

Module 5

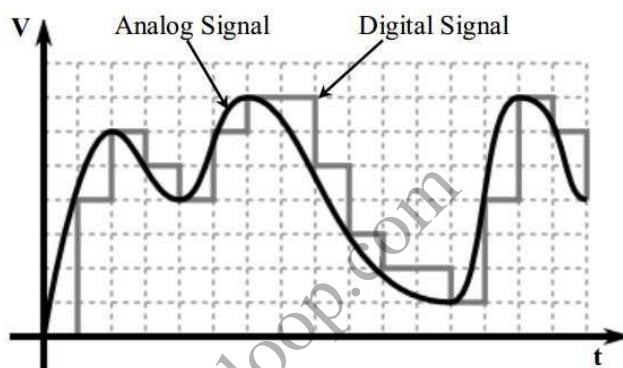
Digital Electronics Fundamentals :

Difference between analog and digital signals, Number Systems-Binary, Hexadecimal, Conversion - Decimal to Binary, Hexadecimal to Decimal and vice versa. Boolean Algebra, Basic and Universal gates, Half adder and Full adder, Multiplexer, Decoder, SR and JK flip flops, Shift register, 3-bit Ripple Counters.

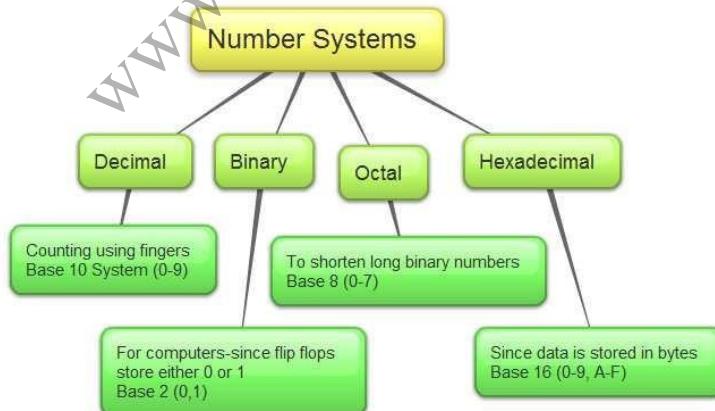
Basic Communication system, Principle of operations of mobile phone.

Difference between analog and digital signals

- **Analog or Continuous Signal:** The signal may acquire any value in a range of the independent variable (time). Example: Analog Signal
- **Discrete or Digital Signal:** The signal can have any value but it would remain constant over periods of time called sampling period. Digital signals or numbers are processed by digital system using the concept of binary numbers and Boolean algebra.

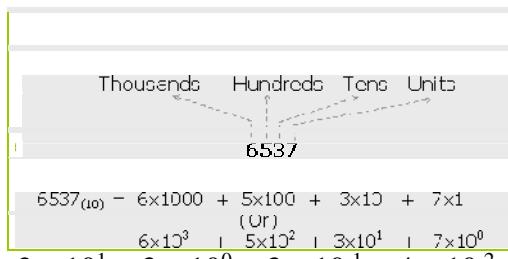


Number Systems: There are four number systems that are used in the digital systems.



1. **Decimal Number System:** It is the most commonly used numbering system in day-to-day life to count, measure and label. Combination of ten digits from 0 to 9 are used to represent any number. In this system the next position to the left from the decimal point represents units, tens, hundreds, thousands etc. and the next position to the right after the decimal point represents $(\frac{1}{10} \text{ or } 10^{-1})$'s, $(\frac{1}{100} \text{ or } 10^{-2})$'s, $(\frac{1}{1000} \text{ or } 10^{-3})$'s, etc.

Digits are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and **Base value:** 10



Example: $123.34 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 3 \times 10^{-1} + 4 \times 10^{-2} = 100 + 20 + 3 + 0.3 + 0.04 = 123.43$

2. **Binary Number System:** In binary numbering systems only two digits 0 and 1 are used to represent any number. It will go like 0, 1, 10, 11, 100, 101, 110, 111, 1000 and so on. It is the numbering system used in computers. In this system the next position to the left from the binary point represents units, 2's, 4's, 8's etc. and the next position to the right after the binary point represents $\left(\frac{1}{2} \text{ or } 2^{-1}\right)$'s, $\left(\frac{1}{4} \text{ or } 2^{-2}\right)$'s, $\left(\frac{1}{8} \text{ or } 2^{-3}\right)$'s etc.

Digits are: 0, 1 and **Base value:** 2

Example: $1010.101 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$
 $= 8 + 0 + 2 + 0 + 0.5 + 0 + 0.125 = 10.625$

3. **Octal Number System:** In octal numbering system combination of eight digits from 0 to 7 are used to represent a number. It will go from 0...7, a two digit sequence is from 10..77 and a three digit sequence is from 100...777 and so on. It is used to shorten the binary numbers.

Digits are: 0, 1, 2, 3, 4, 5, 6, 7 and **Base value:** 8

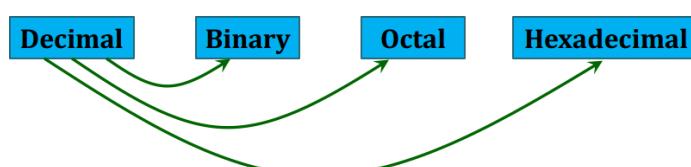
Example: $376 = 3 \times 8^2 + 7 \times 8^1 + 6 \times 8^0 = 192 + 56 + 6 = 254$

4. **Hexadecimal Number System:** In hexadecimal numbering system combination of 16 digits from 0 to 9 and A to F are used to represent a number. It will go from 0...F, a two digit sequence is from 10...FF and a three digit sequence from 100...FFF and so on. It is used to represent data's and memory addresses.

Digits are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F and **Base value:** 16

Example: $2A3B = 2 \times 16^3 + 10 \times 16^2 + 3 \times 16^1 + 11 \times 16^0 = 8192 + 2560 + 48 + 11 = 10811$

Conversions:

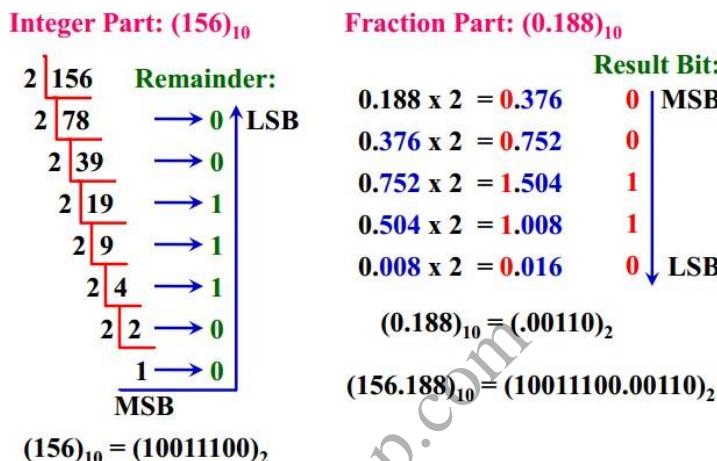


Decimal to Binary:

Integer Part: To convert the decimal integer to binary, divide the number by 2 and the successive quotients by 2. The successive remainders (which can be only 0 or 1) written in reverse order form the equivalent binary number.

Fractional Part: To convert decimal fractions to binary, multiply the number by 2 repeatedly until the fraction product is 0 (or until the desired number of binary places). The whole digits of the multiplication results produce the answer, with first as MSB and the last as LSB.

Example 1: Convert 156.188 decimal to binary.



Example 2: Convert 12.125 decimal to binary.

$$\begin{aligned}
 (12)_{10} &= (?)_2 \\
 12 \div 2 &\Rightarrow Q = 6, R = 0 \\
 6 \div 2 &\Rightarrow Q = 3, R = 0 \\
 3 \div 2 &\Rightarrow Q = 1, R = 1 \\
 (12)_{10} &= (1100)_2
 \end{aligned}$$

$$\begin{aligned}
 0.125 \times 2 &= 0.25 \Rightarrow \text{Carry 0} \\
 0.25 \times 2 &= 0.5 \Rightarrow \text{Carry 0} \\
 0.5 \times 2 &= 1.0 \Rightarrow \text{Carry 1} \\
 (0.122)_{10} &= (0.001)_2
 \end{aligned}$$

Therefore, $(12.125)_{10} = (1100.001)_2$

Assignment Problems:

- Convert the following decimal number to binary up to four binary places.
 - $(47.8125)_{10} = (?)_2$
 - $(100.974)_{10} = (?)_2$
 - $(29.3749)_{10} = (?)_2$
 - $(105.202)_{10} = (?)_2$
 - $(1024.625)_{10} = (?)_2$
 - $(555)_{10} = (?)_2$
 - $(0.825)_{10} = (?)_2$

Binary to Decimal Conversion:

- To convert the binary number to its equivalent decimal, multiply the binary digits by its corresponding weights, and then add them.

Weights from the left of binary point are: $2^0, 2^1, 2^2, 2^3, 2^4$ etc

Weights from the right of binary point are: 2^{-1} , 2^{-2} , 2^{-3} , 2^{-4} , 2^{-5} etc

Example: Convert the binary decimal

Integer Part: $(11011001)_2$

Binary Number: 1 1 0 1 1 0 0 1 ;Binary Number

Place Value: 2⁷ 2⁶ 2⁵ 2⁴ 2³ 2² 2¹ 2⁰ ;Place Value

1×2^0	=	1
0×2^1	=	0
0×2^2	=	0
1×2^3	=	8
1×2^4	=	16
0×2^5	=	0
1×2^6	=	64
1×2^7	=	128

$(11011001)_2 = (217)_{10}$

Fraction Part: (.1011),

Binary Number: 1 0 1 1 ;Place Value: $2^{-1} \ 2^{-2} \ 2^{-3} \ 2^{-4}$

Decimal Calculation:

- $1 \times 2^{-1} = 0.5$
- $0 \times 2^{-2} = 0$
- $1 \times 2^{-3} = 0.125$
- $1 \times 2^{-4} = 0.0625$

Sum: $0.5 + 0 + 0.125 + 0.0625 = 0.625$

$$(.1011)_2 = (.6875)_{10} \quad 0.6875$$

$$(11011001.1011)_2 = (217.6875)_{10}$$

Example: Convert $(1100.001)_2$ to decimal

$$(1100.001)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 8 + 4 + \frac{1}{8} = (12.125)_{10}$$

Assignment:

- Convert the following binary number to equivalent decimal number.

$$(a) (1101)_2 = (?)_{10}$$

$$(b) (10101)_2 = (\)_{10}$$

$$(c) (11001.011)_2 = (?)_{10}$$

$$(d) (11001.110)_2 = (?)_{10}$$

Hexadecimal to Decimal Conversion:

- To convert the hexadecimal number to its equivalent decimal, multiply the hexadecimal digits by its corresponding weights, and then add them.

Weights from the left of hexadecimal point are: $16^0, 16^1, 16^2, 16^3, 16^4$ etc

Weights from the right of hexadecimal point are: $16^{-1}, 16^{-2}, 16^{-3}, 16^{-4}, 16^{-5}$ etc

Example: Convert the octal number $(A37E.5C2)_{16}$ to decimal.

Integer Part: $(A37E)_{16}$

A	3	7	E	;	Hexadecimal Number
16^3	16^2	16^1	16^0	;	Place Value
					$14 \times 16^0 = 14$
					$7 \times 16^1 = 112$
					$3 \times 16^2 = 768$
					$10 \times 16^3 = 40960$

$$(A37E)_{16} = (41854)_{10}$$

Fraction Part: $(.5C2)_{16}$

5	C	2	;	Octal Number
16^{-1}	16^{-2}	16^{-3}	;	Place Value
				$2 \times 16^{-3} = 0.00049$
				$12 \times 16^{-2} = 0.04687$
				$5 \times 16^{-1} = 0.31250$

$$(.5C2)_{16} = (.35986)_{10}$$

$$(A37E.5C2)_{16} = (41854.35986)_{10}$$

A	3	7	E	.	5	C	2	;	Octal Number	
16^3	16^2	16^1	16^0	.	16^{-1}	16^{-2}	16^{-3}	;	Place Value	
\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow			
10×16^3	3×16^2	7×16^1	14×16^0	5×16^{-1}	12×16^{-2}	2×16^{-3}				
$40960 + 768 + 112 + 14 + 0.3125 + 0.04687 + 0.00049$										
$= 41854.35986$										

$$(A37E.5C2)_{16} = (41854.35986)_{10}$$

Example: Convert the hexadecimal number $(3A1.4)_{16}$ to decimal.

$$(3A1.4)_{16} = 3 \times 16^2 + 10 \times 16^1 + 1 \times 16^0 + 4 \times 16^{-1} = 768 + 160 + 1 + \frac{4}{16} = (929.25)_{10}$$

Assignment:

- Convert the following hexadecimal number to equivalent decimal number.
 - $(AF)_{16} = ()_{10}$
 - $(B6A)_{16} = ()_{10}$
 - $(ABC.CD)_{16} = ()_{10}$
 - $(9FC1)_{16} = ()_{10}$

Decimal to Hexadecimal Conversion:

Integer Part: To convert a decimal integer to hexadecimal, divide the number by 16 and the successive quotients by 16. The successive remainders (which can be only 0 to 9 or A to F) written in reverse order form the equivalent octal number.

Fractional Part: To convert decimal fractions to octal, multiply the number by 8 repeatedly until the fraction product is 0 (or until the desired number of octal places). The whole digits of the multiplication results produce the answer, with first as MSD and the last as LSD.

Example: Convert $(10766.342)_{10}$ to hexadecimal.

Integer Part: $(10767)_{10}$	Fraction Part: $(0.342)_{10}$	Result Digit:
$\begin{array}{r} 16 \mid 10767 \\ 16 \quad 672 \\ 16 \quad 42 \\ 2 \end{array}$ <p style="margin-left: 100px;">Remainder:</p> <p style="margin-left: 100px;">→ 15=F → 0 → 10=A</p> <p style="margin-left: 100px;">LSD</p> <p style="margin-left: 100px;">MSD</p> <p>$(10767)_{10} = (2A0F)_{16}$</p>	$\begin{array}{l} 0.342 \times 16 = 5.472 \quad 5 \\ 0.472 \times 16 = 7.552 \quad 7 \\ 0.552 \times 16 = 8.832 \quad 8 \\ 0.832 \times 16 = 13.312 \quad 13=D \\ 0.312 \times 16 = 4.992 \quad 4 \end{array}$ <p style="margin-left: 100px;">↓ LSD</p> <p>$(0.342)_{10} = (.578D4)_{16}$</p>	<p style="color: green;">MSD</p> <p style="color: blue;">↓ LSD</p>
$(10767.342)_{10} = (2A0F.578D4)_{16}$		

Example: Convert $(5386.345)_{10}$ to hexadecimal.

$$5386 \div 16 \Rightarrow Q=336, R=10 \text{ (A)}$$

$$336 \div 16 \Rightarrow Q=21, R=0$$

$$21 \div 16 \Rightarrow Q=1, R=5$$

$$(5386)_{10} = (150A)_{16}$$

$$0.345 \times 16 = 5.52 \Rightarrow \text{Carry } 5$$

$$0.52 \times 16 = 8.32 \Rightarrow \text{Carry } 8$$

$$0.32 \times 16 = 5.12 \Rightarrow \text{Carry } 5$$

$$0.12 \times 16 = 1.92 \Rightarrow \text{Carry } 1$$

$$(0.345)_{10} = (0.5851)_{16}$$

$$\text{Therefore, } (5386.345)_{10} = (150A.5851)_{16}$$

Assignment Problems:

3. Convert the following decimal number to hexadecimal.

(a) $(57345)_{10} = ()_{16}$

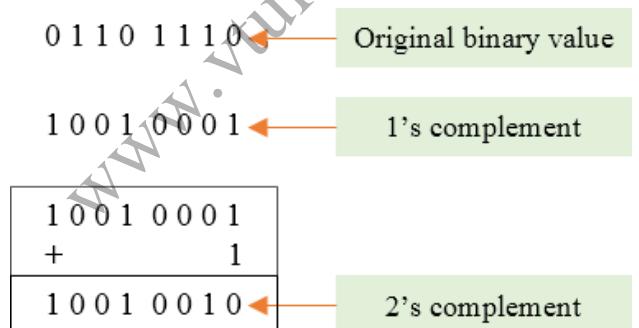
(b) $(342.56)_{10} = ()_{16}$

Relationship between Decimal, Binary, Octal, Hexadecimal number:

Hexadecimal	Decimal	Octal	Binary
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	8	10	1000
9	9	11	1001
A	10	12	1010
B	11	13	1011
C	12	14	1100
D	13	15	1101
E	14	16	1110
F	15	17	1111

Complement of Binary Numbers:

- There are two types of complements for binary numbers and are used to perform subtraction using addition.
 - 1's complement: Obtained by changing all 0's to 1's and all 1's to 0's.
 - 2's complement: Obtained by adding 1 to 1's complement of a number.



Binary Addition:

The rules of binary addition are:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ (with carry 1)}$$

$$\begin{array}{r}
 & 1 & 1 & 1 & 1 & \leftarrow \text{carry} \\
 & 1 & 1 & 1 & 0 & 1 \\
 (+) & 1 & 1 & 0 & 1 & 1 \\
 \hline
 & 1 & 1 & 0 & 0 & 0
 \end{array}$$

Circuit Globe

Example 1: Perform addition of $(11001100)_2$ and $(11011010)_2$

$$\begin{array}{r}
 & 1 & 1 & 1 & \leftarrow \text{Carry} \\
 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
 \hline
 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0
 \end{array}$$

Example 2: Add $(28)_{10}$ and $(15)_{10}$ by converting them into binary.

$$\begin{array}{r} & 1 & 1 & \leftarrow \text{Carry} \\ (28)_{10} = & 1 & 1 & 1 & 0 & 0 \\ (15)_{10} = & 0 & 1 & 1 & 1 & 1 \\ \hline & 1 & 0 & 1 & 0 & 1 & 1 \end{array}$$

Binary Subtraction:

The rules of binary subtraction are:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$0 - 1 = 1$ (Taking barrow from outside)

Examples:

$$\begin{array}{r} 16 = 1 \ 0 \ 0 \ 0 \ 0 \\ - 3 = 0 \ 0 \ 0 \ 1 \ 1 \\ \hline 13 \quad 0 \ 1 \ 1 \ 0 \ 1 \end{array} \quad \begin{array}{r} 6.25 = 1 \ 1 \ 0 . 0 \ 1 \\ - 4.50 = 1 \ 0 \ 0 . 1 \ 0 \\ \hline 1.75 \quad 0 \ 0 \ 1 . 1 \ 1 \end{array} \quad \begin{array}{r} 5 = 0 \ 1 \ 0 \ 1 \\ - 9 = 1 \ 0 \ 0 \ 1 \\ \hline 4 \quad 1 \ 1 \ 0 \ 0 \end{array}$$

Binary Subtraction Using 1's Complement:

1. Take 1's complement of subtrahend and add it to minuend.
2. If carry is generated, then result is positive. Add carry to the result to get final result.
3. If carry is not generated, then result is negative and in 1's complement form.

$$\begin{array}{r} 5 \leftarrow \text{Minuend} - \text{First number in subtraction} \\ - 1 \leftarrow \text{Subtrahend} - \text{Number to be subtracted from another} \\ \hline 4 \leftarrow \text{Difference} - \text{Result of subtraction} \end{array}$$

Example: Subtract $(1010)_2$ from $(1111)_2$

Direct Method:

$$\begin{array}{r} 1 \ 1 \ 1 \ 1 \\ - 1 \ 0 \ 1 \ 0 \\ \hline 0 \ 1 \ 0 \ 1 \end{array} \xrightarrow{\text{1's Complement}} \begin{array}{r} 1 \ 1 \ 1 \ 1 \\ 1 \ 1 \ 1 \ 1 \\ \hline 0 \ 1 \ 0 \ 1 + \\ 1 \ 0 \ 1 \ 0 \ 0 \\ \hline 0 \ 1 \ 0 \ 1 \end{array}$$

Example 1: Perform $(28)_{10} - (19)_{10}$ using 1's complement.

Direct Method:

$$\begin{array}{r} (28)_{10} = 1 \ 1 \ 1 \ 0 \ 0 \\ (19)_{10} = 1 \ 0 \ 0 \ 1 \ 1 \\ \hline (9)_{10} \quad 0 \ 1 \ 0 \ 0 \ 1 \end{array} \xrightarrow{\text{1's Complement}} \begin{array}{r} 1 \ 1 \ 1 \\ 1 \ 1 \ 1 \ 0 \ 0 \\ \hline 0 \ 1 \ 1 \ 0 \ 0 \\ 1 \ 0 \ 1 \ 0 \ 0 \\ \hline 0 \ 1 \ 0 \ 0 \ 1 = 9 \end{array}$$

Example 2: Perform $(15)_{10} - (28)_{10}$ using 1's complement.

Direct Method:

$$\begin{array}{r}
 (15)_{10} = 0 \ 1 \ 1 \ 1 \ 1 \\
 (28)_{10} = 1 \ 1 \ 1 \ 0 \ 0 \\
 \hline
 (-13)_{10} = 1 \ 0 \ 0 \ 1 \ 1
 \end{array}
 \xrightarrow{\text{1's Complement}}
 \begin{array}{r}
 1 \ 1 \ 1 \ 1 \\
 0 \ 1 \ 1 \ 1 \ 1 \\
 0 \ 0 \ 0 \ 1 \ 1 \\
 \hline
 1 \ 0 \ 0 \ 1 \ 0
 \end{array}$$

□ No carry, hence result is negative in 1's complement form.

Original Result: 1 0 0 1 0

1's Complement of Result: 0 1 1 0 1 = 13

Binary Subtraction Using 2's Complement:

1. Take 2's complement of subtrahend and add it to minuend.
2. If carry is generated, then result is positive and discard the carry.
3. If carry is not generated, then result is negative and in 2's complement form.

Example 1: Perform $(28)_{10} - (19)_{10}$ using 2's complement.

Direct Method:

$$\begin{array}{r}
 (28)_{10} = 1 \ 1 \ 1 \ 0 \ 0 \\
 - (19)_{10} = 1 \ 0 \ 0 \ 1 \ 1 \\
 \hline
 (9)_{10} = 0 \ 1 \ 0 \ 0 \ 1
 \end{array}
 \xrightarrow{\text{1's Complement}}
 \begin{array}{r}
 0 \ 1 \ 1 \ 0 \ 0 \\
 + 1 \\
 \hline
 0 \ 1 \ 1 \ 0 \ 1
 \end{array}$$

$$\begin{array}{r}
 1 \ 1 \ 1 \\
 + 1 \ 1 \ 1 \ 0 \ 0 \\
 \hline
 0 \ 1 \ 1 \ 0 \ 1
 \end{array}
 \xrightarrow{\text{2's Complement}}
 \begin{array}{r}
 0 \ 1 \ 1 \ 0 \ 1 \\
 + 1 \\
 \hline
 0 \ 1 \ 0 \ 0 \ 1
 \end{array}$$

Ignore the carry and result is positive:

Example 2: Perform $(15)_{10} - (28)_{10}$ using 2's complement.

Direct Method:

$$\begin{array}{r}
 (15)_{10} = 0 \ 1 \ 1 \ 1 \ 1 \\
 - (28)_{10} = 1 \ 1 \ 1 \ 0 \ 0 \\
 \hline
 (-13)_{10} = 1 \ 0 \ 0 \ 1 \ 1
 \end{array}
 \xrightarrow{\text{1's Complement}}
 \begin{array}{r}
 0 \ 0 \ 0 \ 1 \ 1 \\
 + 1 \\
 \hline
 0 \ 0 \ 1 \ 0 \ 0
 \end{array}$$

$$\begin{array}{r}
 1 \ 1 \\
 + 0 \ 1 \ 1 \ 1 \ 1 \\
 \hline
 0 \ 0 \ 1 \ 0 \ 0
 \end{array}
 \xrightarrow{\text{2's Complement}}
 \begin{array}{r}
 0 \ 1 \ 1 \ 0 \ 0 \\
 + 1 \\
 \hline
 0 \ 1 \ 1 \ 0 \ 1
 \end{array}$$

No carry, hence result is negative in 2's complement: $0 \ 1 \ 1 \ 0 \ 1 = 13$

Perform the following:

- Covert $(57345)_{10} = (\)_{16}$
- Subtract $(28)_{10} - (19)_{10}$ using 2's complement method.

(i) Convert $(57345)_{10} = (?)_{16}$

$$\begin{array}{r}
 16 \boxed{57345} \quad \text{Remainder:} \\
 16 \boxed{3584} \quad \rightarrow 1 \\
 16 \boxed{224} \quad \rightarrow 0 \\
 14 \quad \rightarrow 0
 \end{array}$$

$$(57345)_{10} = (E001)_{16}$$

(ii) Subtract $(28)_{10} - (19)_{10}$ using 2's complement method.

$$\begin{array}{r}
 (28)_{10} = (11100)_2 \\
 -(19)_{10} = (10011)_2 \Rightarrow 2\text{'s Complement:} \\
 \hline
 & 11100 \\
 & +01101 \\
 \hline
 & 1)01001 \\
 & 01101
 \end{array}$$

Discard the carry, result is positive in true form: $010001 = 9$ **Perform the following:**

- i. Subtract $10.0101 - 101.1110$ using 1's complement method.

(i) $10.0101 - 101.1110$; using 1's complement

$$\begin{array}{r}
 (2.3125)_{10} \leftarrow 010.0101 \\
 -(5.8750)_{10} \leftarrow 101.1110 \Rightarrow 1\text{'s Complement:} \\
 \hline
 & 010.0101 \\
 & +010.0001 \\
 \hline
 & 100.0110
 \end{array}$$

No carry hence result is negative in 1's complement form: $100.0110 = 11.1001 = -$

3.5625d

Subtract $(19)_{10}$ from $(15)_{10}$ using 1s and 2s complement methods.**Solution:** $(15)_{10} - (19)_{10}$

$$\begin{array}{r}
 (15)_{10} = (01111)_2 \\
 -(19)_{10} = (10011)_2 \Rightarrow 1\text{'s Complement:} \\
 \hline
 & 01111 \\
 & +01100 \\
 \hline
 & 11011
 \end{array}$$

No carry hence result is negative in 1's complement form: $11011 = 00100 = -4$

$$\begin{array}{r}
 (15)_{10} = (01111)_2 \\
 -(19)_{10} = (10011)_2 \Rightarrow 2\text{'s Complement:} \\
 \hline
 & 01111 \\
 & +01101 \\
 \hline
 & +1 \\
 & 11000 \\
 & 01101
 \end{array}$$

No carry hence result is negative in 2's complement form: $11000 = 00011 + 1 = 00100 = -4$

Subtract the following using 2's complement method:

- i) $(101011)_{(2)}$ from $(111001)_{(2)}$
 ii) $(111001)_{(2)}$ from $(101011)_{(2)}$

Solution: (i) $(101011)_{(2)}$ from $(111001)_{(2)}$

$$\begin{array}{r}
 111001(57) \\
 -101011(43) \Rightarrow 2\text{'s Complement:} \\
 \hline
 & 111001 \\
 & +010101 \\
 \hline
 & +1 \\
 & 1)001110 \\
 & 010101
 \end{array}$$

Discard the carry, result is positive in true form: $001110 = 14$

(ii) $(111001)_{(2)}$ from $(101011)_{(2)}$

$$\begin{array}{r}
 101011(43) \\
 -111001(57) \Rightarrow 2\text{'s Complement: } 000110 \\
 \hline
 +000111 \\
 +1 \\
 \hline
 110010 \\
 000111
 \end{array}$$

No carry hence result is negative in 2's complement form: $110010 = 001101 + 1 = 001110 = -14$

4. Perform the subtraction

- i) $(11010)_2 - (10000)_2$ using 1's complement.
 ii) $(1000100)_2 - (1010100)_2$ using 1's complement.

June/July.2016, 15ELN15/25, 04 Marks**Solution:** (i) $(11010)_2 - (10000)_2$ using 1's complement.

$$\begin{array}{r}
 11010(26) \\
 -10000(16) \Rightarrow 1\text{'s Complement: } 1)01001 \\
 \hline
 +01111 \\
 +1 \\
 \hline
 01010
 \end{array}$$

Add carry to the result: $01010 = +10$ **(ii) $(1000100)_2 - (1010100)_2$ using 1's complement.**

$$\begin{array}{r}
 1000100(68) \\
 -1010100(84) \Rightarrow 1\text{'s Complement: } 1101111 \\
 \hline
 +0101011
 \end{array}$$

No carry hence result is negative in 1's complement form: $1101111 = 0010000 = -16$

5. Perform the following operations using 1's and 2's complement technique.

- i) $(56)_{10} - (79)_{10}$ ii) $(23)_{10} - (18)_{10}$.

Solution: (i) $(56)_{10} - (79)_{10}$ **Using 1's complement:**

$$\begin{array}{r}
 (56)_{10} = (0111000)_2 \\
 (79)_{10} = (1001111)_2 \Rightarrow 1\text{'s Complement: } 1101000 \\
 \hline
 +0110000 \\
 1101000
 \end{array}$$

No carry hence result is negative in 1's complement form: $1101000 = 0010111 = -23$ **Using 2's complement:**

$$\begin{array}{r}
 (56)_{10} = (0111000)_2 \\
 (79)_{10} = (1001111)_2 \Rightarrow 2\text{'s Complement: } 0110000 \\
 \hline
 +0110001 \\
 1101001 \\
 \hline
 0110001
 \end{array}$$

No carry hence result is negative in 2's complement form: $1101001 = 0010110 + 1 = 0010111 = -23$ (ii) $(23)_{10} - (18)_{10}$.**Using 1's complement:**

$$\begin{array}{r}
 (23)_{10} = (10111)_2 \\
 (18)_{10} = (10010)_2 \Rightarrow 1\text{'s Complement: } 1)00100 \\
 \hline
 +01101 \\
 1)00100 \\
 \hline
 +1 \\
 00101
 \end{array}$$

Add carry to the result: $00101 = +5$

Using 2's complement:

$$\begin{array}{r} (23)_{10} = (10111)_2 \\ (18)_{10} = (10010)_2 \Rightarrow 2\text{'s Complement: } 01101 \\ \hline \end{array} \quad \begin{array}{r} 10111 \\ +01110 \\ \hline 1)00101 \\ 01110 \end{array}$$

Discard the carry, result is positive in true form: $00101 = +5$

6. Subtract $(111001)_2$ from $(101011)_2$ using 2's complement method.

Solution: $(101011)_2 - (111001)_2$

$$\begin{array}{r} 0111000(56) \\ -1001111(79) \Rightarrow 2\text{'s Complement: } 0110000 \\ \hline \end{array} \quad \begin{array}{r} 0111000 \\ +0110001 \\ \hline 1101001 \\ 0110001 \end{array}$$

No carry hence result is negative in 2's complement form: $1101001 = 0010110 + 1 = 0010111 = -23$

7. i) Subtract $(1000.01)_2$ from $(1011.10)_2$ using 1's and 2's complement method.
ii) Add $(7AB.67)_{16}$ with $(15C.71)_{16}$.

Solution:

i) $(1011.10)_2 - (1000.01)_2$

Using 1's Complement:

$$\begin{array}{r} 1011.10 (11.50) \\ -1000.01 (08.25) \Rightarrow 1\text{'s Complement: } +0111.10 \\ \hline 0011.01 (03.25) \end{array} \quad \begin{array}{r} 1011.10 \\ +0111.10 \\ \hline 1)0011.00 \\ +1 \\ \hline 0011.01 = 3.25d \end{array}$$

Using 2's Complement:

$$\begin{array}{r} 1011.10 (11.50) \\ -1000.01 (08.25) \Rightarrow 1\text{'s Complement: } 0111.10 \\ \hline 0011.01 (03.25) \end{array} \quad \begin{array}{r} 1011.10 \\ +0111.11 \\ \hline 1)10011.01 = 3.25d \end{array}$$

ii) $(7AB.67)_{16} + (15C.71)_{16}$

$$\begin{array}{r} 7AB.67 \\ +15C.71 \\ \hline 907.D8 \end{array}$$

8. Subtract $(111)_2$ from $(1010)_2$ using 1's and 2's complement method.

Solution:

Using 1's complement:

$$\begin{array}{r} 1010 (10d) \\ -0111 (07d) \Rightarrow 1\text{'s Complement: } +1000 \\ \hline 0011 (03d) \end{array} \quad \begin{array}{r} 1010 \\ +1000 \\ \hline 1)0010 \\ +1 \\ \hline 0011 = 3d \end{array}$$

Using 2's Complement:

$$\begin{array}{r}
 1010 \text{ (10d)} \\
 -0111 \text{ (07d)} \\
 \hline
 0011 \text{ (03d)}
 \end{array} \Rightarrow 1\text{'s Complement: }
 \begin{array}{r}
 1000 \\
 +1 \\
 \hline
 1001
 \end{array}
 \begin{array}{r}
 1010 \\
 +1001 \\
 \hline
 1)0011 = 3d
 \end{array}$$

9. Use 1's complement to perform the binary subtraction
 $01111 - 11010$ repeat by 2's complement method.

Solution:**Using 1's complement:**

$$\begin{array}{r}
 01111 \text{ (15d)} \\
 -11010 \text{ (26d)} \\
 \hline
 1)10101 \text{ (-11d)}
 \end{array} \Rightarrow 1\text{'s Complement: }
 \begin{array}{r}
 01111 \\
 +00101 \\
 \hline
 10100
 \end{array} \Rightarrow 1\text{'s Complement: } 01011 = 11d$$

Using 2's Complement:

$$\begin{array}{r}
 01111 \text{ (15d)} \\
 -11010 \text{ (26d)} \\
 \hline
 1)10101 \text{ (-11d)}
 \end{array} \Rightarrow 1\text{'s Complement: }
 \begin{array}{r}
 00101 \\
 +00110 \\
 \hline
 00110
 \end{array}
 \begin{array}{r}
 01111 \\
 +00110 \\
 \hline
 10101
 \end{array}$$

Since no carry, result is negative and represented in 2's complement form:

$$10101 \Rightarrow 1\text{'s Complement: }
 \begin{array}{r}
 01010 \\
 +1 \\
 \hline
 01011
 \end{array} = 11d$$

Perform subtraction using 2's complement method $1101 - 1010$.

Solution:**Using 2's Complement:**

$$\begin{array}{r}
 1101 \text{ (13d)} \\
 -1010 \text{ (10d)} \\
 \hline
 0011 \text{ (03d)}
 \end{array} \Rightarrow 1\text{'s Complement: }
 \begin{array}{r}
 0101 \\
 +0110 \\
 \hline
 0110
 \end{array}
 \begin{array}{r}
 1101 \\
 +0110 \\
 \hline
 1)0011
 \end{array}$$

Discard the carry, result is positive: $0011 = 3$

Perform the following operations:

a. $(CAD.F1)_{16} + (BE1.54)_{16}$

Solution:

a. $(CAD.F1)_{16} + (BE1.54)_{16}$

Direct Method:

$$\begin{array}{r}
 1 \ 1 \quad 1 \\
 C \ A \ D \ . \ F \ 1 \longrightarrow \ 3 \ 2 \ 4 \ 5 \ . \ 9 \ 4 \ 1 \ 4 \ 0 \ 6 \ 2 \ 5 \ d \\
 + \ B \ E \ 1 \ . \ 5 \ 4 \longrightarrow + \ 3 \ 0 \ 4 \ 1 \ . \ 3 \ 2 \ 8 \ 1 \ 2 \ 5 \ 0 \ 0 \ d \\
 \hline
 1 \ 8 \ 8 \ F \ . \ 4 \ 5 \qquad \qquad \qquad 6 \ 2 \ 8 \ 7 \ . \ 2 \ 6 \ 9 \ 5 \ 3 \ 1 \ 2 \ 5 \ d \\
 \\
 = 6 \ 2 \ 8 \ 7 \ . \ 2 \ 6 \ 9 \ 5 \ 3 \ 1 \ 2 \ 5 \ d
 \end{array}$$

i) Perform $(FC02A)_{16} - (D052)_{16}$ using 16's complement

Solution:

$$\text{i. } (\text{FC02A})_{16} - (\text{D052})_{16}$$

Direct Method:

Discard the carry. Result is Positive = EEEFD8h

Boolean Algebra

Boolean Algebra Laws:

1.	Law of Identity	$A = A$ $\bar{A} = \bar{A}$
2.	Commutative Law	$A \cdot B = B \cdot A$ $A + B = B + A$
3.	Associative Law	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$ $A + (B + C) = (A + B) + C$
4.	Idempotent Law	$A \cdot A = A$ $A + A = A$
5.	Double Negative Law	$\bar{\bar{A}} = A$
6.	Complementary Law	$A \cdot \bar{A} = 0$ $A + \bar{A} = 1$
7.	Law of Insertion	$A \cdot 1 = A$ $A \cdot 0 = 0$
8.	Law of Union	$A + 1 = 1$ $A + 0 = A$
9.	Distributive Law	$A \cdot (B + C) = A \cdot B + A \cdot C$ $A + (BC) = (A + B) \cdot (A + C)$
10.	Law of Absorption	$A \cdot (A + B) = A$ $A + (AB) = A$
11.	Law of Common Identities	$A \cdot (\bar{A} + B) = AB$ $A + (\bar{A}B) = A + B$

De Morgan suggested two theorems:

1. The complement of product is equal to the **sum of complements** of individual.

$$\overline{AB} = \overline{A} + \overline{B} \quad ; \text{for two variables}$$

Proof: Two variables:

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

2. The complement of sum is equal to the **product of complements** of individual.

$$\overline{A + B} = \overline{A} \cdot \overline{B} \quad ; \text{for two variables}$$

Proof: Two variables:

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

State and prove De Morgan's theorem for three variables.

- The complement of **product** is equal to the **sum of complements** of individual.

$$\overline{ABC} = \overline{A} + \overline{B} + \overline{C} \quad ; \text{for three variables}$$

Proof: Three variables:

A	B	C	\overline{A}	\overline{B}	\overline{C}	ABC	\overline{ABC}	-- --
0	0	0	1	1	1	0	1	1
0	0	1	1	1	0	0	1	1
0	1	0	1	0	1	0	1	1
0	1	1	1	0	0	0	1	1
1	0	0	0	1	1	0	1	1
1	0	1	0	1	0	0	1	1
1	1	0	0	0	1	0	1	1
1	1	1	0	0	0	1	0	0

- The complement of **sum** is equal to the **product of complements** of individual.

$$\overline{A + B + C} = \overline{A} \cdot \overline{B} \cdot \overline{C} \quad ; \text{for three variables}$$

Proof: Two variables:

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

State De Morgan's theorem for 4 – variables and prove them by the method of perfect induction.

Solution:

- The complement of **product** is equal to the **sum of complements** of individual.

$$\overline{ABCD} = \overline{A} + \overline{B} + \overline{C} + \overline{D} ; \text{for four variables}$$

Proof: Four variables:

A	B	C	D	\overline{A}	\overline{B}	\overline{C}	\overline{D}	ABCD	\overline{ABCD}	$\overline{A} + \overline{B} + \overline{C} + \overline{D}$
0	0	0	0	1	1	1	1	0	1	1
0	0	0	1	1	1	1	0	0	1	1
0	0	1	0	1	1	0	1	0	1	1
0	0	1	1	1	1	0	0	0	1	1
0	1	0	0	1	0	1	1	0	1	1
0	1	0	1	1	0	1	0	0	1	1
0	1	1	0	1	0	0	1	0	1	1
0	1	1	1	1	0	0	0	0	1	1
1	0	0	0	0	1	1	1	0	1	1
1	0	0	1	0	1	1	0	0	1	1
1	0	1	0	0	1	0	1	0	1	1
1	0	1	1	0	1	0	0	0	1	1
1	1	0	0	0	0	1	1	0	1	1
1	1	0	1	0	0	1	0	0	1	1
1	1	1	0	0	0	0	1	0	1	1
1	1	1	1	0	0	0	0	1	0	0

- The complement of **sum** is equal to the **product of complements** of individual.

$$A + B + C + D = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} ; \text{for four variables}$$

Proof: Two variables:

A	B	C	D	\overline{A}	\overline{B}	\overline{C}	\overline{D}	$A + B + C + D$	$\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}$	---
0	0	0	0	1	1	1	1	0	1	1
0	0	0	1	1	1	1	0	1	0	0
0	0	1	0	1	1	0	1	1	0	0
0	0	1	1	1	1	0	0	1	0	0
0	1	0	0	1	0	1	1	1	0	0
0	1	0	1	1	0	1	0	1	0	0
0	1	1	0	1	0	0	1	1	0	0
0	1	1	1	1	0	0	0	1	0	0
1	0	0	0	0	1	1	1	1	0	0
1	0	0	1	0	1	1	0	1	0	0
1	0	1	0	0	1	0	1	1	0	0
1	0	1	1	0	1	0	0	1	0	0
1	1	0	0	0	0	1	1	1	0	0
1	1	0	1	0	0	1	0	1	0	0
1	1	1	0	0	0	0	1	1	0	0
1	1	1	1	0	0	0	0	1	0	0

1. Simplify, $AB + \overline{AC} + \overline{ABC}(AB + C)$.

Solution:

$$\begin{aligned}
 Y &= AB + \overline{AC} + \overline{ABC}(AB + C) &&; \text{Apply De Morgan Law} \\
 Y &= AB + \overline{A} + \overline{C} + \overline{ABC}AB + \overline{ABC}C = AB + \overline{A} + \overline{C} + \overline{ABC} \\
 Y &= AB + \overline{A} + \overline{A} + \overline{C} + \overline{ABC} &&; \text{using Idempotent law } \overline{A} + \overline{A} = \overline{A} \\
 Y &= (AB + \overline{A}) + \overline{C} + (\overline{A} + \overline{ABC}) &&; \text{Apply distributive law: } A + BC = (A + B)(A + C) \\
 Y &= (A + \overline{A})(B + \overline{A}) + \overline{C} + (\overline{A} + \overline{ABC}) = B + \overline{A} + \overline{C} + \overline{A} + \overline{BC} \\
 Y &= \overline{A} + \overline{C} + (B + \overline{B}C) &&; \text{Apply distributive law: } A + BC = (A + B)(A + C) \\
 Y &= \overline{A} + \overline{C} + (B + C) = \overline{A} + \overline{C} + B + C = \overline{A} + B + 1 = 1
 \end{aligned}$$

OR:

$$\begin{aligned}
 Y &= AB + \overline{AC} + \overline{ABC}(AB + C) &&; \text{Apply De Morgan Law} \\
 Y &= AB + \overline{A} + \overline{C} + \overline{ABC}AB + \overline{ABC}C = AB + \overline{A} + \overline{C} + \overline{ABC} \\
 Y &= AB + \overline{A} + (\overline{C} + \overline{ABC}) &&; \text{Apply distributive law} \\
 Y &= AB + \overline{A} + (\overline{C} + AB)(\overline{C} + C) = AB + \overline{A} + \overline{C} + AB &&; A \text{ is common in 1st \& last terms} \\
 Y &= \overline{A} + \overline{C} + A(B + \overline{B}) = \overline{A} + \overline{C} + A = 1 + \overline{C} = 1
 \end{aligned}$$

2. Simplify $ABC + \overline{ABC} + \overline{ABC} + ABC$

Solution:

$$\begin{aligned}
 Y &= ABC + \overline{ABC} + \overline{ABC} + ABC = BC(A + \overline{A}) + \overline{ABC} + ABC = BC + \overline{ABC} + ABC \\
 Y &= C(B + A\overline{B}) + ABC = C(B + \overline{B})(B + A) + ABC = BC + AC + ABC = BC + A(C + \overline{C}) \\
 Y &= BC + A(C + \overline{C})(C + B) = AB + BC + AC
 \end{aligned}$$

3. Simplify $\overline{AB + A\overline{B}}$ using De Morgan's theorem.

Solution:

$$Y = \overline{AB + A\overline{B}} = \overline{AB} \cdot \overline{A\overline{B}} = (\overline{A} + \overline{B})(A + B) = \overline{AB} + A\overline{B} = A \oplus B$$

4. Simplify the following expressions:

(a) $Y = \overline{AB} + ABC + A(B + \overline{AB})$
(b) $Y = (A + B)(\overline{A} + B)$

Solution:

$$\begin{aligned}
 (a) Y &= \overline{\overline{AB} + ABC + A(B + \overline{AB})} \\
 Y &= \overline{\overline{AB} + ABC} + A(\overline{B + \overline{AB}}) &&; A \text{ common in 1st term \& apply distributive law to 2nd term} \\
 Y &= \overline{A(\overline{B} + BC)} + A(\overline{B + \overline{AB}})(B + A) &&; \text{Apply distributive law to 1st term} \\
 Y &= \overline{A(\overline{B} + B)}(\overline{B} + C) + AB + AA \\
 Y &= \overline{A(\overline{B} + C)} + AB + A &&; \text{Apply DeMorgan law} \\
 Y &= \overline{A} + (\overline{B} + C) + AB + A &&; \text{Apply DeMorgan law} \\
 Y &= \overline{A} \cdot (\overline{B} + C) \cdot \overline{AB} \cdot \overline{A} = A \cdot (\overline{B} + C) \cdot \overline{AB} \cdot \overline{A} = 0
 \end{aligned}$$

(b) $Y = (A + B)(\overline{A} + B)$

$$Y = (A + B)(\bar{A} + B) = A\bar{A} + AB + \bar{A}B + BB = B(A + \bar{A}) + B = B + B = B$$

5. Show that:

i) $A\bar{B}C + B + BD + AB\bar{D} + \bar{A}C = B + C$

ii) $\overline{AB + A + AB} = 0$

iii) $AB + A(B + C) + B(B + C) = B + AC$

Solution: (i) $A\bar{B}C + B + BD + AB\bar{D} + \bar{A}C = B + C$

$$A\bar{B}C + B + BD + AB\bar{D} + \bar{A}C = A\bar{B}C + B(1 + \bar{D}) + AB\bar{D} + \bar{A}C = A\bar{B}C + B + AB\bar{D} + \bar{A}C$$

$$= A\bar{B}C + B(1 + A\bar{D}) + \bar{A}C = A\bar{B}C + B + \bar{A}C = C(A\bar{B} + \bar{A}) + B$$

; Apply distributive law: $(A + BC) = (A + B)(A + C)$

$$= C(A + \bar{A})(\bar{B} + \bar{A}) + B = C(\bar{B} + \bar{A}) + B = C\bar{B} + C\bar{A} + B$$

$$= (C\bar{B} + B) + C\bar{A} ; \text{ Apply distributive law}$$

$$= (C + B)(\bar{B} + B) + C\bar{A} = C + B + C\bar{A} = B + C(1 + \bar{A})$$

$$= B + C$$

(ii) $\overline{\overline{AB} + \bar{A} + AB} = 0$; Apply De Morgan Law

$$\overline{\overline{AB} + \bar{A} + AB} = \overline{\overline{AB}} \cdot \bar{A} \cdot \overline{AB} = AB \cdot A \cdot (\bar{A} + \bar{B}) = AB(\bar{A} + \bar{B}) = A\bar{B} + A\bar{B} = 0$$

(iii) $AB + A(B + C) + B(B + C) = B + AC$

$$AB + A(B + C) + B(B + C) = AB + AB + AC + BB + BC = AB + AC + B + BC$$

$$= B(A + 1) + AC + BC = B + AC + BC = B(1 + C) + AC$$

$$= B + AC$$

Basic and Universal Gates

Digital Circuits:

- A **digital circuit** is a circuit where the signal must be one of two discrete levels. Each level is interpreted as one of two different states (for example, on/off, 0/1, true/false). **Digital circuits** use transistors to create logic gates in order to perform Boolean logic. **Digital circuits** are less susceptible to noise or less degradation in quality than analog circuits. It is also easier to perform error detection and correction with digital signals.
- Digital Circuits are classified into two major categories:
 1. **Combinational Circuits:** The output depends on present input only.
 2. **Sequential Circuits:** The output depend on both present inputs and past outputs. Also sequential circuits have memory.

Logic Gates:

- Logic gates are an electronic circuit which accepts binary input and produces a binary output namely 0 and 1. They are basic building blocks of digital circuits. They are used to create digital circuits and even complex integrated circuits.
- Combination of logic gates form circuits designed with specific tasks
- Examples:* **Adders:** to add binary numbers,
Flip-Flops: set or reset bits of memory,
- Complex integrated circuits are complete circuits ready to perform several functions.
Examples: Microprocessors and Microcontrollers
- There three categories of logic gates:
 - i. **Basic Gates:** NOT, OR & AND
 - ii. **Derived Gates:** EX-OR & EX-NOR
 - iii. **Universal Gates:** NAND & NOR

Basic gates AND, OR and NOT gates with truth tables.

NOT Gate:

- It is an electronic circuit having only one input and only one output. The output signal is always opposite to the input signal and it is a physical realization of **Boolean-Complement** operation. It is also called as **inverter**.

Logic Symbol:



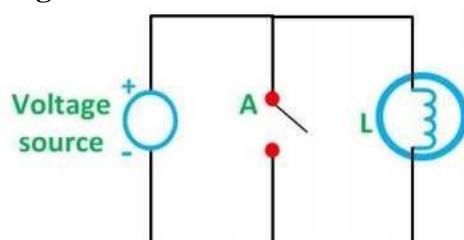
Expression: $Y = \bar{X}$



Truth Table:

Input	Output
X	Y
0	1
1	0

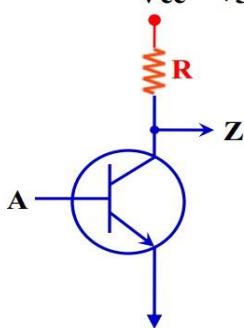
NOT Gate Using Switch:



When Input = 0 ;Switch is Opened
 When Input = 1 ;Switch is Closed

Function Table:

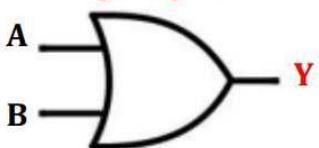
A	L
Open (Low)	ON (High)
Close (High)	OFF (low)

NOT Gate Realization Using Transistor:

- When $+5V$ is applied to A, the transistor will be fully turned ON, drawing maximum collector current, hence the entire $V_{cc} = 5V$ will drop across R, thereby sending $0V$ to Z. Therefore, output $Z = 0$ ($0V$).
- When $0V$ is applied to A, the transistor will be cut-off, hence the entire $V_{cc} = 5V$ will be pulled to Z. Therefore, output $Z = 1$ ($5V$).
- In either cases, it is seen that output is opposite of input.

OR Gate:

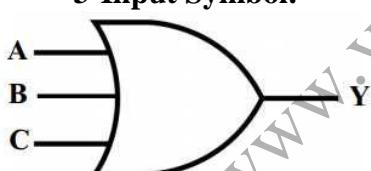
- It is an electronic circuit having two or more inputs and only one output. The output is zero, when all the inputs are zero, otherwise output is one. It is a physical realization of **Boolean-Addition** operation.

2 - Input Symbol

Expression: $Y = A + B$

**Truth Table:**

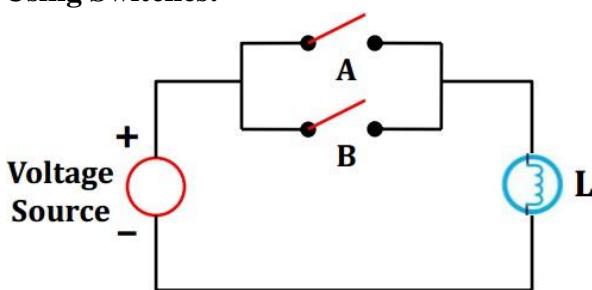
Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

3-Input Symbol:

Expression: $Y = A + B + C$

Truth Table:

Inputs			Output
A	B	C	Y
0	0	0	0
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	1

OR Gate Using Switches:

When Input = 0 ;Switch is Opened

When Input = 1 ;Switch is Closed

Function Table:

A	B	L
Open	Open	OFF
Open	Close	ON
Close	Open	ON
Close	Close	ON

Fig: Realization of OR Gate Using Switches

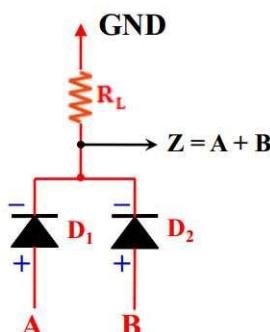
OR Gate Realization Using Diodes:

- When $A = 0$ and $B = 0$:

Both the diodes D_1 & D_2 do not conduct, since they are reverse biased, hence no current flows through R_L , and no voltage develops across R_L . Therefore $Z = 0$ ($0V$).

➤ When A = 0 and B = 1:

Diode D₁ does not conduct, since it is reverse biased. D₂ conducts, since voltage B is high (5V) hence current flows through R_L, then voltage develops across R_L that is approximately equal to voltage B. Therefore Z = 1 (5V).



A and B: Input Voltages (0 or 5V)
Z: Output Voltage (0 or 5V)

Fig: Realization of OR Gate Using Diodes

➤ When A = 1 and B = 0:

Diode D₂ does not conduct, since it is reverse biased. D₁ conducts, since voltage A is high (5V) hence current flows through R_L, then voltage develops across R_L that is approximately equal to voltage A. Therefore Z = 1 (5V).

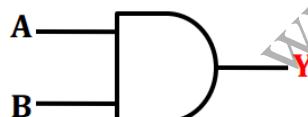
➤ When A = 1 and B = 1:

Both the diodes D₁ & D₂ conducts, since they are forward biased, hence current flows through R_L and voltage approximately equal voltage A or B is developed across R_L. Therefore Z = 1 (5V).

AND Gate:

- It is an electronic circuit having two or more inputs and only one output. The output is one, when all the inputs are one, otherwise output is zero. It is a physical realization of **Boolean-Multiplication** operation.

2 - Input Symbol



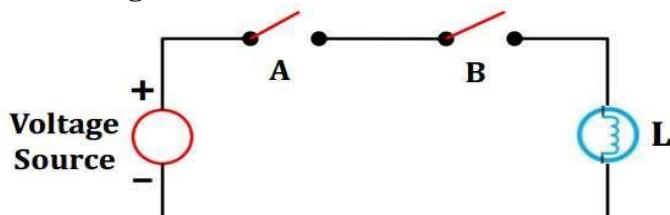
Expression: $Y = A \cdot B$



Truth Table:

Inputs	Output	
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

AND Gate Using Switches:

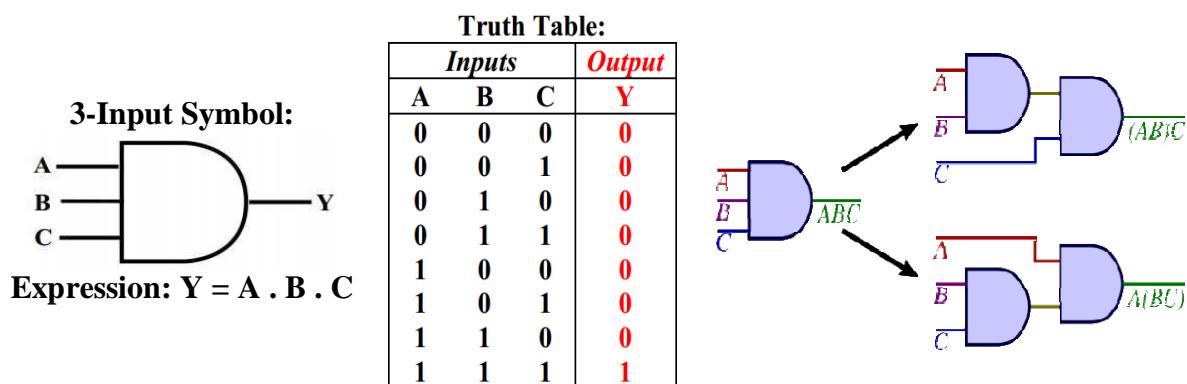


When Input = 0 ;Switch is Opened
When Input = 1 ;Switch is Closed

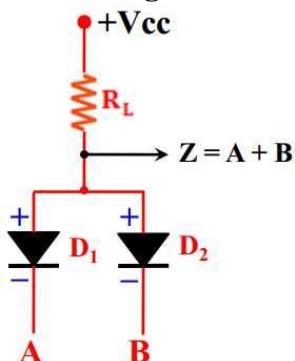
Function Table:

A	B	L
Open	Open	OFF
Open	Close	OFF
Close	Open	OFF
Close	Close	ON

Fig: Realization of AND Gate Using Switches



AND Gate Realization Using Diodes:



A and B: Input Voltages (0 or 5V)
Z: Output Voltage (0 or 5V)

Fig: Realization of AND Gate Using Diodes

➤ When $A = 0$ and $B = 0$:

The cathodes of both diodes D_1 & D_2 are grounded. The diodes get forward biased and hence conduct and current flows through the resistor R_L , then no voltage drop across the diode. Therefore, output $Z = 0$ (0V), practically $Z = 0.7V \approx 0V$.

➤ When $A = 0$ and $B = 1$:

The cathode of diode D_2 is 5V, the diode becomes reverse biased, hence it does not conduct and no current flow through resistor R_L . But the cathode of diode D_1 is grounded, the diode becomes forward biased, hence conduct and current flows through the resistor R_L , then no voltage drop across the diode. Therefore, output $Z = 0$ (0V), practically $Z = 0.7V \approx 0V$.

➤ When $A = 1$ and $B = 0$:

The cathode of diode D_1 is 5V, the diode becomes reverse biased, hence it does not conduct and no current flow through resistor R_L . But the cathode of diode D_2 is grounded, the diode becomes forward biased, hence conduct and current flows through the resistor R_L , then no voltage drop across the diode. Therefore, output $Z = 0$ (0V), practically $Z = 0.7V \approx 0V$.

➤ When $A = 1$ and $B = 1$:

The cathodes of both diodes D_1 & D_2 are connected to 5V. The diodes become reverse biased, hence both does not conduct and no current flows through the resistor R_L , then output voltage pulled to $+V_{cc}$ i.e., 5V. Therefore, output $Z = 1$ (5V).

Exclusive – OR (EX-OR) Gate:

Design of logic circuit, symbol and truth table of exclusive – OR gate.

- It is an electronic circuit having two or more inputs and only one output. The output is zero, when even number of inputs are one, and output is one, when odd number of inputs are one.

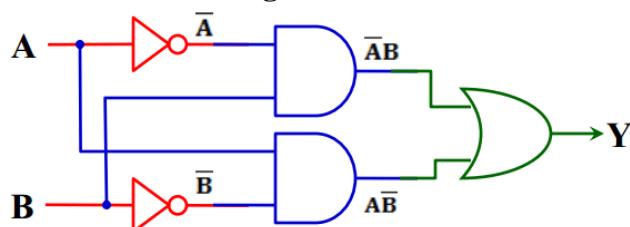
Application: It is used to implement magnitude comparator, gray code converter, adder/subtractor circuits, parity generator, modulo-2 adder etc.

2-Input Symbol:

$$\text{Expression: } Y = \overline{A} \overline{B} + A \overline{B} = A \oplus B$$

**Truth Table:**

Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Logic Circuit:**3-Input Symbol:**

$$\text{Expression:}$$

Truth Table:

Inputs			Output
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Expression:

$$Y = \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot \overline{C} + A \cdot \overline{B} \cdot \overline{C} + A \cdot B \cdot C = (\overline{A} \cdot \overline{B} + A \cdot B)C + (\overline{A} \cdot B + A \cdot \overline{B})\overline{C}$$

$$Y = \overline{A \oplus B} \cdot C + A \oplus B \cdot \overline{C} = A \oplus B \oplus C$$

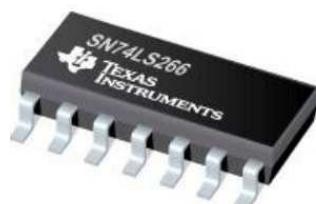
Exclusive - NOR (EX-NOR) Gate:

Symbol, truth table and final expression for NAND and Ex-OR gate (For two I/Ps).

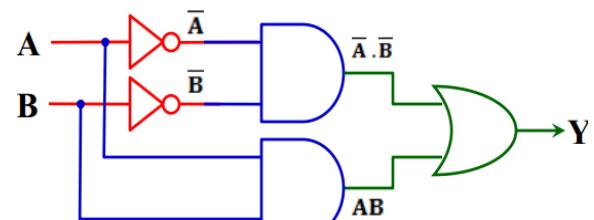
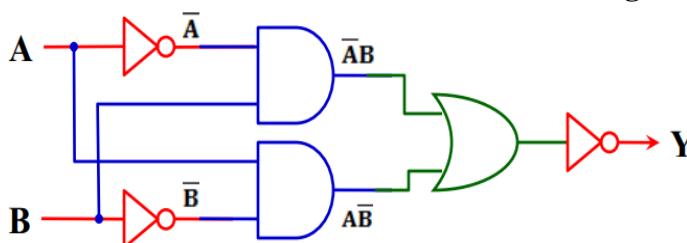
- It is an electronic circuit having two or more inputs and only one output. The output is complement of EX-OR gate output.

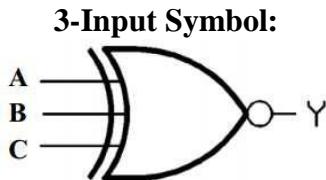
2-Input Symbol:**Expression:**

$$Y = \overline{A} \cdot \overline{B} + AB = \overline{A \oplus B} = A \odot B$$

**Truth Table:**

Inputs		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Logic Circuit:



Truth Table:			
Inputs		Output	
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

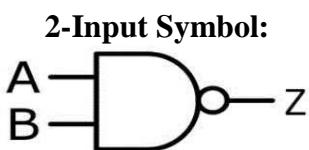
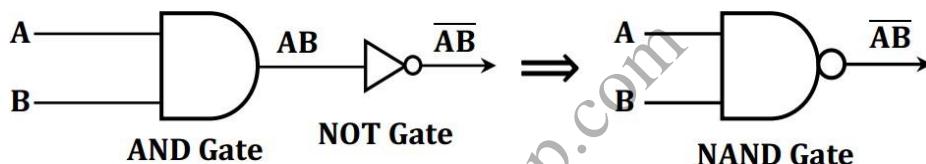
Expression:

$$Y = \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C} = (\overline{A} \cdot \overline{B} + A \cdot B) \overline{C} + (\overline{A} \cdot B + A \cdot \overline{B}) C$$

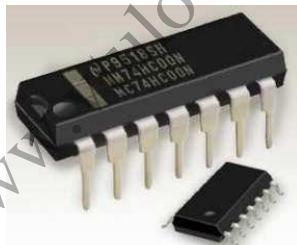
$$Y = \overline{A \oplus B} \cdot \overline{C} + A \oplus B \cdot C = \overline{A \oplus B \oplus C} = A \odot B \odot C$$

Universal Gates:**NAND Gate:**

- It is an electronic circuit having two or more inputs and only one output. The output is complement of AND gate output. It is also called as universal gate, because it can be used to realize all other gates.



Expression: $Z = \overline{AB}$

**Truth Table:**

Inputs		Output
A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

3-Input Symbol:

Expression: $Z = \overline{ABC}$

Truth Table:

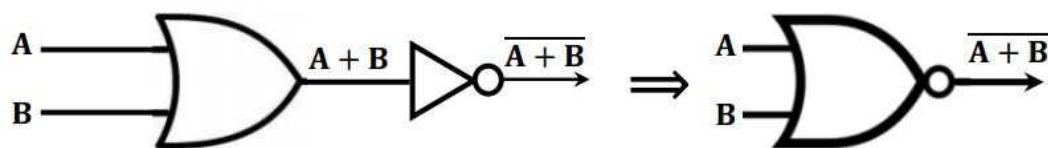
Inputs			Output
A	B	C	Z
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

NOR Gate:

- It is an electronic circuit having two or more inputs and only one output. The output is complement of OR gate output. It is also called as universal gate, because it can be used to realize all other gates.

Shruthi V, Asst Prof, Dept of ECE, GMIT

Page | 24



2-Input Symbol:

Expression: $Z = \overline{A + B}$



Truth Table:

Inputs		Output
A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

3-Input Symbol:

Expression: $Z = \overline{A + B + C}$

Truth Table:

Inputs			Output
A	B	C	Z
0	0	0	1
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	0

Realization of Logic Gates using Universal Gates (NAND & NOR):

NOT Gate: $Y = \overline{A}$	
AND Gate: $Y = AB$	
OR Gate: $Y = A + B$ $Y = \overline{\overline{A + B}}$ $Y = \overline{\overline{A}} \cdot \overline{\overline{B}}$	
NOR Gate: $Y = \overline{A + B}$ $Y = \overline{A} \cdot \overline{B}$	
EX-OR Gate: $Y = \overline{AB} + \overline{AB} = \overline{AB} + \overline{AB} = \overline{AB} \cdot \overline{AB}$ $Y = A \oplus B$	

	$Y = \overline{A \cdot \overline{AB}} \cdot \overline{B \cdot \overline{AB}} = A \cdot \overline{AB} + B \cdot \overline{AB} = A \cdot \overline{AB} + B \cdot \overline{AB}$ $Y = A(\overline{A} + \overline{B}) + B(\overline{A} + \overline{B}) = \overline{AB} + \overline{AB}$ $Y = A \oplus B$
EX-NOR Gate: $Y = \overline{A} \cdot \overline{B} + AB$ $Y = A \odot B$	$Y = \overline{A} \cdot \overline{B} + AB = \overline{\overline{A} \cdot \overline{B}} + \overline{AB} = \overline{A \cdot \overline{B}} \cdot \overline{AB}$ $Y = \overline{A \cdot \overline{AB}} \cdot \overline{B \cdot \overline{AB}} = A \cdot \overline{AB} + B \cdot \overline{AB} = A \cdot \overline{AB} + B \cdot \overline{AB}$ $Y = A(\overline{A} + \overline{B}) + B(\overline{A} + \overline{B}) = \overline{AB} + \overline{AB}$ $Y = A \oplus B$

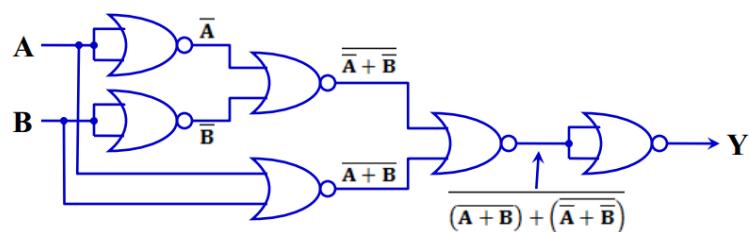
Realization Using NOR Gate:

1. Implement EX-NOR gate using only NOR gates. Dec.2013/Jan.2014, 10ELN15/25, 04 Marks

NOT Gate: $Y = \overline{A}$	$Y = \overline{A + A} = \overline{A}$
AND Gate: $Y = AB$	$Y = AB = \overline{\overline{AB}} = \overline{\overline{A} + \overline{B}}$ $Y = \overline{\overline{A} + \overline{B}} = A \cdot B$
OR Gate: $Y = A + B$	$Y = \overline{\overline{A + B}} = A + B$
NAND Gate: $Y = \overline{AB}$	$Y = \overline{AB} = \overline{\overline{A} + \overline{B}}$ $\overline{\overline{A} + \overline{B}} = A \cdot B$ $Y = \overline{A \cdot B}$
EX-NOR Gate:	$Y = \overline{A} \cdot \overline{B} + AB = \overline{A} \cdot B + A \cdot \overline{B} = A \cdot B \cdot \overline{AB}$

$$\begin{aligned} Y &= \bar{A} \cdot \bar{B} + A \cdot B \\ Y &= A \odot B \\ \text{OR} \\ Y &= \bar{A} \bar{B} + A \bar{B} \\ Y &= A \oplus B \end{aligned}$$

$$Y = (\bar{A} + B) \cdot (\bar{A} + \bar{B}) = \bar{A} + B + \bar{A} + \bar{B}$$

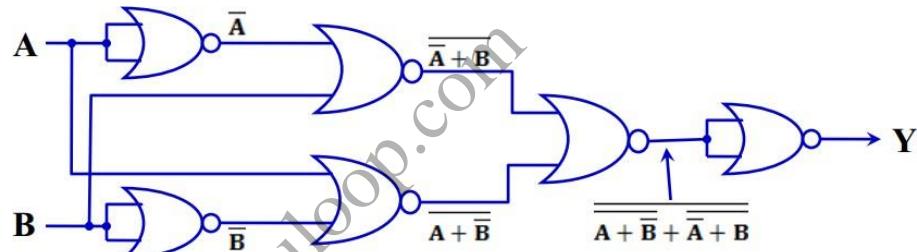


$$\begin{aligned} Y &= A + (\bar{A} + B) \\ &= A + \bar{A} + B \\ &= A \oplus B \end{aligned}$$

$$Y = \bar{A} + A + \bar{B} + B + \bar{A} + B = \bar{A} \cdot (A + B) + \bar{B} \cdot (A + B)$$

$$Y = \bar{A} \bar{B} + A \bar{B} = A \oplus B$$

$$Y = \bar{A} \bar{B} + A \bar{B} = \bar{A} \bar{B} + A \bar{B} = \bar{A} \bar{B} \cdot A \bar{B} = (\bar{A} + B) \cdot (\bar{A} + B) = \bar{A} + \bar{B} + \bar{A} + B$$



$$\begin{aligned} Y &= A + (\bar{A} + B) \\ &= A + \bar{A} + B \\ &= A \oplus B \end{aligned}$$

EX-OR Gate:

$$\begin{aligned} Y &= \bar{A} \bar{B} + A \bar{B} \\ Y &= A \oplus B \end{aligned}$$

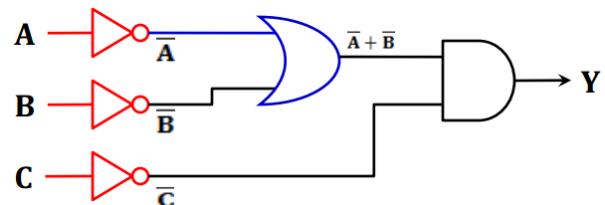
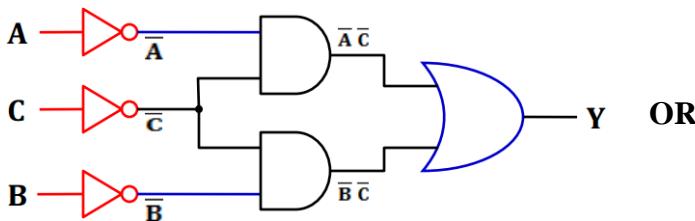
Algebraic Simplification and Realization using Logic Gates:

1. Simplify the expression and realize using basic gates $\bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} \bar{C} + A \bar{B} \bar{C} + A \bar{B} \bar{C}$.

Solution:

$$\begin{aligned} Y &= \bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} \bar{C} + A \bar{B} \bar{C} + A \bar{B} \bar{C} \\ &= \bar{A} \bar{C}(\bar{B} + B) + A \bar{B} \bar{C} \\ &= \bar{A} \bar{C} + A \bar{B} \bar{C} \\ &= (\bar{A} + A\bar{B})\bar{C} \\ &= (\bar{A} + A)(\bar{A} + \bar{B})\bar{C} \\ &= (\bar{A} + \bar{B})\bar{C} \end{aligned}$$

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

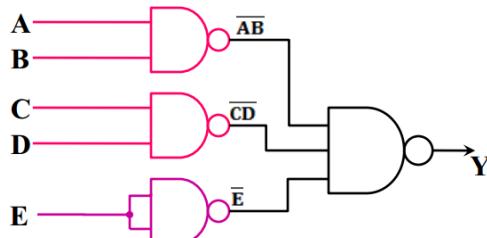


2. Realize $Y = AB + CD + E$ using NAND gate.

Dec.2017/Jan.2018, 17ELN15/25, 06 Marks

Solution:

$$Y = AB + CD + E = \overline{\overline{AB} + \overline{CD} + \overline{E}} = \overline{\overline{AB}} \cdot \overline{\overline{CD}} \cdot \overline{\overline{E}}$$



3. $Y = A + \overline{AB} + AB\bar{C}$ simplify and implement using logic gates and NOR gates.

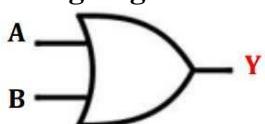
Dec.2017/Jan.2018, 17ELN15/25, 06 Marks

Marks

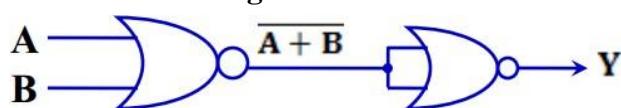
Solution:

$$\begin{aligned} Y &= A + \overline{AB} + AB\bar{C} \\ &= (A + \bar{A})(A + B) + AB\bar{C} \\ &= A + B + AB\bar{C} \\ &= A + B(1 + B\bar{C}) \\ Y &= A + B \end{aligned}$$

Using Logic Gates:



Using NOR Gates:



4. Simplify the following Boolean expressions and realize them using basic gates and universal gates.

$$(a) P = \overline{X}\overline{Y}\overline{Z} + \overline{X}\overline{Y}\overline{Z} + \overline{X}\overline{Y} + X\overline{Y}$$

$$(b) Y = ABC + AB\bar{C} + \bar{A}BC$$

$$(c) Q = (B + CA)(C + \bar{A}B)$$

$$(d) Z = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D$$

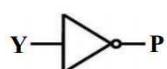
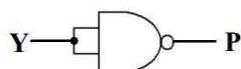
$$(e) Y = ABC + A\bar{B}C + AB\bar{C}$$

$$(f) Z = \overline{AB} + \overline{A}\overline{B}$$

Solution:

$$\begin{aligned} (a) P &= \overline{X}\overline{Y}\overline{Z} + \overline{X}\overline{Y}\overline{Z} + \overline{X}\overline{Y} + X\overline{Y} \\ &= \overline{X}\overline{Y}(\overline{Z} + 1) + X\overline{Y} \\ &= \overline{Y}(X + \overline{X}) \end{aligned}$$

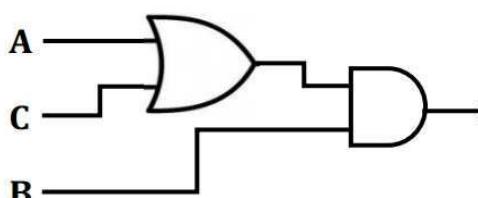
$$P = \overline{Y}$$

Realization:**Basic Gate:****NAND Gate:****NOR Gate:**

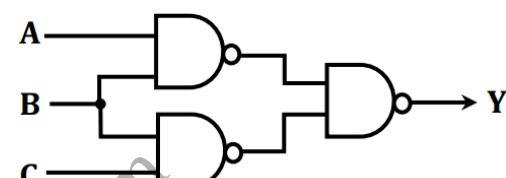
$$\begin{aligned}
 (b) \quad Y &= ABC + AB\bar{C} + \bar{A}BC \\
 &= AB(C + \bar{C}) + \bar{A}BC \\
 &= AB + \bar{A}BC \\
 &= B(A + \bar{A}C) \\
 &= B(A + \bar{A})(A + C) \\
 Y &= AB + BC
 \end{aligned}$$

Realization:**Using Basic Gate:**

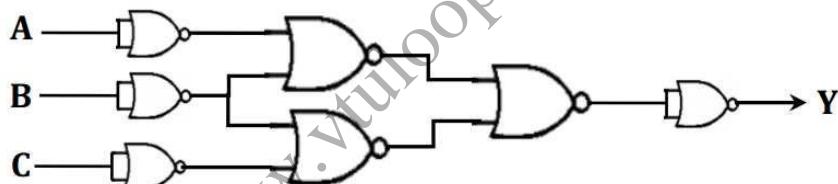
$$Y = AB + BC = B(A + C)$$

**Using NAND Gate:**

$$Y = \overline{\overline{AB} + \overline{BC}} = \overline{\overline{AB} \cdot \overline{BC}}$$

**Using NOR Gate:**

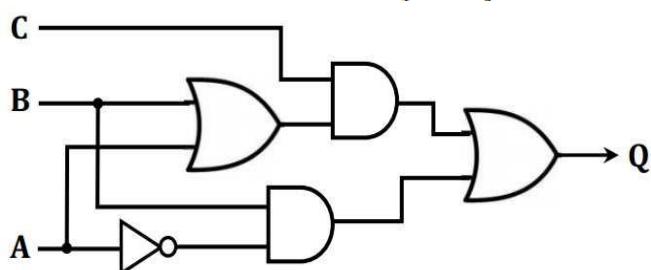
$$Y = \overline{\overline{AB} + \overline{BC}} = \overline{\overline{AB} \cdot \overline{BC}} = (\overline{A} + \overline{B}) \cdot (\overline{B} + \overline{C}) = (\overline{A} + \overline{B}) + (\overline{B} + \overline{C})$$



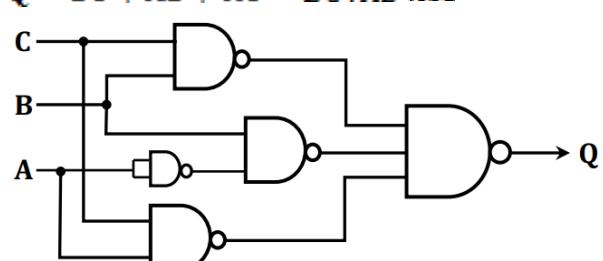
$$\begin{aligned}
 (c) \quad Q &= (B + CA)(C + \overline{AB}) = B(C + \overline{AB}) + AC(C + \overline{AB}) \\
 &= BC + \overline{A}BB + ACC + AC\overline{AB} = BC + \overline{A}B + AC
 \end{aligned}$$

Realization:**Using Basic Gate:**

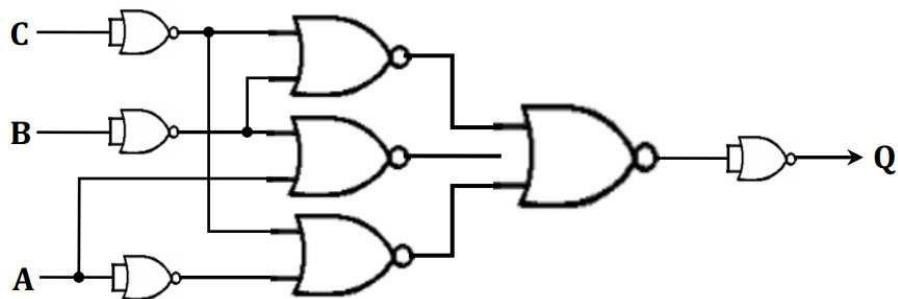
$$Q = BC + \overline{AB} + AC = C(A + B) + \overline{AB}$$

**Using NAND Gate:**

$$Q = \overline{BC + \overline{AB} + AC} = \overline{\overline{BC} \cdot \overline{\overline{AB}} \cdot \overline{AC}}$$

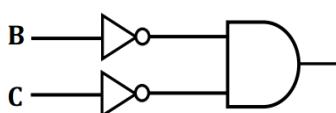
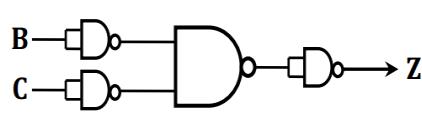
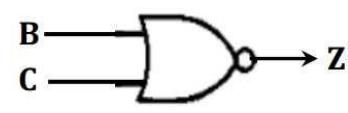
**NOR Gate:**

$$Q = \overline{\overline{BC} + \overline{AB} + \overline{AC}} = \overline{\overline{BC} \cdot \overline{\overline{AB}} \cdot \overline{\overline{AC}}} = (\overline{B} + \overline{C}) + (\overline{A} + \overline{B}) + (\overline{A} + \overline{C})$$



$$(d) Z = \overline{A} \overline{B} \overline{C} \overline{D} + \overline{A} \overline{B} \overline{C} D + A \overline{B} \overline{C} \overline{D} + A \overline{B} \overline{C} D = \overline{B} \overline{C} (\overline{A} \overline{D} + \overline{A} D + A \overline{D} + A D)$$

$$Z = \overline{B} \overline{C} (\overline{A} \overline{D} + \overline{A} D + A \overline{D} + A D) = \overline{B} \overline{C} (\overline{A} [\overline{D} + D] + A [\overline{D} + D]) = \overline{B} \overline{C} (\overline{A} + A) = \overline{B} \overline{C}$$

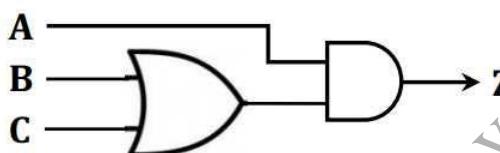
Realization:**Basic Gate:****NAND Gate:****NOR Gate:**

$$(e) Y = ABC + A\overline{B}C + A\overline{B}\overline{C} = AB(C + \overline{C}) + A\overline{B}C = AB + A\overline{B}C = A(B + \overline{B}C)$$

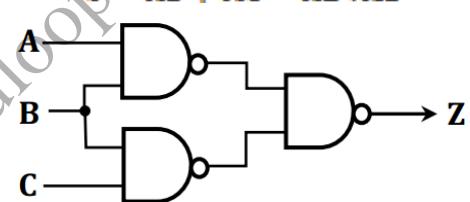
$$Y = A(B + \overline{B})(B + C) = A(B + C) = AB + AC$$

Realization:**Basic Gate:**

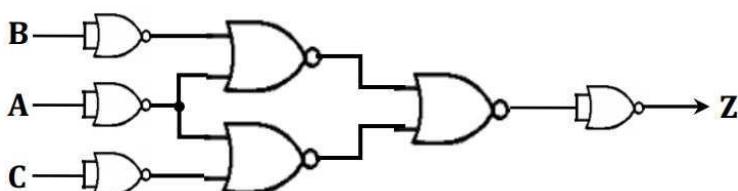
$$Y = AB + AC = A(B + C)$$

**NAND Gate:**

$$Y = \overline{AB} + \overline{AC} = \overline{AB} \cdot \overline{AC}$$

**NOR Gate:**

$$Y = \overline{\overline{AB} + \overline{AC}} = \overline{\overline{AB} \cdot \overline{AC}} = \overline{(\overline{A} + \overline{B})} + \overline{(\overline{A} + \overline{C})}$$



$$(f) Y = \overline{AB} + \overline{A}\overline{B} = \overline{AB} \cdot \overline{\overline{A}\overline{B}}$$

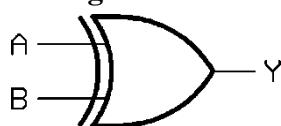
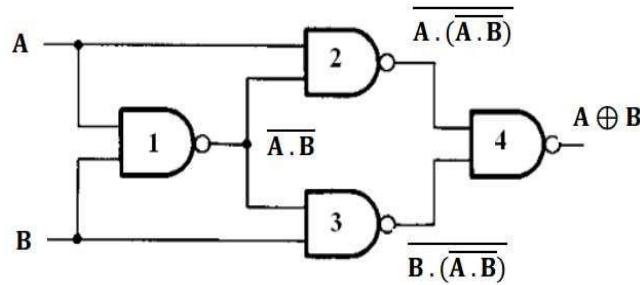
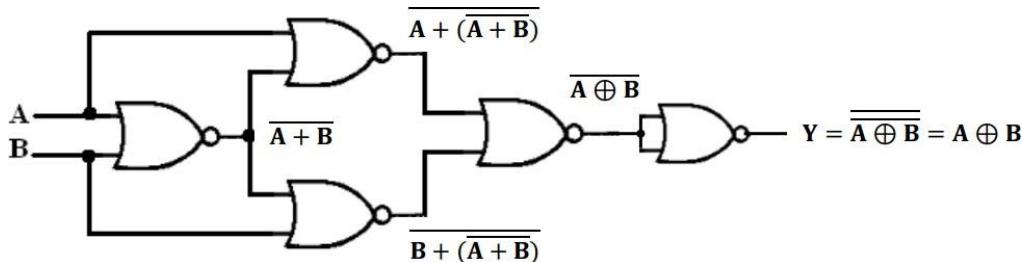
$$= (\overline{A} + \overline{B})(A + B)$$

$$= \overline{AA} + \overline{AB} + \overline{AB} + \overline{BB}$$

$$= \overline{AB} + A\overline{B}$$

$$= \overline{AB} + A\overline{B}$$

$$Y = A \oplus B$$

Realization:**Using Basic Gate:****Using NAND Gate:****Using NOR Gate:**

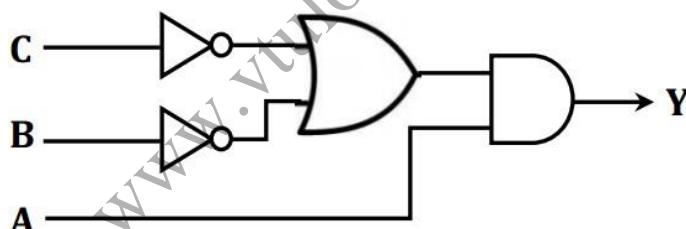
5. Simplify the following expression and realize using basic gates:

$$Y = A(\bar{ABC} + \bar{ABC})$$

June/July.2016, 15ELN15/25, 04

Marks

Solution:
$$\begin{aligned} Y &= A(\bar{ABC} + \bar{ABC}) = A(\bar{A} + \bar{B} + \bar{C}) + A\bar{ABC} = \bar{A}\bar{A} + A\bar{B} + A\bar{C} + A\bar{B}\bar{C} \\ &= A\bar{B}(1 + C) + A\bar{C} = A\bar{B} + A\bar{C} = A(\bar{B} + \bar{C}) \end{aligned}$$

Realization using Basic Gates:

6. Simplify and realize the following expressions using only NAND and NOR.

i) $Y = (A + \bar{B})(B + C)(\bar{C} + \bar{B})$ ii) $Y = AB + AC + BD + CD$.

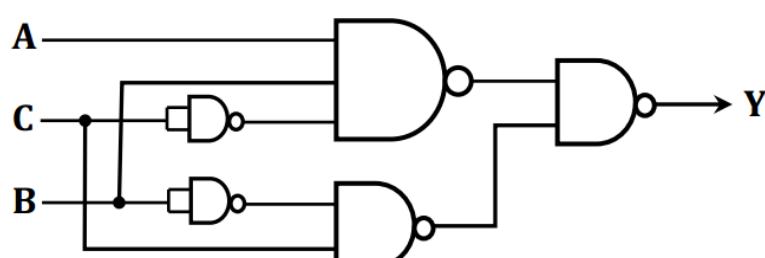
Dec.2015/Jan.2016, 15ELN15/25, 10 Marks

Solution:

(i)
$$\begin{aligned} Y &= (A + \bar{B})(B + C)(\bar{C} + \bar{B}) = (AB + AC + \bar{B}B + \bar{B}C)(\bar{C} + \bar{B}) \\ &= ABC + ABB + ACC + AC\bar{B} + \bar{B}CC + \bar{B}CB = ABC + A\bar{B}C + \bar{B}C \\ &= ABC + \bar{B}C(A + 1) = ABC + \bar{B}C \end{aligned}$$

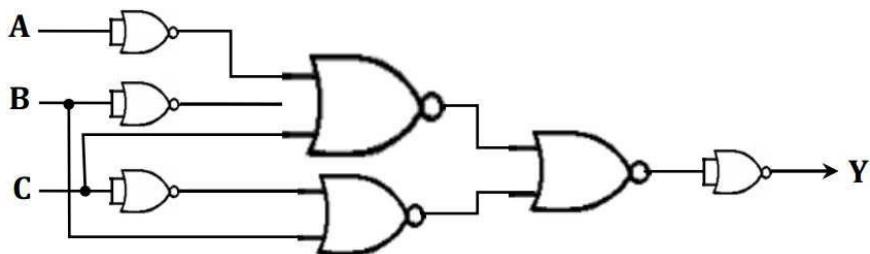
NAND Realization:

$$Y = AB\bar{C} + \bar{B}C = \overline{\overline{ABC} + \overline{BC}} = \overline{\overline{ABC} \cdot \overline{BC}}$$



NOR Realization:

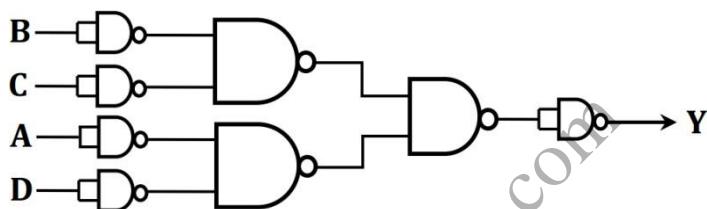
$$Y = \overline{\overline{ABC} + \overline{BC}} = \overline{\overline{ABC} \cdot \overline{BC}} = \overline{(\overline{A} + \overline{B} + C)} + \overline{(B + \overline{C})}$$



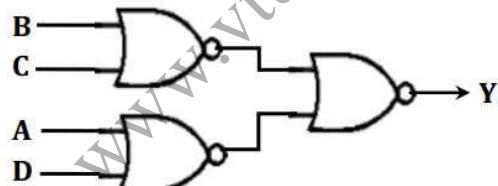
$$(ii) \quad Y = AB + AC + BD + CD = A(B + C) + D(B + C) = (B + C)(A + D)$$

NAND Realization:

$$Y = \overline{(B + C)(A + D)} = \overline{(B + C)} + \overline{(A + D)} = \overline{(\overline{B} \cdot \overline{C})} \cdot \overline{(\overline{A} \cdot \overline{D})}$$

**NOR Realization:**

$$Y = \overline{(B + C)(A + D)} = \overline{(B + C)} + \overline{(A + D)}$$



7. Simplify the given Boolean equation $Y = (A + \overline{B})(CD + E)$ and realize using NAND gates only.

June/July 2017, 15ELN15/25,

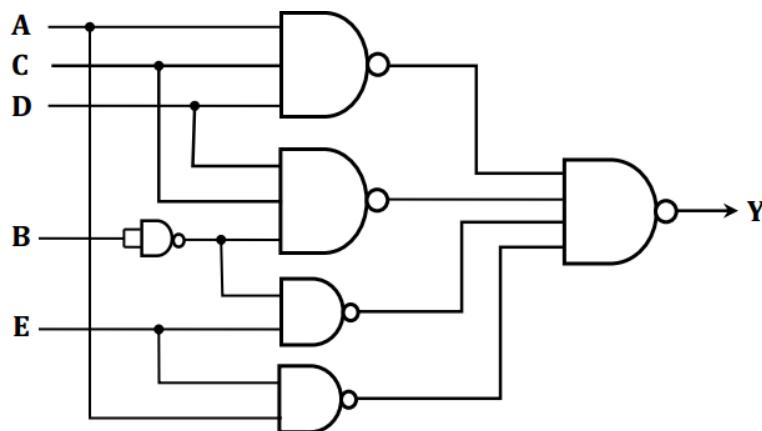
04 Marks

Solution:

$$Y = (A + \overline{B})(CD + E) = ACD + AE + \overline{B}CD + \overline{B}E$$

Realization using NAND gates:

$$Y = \overline{ACD + AE + \overline{B}CD + \overline{B}E} = \overline{\overline{ACD} \cdot \overline{AE} \cdot \overline{\overline{B}CD} \cdot \overline{\overline{B}E}}$$



8. Simplify $P = xy + xyz + xy\bar{z} + \bar{x}yz$. Implement using 2 inputs NAND gates.

Dec.2014/Jan.2015, 10ELN15/25, 06 Marks

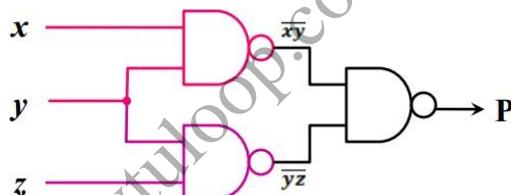
Solution:

$$P = xy + xyz + xy\bar{z} + \bar{x}yz = xy(1+z) + xy\bar{z} + \bar{x}yz = xy + xy\bar{z} + \bar{x}yz = xy(1 + \bar{z}) + \bar{x}yz$$

$$P = xy + \bar{x}yz = y(x + \bar{x}z) = y(x + \bar{x})(x + z) = xy + yz$$

Implementation Using NAND Gates:

$$P = xy + yz = \overline{\overline{xy} + yz} = \overline{\overline{xy}} \cdot \overline{yz}$$



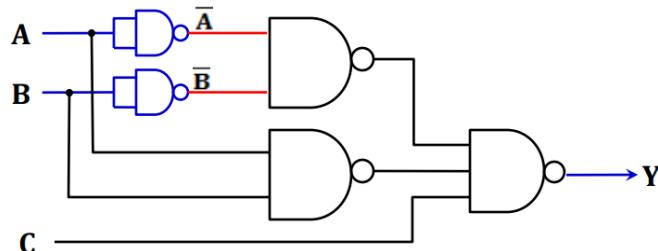
9. Simplify and realize the Boolean expression using two inputs NAND gates only $(A + \bar{B} + C)(\bar{A} + B + C)$.

Dec.2017/Jan.2018, 15ELN15/25, 05 Marks

Solution:

$$\begin{aligned} Y &= (A + \bar{B} + C)(\bar{A} + B + C) = A\bar{A} + AB + AC + \bar{A}\bar{B} + \bar{B}B + \bar{B}C + \bar{A}C + BC + CC \\ &= AB + AC + \bar{A}\bar{B} + \bar{B}C + \bar{A}C + BC + C = AB + AC + \bar{A}\bar{B} + \bar{B}C + \bar{A}C + C \\ &= AB + AC + \bar{A}\bar{B} + \bar{B}C + C = AB + AC + \bar{B}\bar{A} + C = AB + \bar{A}\bar{B} + C \end{aligned}$$

Implementation Using NAND Gates:

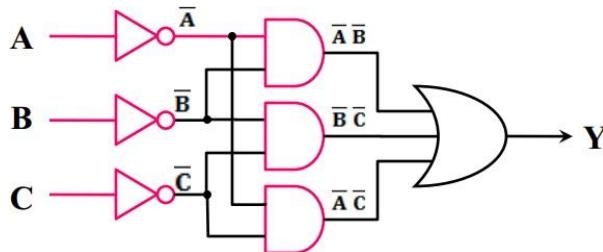


10. Write logic circuit using basic gates for the simplified expression: $(A + B)(B + C)(A + C)$.

Dec.2015/Jan.2016, 10ELN15/25, 04 Marks

Solution:

$$\begin{aligned} Y &= \overline{(A + B)(B + C)(A + C)} ; \text{ Apply De Morgan's Law} \\ &= \overline{A + B} + \overline{B + C} + \overline{A + C} ; \text{ Apply De Morgan's Law to individual terms} \\ &= \overline{A}\bar{B} + \bar{B}\bar{C} + \bar{A}\bar{C} \end{aligned}$$

Implementation Using Basic Gates:

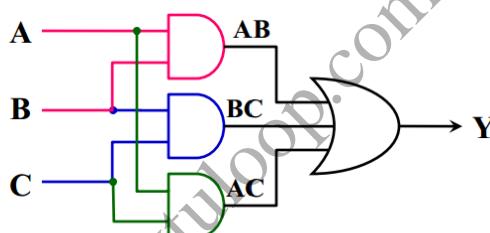
11. Simplify and realize using basic gates:

- i) $ABC + A\bar{B}C + AB\bar{C} + \bar{A}\bar{B}C$
ii) $(A+B)(\bar{A}+\bar{C})(\bar{B}+C)$

Dec.2014/Jan.2015, 14ELN15/25, 06 Marks

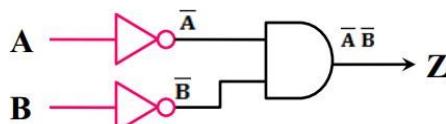
Solution:

$$\begin{aligned} i) \quad Y &= ABC + A\bar{B}C + AB\bar{C} + \bar{A}\bar{B}C \\ Y &= ABC + A\bar{B}C + AB\bar{C} + \bar{A}\bar{B}C = BC(A + \bar{A}) + A\bar{B}C + AB\bar{C} = BC + A\bar{B}C + AB\bar{C} \\ &= C(B + AB) + AB\bar{C} = C(B + \bar{B})(B + A) + AB\bar{C} = BC + AC + AB\bar{C} = BC + A(C + BC) \\ &= BC + A(C + \bar{C})(C + B) = AB + BC + AC \end{aligned}$$

Implementation Using Basic Gates: $Y = AB + BC + AC$ 

ii) $Z = (\bar{A} + B)(\bar{A} + \bar{C})(\bar{B} + C)$

$$\begin{aligned} Z &= (\bar{A} + B)(\bar{A} + \bar{C})(\bar{B} + C) \quad ; \text{Apply De Morgan's law to 1st term and multiply 2nd \& 3rd ms} \\ &= (\bar{A}\bar{B})(\bar{A}\bar{B} + \bar{A}C + \bar{B}\bar{C} + 0) = \bar{A}\bar{B}.\bar{A}\bar{B} + \bar{A}\bar{B}.\bar{A}C + \bar{A}\bar{B}.\bar{B}\bar{C} \\ &= \bar{A}\bar{B} + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} = \bar{A}\bar{B} + \bar{A}\bar{B}(C + \bar{C}) = \bar{A}\bar{B} \end{aligned}$$

Implementation Using Basic Gates: $Z = \bar{A}\bar{B}$ 

12. Simplify the following expressions and implement using only NAND Gates:

June/July.2014, 10ELN15/25,

08 Marks

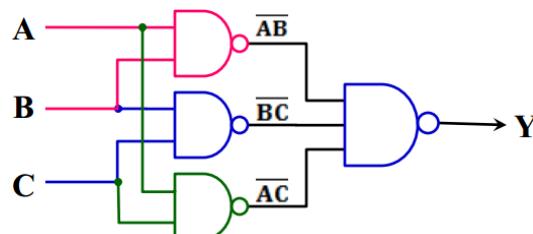
- a. $Y = ABC + A\bar{B}C + AB\bar{C} + \bar{A}\bar{B}C$
b. $Y = \overline{AB} + \overline{AC}$
c. $Y = A + \overline{AB}$

Solution:

$$\begin{aligned} a. \quad Y &= ABC + A\bar{B}C + AB\bar{C} + \bar{A}\bar{B}C = BC(A + \bar{A}) + A\bar{B}C + AB\bar{C} = BC + A\bar{B}C + AB\bar{C} \\ &= C(B + AB) + AB\bar{C} = C(B + \bar{B})(B + A) + AB\bar{C} = BC + AC + AB\bar{C} = BC + A(C + BC) \\ &= BC + A(C + \bar{C})(C + B) = AB + BC + AC \end{aligned}$$

Implementation Using NAND Gates:

$$Y = AB + BC + AC = \overline{\overline{AB} + BC + AC} = \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{AC}}$$

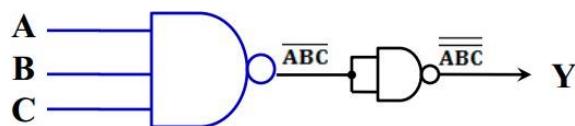


b. $Y = \overline{\overline{AB} + \overline{AC}}$; Apply De Morgan's law

$$Y = \overline{\overline{\overline{AB}} \cdot \overline{\overline{AC}}} = \overline{AB} \cdot \overline{AC} = ABC$$

Implementation Using NAND Gates:

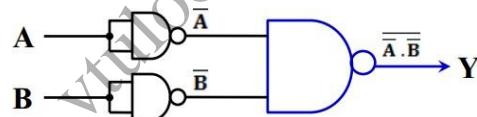
$$Y = ABC = \overline{\overline{ABC}}$$



c. $Y = A + \overline{AB} = (A + \overline{A})(A + B) = A + B$

Implementation Using NAND Gates:

$$Y = A + B = \overline{\overline{A} + \overline{B}} = \overline{\overline{A} \cdot \overline{B}}$$

**Half Adder:**

- The combinational circuit that performs addition of two bits is called **Half Adder**. The half adder operation needs two binary inputs: **Augend & Addend** and provides two binary outputs: **Sum & Carry**.

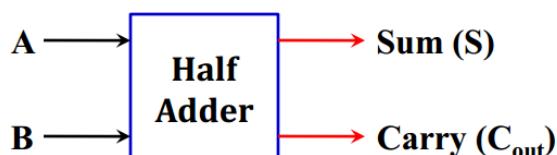
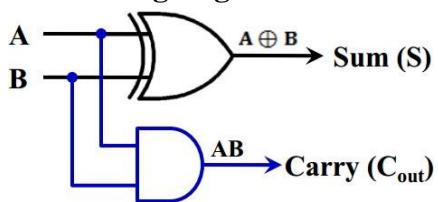
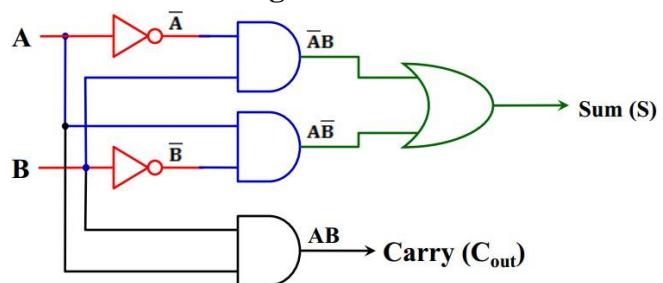


Fig: Block Diagram

Truth Table:

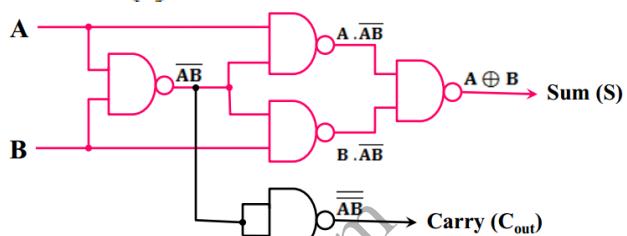
Inputs		Outputs	
A	B	Sum (S)	Carry (C _{out})
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Expressions: Sum (S) = $\overline{AB} + \overline{AB} = A \oplus B$
Carry (C_{out}) = AB

Circuit Diagram:**Using Logic Gates:****Using Basic Gates:****Half Adder using NAND Gates:****Truth Table:**

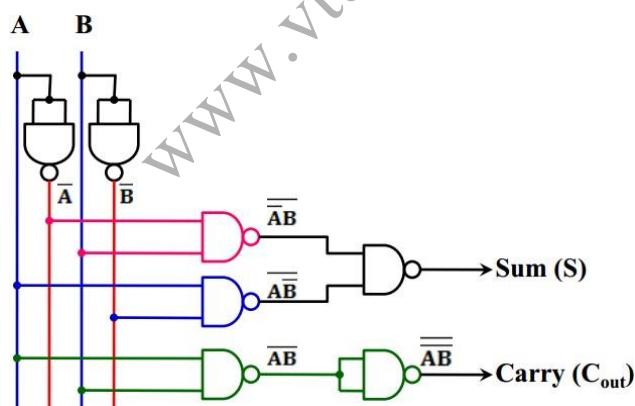
Inputs		Outputs	
A	B	Sum (S)	Carry (C _{out})
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\text{Sum (S)} = A \oplus B \text{ & Carry} = AB = \overline{\overline{AB}}$$



$$\text{Sum} = \overline{AB} + \overline{A}\overline{B} = \overline{\overline{AB}} + \overline{AB} = \overline{AB} \cdot \overline{AB}$$

$$\text{Carry} = AB = \overline{AB}$$



Full Adder:

- The combinational circuit that performs addition of three bits is called **Full Adder**. The full adder has three binary inputs: **Augend**, **Addend** & **Carry** from previous operation and provides two binary outputs: **Sum** and **Carry**.

Block Diagram:**Truth Table:**

Inputs			Outputs	
A	B	C _{in}	S	C _{out}
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

A & B :Represents the two significant bits to be added.

C_{in} :Represent the carry from previous lower significant position.

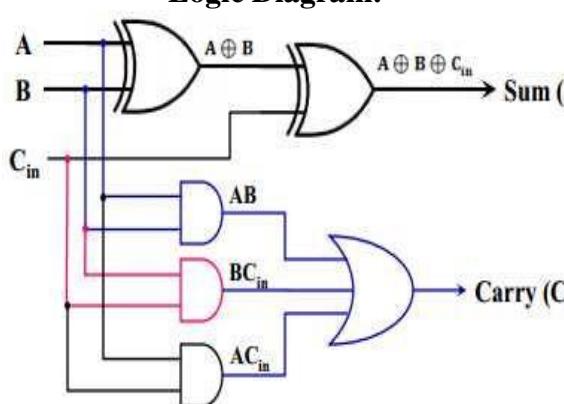
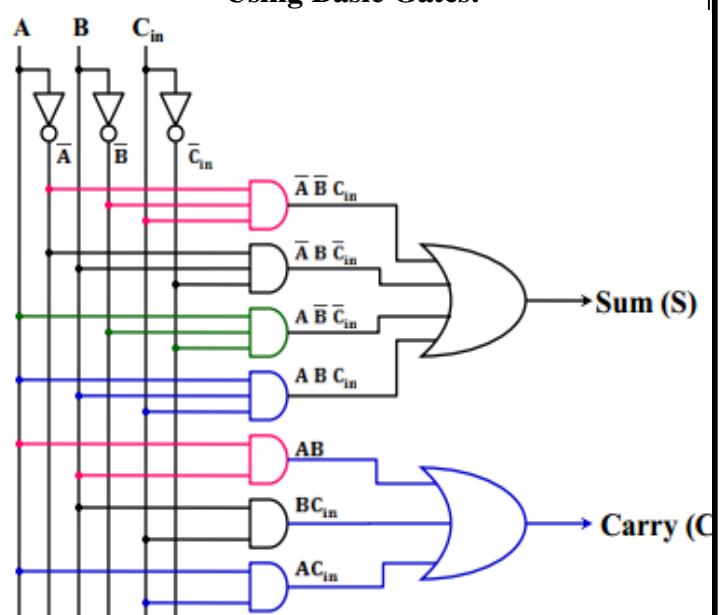
S :Sum bit

C_{out} :Carry to next higher significant position.

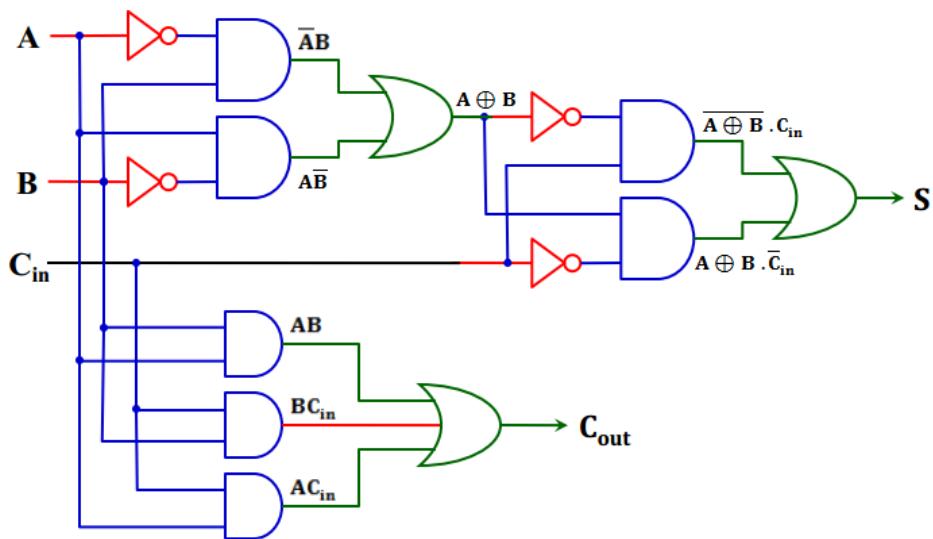
Expressions:

$$\begin{aligned} S &= \overline{A} \overline{B} C_{in} + \overline{A} B \overline{C}_{in} + A \overline{B} \overline{C}_{in} + A B C_{in} = C_{in}(\overline{A} \overline{B} + A B) + \overline{C}_{in}(\overline{A} B + A \overline{B}) \\ &= C_{in}(A \oplus B) + \overline{C}_{in}(A \oplus B) = A \oplus B \oplus C_{in} \end{aligned}$$

$$\begin{aligned} C_{out} &= \overline{A} B C_{in} + A \overline{B} C_{in} + A B \overline{C}_{in} + A B C_{in} = B C_{in}(\overline{A} + A) + A \overline{B} C_{in} + A B \overline{C}_{in} \\ &= BC_{in} + A\overline{B}C_{in} + AB\overline{C}_{in} = C_{in}(B + A\overline{B}) + AB\overline{C}_{in} = C_{in}(B + A)(B + \overline{B}) + ABC\overline{C}_{in} \\ &= BC_{in} + AC_{in} + AB\overline{C}_{in} = BC_{in} + A(C_{in} + B\overline{C}_{in}) = BC_{in} + A(C_{in} + B)(C_{in} + \overline{C}_{in}) \\ &= BC_{in} + AC_{in} + AB = AB + BC_{in} + AC_{in} \end{aligned}$$

Logic Diagram:**Using Basic Gates:**

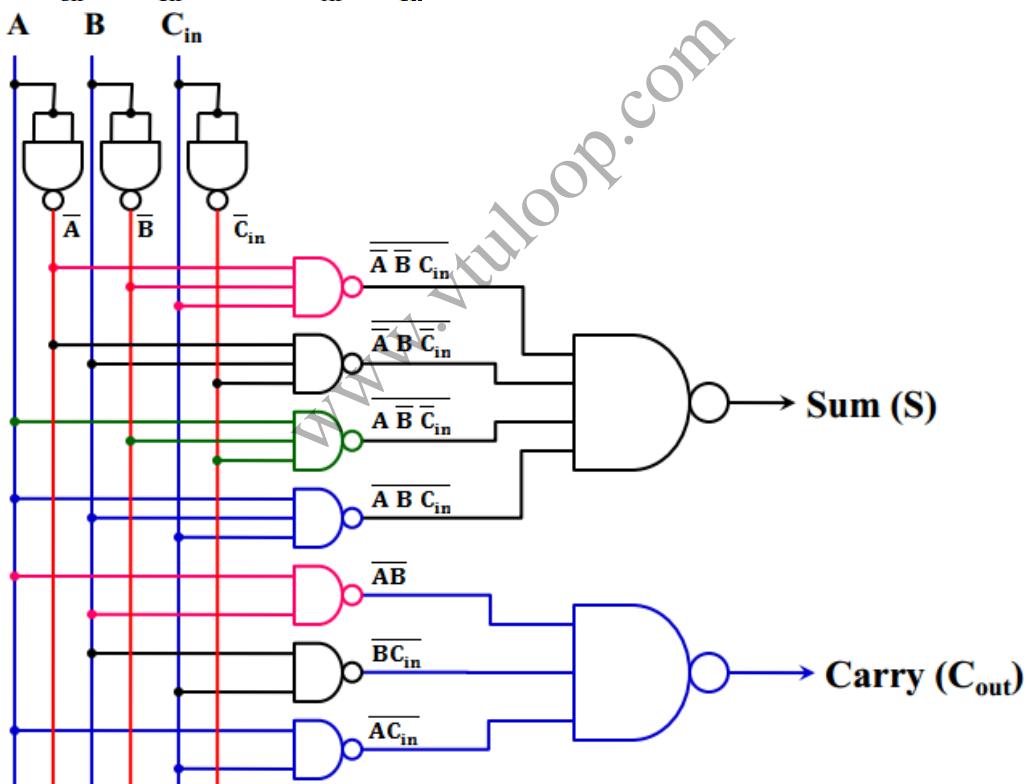
$$S = \overline{A} \overline{B} . C_{in} + A \oplus B . \overline{C}_{in} \text{ and } C_{out} = AB + BC_{in} + AC_{in}$$



Realization Using NAND Gates:

$$S = \overline{\overline{A} \overline{B} C_{in}} + \overline{\overline{A} B \overline{C}_{in}} + \overline{A \overline{B} \overline{C}_{in}} + \overline{A B C_{in}} = \overline{\overline{A} \overline{B} C_{in}} \cdot \overline{\overline{A} B \overline{C}_{in}} \cdot \overline{A \overline{B} \overline{C}_{in}} \cdot \overline{A B C_{in}}$$

$$C = \overline{AB + BC_{in} + AC_{in}} = \overline{AB} \cdot \overline{BC_{in}} \cdot \overline{AC_{in}}$$



Full Adder Using Two Adders:

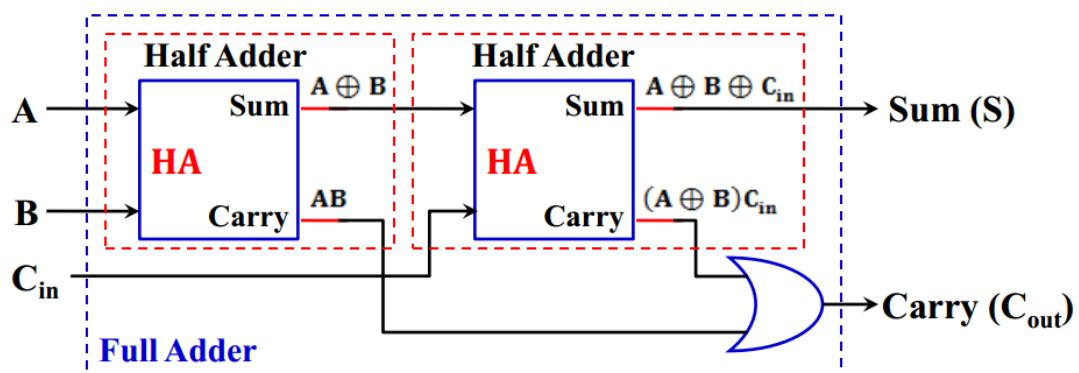


Fig: Block Diagram

Expressions:

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = \bar{A}B C_{in} + A \bar{B} C_{in} + AB \bar{C}_{in} + AB C_{in} = C_{in}(\bar{A}B + A \bar{B}) + AB(\bar{C}_{in} + C_{in})$$

$$C_{out} = (A \oplus B)C_{in} + AB$$

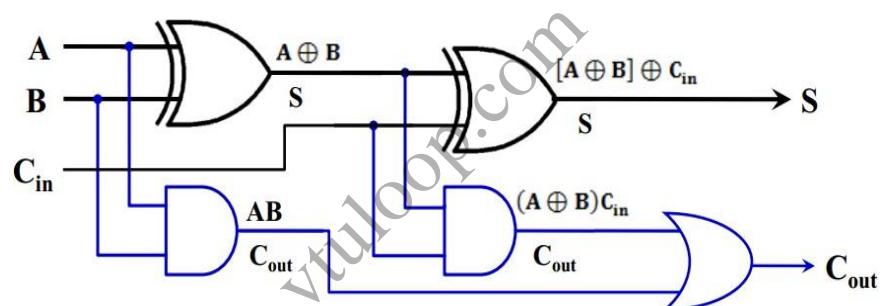


Fig: Realization Using Logic Gates

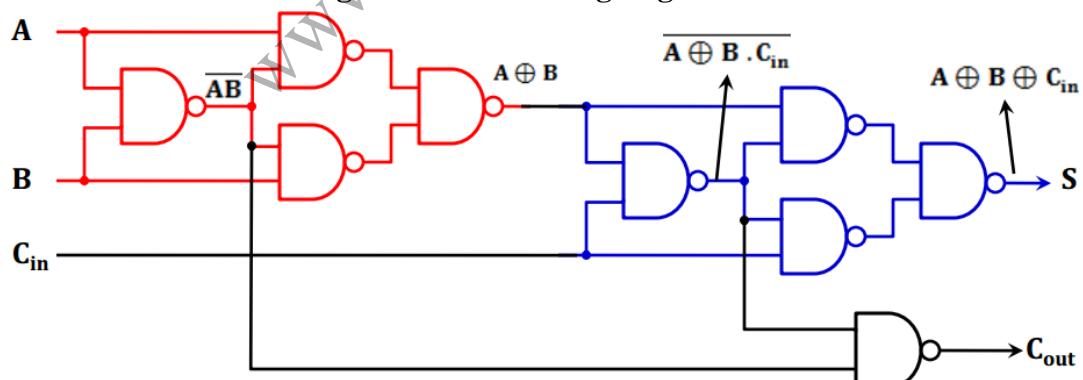


Fig: Realization Using NAND Gates

Problems:

1. Design a combinational circuit that has 3 inputs and one output. The output is at logic 1 or logic high only when even number of inputs are at logic 1.

Expression:

$$Y = \overline{A}BC + A\overline{B}C + AB\overline{C}$$

Truth Table:

Inputs			Outputs
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Logic Diagram Using Basic Gates:

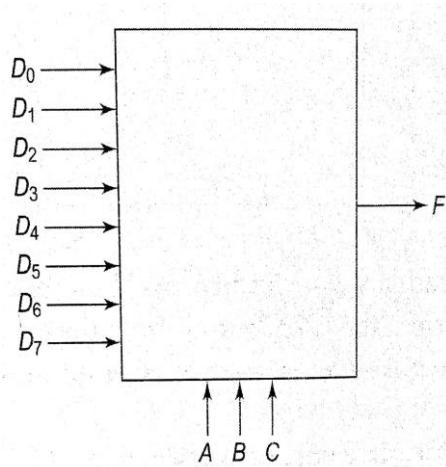
2. Design a logic circuit using basic gates with three inputs A, B & C and one output Y that goes low only when A is high and B and C are different.

Expression:**Truth Table:**

Inputs			Outputs
A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Multiplexers

The objective of a multiplexer is to select one signal from a group of 2^n inputs, to be an output on a single output line. For example 8 –to-1 multiplexer (mux) is as shown in fig. Lines D_0, \dots, D_7 are the data input lines and F is the output line. Lines A, B and C are called the select lines. They are interpreted as a three bit binary number, which is used to choose one of the D lines to be output on the line F .

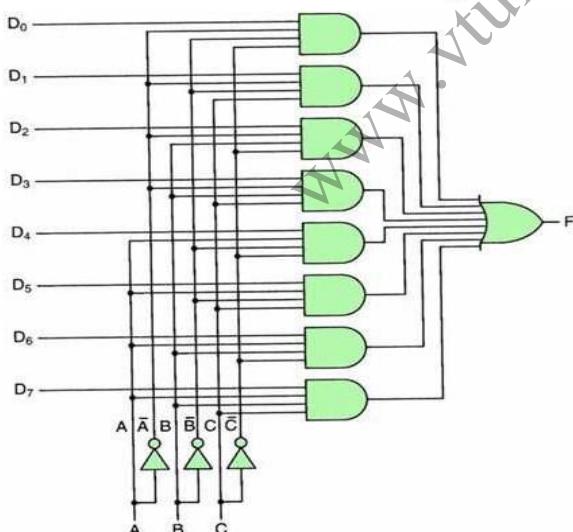


Implementation

A mux can be designed with a regular pattern of AND and OR gates, as shown in fig.

8-to-1 Multiplexer

Logic circuit



Truth Table

A	B	C	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

Logic equation of the circuit:

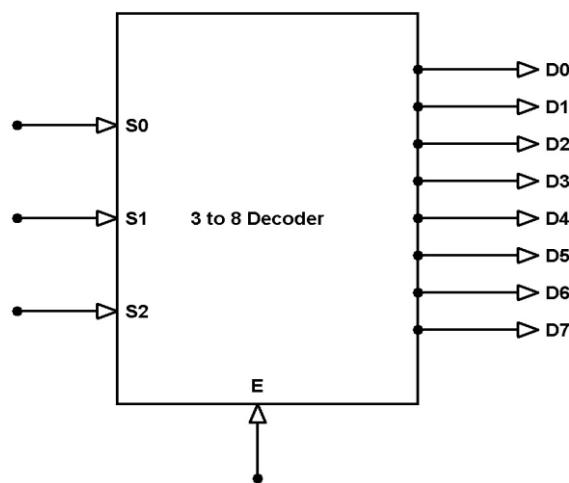
$$Y = A'B'C'.D_0 + A'B'C.D_1 + A'BC'.D_2 + \dots + ABC'.D_6 + ABC.D_7$$

Application

In all types of digital system applications, multiplexers find its immense usage. Since these allows multiple inputs to be connected independently to a single output, these are found in variety of applications including data routing, logic function generators, control sequencers, parallel-to-serial converters, etc.

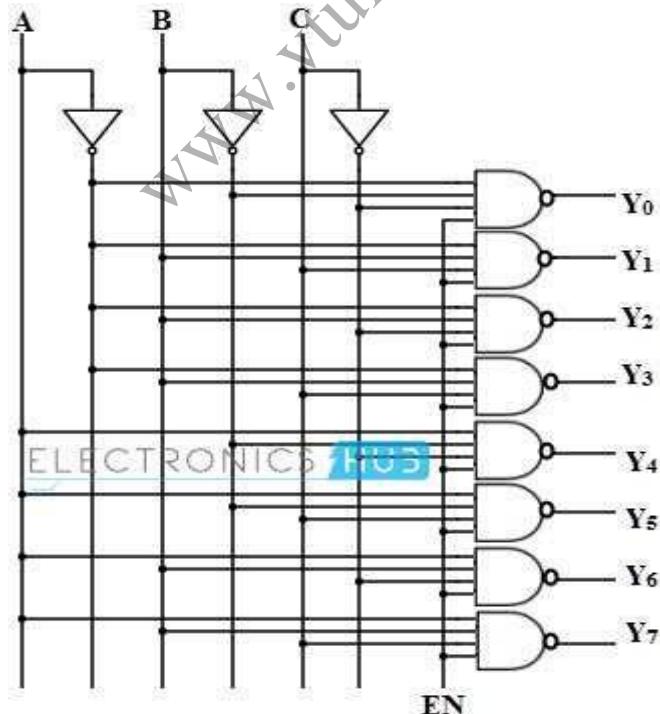
Decoders

The objective of the decoder is to decode an n-bit binary number producing a signal on one of 2^n output lines. Figure shows an 3-to-8 decoder.



Implementation

A Decoder is often implemented with an additional input called an ‘enable’ line. When the line is enabled, the circuit is a decoder. When it is disabled, all the outputs are 0. The same circuit can be used as a de-multiplexer, which directs a single data input line to one of 2^n output lines, depending on the values of n select lines. Fig shows 3 to 8 decoder implementation.



Inputs				Outputs							
EN	A	B	C	Y ₇	Y ₆	Y ₅	Y ₄	Y ₃	Y ₂	Y ₁	Y ₀
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

Applications: Decoder memory address for reads and writes to random access memory.

Introduction to Sequential Circuits

Differences Between Combinational Circuits and Sequential Circuits:

Combinational Logic Circuits:	Sequential Logic Circuits:
<ol style="list-style-type: none"> Output depends only on external inputs Memory is not present No feedback from output to input Terminal behavior is represented by truth table No time constraints on inputs and output changes 	<ol style="list-style-type: none"> Output depends on current inputs and previous outputs Memory is present There is feedback from output to input Terminal behavior is represented by timing diagram Time constraints on inputs and output changes such as setup and hold times, minimum pulse width.



Fig: Block Diagram of Combinational Circuit

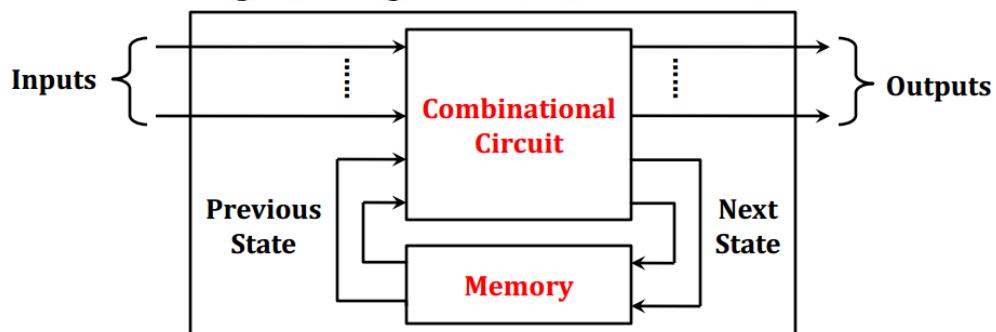


Fig: Block Diagram of Sequential Circuit

Flip-Flop:

- It is a bi-stable sequential circuit, i.e its output has two stable states: **logic 1 and logic 0**. Flip-flops consists of basic bi-stable element with appropriate logic in order to control its state. It requires a clock signal to affect the changes in inputs to output. Clock signal is a rectangular signal with 50% duty cycle as shown below.
- The flip-flop changes its output either at the rising/leading edge or at the falling/trailing edge of the clock signal according to the status of the flip-flop inputs present at that time. The two outputs of flip-flop Q and \bar{Q} are always complement of each other.

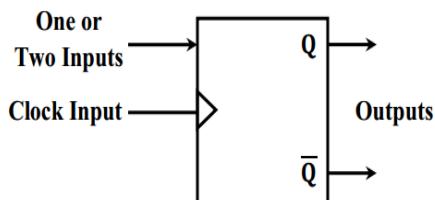


Fig: Block Diagram

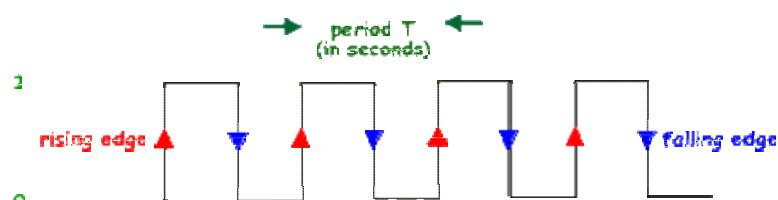
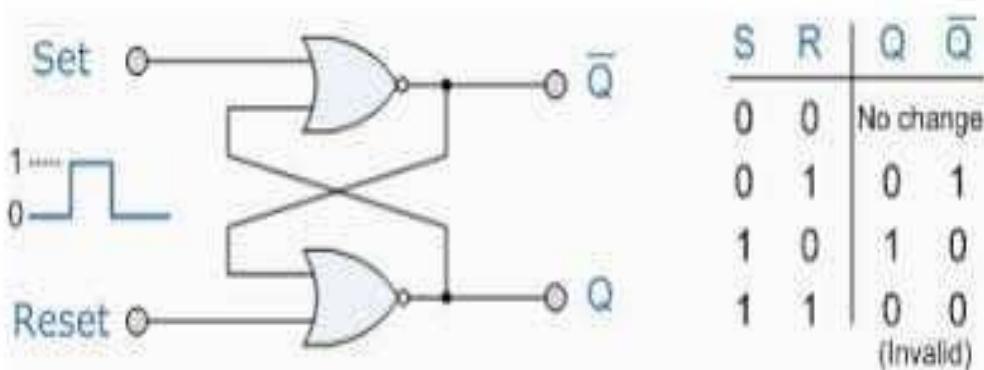


Fig: Clock Signal

- The different types of flip-flops are:
 1. SR Flip-Flop
 2. JK Flip-Flop
 3. D Flip-Flop
 4. T Flip-Flop
 5. Master – Slave Flip-Flop

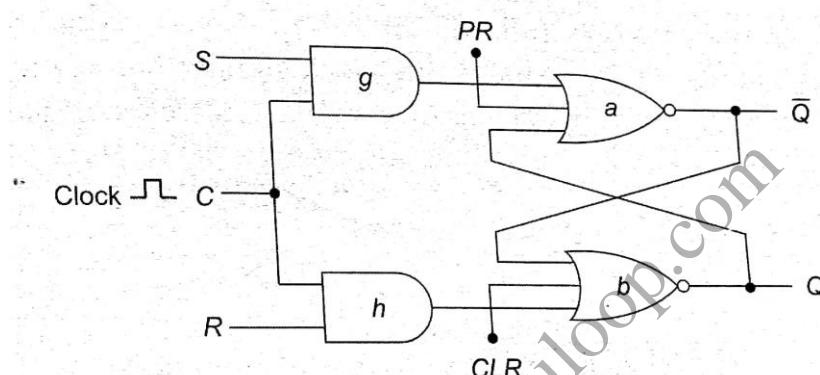
Application:

1. It can be used as a memory element.
2. It can be used to eliminate key de-bounce.
3. It is used as a basic building block in sequential circuits such as counters and registers.
4. It can be used as delay element.

SR Flip Flop

Operation:**Case 1: When S = 0, R = 0**then $Q = Q$ & $\bar{Q} = \bar{Q}$ i.e., No change in output**Case 2: When S = 0, R = 1**then $Q = 0$ & $\bar{Q} = 1$ i.e., Reset the output**Case 3: When S = 1, R = 0**then $Q = 1$ & $\bar{Q} = 0$ i.e., Set the output**Case 4: When S = 1, R = 1**then $Q = 1^*$ & $\bar{Q} = 1^*$ i.e., Forbidden State

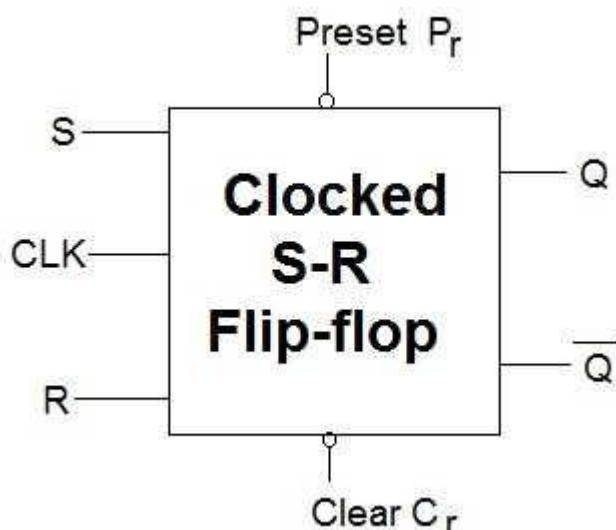
The clock can be applied to a latch so that change in the latch's value can only occur when the clock is in the "1"(high) state. The result is a clocked SR latch shown in fig .



When C=0, the output of both AND gate is zero. So Q cannot change: the latch is non operational.

When C=1, output of gate g is 1. S=S and output of gate h is 1. R=R, the latch is now operational.

The term level triggered is applied to this type of latch to indicate that its ability to change value depends on the level (low or high) of the clock signal.

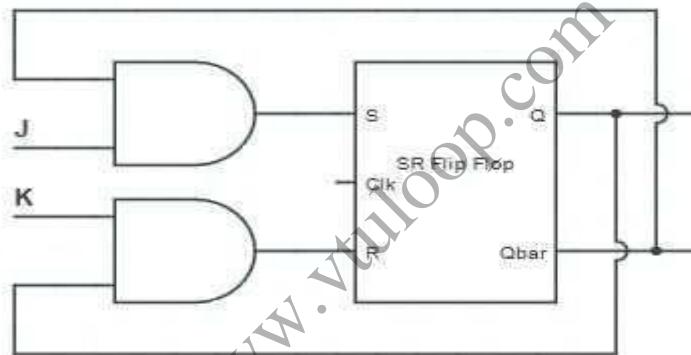
Clear and Preset

CLK	Preset	Clear	S	R	Q	\bar{Q}
X	1	0	X	X	1	0
X	0	1	X	X	0	1
0	0	0	X	X	Q	\bar{Q}
1	0	0	0	0	Q	\bar{Q}
1	0	0	0	1	0	1
1	0	0	1	0	1	0
1	0	0	1	1	0*	0*

When clear and preset are 0 ,the circuit behaves as an SR flip flop.

JK Flip Flop

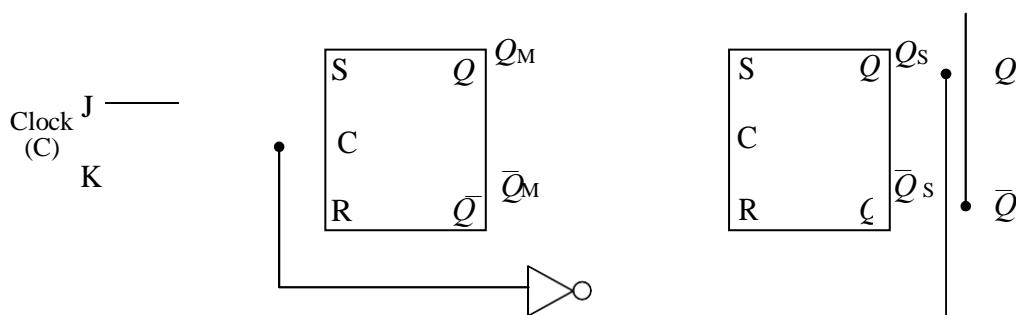
SRFF is converted is converted to JKFF by feeding \bar{Q} to the upper AND and lower AND.



The outputs of various combinations of inputs J,K when a pulse is applied.

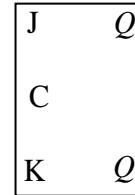
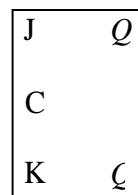
JK Inputs		SR Inputs		Outputs	
J	K	S	R	Q	\bar{Q}
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

Master-slave JK Flip-flop:



Truth Table:

Inputs			Outputs	
J	K	C	Q^+	\bar{Q}^+
0	0	0	Q	\bar{Q}
0	1	0	0	1
1	0	0	1	0
1	1	0	\bar{Q}	Q
X	X	0	Q	\bar{Q}



Logic Symbols:

- The *master-slave JK flip-flop* allow its two information lines to be simultaneously 1, results in toggling the output of the flip-flop, ie if present state=0, then the next state=1, and if present state=1, then next state=0.
- J and K inputs are analogous to S and R inputs of the master-slave SR flip-flop. Additional two and-gates are used to sense and steer the state of the slave.

Working:

Case-I: When $C = 0$, $J = 1$ and $K = 1$,

Assume $Q = 1$, $\bar{Q} = 0$, ie in its 1-state (Set), then $Q = Q_s = 1$, and $\bar{Q} = \bar{Q}_s = 0$

Output of J-input AND-gate = 0 = S, since inputs are: $J = 1$, $\bar{Q} = 0$, and

Output of K-input AND-gate = 1 = R, since inputs are: $K = 1$, $Q = 1$,

Then $S = 0$, and $R = 1$, but master doesn't respond since $C = 0$.

When $C = 0$ to 1; Master enters into reset state, but slave is still in set state, since slave is disabled.

When $C = 1$ to 0; Master transferred to slave, then new state master-slave JK flip-flop is 0-state.

Thus, output of the flip-flop is toggled.

Assume $Q = 0$, $\bar{Q} = 1$, ie in its 0-state (Reset), then $Q = Q_s = 0$, and $\bar{Q} = \bar{Q}_s = 1$

Output of J-input AND-gate = 1 = S, Since inputs are: $J = 1$, $\bar{Q} = 1$, and

Output of K-input AND-gate = 0 = R, Since inputs are: $K = 1$, $Q = 0$,

Then $S = 1$, and $R = 0$, but master doesn't respond since $C = 0$.

When $C = 0$ to 1; Master enters into set state, but slave is still in reset state, since slave is disabled.

When $C = 1$ to 0; Master transferred to slave, then new state master-slave JK flip-flop is 1-state.

Thus, output of the flip-flop is toggled.

Case-2: When J = 1 and K = 0,

Assume flip-flop is in 1-state, ie $Q = Q_s = 1$, and $\bar{Q} = \bar{Q}_s = 0$.

When C = 0, slave is enabled and is in 1-state, then master must also in its 1-state, ie $Q_M = 1$, $\bar{Q}_M = 0$.

Output of J-input AND-gate = 0 = S, Since inputs are: J = 1, $\bar{Q}_s = 0$, and

Output of K-input AND-gate = 0 = R, Since inputs are: K = 0, $Q_s = 1$,

Then S = 0, and R = 0, but master doesn't respond since C = 0.

When C = 0 to 1; Master doesn't change remains in 1-state, ie $Q_M = 1$, $\bar{Q}_M = 0$, since S = 0, R = 0.

When C = 1 to 0; Master transferred to slave, slave also in its 1-state, then output itself in 1-state.

Thus, output of the flip-flop remains in its 1-state.

Assume flip-flop is in 0-state, ie $Q = Q_s = 0$, and $\bar{Q} = \bar{Q}_s = 1$,

When C = 0, slave is enabled and is in 0-state, then master must also in 0-state, ie $Q_M = 0$, $\bar{Q}_M = 1$.

Output of J-input AND-gate = 1 = S, Since inputs are: J = 1, $\bar{Q}_s = 1$, and

Output of K-input AND-gate = 0 = R, Since inputs are: K = 0, $Q_s = 0$,

Then S = 1, and R = 0, but master doesn't respond since C = 0.

When C = 0 to 1; Master enters into 1-state, ie $Q_M = 1$, $\bar{Q}_M = 0$, since S = 1, R = 0.

When C = 1 to 0; Master transferred to slave, slave enters into 1-state, then output itself in 1-state.

Thus, output of the flip-flop enters into 1-state.

Thus, when J = 1 and K = 0, output of the flip-flop remains or enters into 1-state (set).

Case-3: When J = 0 and K = 1,

Assume flip-flop is in 0-state, ie $Q = Q_s = 0$, and $\bar{Q} = \bar{Q}_s = 1$

When C = 0, slave is enabled and is in 0-state, then master must also in 0-state, ie $Q_M = 0$, $\bar{Q}_M = 1$.

Output of J-input AND-gate = 0 = S, Since inputs are: J = 0, $\bar{Q}_s = 1$, and

Output of K-input AND-gate = 0 = R, Since inputs are: K = 1, $Q_s = 0$,

Then S = 0, and R = 0, but master doesn't respond since C = 0.

When C = 0 to 1; Master doesn't change remains in 0-state, ie $Q_M = 0$, $\bar{Q}_M = 1$, since S = 0, R = 0.

When C = 1 to 0; Master transferred to slave, slave also in its 0-state, then output itself in 0-state.

Thus, output of the flip-flop remains in its 0-state.

Assume flip-flop is in 1-state, ie $Q = Q_s = 1$, and $\bar{Q} = \bar{Q}_s = 0$

When C = 0, slave is enabled and is in 1-state, then master must also in 1-state, ie $Q_M = 1$, $\bar{Q}_M = 0$.

Output of J-input AND-gate = 0 = S, Since inputs are: J = 0, $\bar{Q}_s = 0$, and

Output of K-input AND-gate = 1 = R, Since inputs are: K = 1, $Q_s = 1$,

Then S = 0, and R = 1, but master doesn't respond since C = 0.

When C = 0 to 1; Master enters into 0-state, ie $Q_M = 0$, $\bar{Q}_M = 1$, since S = 0, R = 1.

When C = 1 to 0; Master transferred to slave, slave enters into 0-state, then output itself in 0-state.

Thus, output of the flip-flop enters into 0-state.

Thus, when J = 0 and K = 1, output of the flip-flop remains or enters into 0-state (reset).

Case-4: When J = 0 and K = 0,

Assume flip-flop is in 0-state, ie $Q = Q_s = 0$, and $\bar{Q} = \bar{Q}_s = 1$.

When C = 0, slave is enabled and is in 0-state, then master must also in 0-state, ie $Q_M = 0$, $\bar{Q}_M = 1$.

Output of J-input AND-gate = 0 = S, Since inputs are: J = 0, $\bar{Q}_s = 1$, and

Output of K-input AND-gate = 0 = R, Since inputs are: K = 0, $Q_s = 0$,

Then S = 0, and R = 0, but master doesn't respond since C = 0.

When C = 0 to 1; Master doesn't change remains in 0-state, ie $Q_M = 0$, $\bar{Q}_M = 1$, since S = 0, R = 0.

When C = 1 to 0; Master transferred to slave, slave also in its 0-state, then output itself in 0-state.

Thus, output of the flip-flop remains in its 0-state.

Assume flip-flop is in 1-state, ie $Q = Q_s = 1$, and $\bar{Q} = \bar{Q}_s = 0$.

When C = 0, slave is enabled and is in 1-state, then master must also in 1-state, ie $Q_M = 1$, $\bar{Q}_M = 0$.

Output of J-input AND-gate = 0 = S, Since inputs are: J = 0, $\bar{Q}_S = 0$, and

Output of K-input AND-gate = 1 = R, Since inputs are: K = 0, $Q_S = 1$,

Then S = 0, and R = 1, but master doesn't respond since C = 0.

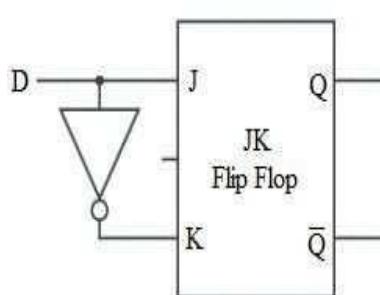
When C = 0 to 1; Master doesn't change remains in its 1-state, ie $Q_M = 1$, $\bar{Q}_M = 0$, since S = 0, R = 0.

When C = 1 to 0; Master transferred to slave, slave enters into 1-state, then output itself in 1-state.

Thus, output of the flip-flop remains in its 1-state.

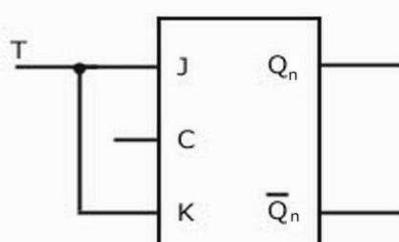
Thus, when J = 0 and K = 0, output of the flip-flop remains in same state as present state

D and T flip flop action is achieved by a JK flip flop



Truth Table:

Inputs		Outputs	
D	C	Q^+	\bar{Q}^+
0	↑	0	1
1	↑	1	0
X	0	Q	\bar{Q}
X	1	Q	\bar{Q}

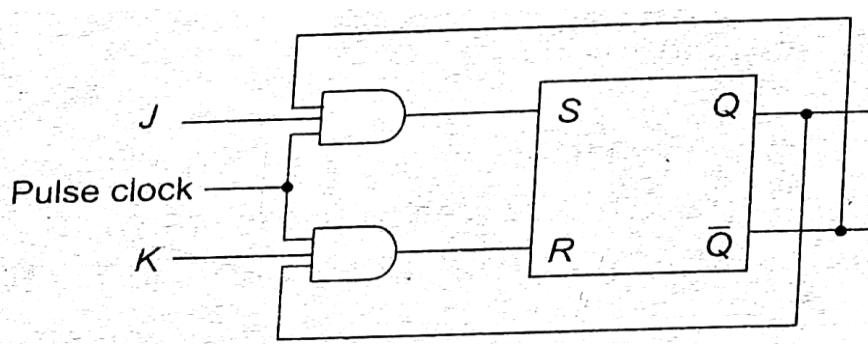


Truth Table:

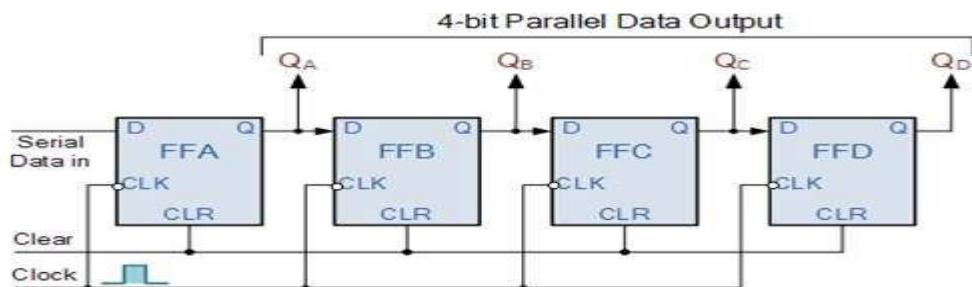
Inputs		Outputs	
D	C	Q^+	\bar{Q}^+
0	↓	0	1
1	↓	1	0
X	0	Q	\bar{Q}
X	1	Q	\bar{Q}

Shift Register

Consider the following circuit shown in fig, containing two RS flip flops. When a clock pulse occurs, the value of the left flip flop is copied to the right flip flop. This circuit is known as a **shift register**.



The working principle of a shift register using D flip flop is as shown in fig.



Let the input sequence be $A_1, A_2, A_3, A_4 = 1101$

The initial state of the register be Q_0, Q_1, Q_2, Q_3

CLK	Serial input	Parallel outputs			
		Q _A	Q _B	Q _C	Q _D
-	-	0	0	0	0
1	1	1	0	0	0
2	0	0	1	0	0
3	1	1	0	1	0
4	1	1	1	0	1

The register is named serial in, parallel out shift register.

Counters

A counter is a sequential circuit that counts the number of input pulses. A counter that counts in terms of binary is called a binary counter. The count output of an n-bit binary counter is 2^n states. Therefore , it can count from 0 to (2^n-1) .

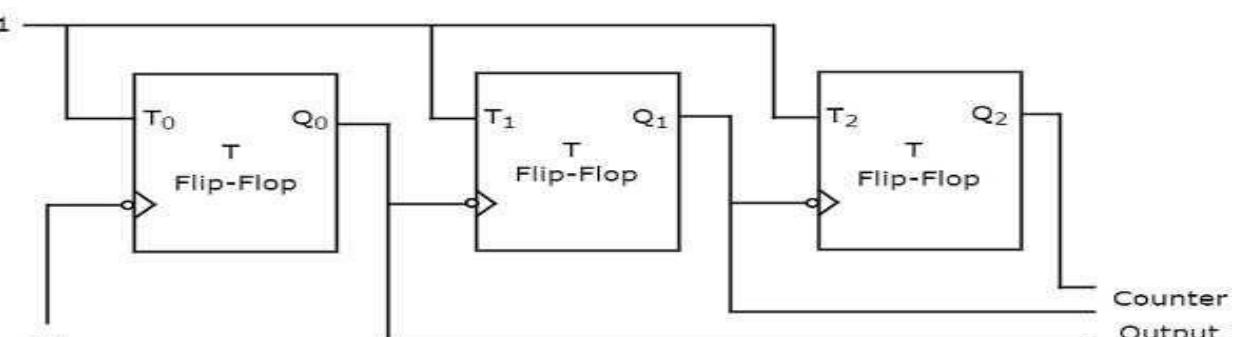
There are two types of counters

1. Asynchronous counter
2. Synchronous counter

In case of an asynchronous counter, the flip flops are clocked sequentially ,while in a synchronous counter, they are clocked simultaneously. Therefore, time delays of each flip flop get added and their count action is much slower than in synchronous counters.

Asynchronous counter

T flip flop are used in these counters. A 3 bit binary ripple counter is shown in fig. The small circles at the back of the clock input stand for the fact that these are triggered by the trailing edge of the input pulse(1 going 0). For toggling, all T inputs are kept high(1). From left to right, the first T flip flop receives pulses from the counter. The other two receive pulses from the output of the preceding T flip flop. Number of states of a counter is referred to as its modulus(m). For an n-bit counter, $m \leq 2^n$

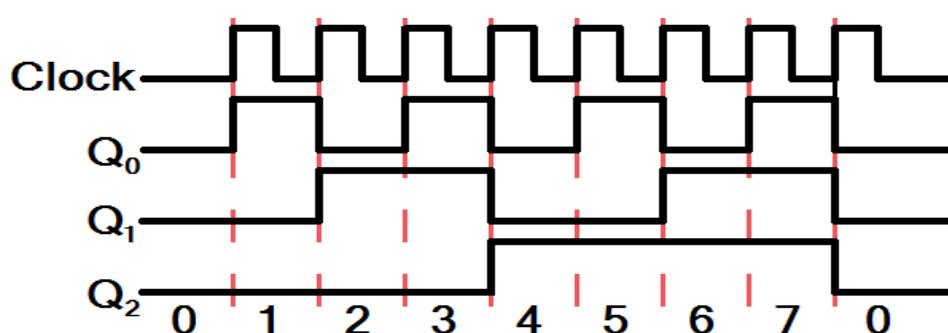


3-bit ripple counter

Truth table of 3-bit ripple counter

Counter State	Q_2	Q_1	Q_0
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

The timing diagram of the counter is presented in fig, It is seen that pulses frequency gets divided by 2 at each stage.



Timing diagram of a 3-bit ripple counter

Basic Communication Systems

Definition: Communication system is defined as the process of exchanging the information between source and destination.

Elements of communication systems are: Information, Transmitter, Communication channel or Medium and Receiver.

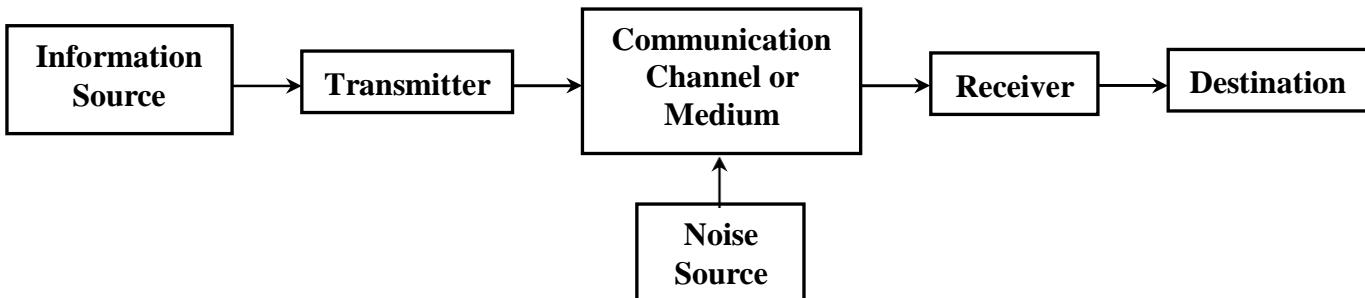


Fig: Elements of Communications Systems

Information Source: Communication systems communicate messages. The messages are come from the information sources. It may contain human voice, picture, code, data, music and their combinations.

Transmitter:

- The transmitter is a collection of electronic circuits designed to convert the information into a signal suitable for transmission over a given communication medium.
- If the messages are non-electrical, immediate transmission is not possible, such messages need to be coded or processed before transmission and also require suitable transducer to convert them into electrical signal.
- Most of the transmitters have built-in amplifier circuits. These circuits amplify the incoming signals before transmission. Built-in circuits has decoder, encoder, transducers etc.

Communication Channel: The communication channel is the medium by which the electronic signal is transmitted from one place to another. The communication medium can be a pair of conducting wire, coaxial cable, optical fiber cable or free space.

Noise:

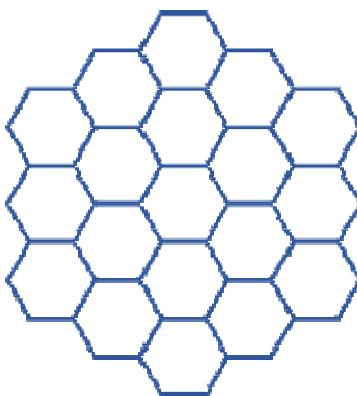
- Noise is a random, undesirable electric energy that enters the communication system via the medium and interfaces with the transmitted energy. Some noise is also produced in the receiver. Noise can be either natural or man-made.
- Natural noise includes noise produced in the nature. Example: From lighting during rainy season, Noise due to radiations produced by the sun and other stars.
- Man-made noise is the noise produced by electric ignition systems of cars, electric motors, fluorescent lights etc.
- Noise can't be completely eliminated, but can be reduced.

Receiver:

- It is a collection of electronic circuits designed to convert the signal back to the original information. It consists of amplifier, detector, mixer, oscillator, transducers etc.

Principle of operation of Mobile communication

A cellular /mobile system provides standard telephone operation by full duplex two way radio at remote locations. It provides a wireless connection to the Public switched Telephone Network (PSTN) from any user location within the radio range of the system.



Cell area

The basic concept behind the cellular radio system is that rather than serving given geographical area within a single transmitter and receiver, the system divides service area into many small areas known as cell as shown in fig.

The typical cell covers only several square kilometers and contain its own receiver and low power transmitter. The cell area shown in fig is an ideal hexagon. However, in a reality they will have circular or other geometric shapes. These areas may overlap and cells may be of different sizes.

A basic cellular system consists of mobile stations, base stations and a mobile switching centre(MSC).The MSC is also known as Mobile Telephone Switching Office(MSTO).

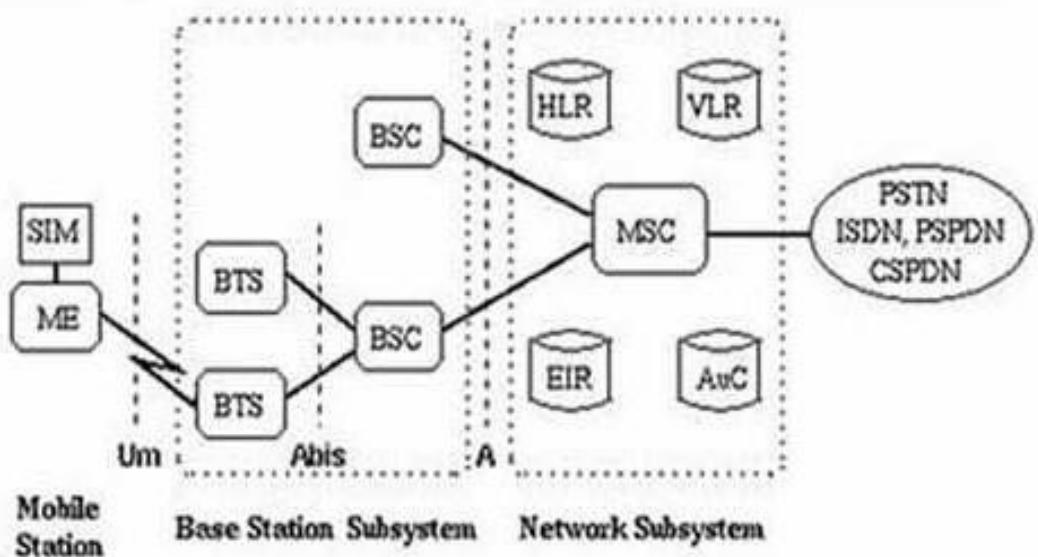
The MSTO controls the cells and provides the interface between each cell and the main telephone office. Each mobile communicates via radio with one of the base stations and may handed off to any other base station throughout the duration of the call.

Each mobile station consists of a transceiver, an antenna and control unit circuit. The base stations consists of several transmitters and receivers which simultaneously handle full duplex communication and generally have towers which support several transmitting and receiving antennas.

The base station serves as a bridge between all mobile users in the cell and connects the simultaneous mobile calls via telephone lines or microwave link to the MSC.

The MSC co-ordinates the activities of all base stations and connects the entire cellular system to the PSTN. Most cellular systems also provide a service known as roaming.

A simple block diagram representing working of mobile networks through GSM is as shown in fig. The cellular system operates in the 800-900 MHz range. The new digital cellular system has even greater capacity. Some of these systems operates in 1.7-1.8GHz bands.

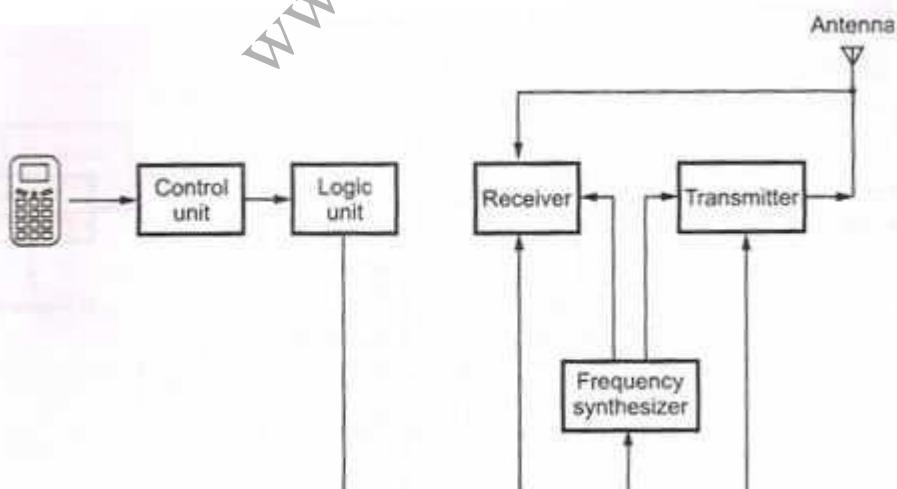


SIM Subscriber Identity Module BSC Base Station Controller MSC Mobile service switching center
 ME Mobile Equipment HLR Home Location Register EIR Equipment Identity Register
 BTS Base Transceiver station VLR Visitor Location Register AuC Authentication Center

Block diagram of GSM System

Cellular Telephone Unit

Figure shows block diagram of a cellular mobile radio unit.



It unit consists of five major parts

1. Transmitter
2. Receiver
3. Synthesizer
4. Logic unit
5. Control unit

Transmitter

It is the low power FM transmitter operating in a frequency range of 825 to 845 MHz. There is 66630 KHz transmit channel. Transmitter produces a deviation of $\pm 12\text{MHz}$. The modulated output is translated upto final transmitter frequency with the help of mixer, whose second input also comes from frequency Synthesizer. The unique feature of high power translator is that output is controllable by the cell site and MTSO (Mobile telephone switching office).

Receiver

The cellular receiver consists of RF amplifier, FM demodulator and filters. An RF amplifier boosts the level of received cell site signal. Received signal is monitored by MTSO. If the signal is weak in the present cells then mobile unit is shifted to other site where the signal is strong.

Frequency Synthesizer

Frequency Synthesizer is used to generate various signals required for transmitter and receiver. When a mobile unit initiates a call, MTSO identifies the user and assigns a frequency channel which is not used by any other mobile in the cell. MTSO sends a unique code for setting channel frequencies.

Logic Unit

Logic unit is microprocessor controlled master control circuit for cellular radio. It basically controls the complete operation of MTSO and mobile unit.

Control Unit

The control unit is a set of speaker ,microphone with touch tone dialing facility and it stores the memory like numbers and dialing features.