Lecture Notes

for

Combinational Logic Circuits

(PHYS4008: Electronics)

Dr. Pawan Kumar

(Assistant Professor)

Department of Physics

Mahatma Gandhi Central University

Motihari-845401, Bihar

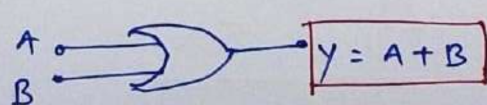
Digital Electronics

* The branch of electronics which deals with digital circuits is called digital electronics.
* Analog signal : A continuously varying signal.
* Digital Signal: A signal which can have only two discrete values.
* An electronic circuit which handles only digital signal is called digital circuits.

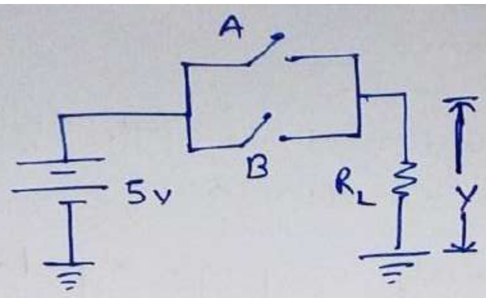
logic system
- +ve logic
- -ve logic

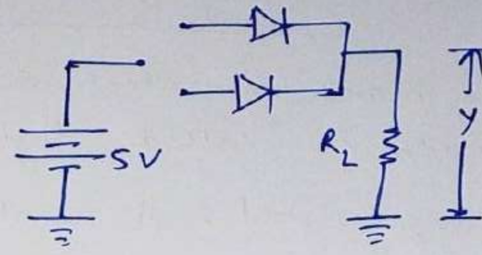
* Number systems
  1. Decimal
  2. Binary
  3. Octal
  4. Hexadecimal
* A digital circuit with one or more input signals but only one output signal is called a logic gate.
* The term "logic" refers to decision making process.
* Three basic logic gates are i) OR gate
  ii) AND "
  iii) NOT "
* OR gate : A logic gate that has two or more inputs but only one output.

  It is called OR gate because output is high if any or all inputs are high.

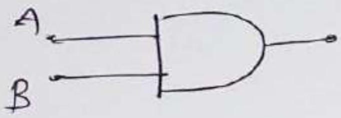


A	B	Y = A+B
0	0	0
0	1	1
1	0	1
1	1	1



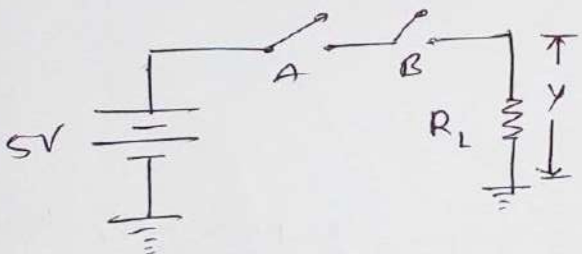


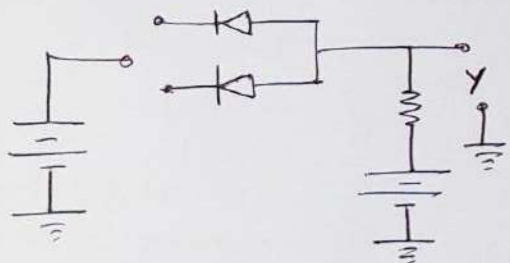
* AND gate. It has two or more inputs but only one output.
* Its output is high only when all inputs are high.



$\boxed{Y = A \cdot B}$

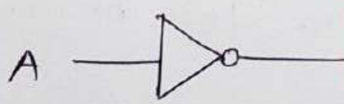
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1





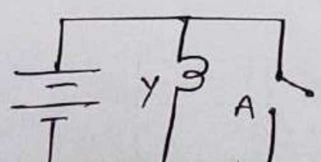
* NOT gate (Inverter)

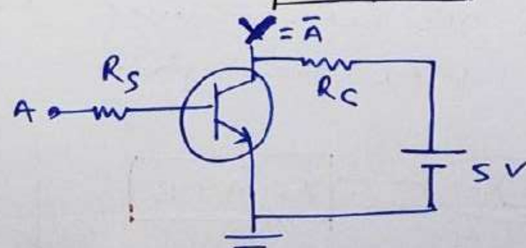
It has only one input and one output, where output is opposite of the input.



$\boxed{Y = \bar{A}}$

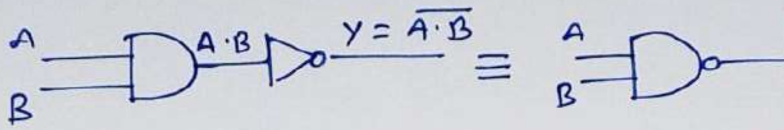
A	Y
0	1
1	0





Combination of Basic logic device

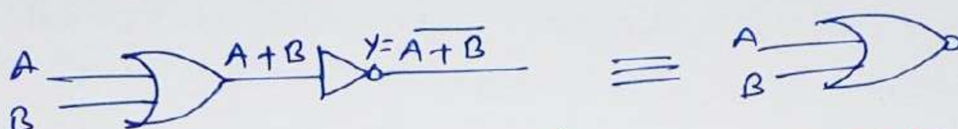
1. NAND gate :



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

The output from NAND gate is always 1 except when all of the inputs are 1.

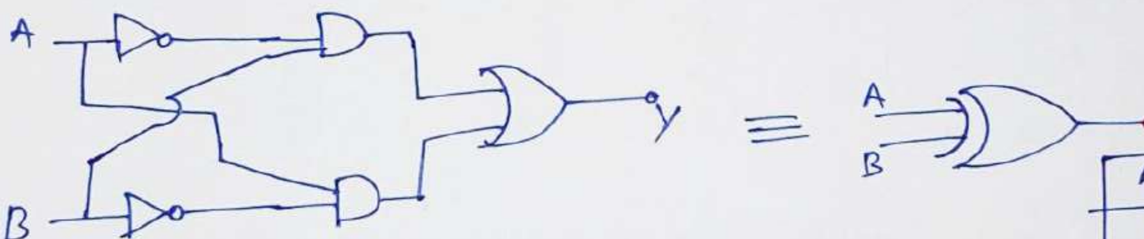
2. NOR gate :



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

The output from a NOR gate is 1 only when all inputs are 0.

3. Exclusive OR gate ($Y = A \oplus B$)

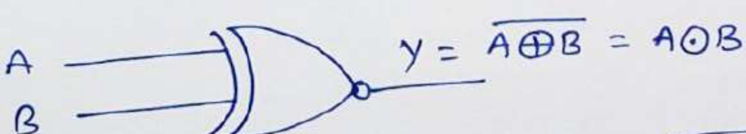


$\boxed{Y = A\bar{B} + \bar{A}B}$

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

4. Exclusive NOR gate. (XNOR)

$\boxed{Y = \bar{A}\bar{B} + AB}$



$\boxed{Y = \overline{A \oplus B} = A \odot B}$

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

De Morgan's theorem.

* $\overline{A+B} = \overline{A} \cdot \overline{B}$
* $\overline{A \cdot B} = \overline{A} + \overline{B}$
* $A + AB = A$
* $1 + B = 1 = 1 + A$
* $A\overline{A} = 0$

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

* $A + \overline{A} = 1$
* NAND and NOR gate acts as universal gate because either NAND or NOR gate can be combined to make all basic gates.

* Decimal to Binary

$23 \div 2 = 11$; remainder $= 1$
$11 \div 2 = 5$; " " $= 1$
$5 \div 2 = 2$; " " $= 1$
$2 \div 2 = 1$; " " $= 0$
$1 \div 2 = 0$; " " $= 1$

$\therefore$ $\boxed{(23)_{10} = (10111)_2}$

$0.55 \times 2 = 0.10$; Carry $= 1$
$0.10 \times 2 = 0.20$; Carry $= 0$
$0.20 \times 2 = 0.40$; " $= 0$
$0.40 \times 2 = 0.80$; " $= 0$
$0.80 \times 2 = 0.60$; " $= 1$
$0.60 \times 2 = 0.20$; " $= 1$

Approx result of 0.55 is

$\boxed{(0.55)_{10} = (0.100011)_2}$

* Binary to Decimal

* $10110 = 1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$

$= 16 + 0 + 4 + 2 + 0 = 22$

$$\boxed{(10110)_2 = (22)_{10}}$$

* $(0.1101)_2 = 1\times2^{-1} + 1\times2^{-2} + 0\times2^{-3} + 1\times2^{-4}$

$= \frac{1}{2} + \frac{1}{4} + 0 + \frac{1}{16}$

$= 0.5 + 0.25 + 0 + 0.0625$

$$\boxed{(0.1101)_2 = (0.8125)_{10}}$$

* Complement methods

Binary substraction can be alternatively carried out by 1's complement method and 2's complement method.

* The 1's complement of binary number is obtained by changing each 0 of the number to 1 and each 1 to 0.

* The 2's complement can be obtained by adding 1 to 1's complement.

$$\begin{cases} \text{2's complement of } 1110 = 0001 + 1 = 0010 \\ \text{1's complement of } 1110 = 0001 \end{cases}$$

* Binary Substraction by 1's complement method.

a) The 1's complement of the number to be substracted is determined.

b) The 1's complement is added to the number from which substraction is desired.

c) ⇒ When there is 1 carry in the last position of the result of addition in step (b), the carry is removed and added to the result without the carry to obtain the final answer.

⇒ If there is no 1 carry in the last place, the result is -ve and is in its 1's complement.

* Binary substraction by 2's complement.

The 2's complement of the number to be substracted is added to the number from which the substraction is desired and the carry in the last place is rejected.

* The 1 carry in the last position of the result in 2's complement method implies that answer is +ve.

* The absence of a 1 carry in the last position indicates that the answer of substraction is -ve and is in 2's complement.

* Octal Number System : It has a base of 8.

$879 \div 8$	$= 109$	; remainder	$= 7$
$109 \div 8$	$= 13$	; "	$= 5$
$13 \div 8$	$= 1$	; "	$= 5$
$1 \div 8$	$= 0$	; "	$= 1$

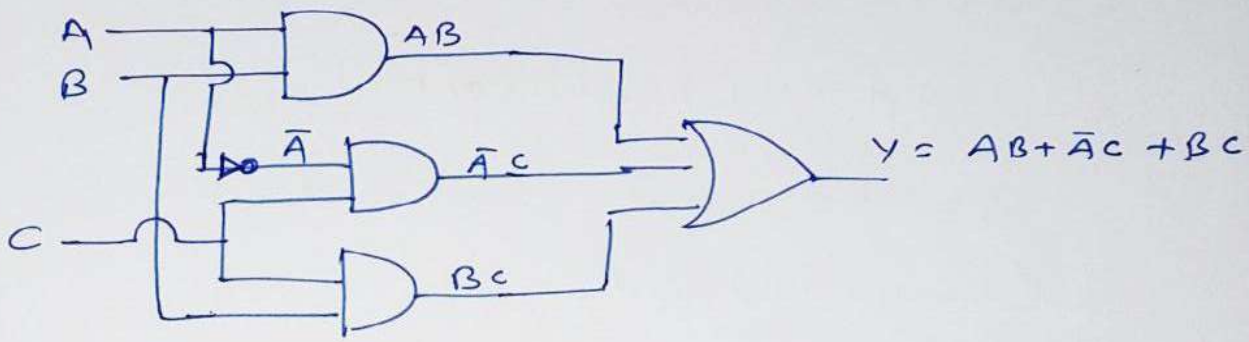
$$\boxed{(879)_{10} = (1557)_8}$$

* Hexadecimal Number System : It uses a base of 16.

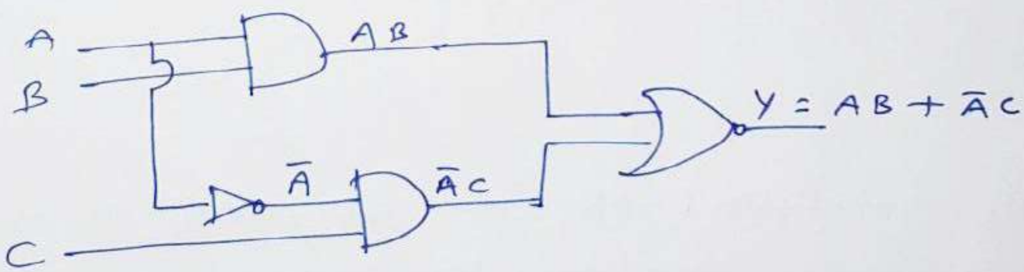
$980 \div 16$	$= 61$	; remainder	$= 4$
$61 \div 16$	$= 3$	; "	$= 13$
$3 \div 16$	$= 0$	; "	$= 3$

$$\boxed{(980)_{10} = (3D4)_{16}}$$

Q Do the circuit implementation of the Boolean equation $Y = AB + \bar{A}C + BC$ and simplify the equation. ~~Plot~~



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



*
$$
\begin{aligned}
A + \bar{A}B &= (A + AB) + \bar{A}B \\
&= (AA + AB) + \bar{A}B \\
&= AA + AB + \bar{A}A + \bar{A}B \\
&= (A + \bar{A})(A + B) \\
&= A + B
\end{aligned}
$$

Ans

Q Simplify $\left(a\cdot b\cdot(c+\overline{b\cdot d}) + \overline{a\cdot b}\right)\cdot c\cdot d$

$= \left(a\cdot b\cdot(c+\bar{b}+\bar{d}) + \bar{a}+\bar{b}\right)\cdot c\cdot d$

$= a\cdot b\cdot c + a\cdot b\cdot\bar{b} + a\cdot b\cdot\bar{d} + \bar{a}+\bar{b})c\cdot d$

$= \left(a\cdot b\cdot c + a\cdot b\cdot\bar{d} + \bar{a}+\bar{b}\right)c\cdot d$

$= a\cdot b\cdot c\cdot d + a\cdot b\cdot\bar{d}\cdot c\cdot d + \bar{a}\cdot c\cdot d + \bar{b}\cdot c\cdot d$

$= \left(a\cdot b + \bar{a}+\bar{b}\right)\cdot c\cdot d$

$= \left(a\cdot b + \overline{a\cdot b}\right)\cdot c\cdot d$

$= c\cdot d$ Ans

Q Convert the octal number $(321)_8$ to their binary equivalent

Sol^n: $(321)_8 =$

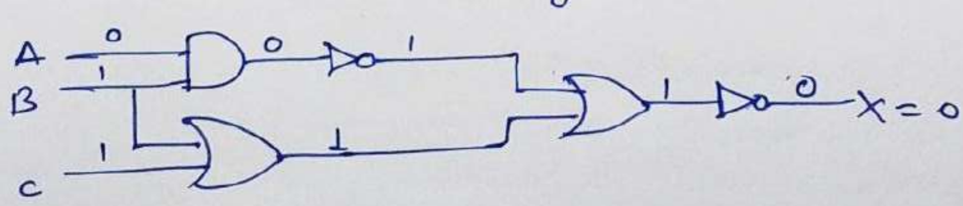
$$\begin{array}{ccc} 3 & 2 & 1 \\ \downarrow & \downarrow & \downarrow \\ 011 & 010 & 001 \end{array}$$

$\therefore (321)_8 = (011010001)_2$

$= (11010001)_2$ Ans

* An exclusive-OR gate is used to detect the failure in one of the identical circuits operating in parallel.

Q Determine the output logic level for a digital circuit shown below, having $A=0$, $B=1$ and $C=1$



* $(A+B)(A+C) = AA + AC + AB + BC$

$= A + AC + AB + BC$

$= A(1+C) + AB + BC$

$= A \cdot 1 + AB + BC$

$= A(1+B) + BC$

$= A \cdot 1 + BC$

$= A + BC$ Ans

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

$= \bar{A}B + AB + BB$

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

$= B$ Ans

* $\left[A\bar{B}\left((C+BD) + \bar{A}\bar{B}\right]C$

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

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

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

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

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

$= \bar{B}C$ Ans

* $(x+y)\left[\overline{\bar{x}(\bar{y}+\bar{z})}\right] + \bar{x}\bar{y} + \bar{x}\bar{z}$

$(x+y)\left[\bar{\bar{x}} + (\overline{\bar{y}+\bar{z}})\right] + \bar{x}\bar{y} + \bar{x}\bar{z}$

$(x+y)(x + \bar{\bar{y}} \cdot \bar{\bar{z}}) + \bar{x}\bar{y} + \bar{x}\bar{z}$

$xx + xyz + xy + yyz + \bar{x}\bar{y} + \bar{x}\bar{z}$ $(\because xx = x)$

$x(1+y+yz) + yz + \bar{x}\bar{y} + \bar{x}\bar{z}$

$x + yz + \bar{x}\bar{y} + \bar{x}\bar{z}$ $(\because 1+y+yz = 1)$

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

$x + \bar{y} + yz + \bar{x}\bar{z}$ | $= x + \bar{y} + 1 = x + \bar{y}$ Ans.

Standard forms of Boolean Expressions

All Boolean expressions, regardless of their form, can be converted into either of two standard forms:

i) Sum of products (SOP) form

ii) Product of Sum (POS) form

i) In Boolean Algebra, a product term (also called min-term) is the product of literals (or Boolean variables). In logic circuits, a product term is produced by AND operation. whereas, sum of the terms are produced by OR operation.

Q* Convert following Boolean expressions to the sum-of-products form (SOP form)

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

$= AC + A\overline{B} + BC + 0$

$= AC + A\overline{B} + BC$

(b) $(A+\overline{B}C)C = AC + \overline{B}CC$

$= AC + \overline{B}C$

* Standard SOP form: The form in which all the variables in the domain appear in each product term of the expression.

eg: $AB\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}D$ is a standard sum-of-products.

whereas, $\overline{A}B\overline{C} + A\overline{B}D + AB\overline{C}D$ is not in standard SOP form.

* Rules for making Standard SOP form: Identify the non-standard product term and multiply each such product term ~~made~~ by a term made up of the sum of a missing variable and its complement.

e.g. $A\bar{B}C + \bar{A}B + AB\bar{C}D$ (non-standard SOP form)

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

$= A\bar{B}CD + A\bar{B}C\bar{D} + (\bar{A}BC + \bar{A}B\bar{C})(D+\bar{D}) + AB\bar{C}D$

$= A\bar{B}CD + A\bar{B}C\bar{D} + \bar{A}BCD + \bar{A}BC\bar{D} + \bar{A}B\bar{C}D + \bar{A}B\bar{C}\bar{D} + AB\bar{C}D$

= Standard SOP form.

ii) Product of Sum (POS) form: A sum term (also called a max-term) is the sum of literals (or Boolean variables). In logic circuit, a sum term is produced by an OR operation only.

* When two or more sum terms are multiplied, the resulting expression is POS.

* Identify the non-standard sum terms in the given expression. To each non-standard sum term, add a product term consisting of a missing variable and its complement.

Q Convert the Boolean expression $(A+\bar{B})(B+C)$ to a standard product of sum form.

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

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

Q If the output is 0 for binary value 000, 001, 100, and 101.

$$\begin{cases} 000 \rightarrow A+B+C \\ 001 \rightarrow A+B+\bar{C} \\ 100 \rightarrow \bar{A}+B+C \\ 101 \rightarrow \bar{A}+B+\bar{C} \end{cases}$$

The resulting standard POS expression for the output

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

Q The remaining combinations of variables (previous Ques) for which output is 1 are 010, 011, 110, and 111.

$$\therefore \quad 010 \rightarrow \bar{A}B\bar{C}$$
$$011 \rightarrow \bar{A}BC$$
$$110 \rightarrow AB\bar{C}$$
$$111 \rightarrow ABC$$

The resulting $\boxed{\text{SOP}}$ expression for the output X is

$$X = \bar{A}B\bar{C} + \bar{A}BC + AB\bar{C} + ABC \qquad \underline{\text{Ans}}$$

The truth table for above two questions.

	A	B	C	output X
*	0	0	0	0
*	0	0	1	0
→	0	1	0	1
→	0	1	1	1
*	1	0	0	0
*	1	0	1	0
→	1	1	0	1
→	1	1	1	1

* POS terms
→ SOP terms

Q Convert the following SOP expression to an equivalent POS expression.

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

Sol^n → The evaluation of each product term is

$$000 + 001 + 011 + 101 + 111$$

As SOP contains 5 terms, so the POS will have other three which are 010, 100, and 110. Hence, the equivalent POS expression is

$$(\bar{A}+\bar{B}+\bar{C})(\bar{A}+\overline{B+C})(\bar{A}+\bar{B}+\bar{C}) \qquad \underline{\text{Ans}}$$

Q Construct a truth table for the standard SOP expression

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

Soln: we note that there are 3 variables in the domain; so there are eight possible combinations of binary values. The binary values that make the product terms in the expression equal to 1 are 001 for $\bar{A}\bar{B}C$, 100 for $A\bar{B}\bar{C}$ and 111 for ABC. For each of these binary values, a 1 is placed in the output column. For each of the remaining binary combinations, a 0 is placed in the output column.

A	B	C	output X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

KARNAUGH MAP (K-map)

This is graphical technique which provides a systematic method for simplifying and manipulating Boolean expressions. In this technique, the information contained in a truth table or available in POS or SOP form is represented on Karnaugh map (K-map).

* Although the technique may be used for any number of variables, it is generally used up to six variables beyond which it becomes very cumbersome.

* In an n-variable K-map, there are 2^n cells. Each cell corresponds to one of the combinations of n variables, since there are 2^n combinations of n variables.

* Two variable Karnaugh map

Truth table

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

Boolean Expression

$$\boxed{Y = \bar{A}\cdot\bar{B} + AB}$$

K-map

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

Three variable karnaugh map

$$Y = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC \leftarrow (\text{SOP})$$

$$Y = (A+B+C)(A+\bar{B}+\bar{C})(\bar{A}+B+\bar{C})(\bar{A}+\bar{B}+C) \leftarrow (\text{POS})$$

C \ AB	00	01	11	10
0	(0) 0	(2) 1	(6) 0	(4) 1
1	(1) 1	(3) 0	(7) 1	(5) 0

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

$$Y = \sum m(1, 2, 4, 7)$$

$$Y = \prod M(0, 3, 5, 6)$$

*** Rules for minimization of SOP form.

* We have to prepare the K-map first and look for combination of ones on the K-map. We have to combine the ones in such a way that the resulting expression is minimum. The following algorithm can be used which will definitely lead to minimized expression.

→ 1. Identify the ones which cannot be combined with any other ones and encircle them. These are essential prime implicants.

→ 2. Identify the ones that can be combined in groups of two, four and eight adjacent ones in only one way. Encircle them in groups.

→ 3. After identifying the essential groups of 2, 4 and 8 ones, if there still remains some ones which have not been encircled then these are to be combined with each other or with other already encircled ones. Any 1 (one) can be included any number of times without affecting expression.

Q Minimize the four variable logic function using K-map

$$f(A,B,C,D) = \sum m\,(0,1,2,3,5,7,8,9,11,14)$$

CD \ AB	00	01	11	10
00	0: 1	4:	12:	8: 1
01	1: 1	5: 1	13:	9: 1
11	3: 1	7: 1	15:	11: 1
10	2: 1	6:	14: 1	10:

$f(A,B,C,D)$

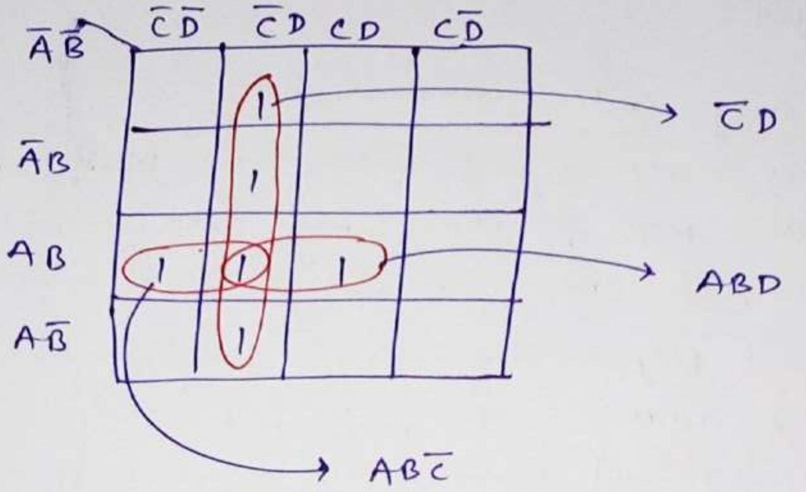
$= ABC\bar{D} + \bar{B}\bar{C} + \bar{B}D + \bar{A}D + \bar{A}\bar{B}$

Ans

The 1 in cell 14 cannot be combined with any other 1. It corresponds to $ABC\bar{D}$.

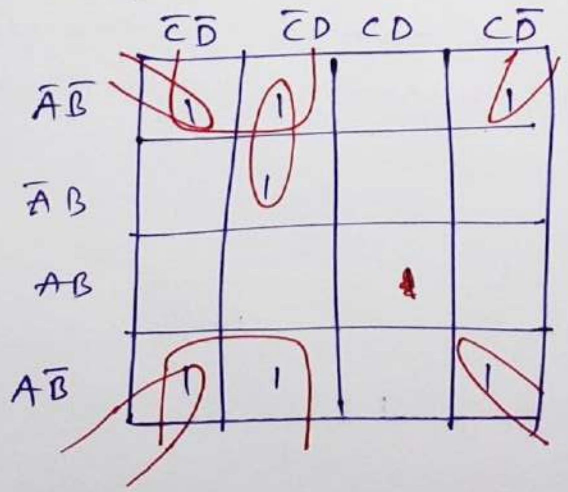
Q Simplify the following SOP expression using the Karnaugh mapping procedure.

$$X = \bar{A}B\bar{C}D + A\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}D + AB\bar{C}D + AB\bar{C}\bar{D} + ABCD$$



Hence $X = ABD + AB\bar{C} + \bar{C}D$ **Ans**

Q Figure shows a K-map of sum of products (SOP) function. Determine the simplified SOP func.



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

Gray Code:

It was designed by Frank Gray at Bell labs and patented in 1953. It is unweighted binary code in which two successive value differ only by 1 bit.

Binary to Gray code conversion:

1. Begin with most significant bit (**MSB**) of binary number. The MSB of the Gray code equivalent is the same as the MSB of given binary number.
2. The second MSB in the gray code number is obtained by adding the MSB and the second MSB of binary and ignoring the carry, if any.
3. The third MSB in the Gray code number is obtained by adding the second MSB and third MSB in the binary number and ignoring carry, if any.
4. The process continues untill we obtain the LSB of the Gray code number by the addition of the LSB and next higher adjacent bit of the binary number.

e.g.

Binary $\boxed{1011}$
Gray code 1___
Binary 1011
Gray code 11__
Binary 1011
Gray code 111_
Binary 1011
Gray code $\boxed{1110}$

Gray to Binary conversion

1. Begin with MSB. The MSB of binary number is the same as the MSB of the Gray code number.
2. The bit next to MSB (2nd MSB) in binary number is obtained by adding the MSB in the binary number to the second MSB in the Gray code number and disregarding the carry if any.
3. The 3rd MSB in the binary number is obtained by adding the 2nd MSB in the binary number to third MSB in the Gray code number. Ignore the carry.
4. The process continues untill we obtain LSB of binary number.

e.g. Gray code $\boxed{1110}$

Binary 1 – – –

Gray code 1110

Binary 10 – –

Gray code 1110

Binary 101 –

Gray code 1110

Binary $\boxed{1011}$

Applications

1. used in Transmission ~~line~~ of digital signals because it minimizes the occurance of error.
2. preferred in angle measuring device. It has also cyclic property suitable for angle
3. used for lebelling of K-map.
4. used to address program memory in Computers for power consumption minimization.

ASCII Code : American Standard Code for Information Exchange. It is used to represent the alphanumeric data in computers, and communication equipments. It was first published as a standard in 1967. It was subsquently updated and published as ANSI X3.4-1968, then as ANSI X3.4-1977 and finally as ANSI X3.4-1986.

* Since it is Seven-bit code, it can be used to represent 128 characters.
* It currently defines 95 printable characters including 26 upper-case letter (A to Z), 26 lower case letters (a to z), 10 numeral (0 to 9) and 33 special characters including mathematical symbols, punctuation marks and space characters.
* In addition, it defines codes for 33 nonprinting, mostly obsolete control characters that affect how text is processed.
* An eight bit version of ASCII code, known as US ASCII-8 or ASCII-8 has also been developed. It can represents 256 characters.

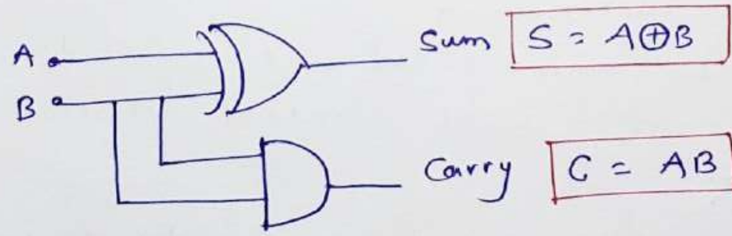
Logic gate circuits

* It can be divided into two categories based on whether they are with feedback sequential logic circuit or without feedback combinational logic circuit.
* Digital electronics is divided into combinational and sequential logic.
* Combinational logic ~~gate~~ output depends on the input levels whereas sequential logic output depends on stored levels of past data and also the present input levels.

* Combinational Logic : combinational logic circuit is used to realize different logic functions using different logic gates.

Adder : It is used to add two or more bits.

* Half adder : It is a combinational logic circuit, which is used to add two bits and generate output as sum (S) and carry (C).



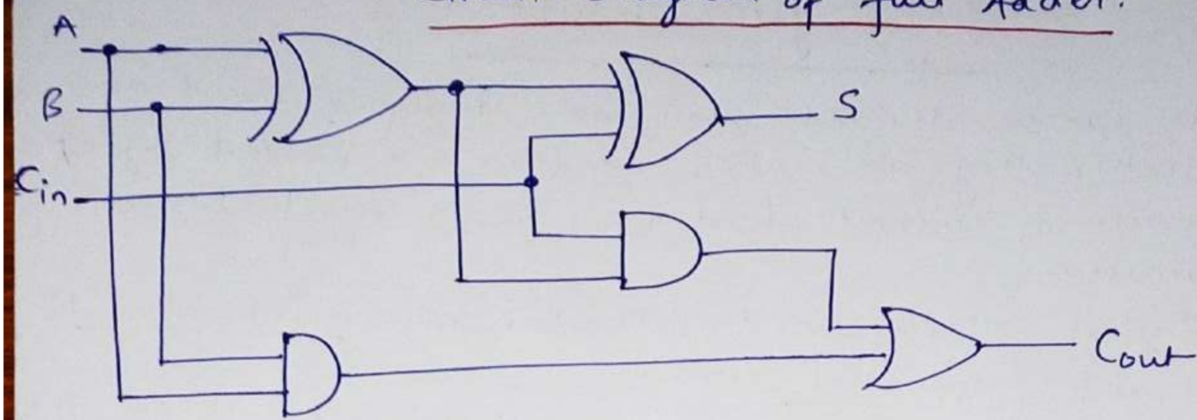
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

* Full Adder : It is used to add three or more bits. The reason for the name full adder is that it can add the carry bit (C_{in}) along with other two inputs A and B.

$$\boxed{S = (A \oplus B) \oplus C_{in}}$$

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

Circuit diagram of full Adder.

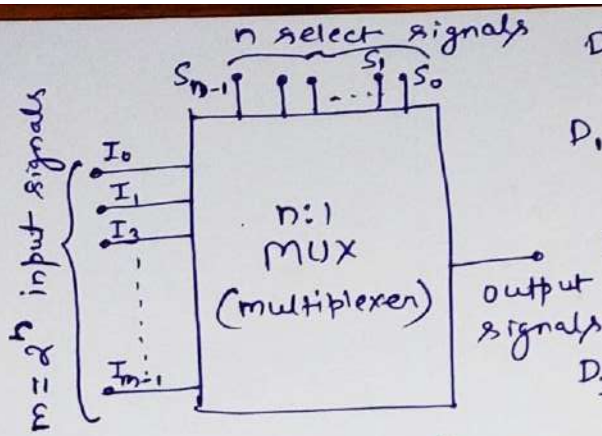


Input			Output	
A	B	C_{in}	C_y	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

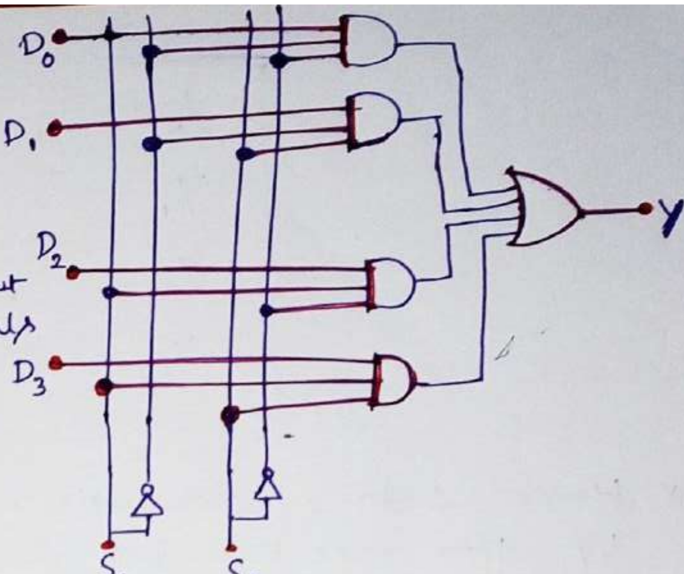
← Truth table of full adder.

Multiplexer

* It is a special combinational circuit that is one of the most widely used standard circuit in digital design.
* The multiplexer (or data selector) is a logic circuit that gates one out of several inputs to single output.
* The selection of input is controlled by a set of select inputs.
* A multiplexer has n number of select inputs, 2^n inputs, and only one output.



Block diagram of a multiplexer

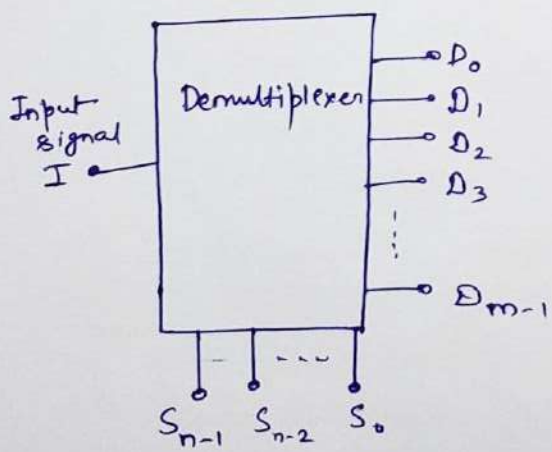


Circuit diagram of 4:1 multiplexer.

$$\boxed{Y = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3}$$

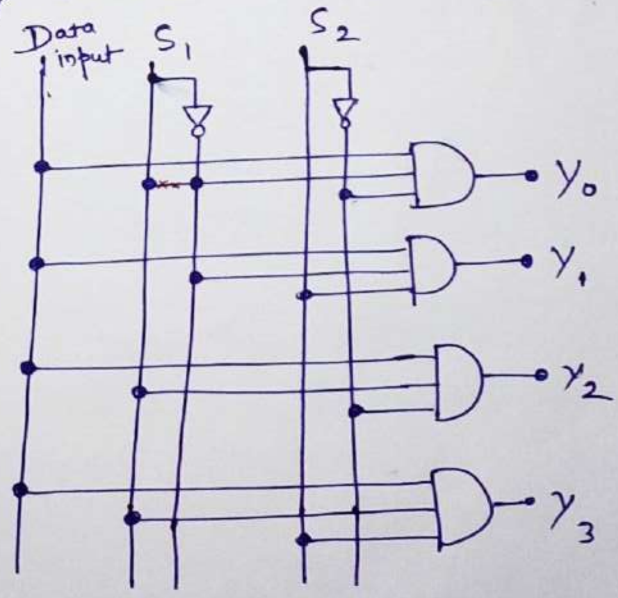
De-multiplexer

Its operation is the opposite to that of the multiplexer. It has n number of select inputs, 2^n outputs and only one ~~select~~ input signal.



$(m = 2^n)$

Block diagram of demultiplexer



Circuit diagram of 1:4 demultiplexer

$$
\begin{aligned}
Y_0 &= \overline{S}_1 \overline{S}_0 D \\
Y_1 &= \overline{S}_1 S_0 D \\
Y_2 &= S_1 \overline{S}_0 D \\
Y_3 &= S_1 S_0 D
\end{aligned}
$$

* Hence 1-to-4 demultiplexer can be implemented by two inverters and four 3-input AND gates. The single input D is applied to all the AND gates. The two select line S_1 & S_0 enable any one AND gate at a time and the data appears at the output of the selected AND gate as shown in figure.

References:

1. Digital Fundamentals by Thomas Floyd
2. Digital principles and Applications by A. P. Malvino and D. P. Leach
3. Modern Digital Electronics by R.P. Jain
4. Digital Design by M. Mano
5. Electronics Fundamentals and Applications by D. Chattopadhyay and P.C. Rakshit
6. Electronic Devices and Circuits by J. Millman and C.C. Halkias
7. Integrated Electronics by J. Millman and C.C. Halkias