PRACTICAL-4

STUDY OF DIFFERENT LOGIC GATES

AIM: To study different logic gates.

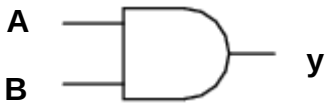
APPARATUS: [1] Digital Trainer
[2] Hook up wires
[3] TTL ICs 7400, 7402, 7404, 7408, 7432, 7486

THEORY:

[1] AND Gate:

The AND gate performs logical multiplication. This operation is represented by a dot or by the absence of an operator. For example, A•B=y or AB=y is read, "A AND B is equal to y". The logical operation AND is interpreted to mean that y=1 if and only if A=1 and B=1; otherwise y=0.

Symbol:



Boolean Expression:

$y = A \bullet B = AB$

Truth Table:

A	B	y
0	0	0
0	1	0
1	0	0
1	1	1

The IC required to investigate the logic behavior of AND gate is 7408 [Quadruple 2-input AND gates]. "Quadruple" means that there are four gates each has 2 inputs within the package.

[2] OR Gate:

The OR gate performs logical addition. This operation is represented by a plus sign. For example, A+B=y is read, "A OR B is equal to y". The logical operation OR is interpreted to mean that y=1 if A=1 or if B=1 or if both A=1 and B=1. If both A=0 and B=0 then y=0. TTL IC7432 [Quadruple 2-input OR gates] is used to perform the experiment.

Symbol:

Boolean Expression:

$$y = A+B$$

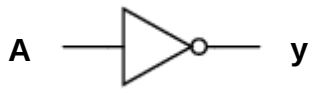
Truth Table:

A	B	y
0	0	0
0	1	1
1	0	1
1	1	1

[3] NOT Gate:

The NOT gate performs a basic logical function called inversion or complementation. This operation is represented by a prime (sometimes by a bar). For example, A'=y (or $\bar{A}$=y) is read, "A NOT is equal to y", meaning that y is what A is not. In other words if A=1, then y=0; but if A=0 then y=1. IC 7474 [Hex Inverters] is used to verify the truth table of NOT gate.

Symbol:



Boolean Expression:

$$y = A' = \bar{A}$$

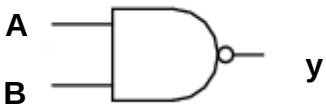
Truth Table:

A	y
0	1
1	0

[4] NAND Gate:

The NAND function is the complement of the AND function and its name is an abbreviation of not-AND. The NAND gate is very popular because it is a "Universal" gate. Any basic gates like AND, OR and NOT or any digital system can be implemented with it. The IC required to verify the truth table of NAND gate is 7400 [Quadruple 2-input NAND gates].

Symbol:



Boolean Expression:

$$y = (AB)'$$

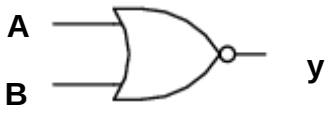
Truth Table:

A	B	y
0	0	1
0	1	1
1	0	1
1	1	0

[5] NOR Gate:

The NOR function is the complement of OR and is an abbreviation of not-OR. The NOR gate is also a "Universal" gate because any Boolean function can be implemented with it. For the experiment of verification of truth table of NOR here we use TTL IC 7402 [Quadruple 2-input NOR gates].

Symbol:



Boolean Expression:

$y = (A+B)'$

Truth Table:

A	B	y
0	0	1
0	1	0
1	0	0
1	1	0

[6] Exclusive-OR (EX-OR) Gate:

The Exclusive-OR (XOR) denoted by the symbol $\oplus$ is a logical operation that performs the following Boolean operation $A \oplus B = AB' + A'B$. It is particularly useful in arithmetic operations and error-detection and correction circuits.

Symbol:

Boolean Expression:

$y = A \oplus B = AB' + A'B$

Truth Table:

A	B	y
0	0	0
0	1	1
1	0	1
1	1	0

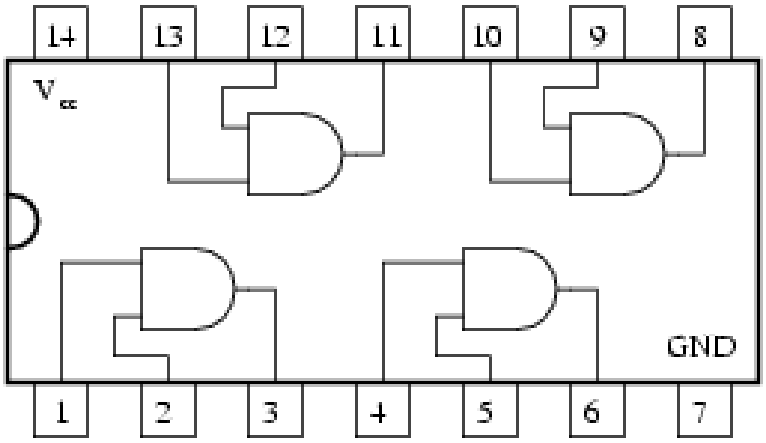
PROCEDURE:

1. Fix the appropriate IC on Digital trainer bread board so that each pin is in different hole and no two pins are connected.
2. Apply Vcc (+5Volts) and Ground to IC using hook up wires through trainer board.
3. Now connect the input pins of any one gate of IC to the supply voltage through switch.
4. Connect the output pin of that gate of the IC to LED on trainer board.
5. Though switch the input can be switched from 0 to 1 value.
6. LED glows if output is 1 and does not glow if output is 0.
7. Verify the Truth table by giving different combinations of inputs.

CONCLUSION:

IC-7408 Quad 2-Input AND Gates

Pin-Diagram

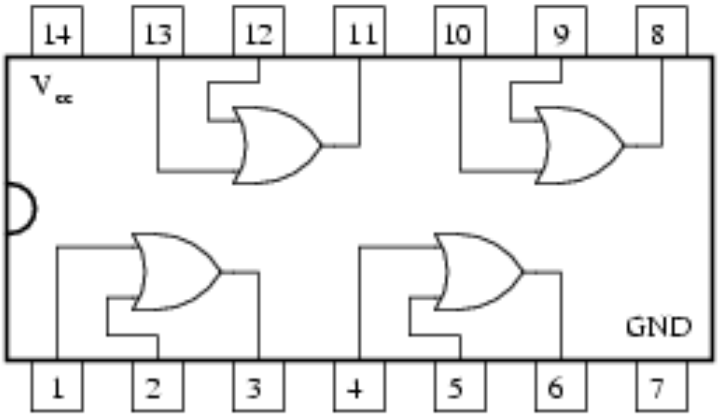


Observation Table

INPUTS		OUTPUTS
A	B	y

IC-7432 Quad 2-Input OR Gates

Pin-Diagram

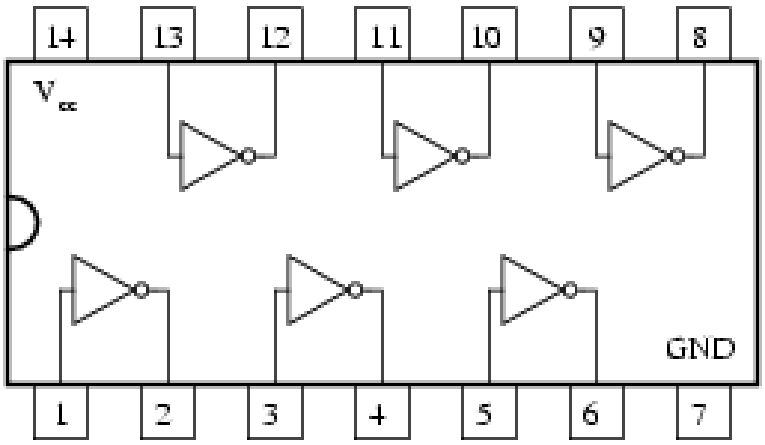


Observation Table

INPUTS		OUTPUTS
A	B	y

IC-7404 Hex Inverters

Pin-Diagram

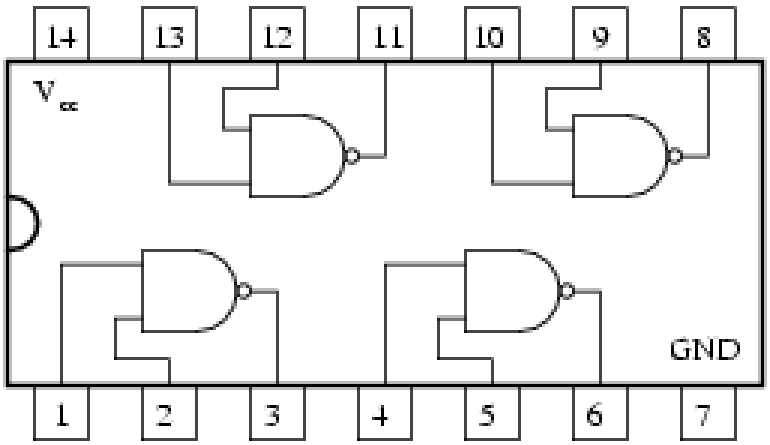


Observation Table

INPUTS	OUTPUTS
A	y

IC-7400 Quad 2-Input NAND Gates

Pin-Diagram

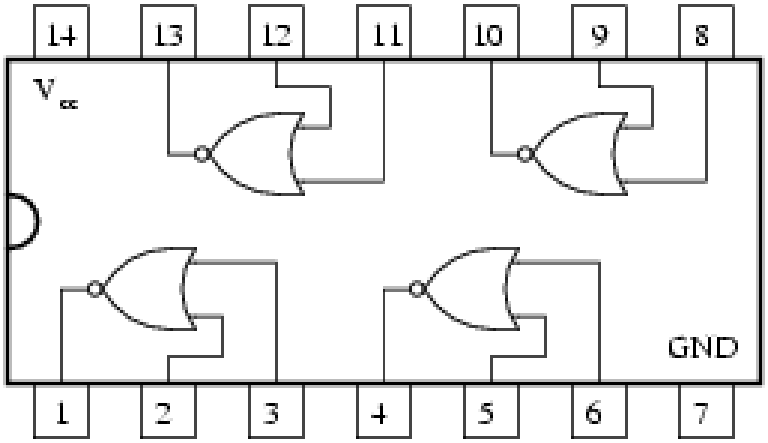


Observation Table

INPUTS		OUTPUTS
A	B	y

IC-7402 Quad 2-Input NOR Gates

Pin-Diagram

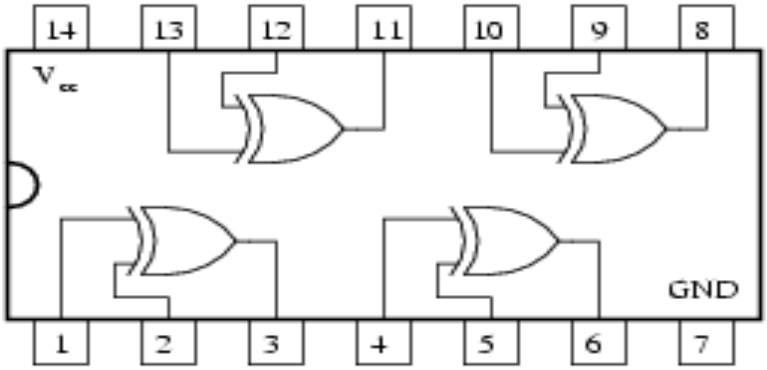


Observation Table

INPUTS		OUTPUTS
A	**B**	**y**

IC-7486 Quad 2-Input EX-OR Gates

Pin-Diagram



Observation Table

INPUTS		OUTPUTS
A	**B**	**y**

PRACTICAL-2

UNIVERSAL GATES

AIM: To study universal gates. (nand ,nor)

APPARATUS: [1] Digital Trainer
[2] Hook up wires
[3] TTL ICs 7400, 7402

THEORY:

The two Universal gates are NAND gate and NOR gate. Using these gates any digital system or Boolean function can be implemented.

[A] NAND gate as a Universal Gate:

$$y = (A \bullet B)'$$

Implementation of different gates using NAND gate:

[1] NOT gate:

$y = A'$
$y = (A \bullet A)'$ ($A \bullet A = A$)

$y = (A \bullet A)'$

[2] AND gate:

$y = A \bullet B$
$y = [\ (A \bullet B)'\ ]'$

$y = [\ (A \bullet B)'\ ]'$

[3] OR gate:

$y = A+B$
$y = [(A+B)']'$
$y = [A' \bullet B']'$

$y = [A' \bullet B']'$

[4] NOR gate:

$y = (A+B)'$
$y = [(A' \bullet B')']'$

$y = [(A' \bullet B')']'$

[5] Ex-OR gate:

$y = A'B+AB'$

= [(A'B+AB')']'
= [AB+A'B']'
= (AB)' • (A'B')'
= (AB)' • (A+B)
= A•(AB)' + B•(AB)'
= [(A•(AB)')' • (B•(AB)')']'

y = [(A•(AB)')' • (B•(AB)')']'

[B] NOR gate as a Universal Gate:

y = (A+B)'

Implementation of different gates using NOR gate:

[1] NOT gate:

y = A'
y = (A+A)' (A+A=A)

y = (A+A)'

[2] OR gate:

y = A+B
y = [(A+B)']'

y = [(A+B)']'

[3] AND gate:

y = A•B
y = [(A•B)']'
y = [A'+B']'

y = [A'+B']'

[4] NAND gate:

y = (A•B)'
y = [(A'+B')']'

y = [(A'+B')']'

[5] Ex-NOR gate:

y = AB+A'B'
= AB+(A+B)'
= [A+(A+B)'] • [B+(A+B)']
= [[A+(A+B)']' + [B+(A+B)']']'

y = [[A+(A+B)']' + [B+(A+B)']']'

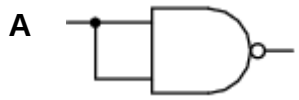
PROCEDURE:

8. For the implementation of the different gates first connect the TTL IC-7400 on Digital trainer bread board so that each pin is in different hole and no two pins are connected.
9. Apply Vcc (+5Volts) and Ground to IC using hook up wires through trainer board.
10. Now connect the terminals of the gates as shown in particular gate design.
11. Then connect the input pins to the supply voltage through switch.
12. Connect the output pin to LED on trainer board.
13. Though switch the input can be switched from 0 to 1 value.
14. LED glows if output is 1 and does not glow if output is 0.
15. Verify the Truth table by giving different combinations of inputs to designed gate.
16. Repeat the steps for IC-7402.

CONCLUSION:

[A] Implementation of different gates using NAND gate:

NOT gate:

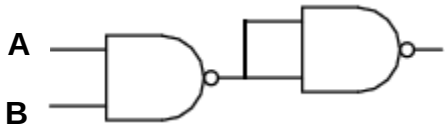


(A•A)' = **A'**

Observation Table

A	y

AND gate:

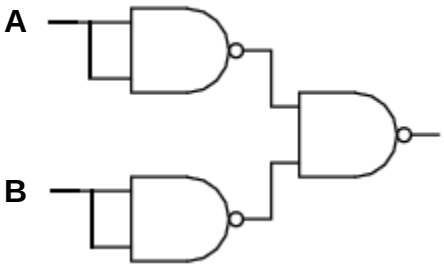


[(AB)']' = **AB**

Observation Table

A	B	y

OR gate:

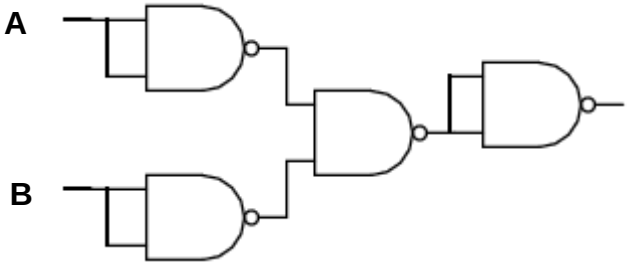


[A'•B']' = **A+B**

Observation Table

A	B	y

NOR gate:



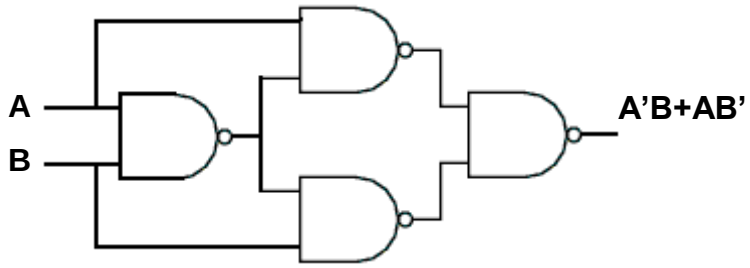
[(A'•B')']'
= (A+B)'

Observation Table

A	B	y

EX-OR gate:

Observation Table



A	B	y

[B] Implementation of different gates using NOR gate:

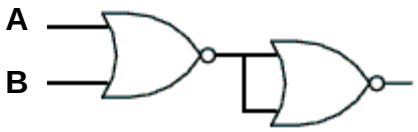
NOT gate:

Observation Table

A	y

OR gate:

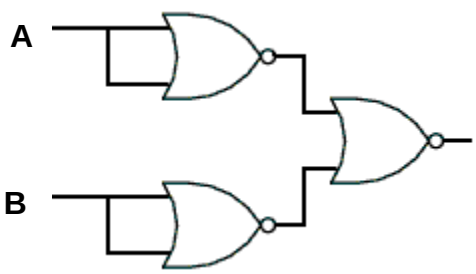
Observation Table



A	B	y

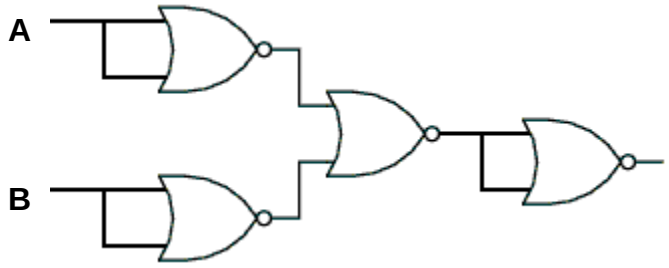
AND gate:

Observation Table



A	B	y

NOR gate:

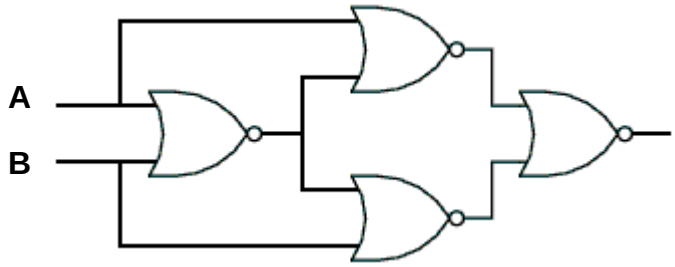


[(A'+B')']'
= (AB)'

Observation Table

A	B	y

EX-OR gate:



A'B+AB'

Observation Table

A	B	y

PRACTICAL-6

HALF & FULL ADDER

AIM: TO DESIGN AND STUDY THE HALF ADDER & FULL ADDER.

APPARATUS: [1] Digital Trainer
[2] Hook up wires
[3] TTL ICs 7486, 7408, 7432

THEORY:

The Adder circuit is a combinational logic circuit. There are two types of adder circuits: (1) Half adder and
(2) Full Adder.

HALF ADDER:

A combinational circuit that performs the addition of two bits is called a half-adder. This circuit needs two binary inputs and two binary outputs. The input variables produce the sum and the carry.

Truth Table:

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Boolean Expression:

The simplified Boolean expressions for the two outputs obtained from the truth table.

$S = x'y + xy'$

$C = xy$

$S = x \oplus y$ and **$C = xy$**

FULL ADDER:

A combinational circuit that performs the addition of three bits is called a Full-adder. This consists of three inputs and two outputs.

Truth Table:

x	y	z	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

Boolean Expression:

The simplified Boolean expressions:

For S:

x \ yz	00	01	11	10
0	0	1	0	1
1	1	0	1	0

∴ S = x'y'z + x'yz' + xy'z' +xyz

For C:

x \ yz	00	01	11	10
0	0	0	1	0
1	0	1	1	1

∴ C = xy + xz + yz

Implementation of a Full-adder with two half adders and an OR gate:

S = x'y'z + x'yz' + xy'z' + xyz
= x' (y'z+yz') + x (y'z'+yz)
= x' (y'z+yz') + x (y'z+yz')'
= x' (y ⊕ z) + x (y ⊕ z)'
= x ⊕ y ⊕ z

∴ S = x ⊕ y ⊕ z

C = x'yz +xy'z +xyz' +xyz
= (x'y + xy') z + xy (z'+z)
= (x ⊕ y) z + xy

∴ C = (x ⊕ y) z + xy

PROCEDURE:

For Half Adder Circuit:

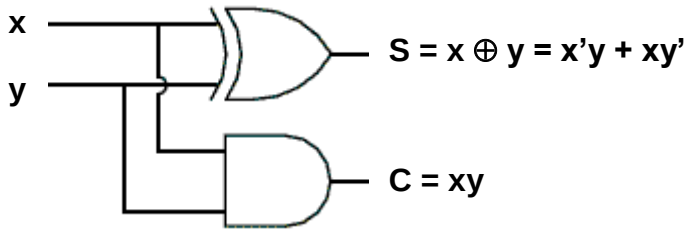
17. Fix the IC 7486 and IC 7408 on Digital trainer bread board so that each pin is in different hole and no two pins are connected.
18. Apply Vcc (+5Volts) and Ground to both ICs using hook up wires through trainer board.
19. Connect the gates as per given circuit design.
20. Connect the input terminals to the supply voltage through switches and output terminal to the LED.
21. Verify the Truth table by giving different combinations of inputs to designed circuit.

For Full Adder Circuit:

1. Fix the IC 7486, IC 7408 and IC 7432 on Digital trainer bread board so that each pin is in different hole and no two pins are connected.
2. Apply Vcc (+5Volts) and Ground to ICs using hook up wires through trainer board.
3. Connect the gates as per given circuit design.
4. Connect the input terminals to the supply voltage through switches and output terminals to the separate LEDs.
5. Verify the Truth table by giving different combinations of inputs to designed circuit.

CONCLUSION:

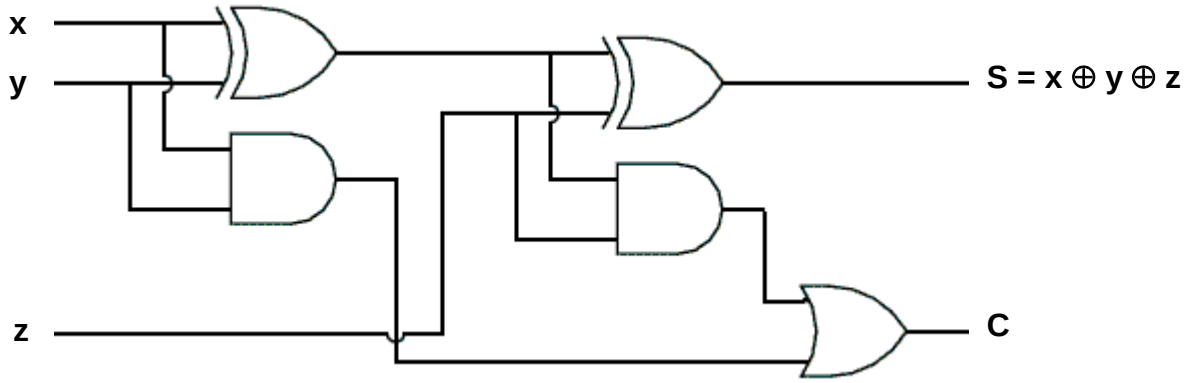
HALF ADDER CIRCUIT



Observation Table

x	y	C	S

FULL ADDER CIRCUIT



Observation Table

x	y	z	C	S

PRACTICAL-4

HALF & FULL SUBTRACTOR

AIM: To design and study the half subtractor and full subtractor.

APPARATUS: [1] Digital Trainer
[2] Hook up wires
[3] TTL ICs 7486, 7408, 7432, 7404

THEORY:

The subtractor circuit is also a combinational logic circuit. There are two types of subtractor circuits: (1) Half subtractor and
(2) Full subtractor.

HALF SUBTRACTOR:

A half subtractor is a combinational circuit that subtracts two bits and produces their difference. It also has an output to specify if a 1 has been borrowed. The half subtractor needs two inputs and two outputs (Difference and Borrow).

Truth Table:

X	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Boolean Expression:

The simplified Boolean expressions for the two outputs obtained from the truth table.

$D = x'y + xy'$

$B = x'y$

$D = x \oplus y$ and **$B = x'y$**

FULL SUBTRACTOR:

A full subtractor is a combinational circuit that performs the subtraction between two bits, taking into account that a 1 may have been borrowed by a lower significant stage. This circuit has three inputs and two outputs.

Truth Table:

x	y	z	B	D
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

Boolean Expression:

The simplified Boolean expressions:

For D:

x \ yz	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$\therefore$ D = x'y'z + x'yz' + xy'z' +xyz

For B:

x \ yz	00	01	11	10
0	0	1	1	1
1	0	0	1	0

$\therefore$ B = x'y + x'z + yz

Implementation of a Full-subtractor with two half subtractors and an OR gate:

D = x'y'z + x'yz' + xy'z' + xyz
= x' (y'z+yz') + x (y'z'+yz)
= x' (y'z+yz') + x (y'z+yz')'
= x' (y $\oplus$ z) + x (y $\oplus$ z)'
= x $\oplus$ y $\oplus$ z

$\therefore$ D = x $\oplus$ y $\oplus$ z

B = x'y'z +x'yz' +x'yz +xyz
= z (x'y'+xy) + x'y (z'+z)
= z (x'y'+xy) + x'y
= z (x $\oplus$ y)' + x'y

$\therefore$ B = (x $\oplus$ y)' z + x'y

PROCEDURE:

For Half Subtractor Circuit:

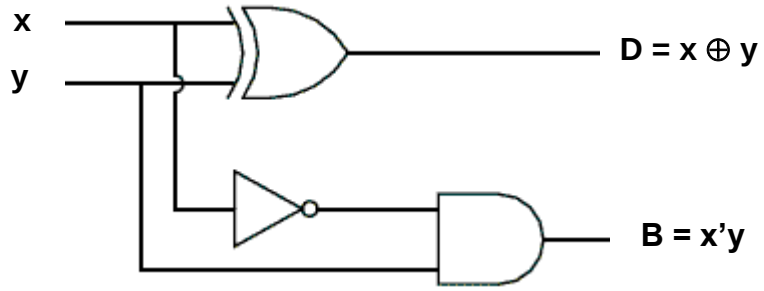
22. Fix the IC 7486, IC 7408 and IC 7404 on Digital trainer bread board so that each pin is in different hole and no two pins are connected.
23. Apply Vcc (+5Volts) and Ground to all ICs using hook up wires through trainer board.
24. Connect the gates as per given circuit design.
25. Connect the input terminals to the supply voltage through switches and output terminal to the LED.
26. Verify the Truth table by giving different combinations of inputs to designed circuit.

For Full Subtractor Circuit:

6. Fix the IC 7486, IC7408, IC 7404 and IC 7432 on Digital trainer bread board so that each pin is in different hole and no two pins are connected.
7. Apply Vcc (+5Volts) and Ground to all ICs using hook up wires through trainer board.
8. Connect the gates as per given circuit design.
9. Connect the input terminals to the supply voltage through switches and output terminals to the separate LEDs.
10. Verify the Truth table by giving different combinations of inputs to designed circuit.

CONCLUSION:

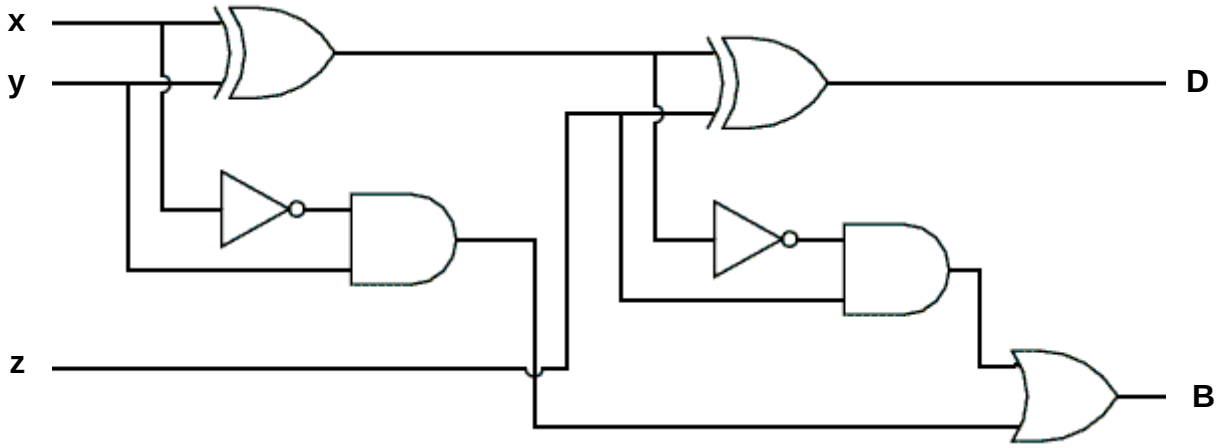
HALF SUBTRACTOR CIRCUIT



Observation Table

x	y	B	D

FULL SUBTRACTOR CIRCUIT



Observation Table

x	y	z	B	D

PRACTICAL-5

CODE CONVERTERS

AIM: To design and study the Code Converters:
(a) Binary to Gray Code Converter
(b) Gray to Binary Code Converter

APPARATUS: [1] Digital Trainer
[2] Hook up wires
[3] TTL IC 7486

THEORY:

A code converter is a combinational circuit that makes the two systems compatible even though each uses a different binary code. Using code converters between two systems of different codes, we can use the output of one system as input to another.

[A] BINARY TO GRAY CODE CONVERTER:

Truth Table:

Binary Input				Gray Output			
A	**B**	**C**	**D**	**w**	**x**	**y**	**z**
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

Boolean Expressions:
(1) For w:

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
10	1	1	1	1
11	1	1	1	1

∴ **w = A**

(2) For x:

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
10	0	0	0	0
11	1	1	1	1

∴ x = A'B + AB'
∴ **x = A ⊕ B**

(3) For y:

AB \ CD	00	01	11	10
00	0	0	1	1
01	1	1	0	0
10	1	1	0	0
11	0	0	1	1

∴ y = BC' + B'C
∴ **y = B ⊕ C**

(4) For z:

AB \ CD	00	01	11	10
00	0	1	0	1
01	0	1	0	1
10	0	1	0	1
11	0	1	0	1

∴ z = C'D + CD'
∴ **z = C ⊕ D**

[B] GRAY TO BINARY CODE CONVERTER:

Truth Table:

Gray Input				Binary Output			
A	**B**	**C**	**D**	**w**	**X**	**y**	**z**
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	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
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

Boolean Expressions:
(1) For w:

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
10	1	1	1	1
11	1	1	1	1

∴ w = A

(2) For x:

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
10	0	0	0	0
11	1	1	1	1

∴ x = A'B + AB'
∴ x = A ⊕ B

(3) For y:

AB \ CD	00	01	11	10
00	0	0	1	1
01	1	1	0	0
10	0	0	1	1
11	1	1	0	0

$\therefore$ y = A'B'C + A'BC' + ABC + AB'C'
= A' (B'C+BC') + A (BC+B'C')
= A' (B⊕C) + A (B⊕C)'
= A⊕ B ⊕ C
$\therefore$ **y = x ⊕ C**

(4) For z:

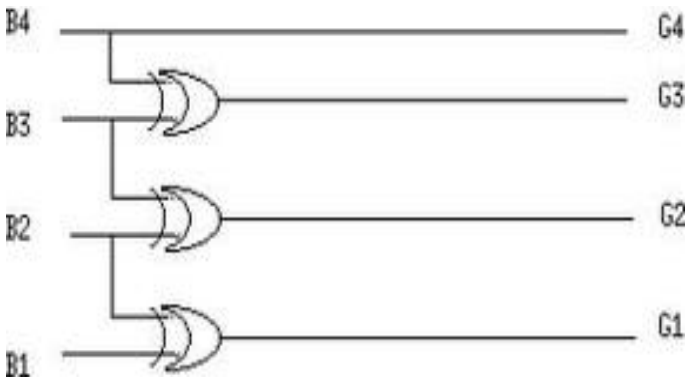
AB \ CD	00	01	11	10
00	0	1	0	1
01	1	0	1	0
10	0	1	0	1
11	1	0	1	0

$\therefore$ z = A'B'C'D+A'B'CD'+A'BC'D'+A'BCD+ABC'D+ABCD'+AB'C'D'+AB'CD
= A'B' (C⊕D) + AB(C⊕D) + A'B(C⊕D) + AB'(C⊕D)
= (C⊕D) (A⊕B') + (A⊕B) (C⊕D)'
= A ⊕ B ⊕ C ⊕ D
$\therefore$ **z = y ⊕ D**

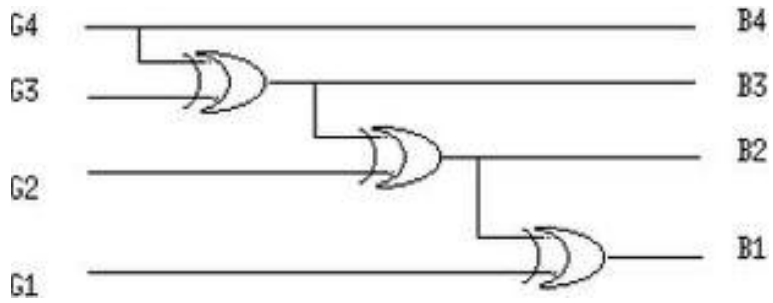
PROCEDURE:

1. Design the code converter circuits on the basis of Boolean function derived.
2. Fix the IC 7486 on Digital trainer bread board so that each pin is in different hole and no two pins are connected.
3. Apply Vcc (+5Volts) and Ground to IC using hook up wires through trainer board.
4. Connect the gates as per design formed in step-1.
5. Connect the input terminals to the supply voltage through switches and output terminals to the LEDs.
6. Verify the output by giving different combinations of inputs to designed circuit.

CONCLUSION:



Binary to Gray Code Converter Circuit



Gray to Binary Code Converter Circuit

PRACTICAL- 6

DECODERS

AIM: To study the Decoders [2 to 4 line and 3 to 8 line]

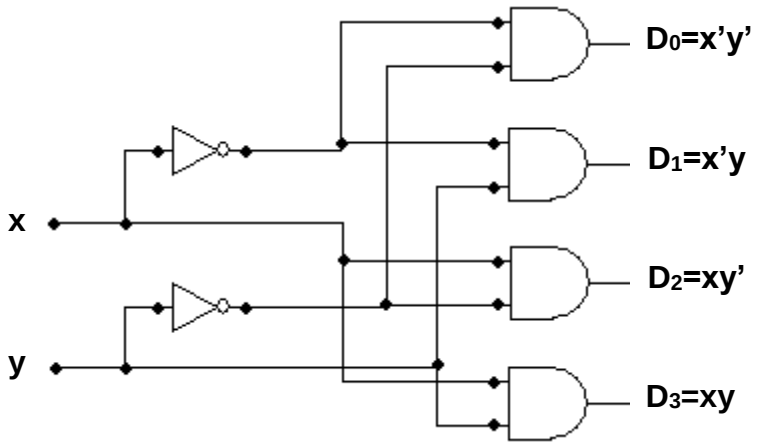
APPARATUS: [1] Digital Trainer
[2] Hook up wires
[3] IC 74138 and IC 74139

THEORY:

A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines. Their purpose is to generate the 2^n (or less) minterms of n input variables.

If the n-bit decoded information has unused or don't care combinations, the decoder output will have less than 2^n outputs. The decoders here are called n-to-m line decoders where $m \leq 2^n$. Decoders such as 2x4, 3x8, 4x16 are available. Decoders are used for implementing the Boolean functions. The name decoder is also used in conjunction with same code converters such as a BCD to Seven segment decoder.

In 2-to-4 line decoder the 2 inputs are decoded into 4 outputs, each output representing one of the minterms of the 2-input variables.



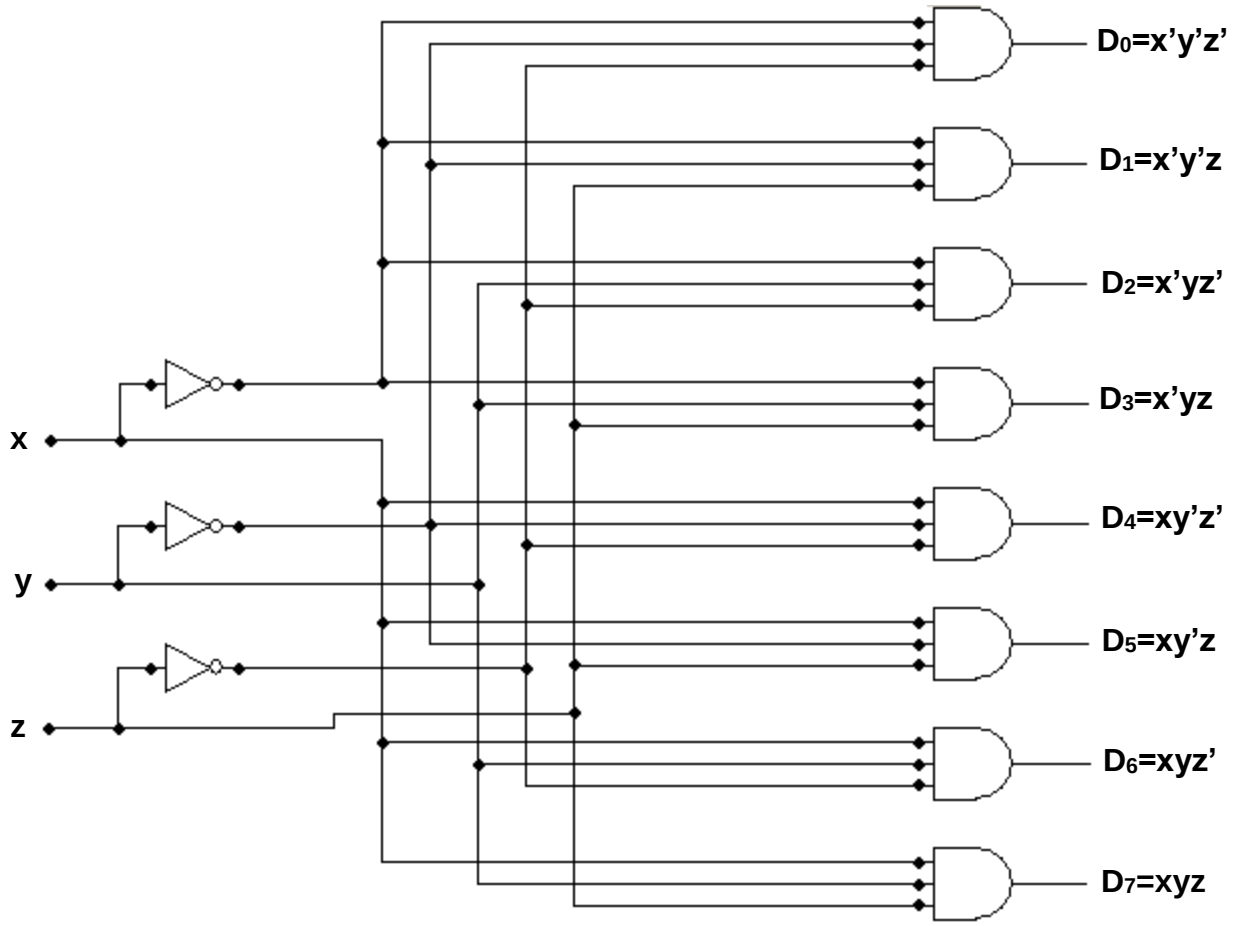
2-to-4 line Decoder Logic Diagram

Truth Table:

Inputs		Outputs			
x	y	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

If we consider the 3-to-8 line decoder, then 3 inputs are decoded into 8 outputs.

The 3-to-8 line decoder circuit is shown here. The three inverters provide the complement of the inputs and each one of the eight AND gates generate one of the minterms. It is also possible to construct the decoders by using NAND gates, since it generates the AND operation with an inverted output.



3-to-8 line Decoder Logic Diagram

The operation of the decoder is classified from its input-output relationship listed in below truth table.

Truth Table:

Inputs			Outputs							
x	**y**	**z**	**D_0**	**D_1**	**D_2**	**D_3**	**D_4**	**D_5**	**D_6**	**D_7**
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

PROCEDURE:

For 2-to-4 Decoder Circuit:

1. Fix the IC 74139 on Digital trainer breadboard so that each pin is in different hole and no two pins are connected.
2. Apply Vcc (+5Volts) and Ground to IC using hook up wires through trainer board.
3. From dual decoders in the IC, select one decoder and connect the Enable and select pins of that decoder to the switches.
4. Connect the output terminals of that decoder to the LED.
5. Observe the output and verify the Truth table by giving different combinations of inputs.

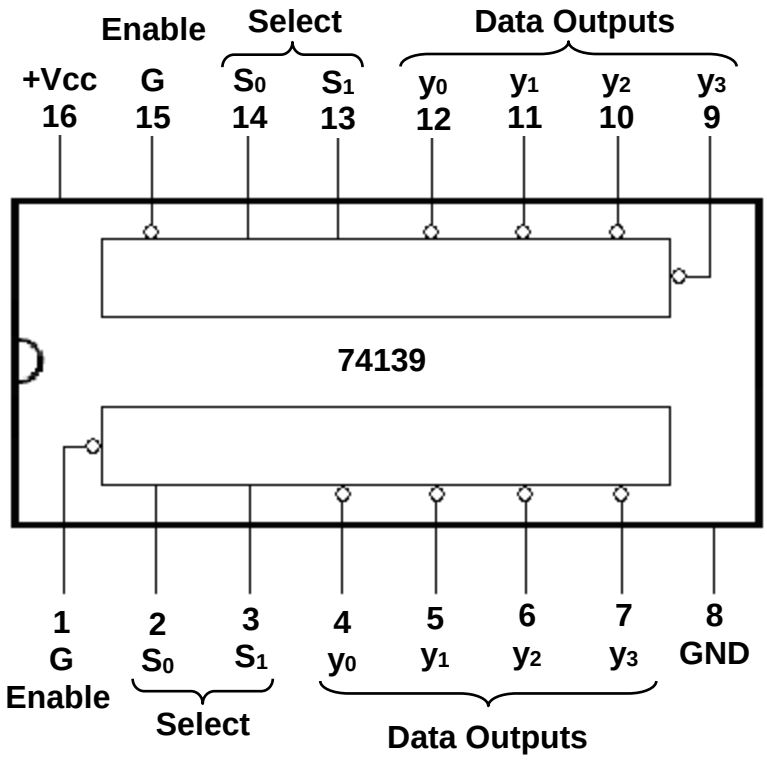
For 3-to-8 Decoder Circuit:

1. Fix the IC 74138 on Digital trainer breadboard so that each pin is in different hole and no two pins are connected.
2. Apply Vcc (+5Volts) and Ground to IC using hook up wires through trainer board.
3. Connect the Enables and select pins of the IC to the input switches.
4. Connect the output terminals to the LED.
5. Observe the output and verify the Truth table by giving different combinations of inputs.

CONCLUSION:

IC-74139 Dual 2-to-4 line Decoder

Pin-Diagram

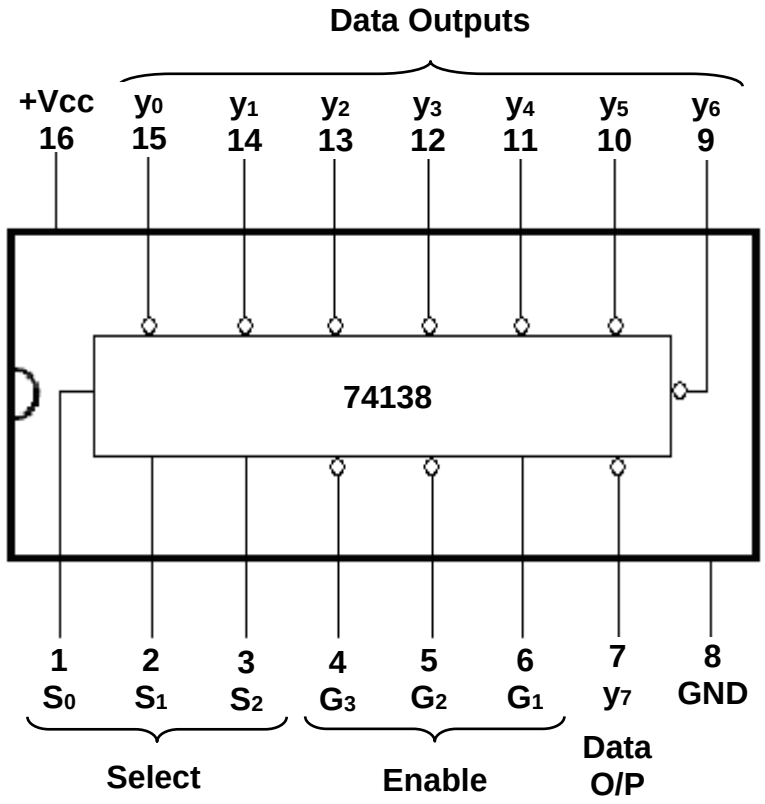


Observation Table

G	S_1	S_0	y_0	y_1	y_2	y_3
1	X	X				
0	0	0				
0	0	1				
0	1	0				
0	1	1				

IC-74138 3-to-8 line Decoder

Pin-Diagram



Observation Table

G_1	G_2	G_3	S_2	S_1	S_0	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	X	X	X	X	X								
X	1	X	X	X	X								
X	X	1	X	X	X								
1	0	0	0	0	0								
1	0	0	0	0	1								
1	0	0	0	1	0								
1	0	0	0	1	1								
1	0	0	1	0	0								
1	0	0	1	0	1								
1	0	0	1	1	0								
1	0	0	1	1	1								

PRACTICAL-7

8X1 Multiplexer and 1X8 Demultiplexer

AIM: To study the 8X1 Multiplexer and 1X8 Demultiplexer.

APPARATUS: [1] Digital Trainer
[2] Hook up wires
[3] IC 74151 and IC 74138

THEORY:

MULTIPLEXER:

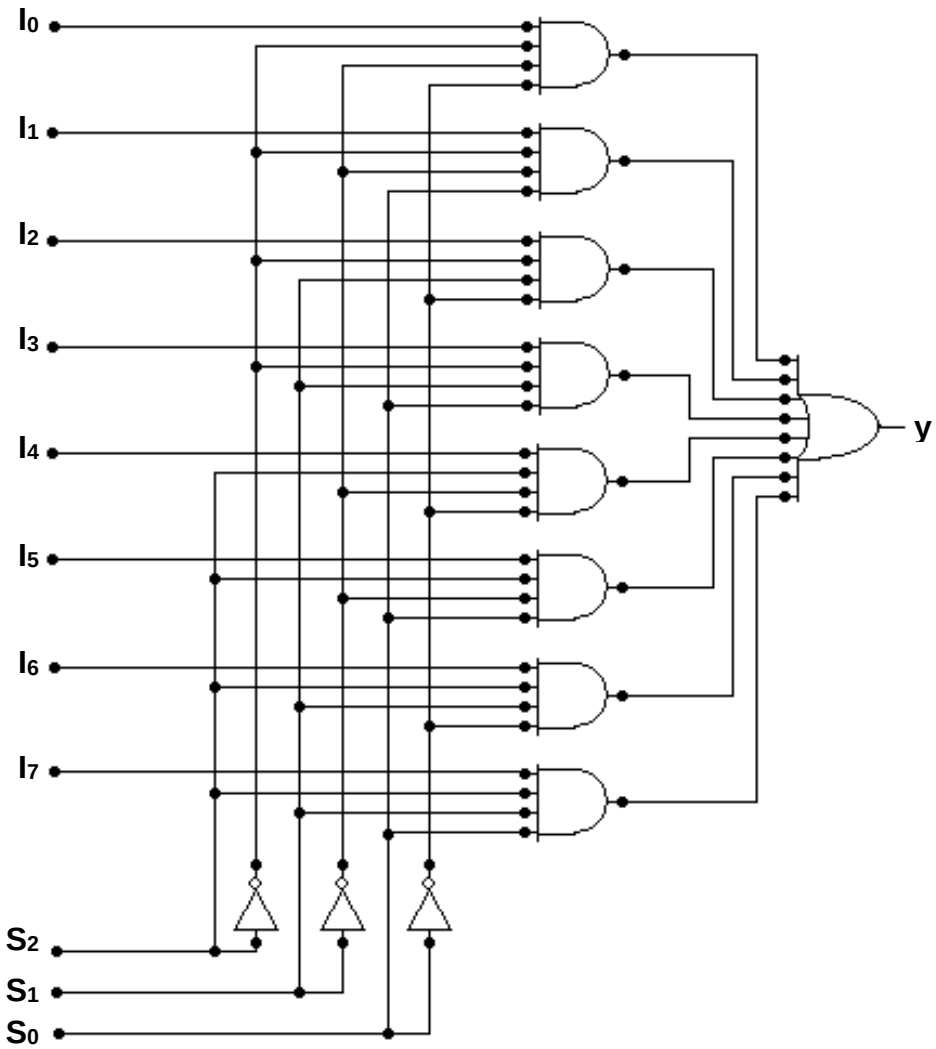
Multiplexing means transmitting a large number of information units over a smaller number of channels or lines.

A digital multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line.

The selection of a particular input line is controlled by a set of selection lines. Normally there are 2^n input lines and n selection lines whose bit combinations determine which input is selected.

A multiplexer is also called a *data selector* since it selects one of the many inputs and steers the binary information to the output line.

The size of a multiplexer is specified by the number 2^n of its input lines and the single output line. It is then implied that it also contains n selection lines. A multiplexer is often abbreviated as MUX.



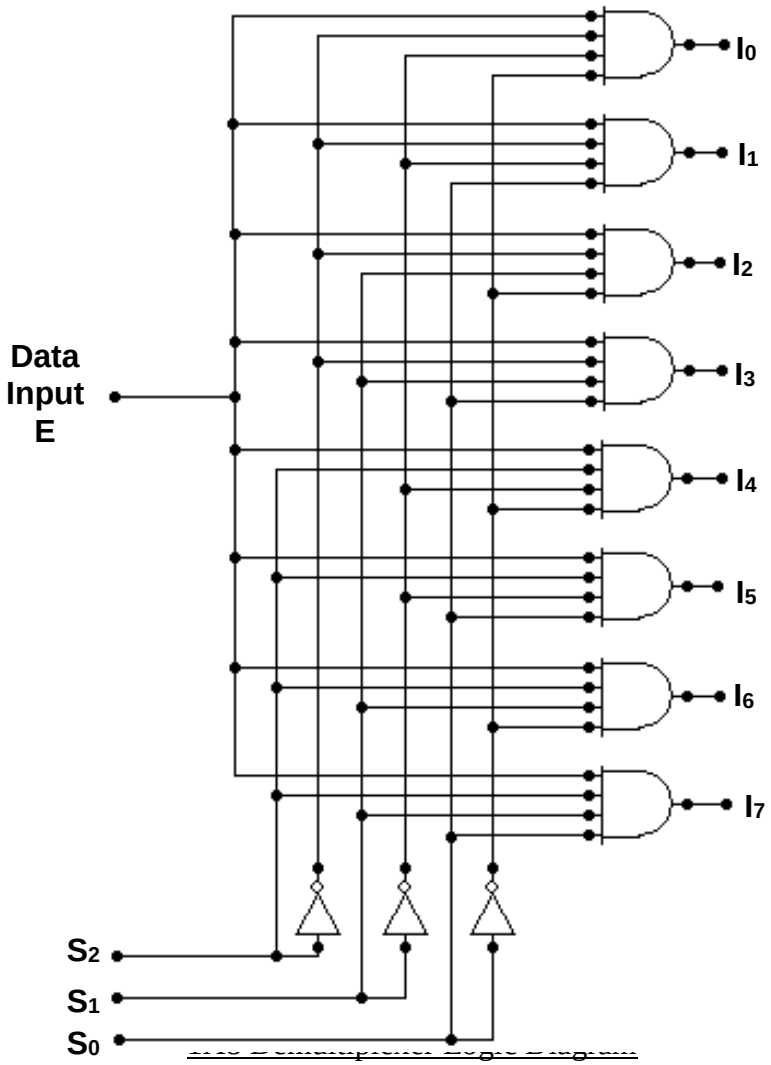
8X1 Multiplexer Logic Diagram

Truth Table:

S_2	S_1	S_0	y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

DEMULTIPLEXER:

A Demultiplexer is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines. The selection of a specific output line is controlled by the bit values of n selection lines. A decoder with an enable input can function as a Demultiplexer, if the enable line is taken as a data input line and input lines of decoder are taken as the selection lines.



4x16 Demultiplexer Logic Diagram

Truth Table:

E	S_2	S_1	S_0	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
1/0	0	0	0	1/0	0	0	0	0	0	0	0
1/0	0	0	1	0	1/0	0	0	0	0	0	0
1/0	0	1	0	0	0	1/0	0	0	0	0	0
1/0	0	1	1	0	0	0	1/0	0	0	0	0
1/0	1	0	0	0	0	0	0	1/0	0	0	0
1/0	1	0	1	0	0	0	0	0	1/0	0	0
1/0	1	1	0	0	0	0	0	0	0	1/0	0
1/0	1	1	1	0	0	0	0	0	0	0	1/0

PROCEDURE:

For 8X1 Multiplexer:

1. Fix the IC 74151 on Digital trainer breadboard so that each pin is in different hole and no two pins are connected.
2. Apply Vcc (+5Volts) and Ground to IC using hook up wires through trainer board.
3. Connect the select lines and strobe pin of the IC to the input switches.
4. Then connect the selected data input pins of IC to the switches.
5. Connect the output pins of the IC to the LEDs.
6. Observe the output and verify the Truth table by giving different inputs.

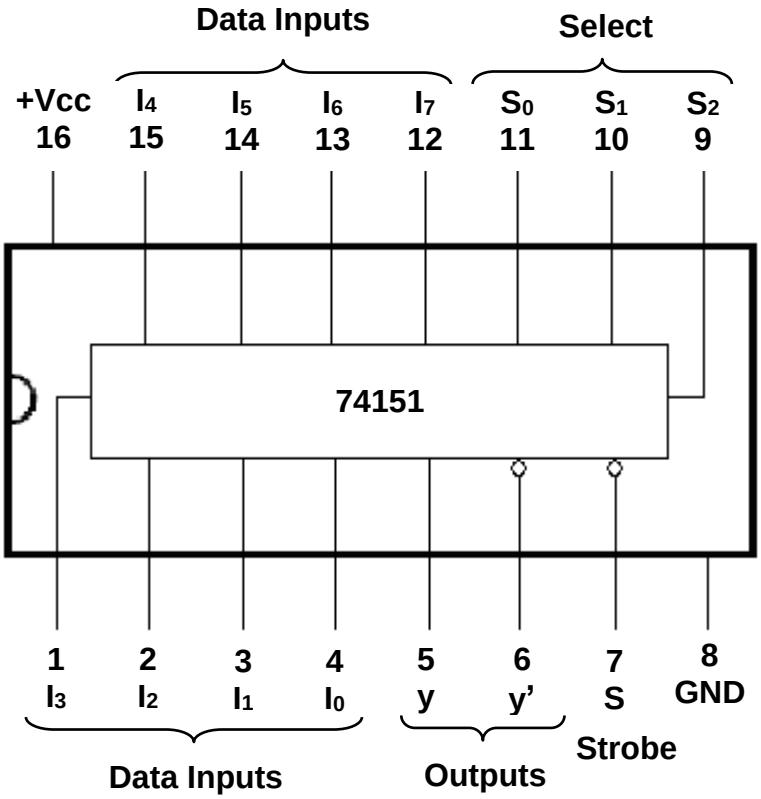
For 1X8 Demultiplexer:

1. Fix the IC 74138 on Digital trainer breadboard so that each pin is in different hole and no two pins are connected.
2. Apply Vcc (+5Volts) and Ground to IC using hook up wires through trainer board.
3. From Enables G_1, G_2 and G_3, short G_2 and G_3 and connect with input switch. Apply logic 0 to G_2 and G_3 using switch.
4. Use Enable G_1 as a data input and connects to the input switch. Also connect the select lines of the IC to the input switches.
5. Connect the output pin of the IC to the LED.
6. Observe the output and verify the Truth table by giving different inputs.

CONCLUSION:

IC-74151 8x1 Multiplexer

Pin-Diagram

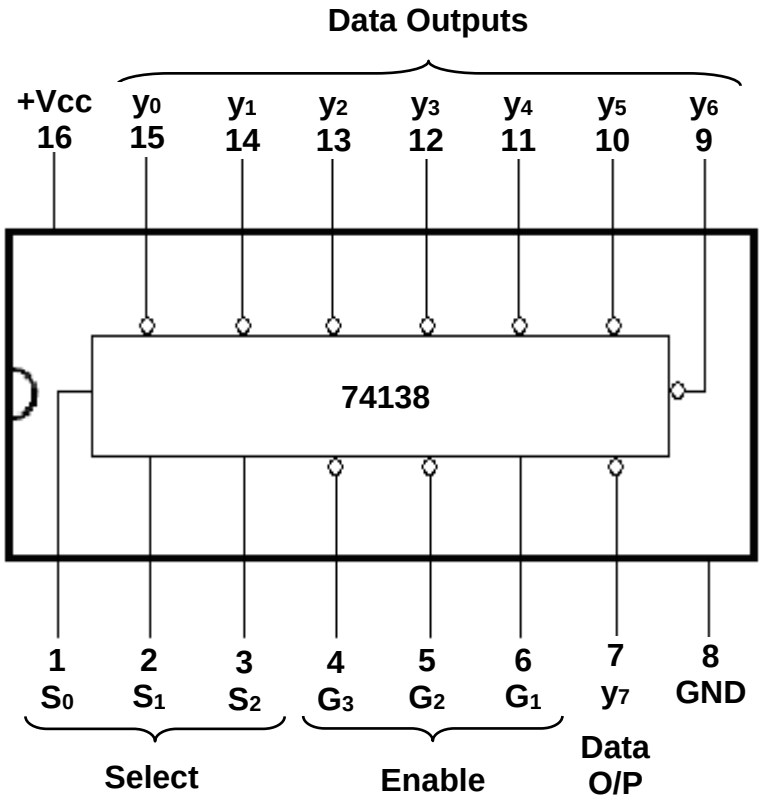


Observation Table

Strobe	Select Lines			Outputs	
S	S_2	S_1	S_0	y	y'
1	X	X	X		
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		

IC-74138 3-to-8 line Decoder

Pin-Diagram



Observation Table

G_1	S_2	S_1	S_0	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0/1	0	0	0								
0/1	0	0	1								
0/1	0	1	0								
0/1	0	1	1								
0/1	1	0	0								
0/1	1	0	1								
0/1	1	1	0								
0/1	1	1	1								

PRACTICAL-8

FLIP-FLOP

AIM: To study and verify the Different Flip-Flops. (D, JK and T Flip-Flops).

APPARATUS: [1] Digital Trainer
[2] Hook up wires
[3] IC 7474 and IC 7476

THEORY:

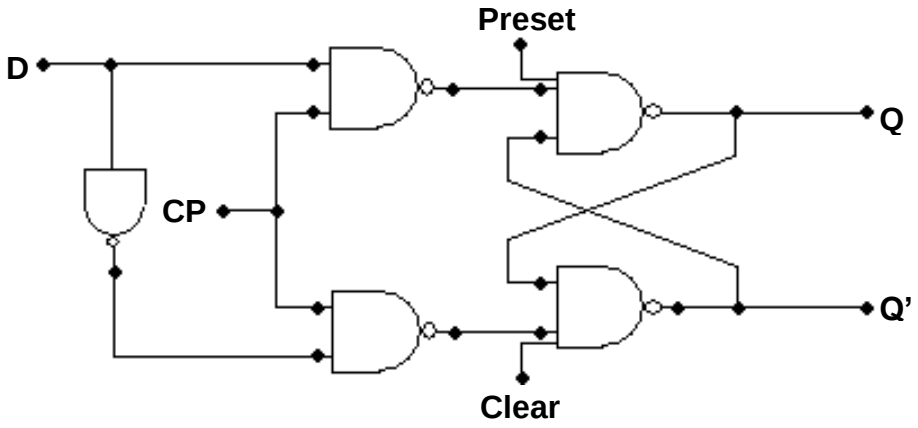
A sequential circuit consists of a combinational circuit to which memory elements are connected to form a feedback path. The memory elements are devices capable of storing binary information within them.

The memory elements used in clocked sequential circuits are called Flip-Flops. These circuits are binary cells capable of storing one bit of information. A Flip-Flop circuit has two outputs, one for the normal value and one for the complement value of the bit stored in it. The major differences among various types of Flip-flops are in the number of inputs they possess and in the manner in which the inputs affect the binary state.

[1] D Flip-Flop:

The D Flip-Flop receives the designation from its ability to transfer "data" into a Flip-Flop. The logic diagram (with NAND gates), characteristics table and symbol for a clocked D Flip-Flop are shown below:

Logic Diagram:

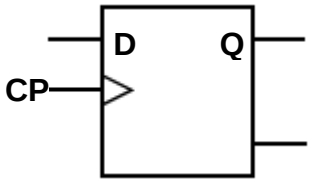


Characteristic Table:

Q(t)	D	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

Q(t+1) = D

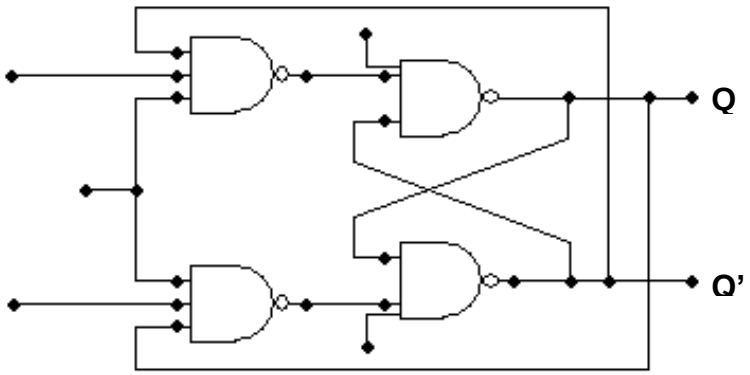
Symbol:



[2] JK Flip-Flop:

Many digital systems are constructed entirely with JK Flip-Flops because they are the most versatile available. JK Flip-Flop is used for any general application. A JK Flip-Flop is a refinement of the RS Flip-Flop in that the indeterminate state of the RS type is defined in the JK type. When J=1 and K=1, the Flip-Flop toggles i.e. goes to the opposite state. The logic diagram, characteristic table and symbol for a clocked JK Flip-Flop are shown below:

Logic Diagram:

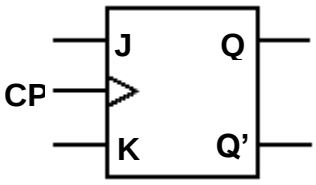


Characteristic Table:

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q'(t)

Q(t+1) = JQ'(t) + K'Q(t)

Symbol:



[3] T Flip-Flop:

The T Flip-Flop is a single input version of the JK Flip-Flop. The T Flip-Flop is obtained from a JK type if both inputs are tied together. The designation T comes from the ability of the

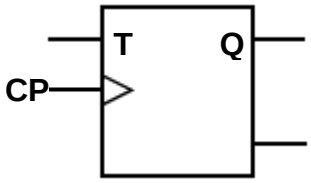
Flip-Flop to "toggle" or change state. The symbol, characteristic table and equation of the T Flip-Flop are shown here:

Characteristic Table:

Q(t)	T	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

Q(t+1) = TQ'(t) + T'Q(t)

Symbol:



PROCEDURE:

[For D Flip-Flop]

1. Apply Vcc (+5Volts) and Ground to IC using hook up wires through trainer board.
2. From the dual D Flip-flop in the IC, select one D Flip-flop and connect input D, clear and preset pins to the input switches.
3. Connect outputs of that Flip-Flop in the IC to the LED.
4. From the function generator or manually (From Trainer Board) apply clock pulse to the clock pin of the Flip-Flop.
5. Verify the table of D Flip-Flop giving different input conditions.

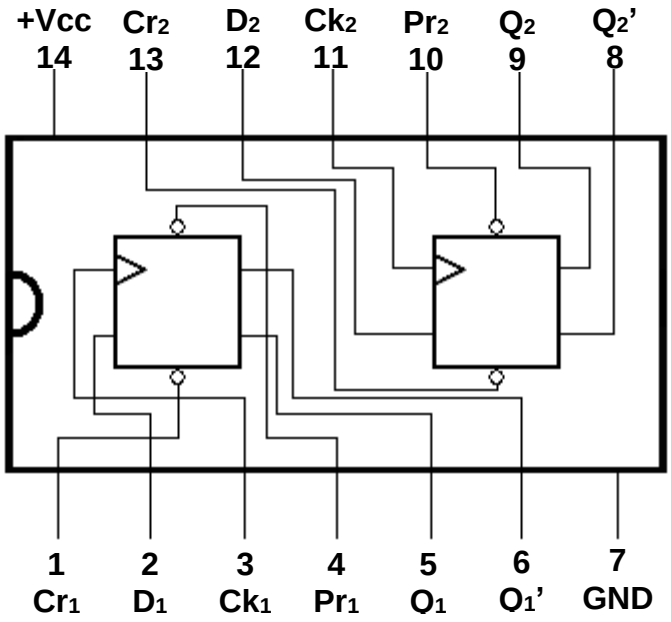
[For JK and T Flip-Flop]

1. Apply Vcc (+5Volts) and Ground to IC using hook up wires through trainer board.
2. From the dual JK Flip-flop in the IC, select one JK Flip-flop and connect inputs J, K, clear and preset pins to the input switches.
3. Connect outputs of that Flip-Flop in the IC to the LED.
4. From the function generator or manually (From Trainer Board) apply clock pulse to the clock pin of the Flip-Flop.
5. Verify the table of JK Flip-Flop giving different input conditions.
6. For T Flip-Flip, first short J and K inputs and make single input that is input T. Then the above steps and verify the table for T Flip-Flop.

CONCLUSION:

IC-7474 Dual D Flip-Flops

Pin-Diagram

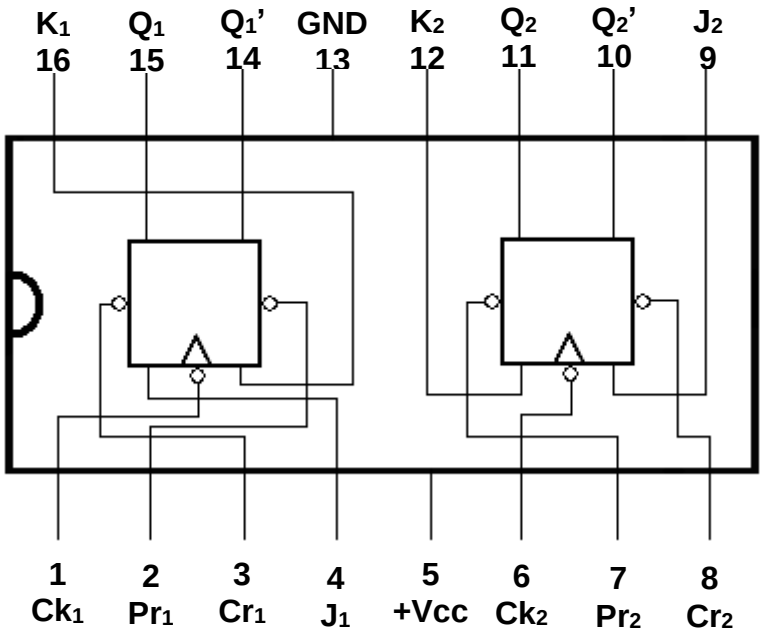


Observation Table

Ck	Cr	Pr	D	Q(t+1)
X	0	1	X	
X	1	0	X	
I	1	1	0	
I	1	1	1	

IC-7476 Dual JK Flip-Flops

Pin-Diagram



Observation Table
(For JK Flop-Flop)

Ck	Cr	Pr	J	K	Q(t+1)
X	0	1	X	X	
X	1	0	X	X	
I	1	1	0	0	
I	1	1	0	1	
I	1	1	1	0	
I	1	1	1	1	

Observation Table
(For T Flip-Flop)

Ck	Cr	Pr	T	Q(t+1)
X	0	1	X	
X	1	0	X	
I	1	1	0	
I	1	1	1	

PRACTICAL-9

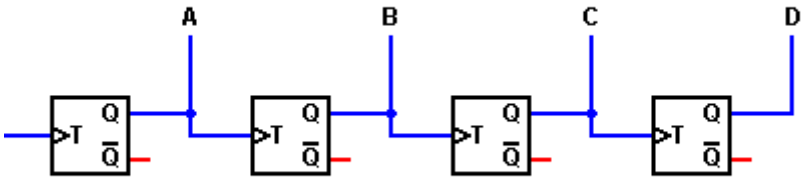
DECADE COUNTER

AIM: To study the function of Decade counter

THEORY:

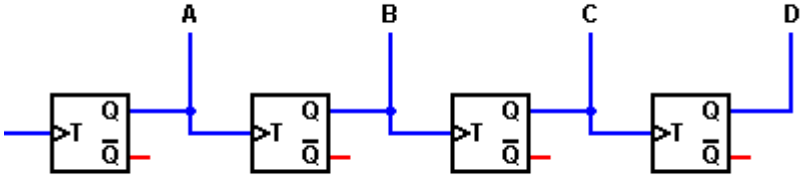
One common requirement in digital circuits is *counting*, both forward and backward. Digital clocks and watches are everywhere, timers are found in a range of appliances from microwave ovens to VCRs, and counters for other reasons are found in everything from automobiles to test equipment.

Although we will see many variations on the basic counter, they are all fundamentally very similar. The demonstration below shows the most basic kind of binary counting circuit.



In the 4-bit counter to the right, we are using edge-triggered master-slave flip-flops similar to those in the Sequential portion of these pages. The output of each flip-flop changes state on the falling edge (1-to-0 transition) of the T input.

The count held by this counter is read in the reverse order from the order in which the flip-flops are triggered. Thus, output D is the high order of the count, while output A is the low order. The binary count held by the counter is then DCBA, and runs from 0000 (decimal 0) to 1111 (decimal 15). The next clock pulse will cause the counter to try to increment to 10000 (decimal 16). However, that 1 bit is not held by any flip-flop and is therefore lost. As a result, the counter actually reverts to 0000, and the count begins again.



In future pages on counters, we will use a different input scheme, as shown to the left. Instead of changing the state of the input clock with each click, you will send one complete clock pulse to the counter when you click the input button. The button image will reflect the

state of the clock pulse, and the counter image will be updated at the end of the pulse. To provide a clear view without taking excessive time, each clock pulse has a duration or pulse width of 300 ms (0.3 second). The demonstration system will ignore any clicks that occur within the duration of the pulse

A major problem with the counters shown on this page is that the individual flip-flops do not all change state at the same time. Rather, each flip-flop is used to trigger the next one in the series. Thus, in switching from all 1s (count = 15) to all 0s (count wraps back to 0), we don't see a smooth transition. Instead, output A falls first, changing the apparent count to 14. This triggers output B to fall, changing the apparent count to 12. This in turn triggers output C, which leaves a count of 8 while triggering output D to fall. This last action finally leaves us with the correct output count of zero. We say that the change of state "ripples" through the counter from one flip-flop to the next. Therefore, this circuit is known as a "ripple counter."

This causes no problem if the output is only to be read by human eyes; the ripple effect is too fast for us to see it. However, if the count is to be used as a selector by other digital circuits (such as a multiplexer or demultiplexer), the ripple effect can easily allow signals to get mixed together in an undesirable fashion. To prevent this, we need to devise a method of causing all of the flip-flops to change state at the same moment. That would be known as a "synchronous counter" because the flip-flops would be synchronized to operate in unison. That is the subject of the next page in this series.

DM7490A: Decade and Binary Counter

General Description

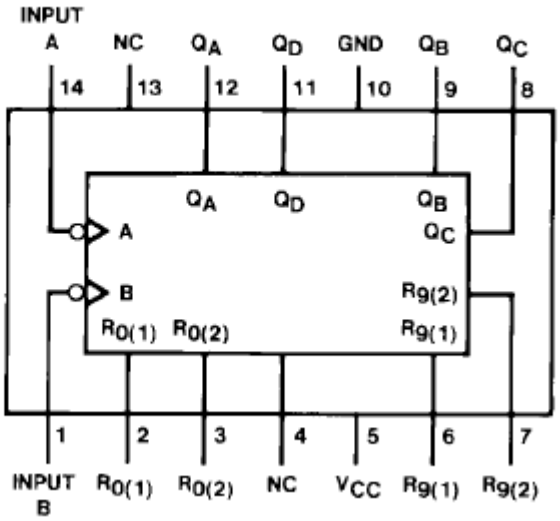
The DM7490A monolithic counter contains four master slave flip-flops and additional gating to provide a divide-by two counter and a three-stage binary counter for which the count cycle length is divide-by-five. The counter has a gated zero reset and also has gated set-to-nine inputs for use in BCD nine’s complement applications. To use the maximum count length (decade or four-bit binary), the B input is connected to the QA output. The input count pulses are applied to input A and the outputs are as described in the appropriate Function Table. A symmetrical divide-by-ten count can be obtained from the counters by connecting the QD output to the A input and applying the input count to the B input which gives a divide-by-ten square wave at output QA.

Features

[1] Typical power dissipation 145 mW
[2] Count frequency 42 MHz

CONCLUSION:

Connection Diagram



Function Tables

BCD Count Sequence (Note 1)

Count	Outputs			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H

BCD Bi-Quinary (5-2) (Note 2)

Count	Outputs			
	Q_A	Q_D	Q_C	Q_B
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	H	L	L	L
6	H	L	L	H
7	H	L	H	L
8	H	L	H	H
9	H	H	L	L

Reset/Count Function Table

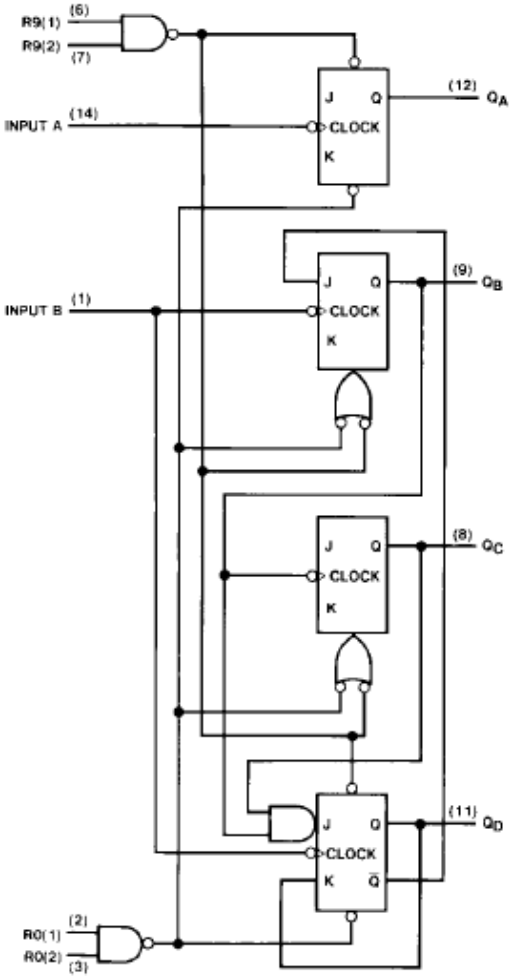
Reset Inputs				Outputs			
R0(1)	R0(2)	R9(1)	R9(2)	Q_D	Q_C	Q_B	Q_A
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
X	L	X	L	COUNT			
L	X	L	X	COUNT			
L	X	X	L	COUNT			
X	L	L	X	COUNT			

H = HIGH Level
L = LOW Level
X = Don't Care

Note 1: Output QA is connected to input B for BCD count.

Note 2: Output QD is connected to input A for bi-quinary count

Logic Diagram



The J and K inputs shown without connection are for reference only and are functionally at a HIGH level.

PRACTICAL-10

SHIFT REGISTER

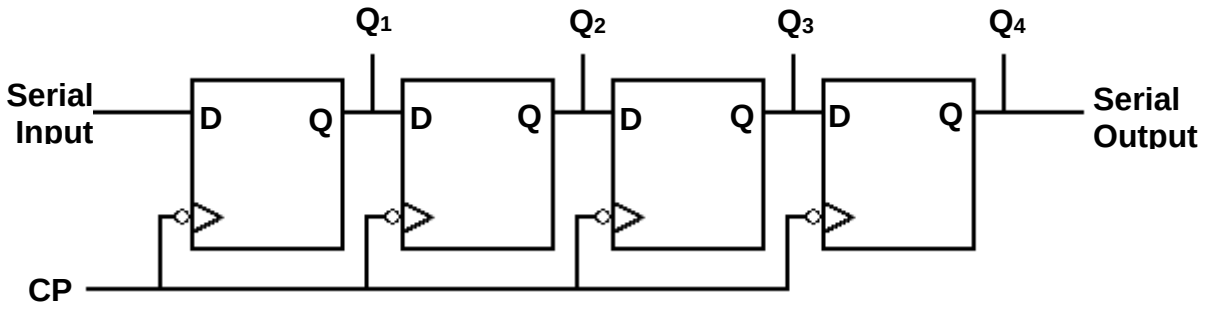
AIM: To study the Shift Register.

THEORY:

A register is a group of binary cells suitable for holding binary information. A group of Flip-Flops constitutes a register. Since each Flip-Flop is a binary cell capable of storing one bit of information. An n-bit register has a group of n Flip-Flops and is capable of storing any binary information containing n bits.

A register capable of shifting its binary information either to the right or to the left is called a shift register. The logical configuration of a shift register consists of a chain of Flip-Flops connected in cascade, with the output of one Flip-Flop connected to the input of the next Flip-Flop. All Flip-Flops receive a common clock pulse that causes the shift from one stage to the next.

The simplest possible shift register is one that uses only Flip-Flops as shown below:



Shift Register Logic Diagram

Above register shifts its contents to the right. A register that can shift in only one direction is called a unidirectional shift register and a register capable of shifting both right and left is called a bi-directional shift register.

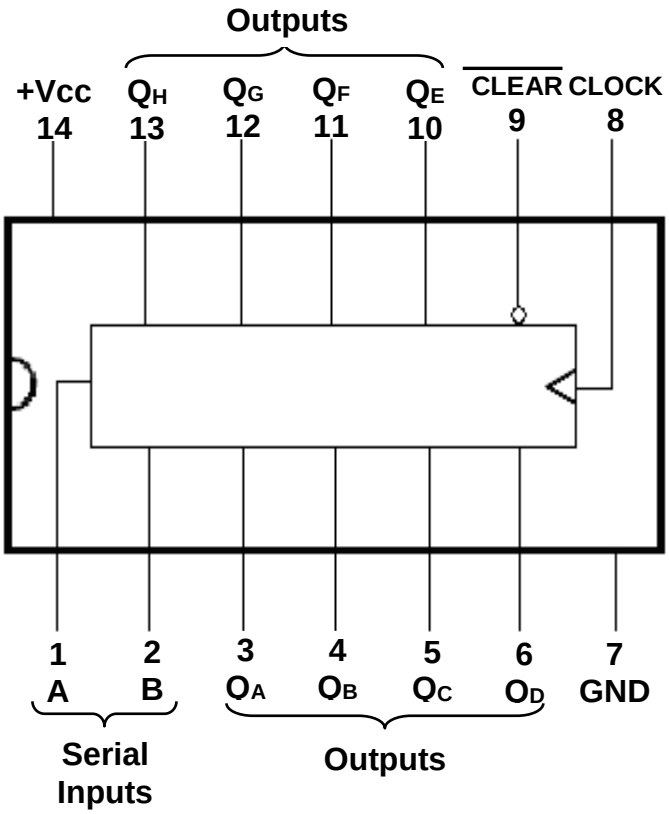
The transfer of new information into a register is referred to as loading the register. If all the bits of the register are loaded simultaneously with a single clock pulse, we say that the loading is done in parallel. If a parallel-load capability is added to a shift register, then data entered in parallel can be taken out in serial fashion by shifting the data stored in the register.

If the register has both shift and parallel-load capabilities, it is called a shift register with parallel load or universal shift register.

CONCLUSION:

IC-74164 8-bit Parallel Output Serial Shift Register

Pin-Diagram



Observation Table

CLEAR	CLOCK	A	B	Q_A Q_B ….. Q_H
0	X	X	X	0 0 …… 0
1	I	L	X	L Q_A ……Q_G
1	I	X	L	L Q_A ……Q_G
1	I	H	H	H Q_A ……Q_G

PRACTICAL -11

OVERVIEW OF MICRO PROCESSOR

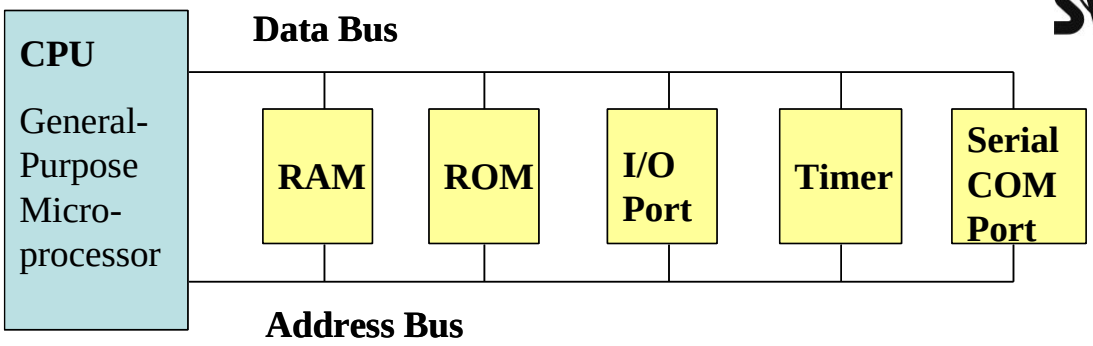
AIM- To Study various types of microprocessors

THEORY-A microprocessor incorporates most or all of the functions of a computer's central processing unit (CPU) on a single integrated circuit (IC, or microchip). The first microprocessors emerged in the early 1970s and were used for electronic calculators, using binary-coded decimal (BCD) arithmetic in 4-bit words. Other embedded uses of 4-bit and 8-bit microprocessors, such as terminals, printers, various kinds of automation etc. Affordable 8-bit microprocessors with 16-bit addressing also led to the first general-purpose microcomputers from the mid-1970s on.

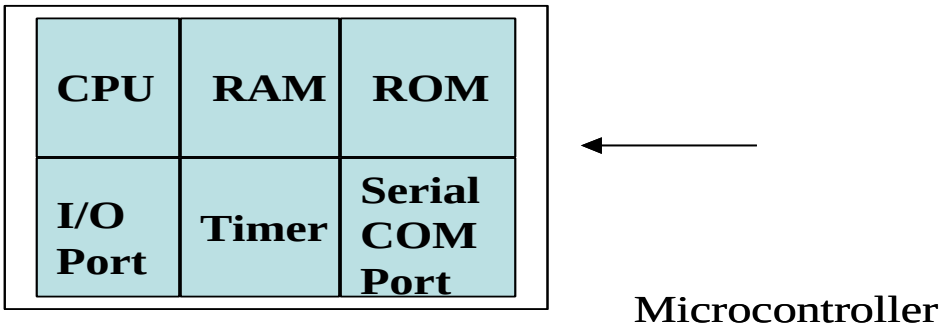
Microcontroller is a programmable digital processor with necessary peripherals. Both microcontrollers and microprocessors are complex sequential digital circuits meant to carry out job according to the program / instructions. Sometimes analog input/output interface makes a part of microcontroller circuit of mixed mode(both analog and digital nature). A microcontroller can be compared to a Swiss knife with multiple functions incorporated in the same IC

Microcontrollers Vs Microprocessors

1. A microprocessor requires an external memory for program/data storage. Instruction execution requires movement of data from the external memory to the microprocessor or vice versa. Usually, microprocessors have good computing power and they have higher clock speed to facilitate faster computation.
2. A microcontroller has required on-chip memory with associated peripherals. A microcontroller can be thought of a microprocessor with inbuilt peripherals.
3. A microcontroller does not require much additional interfacing ICs for operation and it functions as a standalone system. The operation of a microcontroller is multipurpose, just like a Swiss knife.
4. Microcontrollers are also called embedded controllers. A microcontroller clock speed is limited only to a few tens of MHz. Microcontrollers are numerous and many of them are application specific.



General-Purpose Microprocessor System



Microcontroller

AVR Processors: The **AVR** is a modified Harvard architecture 8-bit RISC single chip microcontroller which was developed by Atmel in 1996. The AVR was one of the first microcontroller families to use on-chip flash memory for program storage, as opposed to One-Time Programmable ROM, EPROM, or EEPROM used by other microcontrollers at the time.

ARM Processor: The **ARM** is a 32-bit reduced instruction set computer (RISC) instruction set architecture (ISA) developed by ARM Holdings. It was known as the **Advanced RISC Machine**, and before that as the **Acorn RISC Machine**. The ARM architecture is the most widely used 32-bit ISA in terms of numbers produced.[1][2] They were originally conceived as a processor for desktop personal computers by Acorn Computers, a market now dominated by the x86 family used by IBM PC compatible computers. The relative simplicity of ARM processors made them suitable for low power applications. This has made them dominant in the mobile and embedded electronics market as relatively low cost and small microprocessors and microcontrollers.

PIC Controller

PIC is a family of Harvard architecture microcontrollers made by Microchip Technology, derived from the PIC1640[1] originally developed by General Instrument's Microelectronics Division. The name PIC initially referred to "**Programmable Interface Controller**".

PICs are popular with both industrial developers and hobbyists alike due to their low cost, wide availability, large user base, extensive collection of application notes, availability of low cost or free development tools, and serial programming (and re-programming with flash memory) capability.

CONCLUSION: