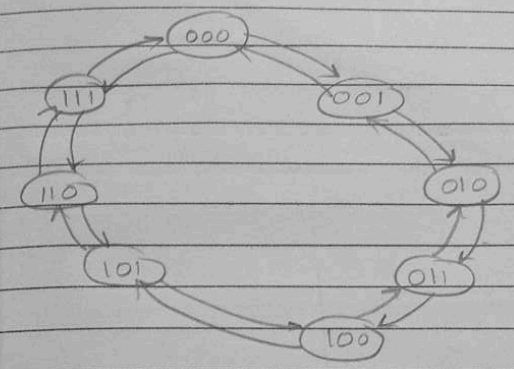
Theory

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. An up/down counter is one that is capable of progressing in increasing order or descending order through a certain sequence. An up/down counter is also called bidirectional counter. Usually up/down operation of the counter is controlled by up/down signal. When this

Input	Present state			Next state			A		B		C	
	Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+1}	J_A	K_A	J_B	K_C	J_C	K_C
	0	0	0	1	1	1	1	x	1	x	1	x
0	0	1	1	1	1	0	x	0	x	0	x	x
0	1	1	0	1	0	1	x	0	x	1	1	1
0	1	1	0	1	0	0	x	0	0	x	x	x
0	1	0	1	1	0	1	x	1	1	x	1	1
0	1	0	0	0	1	1	0	x	x	0	x	x
0	1	0	1	0	1	0	0	x	x	1	1	1
0	0	1	1	0	0	1	0	x	0	x	x	x
0	0	1	0	0	0	0	0	x	0	x	1	1
0	0	0	1	0	0	1	0	x	0	x	1	x
0	0	0	0	0	1	1	0	x	1	x	x	1
1	0	0	1	0	1	0	0	x	x	0	1	x
1	0	1	0	0	0	1	x	x	x	1	x	1
1	0	1	1	0	0	0	x	0	0	x	1	x
1	1	0	0	1	1	1	x	0	1	x	x	1
1	1	0	1	1	1	0	x	0	x	0	1	x
1	1	1	0	1	0	1	x	0	x	1	x	1
1	1	1	1	0	0	0	x	1				

signal is high counter goes through up sequence and when up/down signal is low counter follows reverse sequence

State diagram



Procedure

(i) Connections are given as per circuit diagram

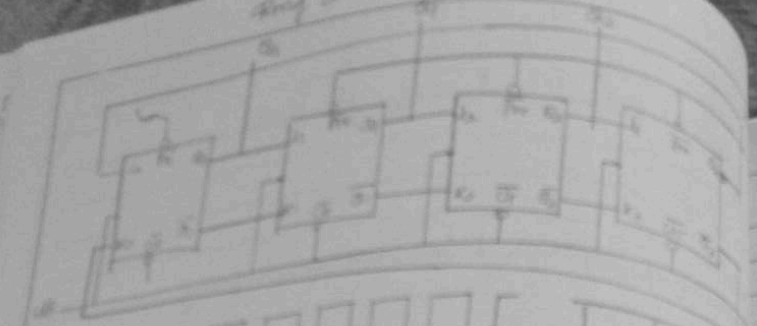
(ii) Logical inputs are given as per circuit diagram.

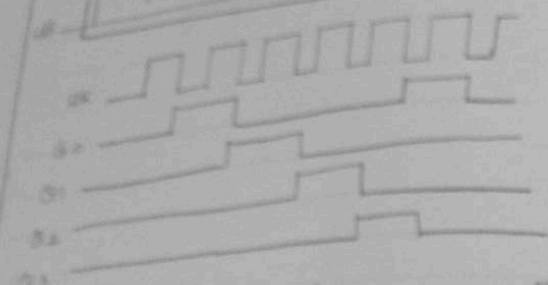
(iii) Observe the output and verify the truth table

Result:

Result of the experiment has been successfully obtained.

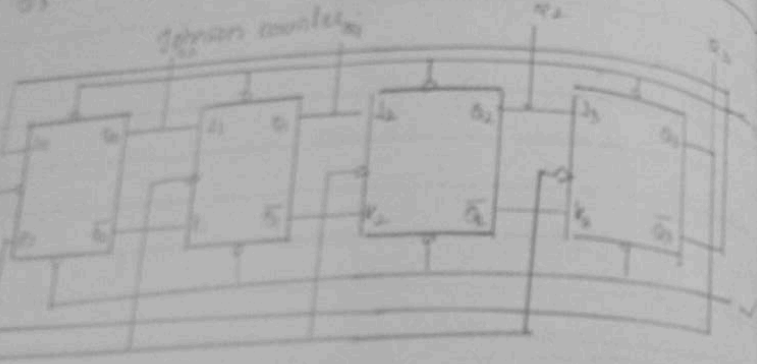
Ring counter

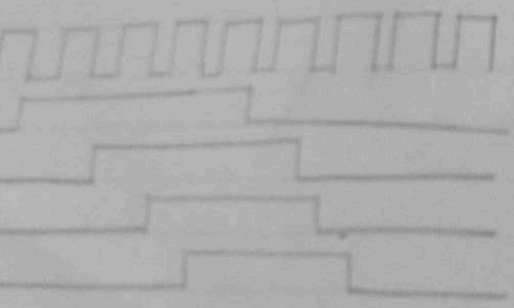




CLK	Q_0	Q_1	Q_2	Q_3
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Johnson counter





CLK	Q_0	Q_1	Q_2	Q_3
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
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

Ring counter / Johnson counter
Experiment - 9

Aim:
To design and setup four bit johnson and ring counter using J-K FF

Components required

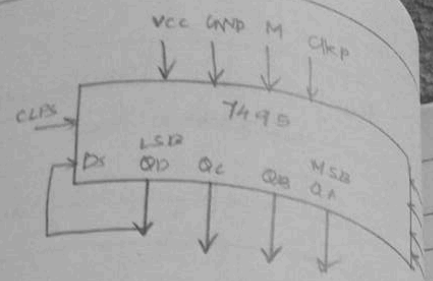
Digital IC trainer kit IC 7476

Theory
Ring counter and johnson counter are basically shift registers

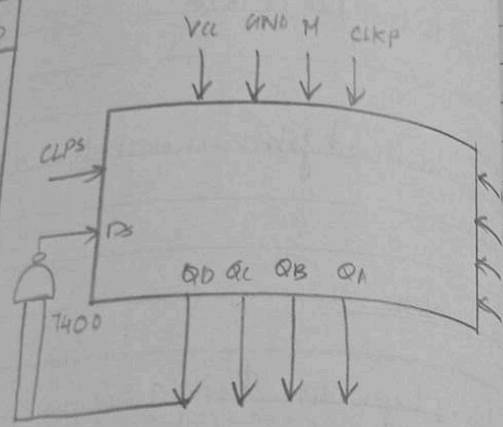
Ring counter

It is made by connecting Q & $\bar{Q}$ output of one JK FF to J & K input of next FF respectively. The Q of final FF is connected to the input of first FF. To start the counter the first FF is set by using present facility and the remaining FF are reset input. When the clock arrives the set condition continues to shift around the ring. As it can be seen from the truth table there are four unique output states for this counter. The modulus value of a ring counter is n, where n is the number of flip flops. Ring counter is called divided by n counter where N is the number of FF

CP	QA	QB	QC	QD
t0	1	0	0	0
t1	0	1	0	0
t2	0	0	1	0
t3	0	0	0	1
t4	1	0	0	0



CP	QA	QB	QC	QD
t0	1	0	0	0
t1	1	1	0	0
t2	1	1	1	0
t3	1	1	1	1
t4	0	1	1	1
t5	0	0	1	1
t6	0	0	1	1
t7	0	0	0	1
t8	1	0	0	0



/ /

Johnson counter (twisted ring counter)

The modulus value of a ring counter can be doubled by making a small change in the ring counter circuit. The $\bar{Q}$ and Q of the last FF's are connected to the J and K input of the first FF respectively. This is Johnson counter. Initially the FF's are reset. After first clock pulse FF is set and the remaining FF's are reset. After the eight clock pulse all the FF's are reset. There are eight different conditions creating a mode 8 Johnson counter. Johnson counter is called a twisted ring counter or divide by 2N counter.

Ring counter using IC-7495

Components required

SL NO	Component	IC Number	Quantity
1	Ring counter	7495	1
2	NAND gate	7400	1.
	Patch chord		
	Trainer kit		

Procedure.

1. Rig up the circuit as shown in the schematic, Ds is not given as input.

/ /

2. Load data parallely with clkp and M=1

3. Then make M=0, clks=cp.

4. Verify the whether the ckt works as a Johnson counter or twisted ring counter.

Result:

Result of the experiment has been successfully obtained.