

MODULE - 3

Syllabus:

Boolean Algebra and logic circuits: Binary numbers, Number Base conversions, Octal and Hexa decimal numbers, complements, Basic definitions, Axiomatic definition of Boolean Algebra, Basic theorems, Properties of Boolean algebra, Boolean functions, Canonical and standard forms, other logic operations Digital logic gates.

Combinational logic: Introduction, design procedure, Adders - Half adder, Full adder.

Binary Numbers:

- Binary number system is one of the type of number representation techniques. It is most popular and used in digital systems.
- Binary system is used for representing binary quantity that has only two states or possible conditions.
Ex: A switch has only two states: open or close.
- In Binary system, there are only two possible digital values i.e 0 and 1. So binary number system is base-2 or radix-2 number system.

The general form is

$$b_m \cdot b_{m-1} \dots b_3 b_2 b_1 b_0 \cdot b_{-1} b_{-2} \dots b_{-n}$$

and its decimal equivalent is

$$b_m \times 2^m + b_{m-1} \times 2^{m-1} + \dots + b_3 \times 2^3 + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0 \\ + b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + \dots + b_{-n} \times 2^{-n}$$

Ex: $(110101.1101)_2$ or $(110101.1101)_b$

Decimal Number System:

- In regular day to day transactions, we see decimal number system in which only ten digits are used i.e., 0 to 9.
- Decimal number system is base -10 or radix-10 number system.

The general form is

$$d_m.d_{m-1} \dots d_3 d_2 d_1 d_0 . d_{-1} d_{-2} \dots d_{-n}$$

and its equivalent is,

$$d_m \times 10^m + d_{m-1} \times 10^{m-1} + \dots + d_3 \times 10^3 + d_2 \times 10^2 + d_1 \times 10^1 + d_0 \times 10^0 + d_{-1} \times 10^{-1} + d_{-2} \times 10^{-2} + \dots + d_{-n} \times 10^{-n}$$

Ex: $(2504.18)_{10}$ or $(2504.18)_d$

Octal Number System:

- Radix-8 system is called as Octal system. The system needs 8 digits (0 to 7).

The general form is

$$O_m.O_{m-1} \dots O_3 O_2 O_1 O_0 . O_{-1} O_{-2} \dots O_{-n}$$

and its equivalent is

$$O_m \times 8^m + O_{m-1} \times 8^{m-1} + \dots + O_3 \times 8^3 + O_2 \times 8^2 + O_1 \times 8^1 + O_0 \times 8^0 + O_{-1} \times 8^{-1} + O_{-2} \times 8^{-2} + \dots + O_{-n} \times 8^{-n}$$

Ex: $(736.52)_8$ or $(736.52)_o$

Hexadecimal Number System:

- The radix-16 system is called hexadecimal system. This needs 16 digits (0 to 9 and A to F).

The general form is

$$H_m.H_{m-1} \dots H_3 H_2 H_1 H_0 . H_{-1} H_{-2} H_{-3} \dots H_{-n}$$

and its equivalent is

$$H_m \times 16^m + H_{m-1} \times 16^{m-1} + \dots + H_3 \times 16^3 + H_2 \times 16^2 + H_1 \times 16^1 + H_0 \times 16^0 + H_{-1} \times 16^{-1} + H_{-2} \times 16^{-2} + \dots + H_{-n} \times 16^{-n}$$

$$+ H_{-1} \times 16^{-1} + H_{-2} \times 16^{-2} + \dots + H_{-n} \times 16^{-n}$$

Ex: $(A3F2.5E)_{16}$ or $(A3F2.5E)_H$.

Number Base Conversion:

→ Decimal to any Base - r:

- * Decimal to Binary - divide by 2.
- * Decimal to Octal - divide by 8.
- * Decimal to Hexadecimal - divide by 16.

→ Any Base - r to Decimal:

- * Binary to decimal - Multiply by powers of 2.
- * Octal to decimal - Multiply by powers of 8.
- * Hexadecimal to decimal - Multiply by powers of 16.

→ Binary to Octal - Split the given binary numbers into 3 bits and write the corresponding octal equivalent.

→ Octal to Binary - Write the corresponding binary equivalent for the given Octal number.

→ Binary to Hexadecimal - Split the given binary numbers into 4 bits and write the corresponding Hexadecimal equivalent.

→ Hexadecimal to Binary - Write the corresponding binary equivalent for the given Hexadecimal number.

→ Octal to Hexadecimal - Firstly convert the octal number into Binary and the obtained Binary value into the Hexadecimal number.

→ Hexadecimal to Octal - Firstly convert the given Hexadecimal number into Binary and the obtained binary value into the corresponding Octal numbers.

Table: Number with different bases.

Decimal (base-10)	Binary (base-2)	Octal (base-8)	Hexadecimal (base-16)
0	0000	000-0	0
1	0001	001-1	1
2	0010	010-2	2
3	0011	011-3	3
4	0100	100-4	4
5	0101	101-5	5
6	0110	110-6	6
7	0111	111-7	7
8	1000		8
9	1001		9
10	1010		A
11	1011		B
12	1100		C
13	1101		D
14	1110		E
15	1111		F

* Decimal to Binary Conversion.

① $(398.75)_{10} = (?)_2$

$\swarrow$ Fractional part.
 $\searrow$ Integer part.

Integer part.

$$\begin{array}{r}
 2 \overline{) 398} \\
 \underline{2 199} \\
 2 \overline{) 99} \\
 \underline{2 99} \\
 2 \overline{) 49} \\
 \underline{2 49} \\
 2 \overline{) 4} \\
 \underline{2 4} \\
 2 \overline{) 12} \\
 \underline{2 12} \\
 2 \overline{) 6} \\
 \underline{2 6} \\
 2 \overline{) 3} \\
 \underline{2 3} \\
 1
 \end{array}$$

$$(398)_{10} = (110001110)_2$$

Fractional Part.

$$0.75 \times 2 = 1.50$$

$$0.5 \times 2 = 1.00$$

$$(0.75)_{10} = (0.11)_2$$

Combining both parts

$$(398.75)_{10} = (110001110.11)_2$$

2) $(172.625)_{10} = (?)_2$.

$$\begin{array}{r}
 2 \overline{) 172} \\
 \underline{2 86 - 0} \\
 2 \overline{) 43} \\
 \underline{2 21 - 1} \\
 2 \overline{) 10} \\
 \underline{2 5 - 0} \\
 2 \overline{) 2} \\
 \underline{2 1 - 0} \\
 1 - 0
 \end{array}$$

$$\begin{array}{l}
 0.625 \times 2 = 1.25 \\
 0.25 \times 2 = 0.50 \\
 0.50 \times 2 = 1.00
 \end{array}$$

$$(172.625)_{10} = (\underline{10101100.101})_2$$

3) $(725.25)_{10} = (?)_2$

$$\begin{array}{r}
 2 \overline{) 725} \\
 \underline{2 362 - 1} \\
 2 \overline{) 181} \\
 \underline{2 90 - 1} \\
 2 \overline{) 45} \\
 \underline{2 22 - 1} \\
 2 \overline{) 11} \\
 \underline{2 5 - 0} \\
 2 \overline{) 2} \\
 \underline{2 1 - 0} \\
 1 - 0
 \end{array}$$

$$\begin{array}{l}
 0.25 \times 2 = 0.50 \\
 0.50 \times 2 = 1.00
 \end{array}$$

$$(725.25)_{10} = (\underline{1011010111.01})_2$$

* Decimal to Octal Conversion.

① $(658.825)_{10} = (?)_8$

$$\begin{array}{r}
 8 \overline{) 658} \\
 \underline{8 82 - 2} \\
 8 \overline{) 10} \\
 \underline{8 1 - 2} \\
 1 - 2
 \end{array}$$

$$\begin{array}{l}
 0.825 \times 8 = 6.6 \\
 0.6 \times 8 = 4.8 \\
 0.8 \times 8 = 6.4
 \end{array}$$

$$(658.825)_{10} = (\underline{1222.646})_8$$

② $(2003)_{10} = (?)_8$

$$\begin{array}{r} 8 \overline{) 2003} \\ 8 \overline{) 250-3} \\ 8 \overline{) 31-2} \\ 3-7 \end{array}$$

$$(2003)_{10} = \underline{\underline{(3123)_8}}$$

③ $(11582.875)_{10} = (?)_8$

$$\begin{array}{r} 8 \overline{) 11582} \\ 8 \overline{) 1447-6} \\ 8 \overline{) 180-7} \\ 8 \overline{) 22-4} \\ 2-6 \end{array}$$

$$0.875 \times 8 = 7.00$$

$$(11582.875)_{10} = \underline{\underline{(26476.7)_8}}$$

④ $(934.705)_{10} = (?)_8$

$$\begin{array}{r} 8 \overline{) 934} \\ 8 \overline{) 116-6} \\ 8 \overline{) 14-4} \\ 1-6 \end{array}$$

$$0.705 \times 8 = 5.64$$

$$0.64 \times 8 = 5.12$$

$$0.12 \times 8 = 0.96$$

$$(934.705)_{10} = \underline{\underline{(1646.550\dots)_8}}$$

* Decimal to Hexadecimal Conversion:

① $(7084.95)_{10} = (?)_{16}$

$$\begin{array}{r} 16 \overline{) 7084} \\ 16 \overline{) 442-C} \\ 16 \overline{) 27-A} \\ 1-B \end{array}$$

$$0.95 \times 16 = 15.2$$

$$0.2 \times 16 = 3.2$$

$$0.2 \times 16 = 3.2$$

$$(7084.95)_{10} = \underline{\underline{(1BAC.F33\dots)_{16}}}$$

② $(4475.85)_{10} = (?)_{16}$

$$\begin{array}{r} 16 \overline{) 4475} \\ 16 \overline{) 279-D} \\ 16 \overline{) 17-7} \\ 1-1 \end{array}$$

$$0.8 \times 16 = 13.6$$

$$0.6 \times 16 = 9.6$$

$$0.6 \times 16 = 9.6$$

$$\underline{\underline{(17D.D99\dots)_{16}}}$$

$$(3) \quad (894867.0368)_{10} = (?)_{16}$$

$$\begin{array}{r} 16 \overline{) 894867} \\ 16 \overline{) 55929-3} \\ 16 \overline{) 3495-9} \\ 16 \overline{) 218-7} \\ \quad \quad \quad 13-A \end{array}$$

$$0.368 \times 16 = 5.888$$

$$0.88 \times 16 = 14.208$$

$$0.208 \times 16 = 3.328$$

$$(894867.0368)_{10} = \underline{\underline{(DA793.5E3\dots)_{16}}}$$

* Binary to Decimal Conversion:

$$(1) \quad (11101.01)_2 = (?)_{10}$$

$$= (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) + (0 \times 2^{-1}) + (1 \times 2^{-2})$$

$$= 16 + 8 + 4 + 0 + 1 + 0 + 0.25$$

$$(11101.01)_2 = (29.25)_{10}$$

$$(2) \quad (101010.101)_2 = (?)_{10}$$

$$= (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) + (1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3})$$

$$= 32 + 0 + 8 + 0 + 2 + 0 + 0.5 + 0 + 0.125$$

$$(101010.101)_2 = (42.625)_{10}$$

* Octal to decimal Conversion:

$$(1) \quad (327.56)_8 = (?)_{10}$$

$$= (3 \times 8^2) + (2 \times 8^1) + (7 \times 8^0) + (5 \times 8^{-1}) + (6 \times 8^{-2})$$

$$= 192 + 16 + 7 + 0.625 + 0.09375$$

$$(327.56)_8 = (215.71875)_{10}$$

$$(2) \quad (107.70)_8 = (?)_{10}$$

$$= (1 \times 8^2) + (0 \times 8^1) + (7 \times 8^0) + (7 \times 8^{-1}) + (0 \times 8^{-2})$$

$$= 64 + 0 + 7 + 0.875 + 0$$

$$(107.70)_8 = (71.875)_{10}$$

* Hexadecimal to Decimal Conversion:

① $(ABC.90)_{16} = (?)_{10}$

$$= (A \times 16^2) + (B \times 16^1) + (C \times 16^0) + (9 \times 16^{-1}) + (0 \times 16^{-2})$$

$$= (2560 + 176 + 12 + 0.5625 + 0)$$

$$(ABC.90)_{16} = (2748.5625)_{10}$$

② $(3A.2F)_{16} = (?)_{10}$

$$= (3 \times 16^1) + (A \times 16^0) + (2 \times 16^{-1}) + (F \times 16^{-2})$$

$$= 48 + 10 + 0.125 + 0.0585937$$

$$(3A.2F)_{16} = (58.18359)_{10}$$

Complements:

- Complements are used in digital computers for simplifying the subtraction operation and for logical manipulation.
- There are 2 types of complements for each base- r system:
 - 1) the radix complements (or) r 's complement
 - 2) the diminished radix complements (or) $(r-1)$'s complement.
- Depending upon the base or radix of different number systems, the complement representations are,

Binary Number System $\left\{ \begin{array}{l} 1^s \text{ complement} \\ 2^s \text{ complement} \end{array} \right.$

Decimal Number system $\left\{ \begin{array}{l} 9^s \text{ complement} \\ 10^s \text{ complement} \end{array} \right.$

Complement of Binary Numbers:

1^s Complement: 1^s complement of a Binary number is the number that results when we change all 1^s to zeros and the zeros to ones.

2's complement: 2's complement is the binary number that results when we add 1 to the 1's complement. It is given as,

$$2's \text{ complement} = 1's \text{ complement} + 1.$$

Subtraction of Binary Numbers:

To perform Subtraction of Binary Numbers in 1's Complement:

For Example: Operation $A - B$ is performed using following steps:

Step 1: Take 1's complement of B

Step 2: 1's complement of B + A $\rightarrow$ Result.

Step 3: If carry is generated then the result is positive. Add carry to the result to get the final result.

Step 4: If carry is not generated then the result is negative and take the 1's complement of result.

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

Soln: $(28)_{10} = (011100)_2 - A$

$(19)_{10} = (010011)_2 - B$

Take 1's complement of B

$$\begin{array}{r} 011100 \\ \downarrow \\ 1001100 \end{array}$$

Add A $+ 011100$

$$\begin{array}{r} 1001100 \\ \text{Carry} \rightarrow +1 \end{array}$$

$$(001001)_2 = (9)_{10}$$

$$\boxed{(28)_{10} - (19)_{10} = (9)_{10}}$$

(2) $(56)_{10} - (79)_{10}$.

$$A = (56)_{10} = (0111000)_2$$

$$B = (79)_{10} = (01001111)_2$$

$$1^s \text{ comp of } B = 10011000$$

$$\text{Add } A + \begin{array}{r} 0111000 \\ 10011000 \\ \hline 11101000 \end{array}$$

No carry, so take 1^s comp of the result.

$$(-00010111)_2 = (-23)_{10}$$

$$\therefore \boxed{(56)_{10} - (79)_{10} = (-23)_{10}}$$

(3) $(101011)_2 - (10000)_2$

$$A = (101011)_2$$

$$B = (10000)_2$$

$$1^s \text{ comp of } B = 1001111$$

$$\text{Add } A + \begin{array}{r} 1001111 \\ 101011 \\ \hline 1011010 \end{array}$$

$$\text{Sign bit } 1011010$$

$$\text{Carry } +1 \quad (011011)_2 = (27)_{10}$$

(4) $(1000.01)_2$ from $(1011.10)_2$.

$$A = 1011.10$$

$$B = 1000.01$$

$$(1011.10)_2 - (1000.01)_2 = (0011.01)_2$$

$$1^s \text{ comp of } B = 00111.10$$

$$\text{Add } A = \begin{array}{r} 1011.10 \\ 00111.10 \\ \hline 10011.00 \end{array}$$

$$\text{Carry } 10011.00 + 1 = (0011.01)_2$$

To perform Subtraction of Binary Numbers in 2^s Complement:

For example: operation $A - B$ is performed using following steps.

Step 1: Take 2^s complement of B

Step 2: 2^s complement + $A \rightarrow$ Result.

Step 3: If carry is generated, then the result is positive.
Carry is ignored.

Step 4: If carry is not generated, then the result is Negative and take 2^s complement of the result.

① Perform $(39)_{10} - (48)_{10}$ using 2^s complement representation

$$A = (39)_{10} = (100111)_2$$

$$B = (48)_{10} = (110000)_2$$

$$\text{Take } 2^s \text{ comp of } B = \begin{array}{r} 00000 \\ 001111 \\ + 1 \end{array}$$

$$\begin{array}{r} 010000 \\ 001111 \\ + 1 \\ \hline 010000 \end{array} \rightarrow 2^s \text{ comp of } B.$$

$$\text{Add } A + \begin{array}{r} 100111 \\ 010000 \\ \hline 110111 \end{array}$$

$$\begin{array}{r} 110111 \\ \hline 110111 \end{array} \rightarrow \text{No carry.}$$

$$\text{Take } 2^s \text{ comp of result} \rightarrow \begin{array}{r} 001000 \\ + 1 \end{array}$$

$$\begin{array}{r} 001000 \\ + 1 \\ \hline 001001 \end{array} \rightarrow (-001001)_2 = (-9)_{10}.$$

$$\therefore \boxed{(39)_{10} - (48)_{10} = (-9)_{10}}$$

② $(78)_{10} - (65)_{10}$

$$A = (78)_{10} = (1001110)_2$$

$$B = (65)_{10} = (1000001)_2$$

Take 2^s comp of B = 0 1 1 1 1 0

$$\begin{array}{r} \textcircled{0} \textcircled{1} \textcircled{1} \textcircled{1} \textcircled{1} \textcircled{0} \\ + 1 \\ \hline \end{array}$$

$$\text{Add A} + \begin{array}{r} 1001110 \\ \hline \end{array}$$

$$\begin{array}{r} \textcircled{1} 0001101 \\ \hline \end{array}$$

Carry ignored.

$$(+0001101)_2 = (13)_{10}$$

$$\therefore [(78)_{10} - (65)_{10} = (13)_{10}]$$

③ Subtract $(111001)_2$ from $(101011)_2$

$$(101011)_2 - (111001)_2$$

$$A = (101011)_2$$

$$B = (111001)_2$$

Take 2^s comp of B = 000110

$$\begin{array}{r} \textcircled{0} \textcircled{0} \textcircled{0} \textcircled{1} \textcircled{1} \textcircled{0} \\ + 1 \\ \hline \end{array}$$

$$\text{Add A} + \begin{array}{r} 101011 \\ \hline \end{array}$$

$$\begin{array}{r} 110010 \\ \hline \end{array}$$

No carry, Take 2^s comp.

$$\begin{array}{r} \textcircled{1} \\ 001101 \\ \hline \end{array}$$

$$+ 1$$

$$(101011)_2 - (111001)_2 = (-001110)_2$$

④ $(1000100)_2 - (1010100)_2$

$$A = (1000100)_2$$

$$B = (1010100)_2$$

Take 2^s comp of $B = 0101011$

$$\begin{array}{r} 0101011 \\ + 1 \\ \hline 0101100 \end{array}$$

Add A + $\begin{array}{r} 1000100 \\ \hline 1110000 \end{array}$

No carry, Take 2^s comp

$$\begin{array}{r} 0001111 \\ + 1 \\ \hline 0010000 \end{array}$$

$$(-0010000)_2 = (-16)_{10}.$$

Subtraction of Decimal Numbers:

To perform subtraction of decimal numbers in q^s complement.

→ The q^s complement of a decimal number may be obtained by subtracting each digit of that number from q .

Ex: $(354)_{10} \rightarrow q^s$ complement is

$$\begin{array}{r} 999 \\ - 354 \\ \hline 645 \end{array}$$

$(645)_{10} \rightarrow q^s$ complement.

Operation $A - B$ is performed using following steps.

Step 1: Find the q^s complement of B .

Step 2: Add q^s complement of $B + A \rightarrow$ Result.

Step 3: If carry is generated, then the result is positive. Add carry to the result to get the final result.

Step 4: If carry is not generated, then the result is negative and take q^s complement of the result.

① Perform $(487)_{10} - (354)_{10}$ using 9^s complement method

$$A = (487)_{10}$$

$$B = (354)_{10}$$

Take 9^s comp of B $\rightarrow$

$$\begin{array}{r} 9 \ 9 \ 9 \\ - \ 3 \ 5 \ 4 \\ \hline \end{array}$$

$$\begin{array}{r} \textcircled{1} \ \textcircled{1} \\ \hline 6 \ 4 \ 5 \end{array}$$

$(645)_{10} \rightarrow 9^s \text{ comp of } B$

$$\text{Add } A + \begin{array}{r} 4 \ 8 \ 7 \\ \hline \end{array}$$

$$\begin{array}{r} \textcircled{1} \ 1 \ 3 \ 2 \\ \hline \end{array}$$

$$\begin{array}{r} \text{---} \rightarrow +1 \\ \hline \end{array}$$

$$\begin{array}{r} 1 \ 3 \ 3 \\ \hline \end{array}$$

$$\therefore \boxed{(487)_{10} - (354)_{10} = (133)_{10}}$$

② $(309)_{10} - (1447)_{10}$

$$A = (309)_{10}$$

$$B = (1447)_{10}$$

Take 9^s comp of B $\rightarrow$

$$\begin{array}{r} 9 \ 9 \ 9 \ 9 \\ - \ 1 \ 4 \ 4 \ 7 \\ \hline \end{array}$$

$$\begin{array}{r} 8 \ 5 \ 5 \ 2 \\ \hline \end{array}$$

$\rightarrow 9^s \text{ comp of } B$

$$\text{Add } A + \begin{array}{r} 3 \ 0 \ 9 \\ \hline \end{array}$$

$$\begin{array}{r} 8 \ 8 \ 6 \ 1 \\ \hline \end{array}$$

No carry, Take 9^s comp of result.

$$\begin{array}{r} 9 \ 9 \ 9 \ 9 \\ - \ 8 \ 8 \ 6 \ 1 \\ \hline \end{array}$$

$$\begin{array}{r} (- \ 1 \ 1 \ 3 \ 8)_{10} \end{array}$$

$$\therefore \boxed{(309)_{10} - (1447)_{10} = (-1138)_{10}}$$

③ $(357.434)_{10} - (1048.05)_{10}$

$$A = (357.434)_{10}$$

$$B = (1048.05)_{10}$$

Take 9^s comp of $B =$

$$\begin{array}{r} 9999.99 \\ 1048.05 \\ \hline 8951.94 \end{array} \rightarrow \text{signed}$$

$$\begin{array}{r} \text{Add } A + \\ 357.434 \\ \hline 9309.383 \end{array}$$

No carry, Take 9^s comp of result

$$\begin{array}{r} 9999.999 \\ 9309.383 \\ \hline (-0690.616)_{10} \end{array}$$

$$\therefore (357.434)_{10} - (1048.05)_{10} = (-0690.616)_{10}$$

To perform Subtraction of Decimal Numbers using 10^s Complement:

→ To obtain 10^s complement of a decimal number, add 1 to its 9^s complement.

Ex: obtain 10^s complement for $(354)_{10}$

$$\begin{array}{r} 9^s \text{ comp} \rightarrow 999 \\ 354 \\ \hline 645 \\ \text{Add } 1 \\ + 1 \\ \hline \end{array}$$

$$(646)_{10} \rightarrow \underline{10^s \text{ complement}}$$

Operation $A - B$ is performed using following steps.

Step 1: Find the 10^s complement of B .

Step 2: 10^s complement of $B + A \rightarrow$ Result.

Step 3: If carry is generated, then the result is positive.
And the carry is ignored.

Step 4: If carry is not generated, then the result is negative. and take 10^5 complement of result.

① Perform $(8437)_{10} - (39)_{10}$ using 10^5 complement method.

$$A = (8437)_{10}$$

$$B = (39)_{10}$$

Take 10^5 complement of $B = 9999$

$$\begin{array}{r} 9999 \\ - 39 \\ \hline 9960 \end{array}$$

$$\text{Add } 1 \rightarrow \begin{array}{r} 9960 \\ + 1 \\ \hline 9961 \end{array}$$

$9961 \rightarrow 10^5$ complement

$$\text{Add } A + 8437$$

$$\begin{array}{r} 8437 \\ + 9961 \\ \hline 18398 \end{array}$$

Discard carry.

$$\therefore (8437)_{10} - (39)_{10} = (8398)_{10}$$

② $(351.434)_{10} - (1048.05)_{10}$.

$$A = (351.434)_{10}$$

$$B = (1048.05)_{10}$$

Take 10^5 complement of $B = 9999.999$

$$\begin{array}{r} 9999.999 \\ - 1048.050 \\ \hline 8951.949 \end{array}$$

$$\text{Add } 1 \rightarrow \begin{array}{r} 8951.949 \\ + 1 \\ \hline 8951.950 \end{array}$$

$$\text{Add } A + \begin{array}{r} 351.434 \\ + 8951.950 \\ \hline 9309.384 \end{array}$$

No carry, Take 10^5 complement of Result

$$\begin{array}{r} 9999.999 \\ 9309.384 \\ \hline 0690.615 \\ + 1 \\ \hline 0690.616 \end{array}$$

$$\therefore (-0690.616)_{10}$$

③ Subtract $(72532)_{10} - (3250)_{10}$.

$$A = (72532)_{10}$$

$$B = (3250)_{10}$$

Take 10^5 complement of B =

$$\begin{array}{r} 9999 \\ - 3250 \\ \hline 6749 \\ + 1 \\ \hline 6750 \end{array}$$

Add A

$$\begin{array}{r} + 72532 \\ \hline 69282 \end{array}$$

Discard carry.

$$(72532)_{10} - (3250)_{10} = (69282)_{10}$$

④ Given two binary numbers $X = 1010100$ and $Y = 1000011$. Perform the subtraction a) $X - Y$ b) $Y - X$ using 1^s & 2^s complement.

$$X = 1010100$$

$$Y = 1000011$$

using 1^s complement.

a) $X - Y$

Take 1^s of $Y =$

$$\begin{array}{r} 0000 \\ 0111100 \end{array}$$

Add $X =$

$$\begin{array}{r} 1010100 \\ \hline \end{array}$$

$$\begin{array}{r} 0010000 \\ \hline \end{array}$$

→ +1

$$\underline{X - Y = (0010001)_2}$$

b) $Y - X$

Take 1^s comp of $X =$

$$\begin{array}{r} 00 \\ 0101011 \end{array}$$

Add $Y =$

$$\begin{array}{r} 1000011 \\ \hline \end{array}$$

$$\begin{array}{r} 1101110 \\ \hline \end{array}$$

No carry, Take 1^s comp of result

$$Y - X = -(0010001)_2$$

using 2^s complement.

a) $X - Y$.

Take 2^s comp of $Y = 0111100$

$$\begin{array}{r} \textcircled{1}\textcircled{1}\textcircled{1}\textcircled{1} \\ 0111100 \\ + 1 \end{array}$$

Add $X +$

$$\begin{array}{r} 1010100 \\ \hline \end{array}$$

$$\textcircled{1}0010001$$

Ignore carry,

$$\underline{X - Y = (0010001)_2}$$

b) $Y - X$

Take 2^s comp of $X = 0101011$

$$\begin{array}{r} \textcircled{1}\textcircled{1} \\ 0101011 \\ + 1 \end{array}$$

Add $Y +$

$$\begin{array}{r} 1000011 \\ \hline 1101111 \end{array}$$

No carry, Take 2^s comp of result

$$\begin{array}{r} 0010000 \\ + 1 \end{array}$$

$$(-0010001)_2$$

$$\underline{Y - X = (-0010001)_2}$$

Boolean Algebra:

Definition: Boolean algebra is a system of mathematical logic. It is an algebraic system consisting on the set of elements (0, 1) two binary operators called OR, AND and one unary operator NOT.

Basic operations:

Three basic operations involved in Boolean algebra are i) AND ii) OR iii) NOT - complement.

i) AND - Multiplication ($\cdot$)

$$Y = A \cdot B$$

Truth-Table:

A	B	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

* In AND Operation, the output is 1 if both A and B are 1. Otherwise it is zero.

ii) OR - Addition Symbol ($+$)

$$Y = A + B$$

Truth-Table:

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

* In OR operation, the output is 1, if either A or B or both are 1.

iii) NOT - complement ($- /$)

$$Y = \bar{A}$$

Truth-Table:

A	Y
0	1
1	0

* In NOT operation, the output is complement of the input.

Duality theorem:

→ It states that every Boolean algebraic expression remains valid if the operator (+ and $\cdot$) and the identity elements (0 and 1) are interchanged.

→ If the dual of Boolean expression is required, we simply interchange OR and AND operators and replace 0's by 1's and 1's by 0's.

Boolean Theorems and Identities:

S.No	NAME	AND form	OR form.
1.	Identity Law	$1 \cdot A = A$	$0 + A = A$
2.	Null Law	$0 \cdot A = 0$	$1 + A = 1$
3.	Idempotent Law	$A \cdot A = A$	$A + A = A$
4.	Inverse Law	$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$
5.	Commutative Law	$AB = BA$	$A + B = B + A$
6.	Associative Law	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A + B) + C = A + (B + C)$
7.	Distributive Law	$A + B \cdot C = (A + B) \cdot (A + C)$	$A \cdot (B + C) = A \cdot B + A \cdot C$
8.	Absorption or redundancy Law	$A \cdot (A + B) = A$	$A + AB = A$
9.	Involution Law	$\bar{\bar{A}} = A$	$\bar{\bar{A}} = A$
10.	De-Morgan's Theorem	$\overline{A \cdot B} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A} \cdot \bar{B}$

Prove the commutative law of Boolean Algebra using truth Table.

i) $A \cdot B = B \cdot A$ ii) $A + B = B + A$

(i) $A \cdot B = B \cdot A$

A	B	$A \cdot B$	$B \cdot A$
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

Hence $A \cdot B = B \cdot A$

Associative Law:

i) $A + (B + C) = (A + B) + C$

A	B	C	$B + C$	$A + (B + C)$	$A + B$	$(A + B) + C$
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

Hence $A + (B + C) = (A + B) + C$

2) $A(BC) = (AB)C$

A	B	C	BC	$A(BC)$	AB	$(AB)C$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	1	0
1	1	1	1	1	1	1

1	1	0	0	0	0	0
1	1	1	1	1	1	1

Hence $A(BC) = (AB)C$

Distributive law!

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

A	B	C	B+C	A(B+C)	AB	AC	AB+AC
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

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

ii) $A+BC = (A+B)(A+C)$

A	B	C	BC	A+BC	A+B	A+C	(A+B)(A+C)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Hence $A+BC = (A+B)(A+C)$

* State and prove a) distributive b) absorption law of Boolean algebra using identities.

Soln: a) distributive law.

P) $A+BC = (A+B)(A+C)$

Soln: $RHS = (A+B)(A+C)$

$$= A \cdot A + A \cdot C + A \cdot B + B \cdot C$$

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

$$= A(1+C+B) + BC \quad \because 1+A=1$$

$$= A \cdot 1 + BC$$

$$= A + BC$$

$$= LHS.$$

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

$$RHS = AB+AC$$

$$= A(B+C)$$

$$= LHS$$

b) Absorption Law:

i) $A+AB = A$

$$LHS = A+AB$$

$$= A(1+B)$$

$$= A(1)$$

$$= A$$

$$= RHS.$$

ii) $A(A+B) = A$

$$LHS = A(A+B)$$

$$= AA+AB$$

$$= A+AB$$

$$= A(1+B)$$

$$= A$$

$$= RHS$$

De-Morgan's Theorem:

It states that,

$$i) \overline{AB} = \overline{A} + \overline{B} \quad ii) \overline{A+B} = \overline{A} \cdot \overline{B} \text{ for 2 variables.}$$

- i) The product of complement is equal to the sum of individual complements.
- ii) The sum of complements is equal to the product of individual complements.

For example, for 3 Variables.

$$(i) \overline{ABC} = \overline{A} + \overline{B} + \overline{C}$$

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

For 4-Variables:

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

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

* State and prove the De-Morgan's theorem for two variables using truth table.

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

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

column 4 & 7 are equal. Hence,

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

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

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

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

column 4 and 7 are equal, Hence

$$\overline{AB} = \overline{A+B} \text{ proved.}$$

* Prove the De-Morgan's theorem for 3-variables (A,B,C) using truth table.

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

$$(ii) \overline{(ABC)} = \overline{A} + \overline{B} + \overline{C}$$

Simplification of Boolean functions.

* Simplify the following Boolean expressions.

i) $A + AC$

ii) $A + \overline{A}B + AB\overline{C}$

Soln: $A + AC$
 $= A(1+C) \{1+C=1\}$
 $= A \cdot 1$
 $= A$
A

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

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

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

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

iii) $\overline{A}C + \overline{A}\overline{C}$
 $= \overline{A}C + \overline{A} + \overline{C}$
 $= \overline{A}(C+1) + \overline{C}$
 $= \overline{A} \cdot 1 + \overline{C}$
 $= \overline{A} + \overline{C}$
 $\overline{A} + \overline{C}$

$$= 1 \cdot (A+B)$$

$$= A+B$$
A+B

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

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

vi) $AB + \overline{A} + \overline{A}B$
 $= AB + \overline{A} + \overline{A} + \overline{B}$ {De-Morgan}
 $= AB + \overline{A} + \overline{B}$
 $= AB + \overline{AB} = 1$
1

Vii) $(B + \bar{C})(\bar{B} + C) + \overline{A + B + \bar{C}}$

$$= B\bar{B} + BC + \bar{C}\bar{B} + \bar{C}C + \bar{A} \cdot \bar{B} \cdot \bar{C} \quad \{\text{De-Morgan's}\}$$

$$= 0 + BC + \bar{B}\bar{C} + 0 + A\bar{B}\bar{C}$$

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

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

$$= \bar{B}\bar{C} + C[(A + B)(B + \bar{B})] \quad \{\text{Distributive law}\}$$

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

$$= \bar{B}\bar{C} + BC + AC.$$

Viii) prove that $\overline{AB + \bar{A} + AB} = 0.$

Soln: LHS = $\overline{AB + \bar{A} + AB}$

$$= \overline{AB} \cdot \bar{\bar{A}} \cdot \overline{AB}$$

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

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

$$= AB\bar{A} + AB\bar{B} \quad A \cdot \bar{A} = 0, \quad B \cdot \bar{B} = 0$$

$$= 0 + 0$$

$$= 0 = \text{RHS.}$$

(ix) Prove that $AB + A + AB = A$

$$\text{LHS} = AB + A + AB$$

$$= AB + A \quad A + A = A$$

$$= A(1 + B)$$

$$= A$$

(x) $\bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}$

$$= \bar{X}\bar{Y}(\bar{Z} + Z + 1) + X\bar{Y} \quad 1 + A = 1$$

$$= \bar{X}\bar{Y} + X\bar{Y}$$

$$= \bar{Y}(\bar{X} + X) \quad X + \bar{X} = 1$$

$$= \bar{Y}$$

Prove the following identities:

1)

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

$$\text{LHS} = A + \bar{A}B$$

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

$$= (A + B) = \text{RHS.}$$

2)

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

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

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

$$= 0 + AC + \bar{A}B + BC(A + \bar{A}) \quad \{A + \bar{A} = 1\}$$

$$= AC + \bar{A}B + ABC + \bar{A}BC$$

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

$$= AC + \bar{A}B = \text{RHS}$$

3)

$$(A + C)(A + D)(B + C)(B + D) = AB + CD$$

$$\text{LHS} = (A + C)(A + D)(B + C)(B + D)$$

$$= (A + CD)(B + CD)$$

$$= AB + CD$$

$$= \text{RHS.}$$

Simplify the following Boolean expressions to a minimum number of literals.

1)

$$\overline{XY + XYZ + X(Y + X\bar{Y})}$$

$$\text{LHS} = \overline{XY + XYZ} \cdot \overline{X(Y + X\bar{Y})}$$

$$= (\overline{XY + XYZ}) \cdot \bar{X} + \overline{(Y + X\bar{Y})}$$

$$= XY(1 + Z) \cdot \bar{X} + (Y + X)$$

$$= (XY)(\bar{X} + \bar{Y}\bar{X})$$

$$= X\bar{X}Y + XY\bar{Y}\bar{X}$$

$$= 0 + 0$$

$$= 0$$

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

$$\begin{aligned}
 2) \quad & ABC + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC \\
 &= AB(C + \bar{C}) + A\bar{B}C + \bar{A}BC \\
 &= AB + A\bar{B}C + \bar{A}BC \\
 &= A(B + \bar{B}C) + \bar{A}BC \quad 1 + \bar{C} = 1 \\
 &= A(B + \bar{B})(B + C) + \bar{A}BC \quad A + BC = (A + B)(A + C) \\
 &= A(B + C) + \bar{A}BC = AB + AC + \bar{A}BC \\
 &= C(A + \bar{A}B) + AB = C[(A + \bar{A})(A + B)] + AB \\
 &= C(A + B) + AB \\
 &= \underline{AB + BC + CA}
 \end{aligned}$$

$$\begin{aligned}
 3) \quad & F = B[(A + B')(B + C)] \\
 &= B[AB + AC + B'B + B'C] \quad BB' = 0 \\
 &= ABB + ABC + BB'B + BB'C \quad BB = B \\
 &= AB + ABC + 0 + 0 \quad 1 + C = 1 \\
 &= AB[1 + C] \\
 &= \underline{AB}
 \end{aligned}$$

$$\begin{aligned}
 4) \quad & Y = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D \\
 &= \bar{B}\bar{C}[\bar{A}\bar{D} + \bar{A}D + A\bar{D} + AD] \\
 &= \bar{B}\bar{C}[\bar{A}(\bar{D} + D) + A(\bar{D} + D)] \quad \bar{D} + D = 1 \\
 &= \bar{B}\bar{C}[\bar{A} + A] \quad \Rightarrow \bar{A} + A = 1 \\
 &Y = \underline{\bar{B}\bar{C}}
 \end{aligned}$$

$$\begin{aligned}
 5) \quad & (wx + \bar{w}y')(x + w) + wx(x' + y') \\
 &= wx + \bar{w}y'x + wxw + \bar{w}y'w + wxx' + \bar{w}xy' \\
 &= wx + \bar{w}xy' + wx + \bar{w}y' + 0 \\
 &= wx + \bar{w}xy' + \bar{w}y' \\
 &= wx + \bar{w}y'(x + 1) = wx + \bar{w}y' = \underline{w(x + y')}
 \end{aligned}$$

$$\begin{aligned}
 6) \quad Y &= \overline{A}\overline{B} + \overline{A}\overline{C} + A\overline{B}\overline{C} + (\overline{A}\overline{B} + \overline{C}) \\
 &= \overline{A} + \overline{B} + \overline{A} + \overline{C} + A(\overline{B} + \overline{C}) + (\overline{A}\overline{B} \cdot \overline{C}) \\
 &= \overline{A} + \overline{B} + \overline{C} + A\overline{B} + A\overline{C} + (\overline{A} + \overline{B})\overline{C} \\
 &= \overline{A} + \overline{B} + \overline{C} + A\overline{B} + A\overline{C} + \overline{A}\overline{C} + \overline{B}\overline{C} \\
 &= \overline{A}(1 + \overline{C}) + \overline{B}(1 + \overline{C}) + \overline{C}(1 + A) + A\overline{B} \\
 &= \overline{A} + \overline{B} + \overline{C} + A\overline{B} \\
 &= \overline{A} + \overline{C} + \overline{B}(1 + A)
 \end{aligned}$$

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

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

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

$$1 + \overline{C} = 1$$

$$1 + A = 1$$

$$\underline{Y = \overline{A} + \overline{B} + \overline{C}}$$

$$\begin{aligned}
 7) \quad Y &= A(\overline{A}\overline{B}\overline{C} + A\overline{B}\overline{C}) \\
 &= A[\overline{A} + \overline{B} + \overline{C} + A\overline{B}\overline{C}] \\
 &= A\overline{A} + A\overline{B} + A\overline{C} + A\overline{B}\overline{C} \\
 &= 0 + A\overline{B} + A\overline{C} + A\overline{B}\overline{C} \\
 &= A\overline{B}(1 + \overline{C}) + A\overline{C} \\
 &= A\overline{B} + A\overline{C}
 \end{aligned}$$

$$A\overline{A} = 0$$

$$AA = A$$

$$1 + \overline{C} = 1$$

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

$$8) \text{ Show that } \overline{A}\overline{B}C + B + B\overline{D} + AB\overline{D} + \overline{A}C = B + C.$$

$$\text{LHS} = \overline{A}\overline{B}C + B + B\overline{D} + AB\overline{D} + \overline{A}C$$

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

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

$$= C[\overline{A}\overline{B} + \overline{A}] + B$$

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

$$= C[\overline{A} + \overline{B}] + B$$

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

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

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

$$1 + A = 1$$

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

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

$$B + \overline{B} = 1$$

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

$$= B + C$$

$$= \text{RHS.}$$

$$\begin{aligned} 9) \quad & \underline{xyz} + \underline{x'y} + \underline{xy z'} \\ &= xy(z + z') + x'y \quad \{ z + z' = 1 \\ &= xy + x'y \quad \{ x + x' = 1 \\ &= y(x + x') \\ &= \underline{y}. \end{aligned}$$

$$\begin{aligned} 10) \quad & xz + x'zy \\ &= z(x + x'y) \quad \{ x + x'y = (x + x')(x + y) \\ &= z(x + x')(x + y) \quad \{ x + x' = 1 \\ &= z \underline{(x + y)} \end{aligned}$$

$$\begin{aligned} 11) \quad & y(wz' + wz) + xy \\ &= wyz' + wyz + xy \\ &= wy(z + z') + xy \quad \{ z + z' = 1 \\ &= wy + xy \\ &= y \underline{(x + w)} \end{aligned}$$

$$\begin{aligned} 12) \quad & x'y' + xy + x'y \\ &= x'(y' + y) + xy \\ &= x' + xy. \end{aligned}$$

$$\begin{aligned} 13) \quad & (x + y)(x + y') = x + xy' + xy + yy' \\ &= x(1 + y') + xy + 0 \quad \{ yy' = 0 \\ &= x + xy \quad \{ 1 + y' = 1 \\ &= x(1 + y) = \underline{x}. \quad \{ 1 + y = 1 \end{aligned}$$

21) Simplify the Boolean expression and draw its logic circuit or implement using basic gates.

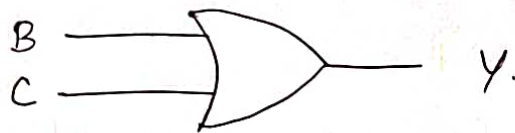
$$Y = (\bar{A} + B + C)(A + B + C)$$

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

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

$$= B \cdot 1 + C \cdot 1$$

$$Y = \underline{\underline{B + C}}$$



Complement of a Function:

* Find the complement of the functions $f_1 = x'yz' + x'y'z$ and $f_2 = x(y'z' + yz)$.

Solⁿ: $f_1 = x'yz' + x'y'z$

$$(f_1)' = (x'yz' + x'y'z)'$$

$$= (x'yz')' \cdot (x'y'z)'$$

$$= (x + y' + z)(x + y + z')$$

$$f_2 = x(y'z' + yz)$$

$$(f_2)' = [x(y'z' + yz)]'$$

$$= x' + (y'z' + yz)'$$

$$= x' + (y'z')' \cdot (yz)'$$

$$= x' + (y + z)(y' + z')$$

i) $f_1 = a\bar{b} + c$

$$(f_1)' = (a\bar{b} + c)' = (a\bar{b})' \cdot c' = (a' + b)c'$$

Apply De-Morgan's theorem

$$\begin{aligned}
 14) \quad & x'y + xy' + xy + x'y' \\
 &= x'(y+y') + x(y'+y) \\
 &= x' + x \\
 &= \underline{\underline{1}}
 \end{aligned}$$

$$\begin{aligned}
 y+y' &= 1 \\
 x+x' &= 1
 \end{aligned}$$

$$\begin{aligned}
 15) \quad & x' + xy + xz' + xy'z' = x' + xy + xz'(1+y') \\
 &= \underline{\underline{x' + xy + xz'}} \quad 1+y'=1
 \end{aligned}$$

$$\begin{aligned}
 16) \quad & xy' + y'z' + x'z' = \\
 &= \overbrace{xy' + x'z'} + y'z' \\
 &= xy' + x'z' \quad \left\{ \begin{array}{l} \text{consensus theorem} \\ y'z' - \text{neglected} \end{array} \right\}
 \end{aligned}$$

$$\begin{aligned}
 17) \quad & xy + x'z + yz \\
 &= xy + x'z + yz(x+x') \quad \text{Add } (x+x') \\
 &= xy + x'z + xyz + x'yz \\
 &= xy(1+z) + x'z(1+y) \quad \begin{array}{l} 1+y=1 \\ (1+z)=1 \end{array} \\
 &= xy + x'z
 \end{aligned}$$

$$\begin{aligned}
 18) \quad & abc + a'b + abc' \\
 &= ab(c+c') + a'b \\
 &= ab + a'b \\
 &= b(a+a') \\
 &= \underline{\underline{b}}
 \end{aligned}$$

$$\begin{aligned}
 19) \quad & x'yz + xz \\
 &= z(x'y + x) \\
 &= z[(x+x')(y+y)] \\
 &= z(x+y)
 \end{aligned}$$

$$\begin{aligned}
 20) \quad & (x+y)'(x'+y') = x'y'(x'+y') \\
 &= x'x'y' + x'y'y' \\
 &= x'y' + x'y' \\
 &= x'y' //
 \end{aligned}$$

$$\phi_2 = (a+b)(\bar{a}+c)$$

$$(\phi_2)' = [(a+b)(\bar{a}+c)]'$$

$$= (a+b)' + (\bar{a}+c)'$$

$$(\phi_2)' = (a' \cdot b') + (a \cdot c')$$

* Find the complement of the functions ϕ_1 and ϕ_2 by taking their duals and complementing each literal.

i) $\phi_1 = x'yz' + x'y'z$ ii) $\phi_2 = x(y'z' + yz)$

Soln. (i) $\phi_1 = x'yz' + x'y'z$

The dual of ϕ_1 is $(x'+y+z')(x'+y'+z)$

Complement of each literal is

$$(x+y'+z)(x+y+z') = (\phi_1)'$$

(ii) $\phi_2 = x(y'z' + yz)$

The dual of ϕ_2 is $x + (y'+z')(y+z)$

Complement of each literal is

$$x' + (y+z)(y'+z') = (\phi_2)'$$

Canonical and Standard forms

Minterms and Maxterms

* Literal: A literal is a boolean variable or its complement. If x be a binary variable then both x and $\bar{x}$ would be literals.

* Minterm: A minterm is a Boolean AND function containing exactly one instance of each input variable or its inverse.

* Maxterm: A maxterm is a Boolean OR function with exactly one instance of each variable or its inverse.

For a combinational logic circuits with n inputs variable, there are 2^n possible minterms & 2^n possible maxterms.

Minterms and Maxterms for three Binary values:

			minterms		Maxterms	
x	y	z	Term	Designation	Term	Designation
0	0	0	$x'y'z'$	m_0	$x+y+z$	M_0
0	0	1	$x'y'z$	m_1	$x+y+z'$	M_1
0	1	0	$x'yz'$	m_2	$x+y'+z$	M_2
0	1	1	$x'yz$	m_3	$x+y'+z'$	M_3
1	0	0	$xy'z'$	m_4	$x'+y+z$	M_4
1	0	1	$xy'z$	m_5	$x'+y+z'$	M_5
1	1	0	xyz'	m_6	$x'+y'+z$	M_6
1	1	1	xyz	m_7	$x'+y'+z'$	M_7

* Consider an example, as shown in the table below.

x	y	z	function (f_1)	function (f_2)
0	0	0	0	0
0	0	1	1	0
0	1	0	0	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

Note:

Form a minterm of each combination of the variables that produces a 1 in function. & then form the OR of all minterms

from table, minterms of function f_1 is

$$f_1 = x'y'z + xy'z' + xyz$$

$$f_1 = \underline{m_1} + m_4 + m_7$$

minterms of function f_2 is

$$f_2 = x'yz + xy'z + xyz + xyz' = m_3 + \underline{m_5} + m_6 + m_7$$

Note: Form a maxterm of each combination of the variables that produces a '0' in function and then form the AND of all maxterms.

$\therefore$ Maxterm of function f_1 is

$$f_1 = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z)$$

$$f_1 = \underline{M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6}$$

Maxterm of function f_2 is

$$f_2 = (x+y+z)(x+y+z')(x+y'+z)(x'+y+z)$$

$$f_2 = \underline{M_0 \cdot M_1 \cdot M_2 \cdot M_4}$$

Canonical form:

Boolean functions expressed as a sum of minterms or product of maxterms are said to be in canonical form.

Sum of Minterms:

- ① Express the Boolean function $F = A + B'C$ in a sum of minterms.

Soln: The function has three variables A, B and C. The first term A is missing two variables, $F = A + B'C \rightarrow$ ①

$$\therefore A = A(B+B') = AB + AB'$$

This is still missing one variable,

$$A = AB(C+C') + AB'(C+C')$$

$$A = ABC + ABC' + AB'C + AB'C' \rightarrow$$
 ②

The second term $B'C$ is missing one variable,

$$B'C = B'C(A+A') = AB'C + A'B'C \rightarrow$$
 ③

Sub ② & ③ in ①.

$$\textcircled{1} \rightarrow F = ABC + ABC' + \underline{AB'C} + AB'C' + \underline{A'B'C} + AB'C$$

$$A+A=A$$

$$F = ABC + ABC' + AB'C + AB'C' + A'B'C$$

$$F = m_1 + m_4 + m_5 + m_6 + m_7$$

$$F(A, B, C) = \sum (1, 4, 5, 6, 7)$$

Truth Table: $Y = A + B'C$

A	B	C	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Note: Σ denotes ORing of terms.

Π denotes ANDing of terms.

Product of Maxterms:

- ① Express the Boolean function $F = xy + x'z$ in a product of Maxterm form.

Soln: $F = xy + x'z \longrightarrow \textcircled{1}$

The given function is in Sum of product.
So, first convert the function into OR terms using distributive law.

$$F = xy + x'z = (xy + x')(xy + z) \\ = (x + x')(y + x')(x + z)(y + z)$$

$$F = (x' + y)(x + z)(y + z) \longrightarrow \textcircled{2}$$

The function has three variables x, y and z .
Each OR term is missing one variable;

$$\therefore x' + y = x' + y + zz' = (x' + y + z)(x' + y + z')$$

$$x + z = x + z + yy' = (x + z + y)(x + z + y')$$

$$y + z = y + z + xx' = (x + y + z)(x' + y + z)$$

Combining all the terms and removing those that appear more than once, we obtain,

$$F = (x+y+z)(x+y'+z)(x'+y+z)(x'+y+z')$$

$$F = M_0 \cdot M_2 \cdot M_4 \cdot M_5$$

$$\therefore F(x, y, z) = \Pi(0, 2, 4, 5)$$

Conversion between Canonical forms:

Consider the function,

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$

This has a complement that can be expressed as

$$F'(A, B, C) = \Sigma(0, 2, 3) = m_0 + m_2 + m_3$$

Now, if we take complement of F' by De Morgan's Theorem,

$$(F')' = F = (m_0 + m_2 + m_3)' = m_0' \cdot m_2' \cdot m_3'$$

$$F = M_0 \cdot M_2 \cdot M_3$$

$$F = \Pi(0, 2, 3)$$

Ex: Consider, the Boolean expression

$$F = xy + x'z$$

Truth Table:

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

from the truth table, The function expressed in Sum of minterms is $F(x, y, z) = \Sigma(1, 3, 6, 7)$

The function expressed in product of maxterm is

$$F(x, y, z) = \Pi(0, 2, 4, 5).$$

Standard forms:

* Another way to express Boolean functions is in standard form.

* There are two types of standard forms

- sum of products
- product of sums.

① Sum of products:

The SOP is a Boolean expression containing AND terms, called product terms, of one or more literals each. The sum denotes the ORing of these terms.

Function expressed in sum of products is

$$F_1 = y' + xy + x'y z'.$$

② Product of Sums:

The POS is a Boolean expression containing OR terms, called sum terms. Each term may have any number of literals. The product denotes the ANDing of these terms.

Function expressed in product of sums is

$$F_2 = x(y' + z)(x' + y + z' + w)$$

Other logic Operations:

* When the binary operators AND and OR are placed between two variables x and y , they form two Boolean functions $x \cdot y$ and $x + y$ respectively.

* It was stated that there are 2^n possible functions for

for n binary Variables.

* For two Variables, $n=2$ and the number of possible Boolean function is 16.

* The truth table for the 16 functions formed with two binary variables x and y are listed below.

x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

* Boolean Expression for 16 functions of two Variables.

Boolean function	operator symbol	Name	Comments
$F_0 = 0$	—	Null	Binary Constant 0
$F_1 = xy$	$x \cdot y$	AND	x and y .
$F_2 = xy'$	x/y	Inhibition	x but not y
$F_3 = x$		Transfer	x
$F_4 = x'y$	y/x	Inhibition	y but not x
$F_5 = y$		Transfer	y
$F_6 = xy' + x'y$	$x \oplus y$	Exclusive-OR	x or y but not both
$F_7 = x + y$	$x + y$	OR	x or y
$F_8 = (x + y)'$	$x \downarrow y$	NOR	NOT-OR
$F_9 = xy + x'y'$	$x \odot y$	Equivalence	x equals y
$F_{10} = y'$	y'	Complement	Not y
$F_{11} = x + y'$	$x \leq y$	Implication	If y then x
$F_{12} = x'$	x'	Complement	Not x
$F_{13} = x' + y$	$x \geq y$	Implication	If x then y
$F_{14} = (xy)'$	$x \uparrow y$	NAND	Not-AND
$F_{15} = 1$		Identity	Binary Constant -1

Digital logic gates:

A logic gate is a digital device that performs a Boolean function, a logical operation performed on one or more binary inputs that produces a single binary output.

There are different types of logic gates listed below.

① Buffer Gate:

The output follows the input.

Symbol:



Logic Expression:

$$Y = A$$

Truth Table:

A	Y
0	0
1	1

② NOT Gate:

The output is the complement of input.

Symbol:



Logic Expression:

$$Y = \bar{A}$$

Truth Table:

A	Y
0	1
1	0

③ AND Gate:

And gate states that if both the inputs are high, the output will be high.

Symbol:



Logic Expression:

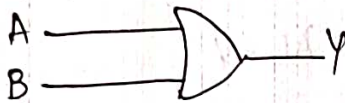
$$Y = A \cdot B$$

Truth Table:

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

- ④ OR gate: The output is high, if any one input is high. It performs addition operation.

Symbol:



Logic Expression:

$$Y = A + B$$

Truth Table:

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

- ⑤ NAND Gate: If any one input is low, the output will be high.

Symbol:



Logic Expression:

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

Truth Table:

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

- ⑥ NOR Gate: When both inputs are low, the output will be high.

Symbol:



Logic Expression:

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

Truth Table:

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

- 7) X-OR Gate: The output will be high if any one input low and high.

Symbol:Truth Table:

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

Logic Expression:

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

$$Y = A \oplus B$$

- 8) X-NOR Gate: The output will be high, when both the inputs are low and high.

Symbol:Truth Table:

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

Logic Expression:

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

$$Y = A \odot B$$

Combinational Logic :

Introduction:

- * logic circuits for digital systems may be combinational or sequential.
- * A combinational circuit consists of logic gates whose outputs are determined directly from present combination of inputs at any time.
- * Sequential circuits employ memory elements in addition to logic gates. Their outputs are a function of the inputs and the state of the memory elements.

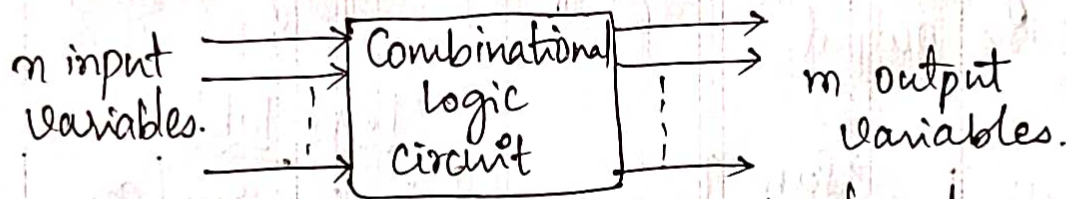


fig: Block diagram of Combinational circuit.

- * For n input variables, there are 2^n possible combinations of binary input values. For each possible input combination, there is one and only one possible output combination. (m).

Design procedure:

The design procedure involves the following steps.

- 1) The problem is stated.
- 2) The number of available input variables and required output variables is determined.
- 3) The input and output variables are assigned letter symbols.
- 4) The truth table that defines the required relationships between inputs and outputs is derived.
- 5) The Simplified Boolean function for each output is obtained.

6) The logic diagram is drawn.

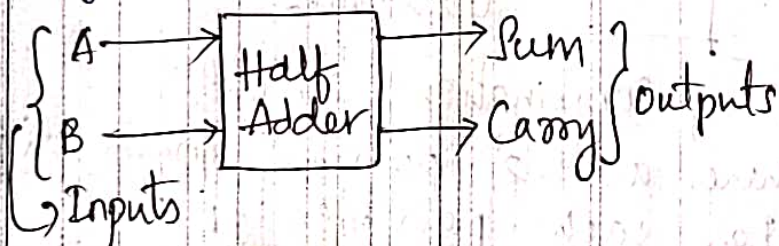
ADDERS:

- * A Digital computers perform a variety of information-processing tasks.
- * Arithmetic operations are the most commonly used functions.
- * Ex: The addition of two binary digits.

Half Adders

- * A combinational circuit that performs addition of two bits. These circuits needs two binary inputs and two binary outputs.

Symbol:

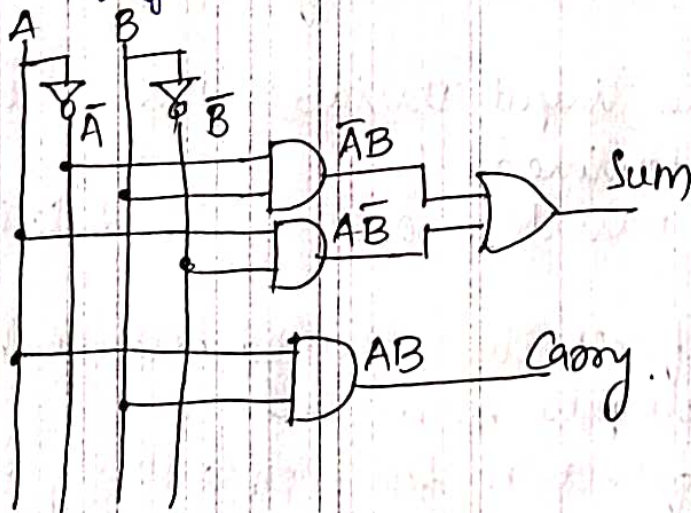


Boolean Expression:

$$\text{Sum} = \bar{A}B + A\bar{B} = A \oplus B$$

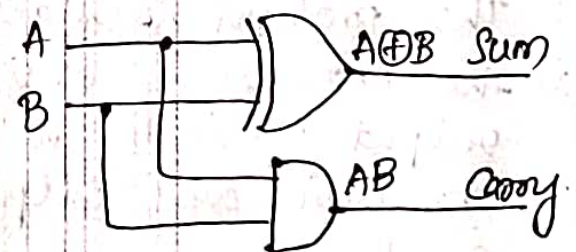
$$\text{Carry} = AB$$

logic diagram:



Truth Table:

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

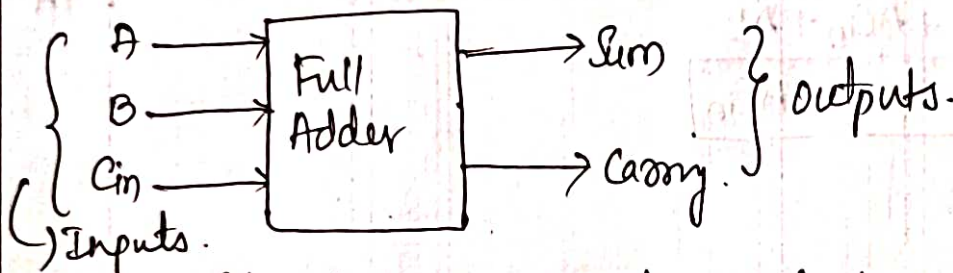


Drawbacks:

- * In half adder, they cannot accept a carry bit from previous stage, meaning that they cannot be chained together to add multi-bit numbers.

Full Adder:

- * A full adder is a combinational circuit that performs the arithmetic sum of three input bits (a two significant bits and a carry from previous state).

Block diagram:

- * It consists of three inputs and two outputs.
- * Two input variables denoted by A and B represents the two significant bits to be added.
- * Third input C_{in} represents the carry from the previous lower significant position.

Truth Table.

A	B	C_{in}	Sum	Carry
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

Boolean Expression:

$$\text{Sum} = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$$

$$\text{Carry} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

Implementation using basic gates:

$$\text{Sum} = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$$

$$\text{Carry} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

$$\text{Add } ABC_{in} + ABC_{in}$$

$$\text{Carry} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in} + ABC_{in} + ABC_{in}$$

Rearrange the terms.

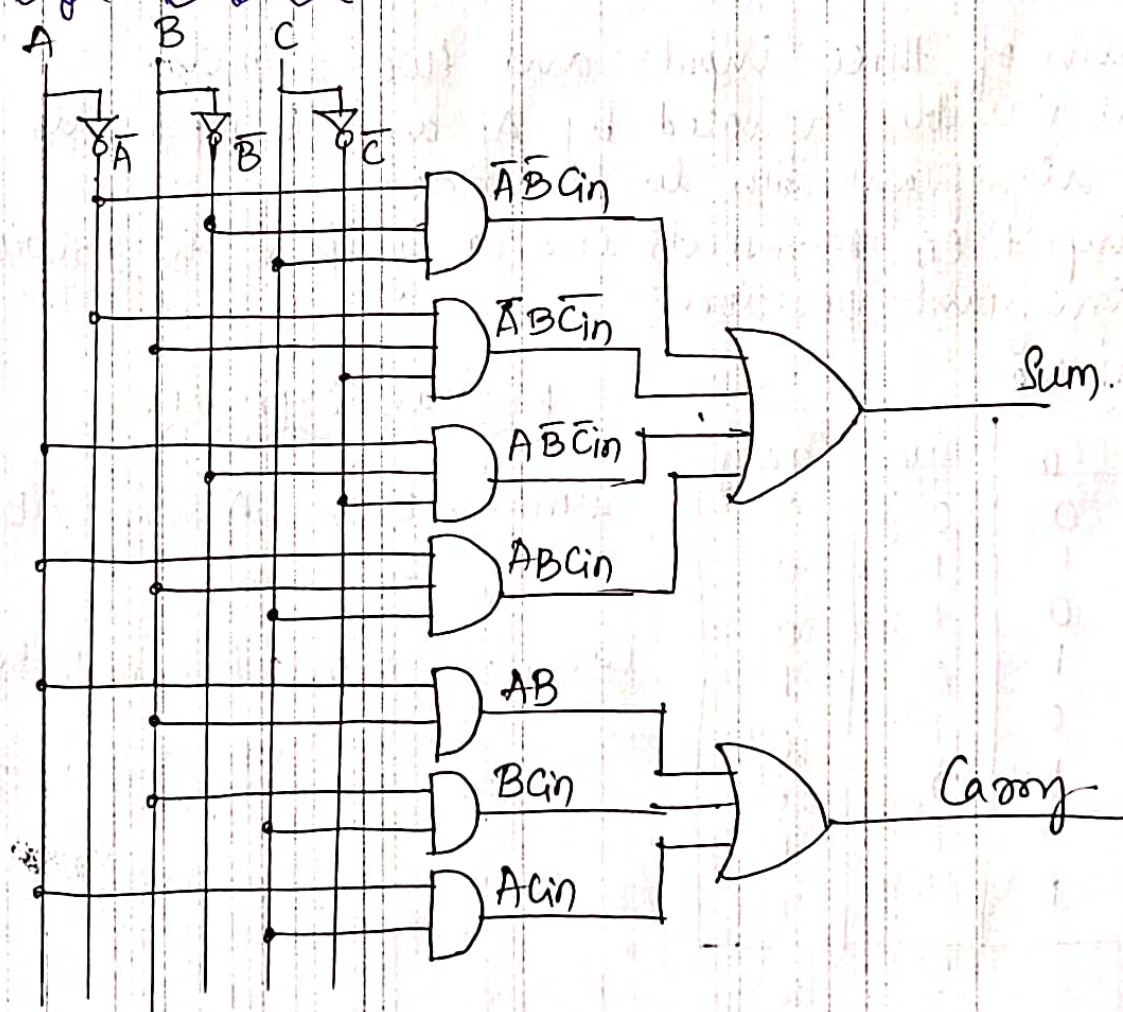
$$= \bar{A}BC_{in} + ABC_{in} + A\bar{B}C_{in} + ABC_{in} + AB\bar{C}_{in} + ABC_{in}$$

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

$$= BC_{in} + AC_{in} + AB$$

$$\text{Carry} = AB + BC_{in} + AC_{in}$$

Logic diagram:



Implementation of Full Adder using 2 Half Adder and 1 OR-gate.

Boolean Expression.

$$\text{Sum} = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$$

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

$$= C_{in}(A \odot B) + \bar{C}_{in}(A \oplus B)$$

$$= C_{in}(\overline{A \oplus B}) + \bar{C}_{in}(A \oplus B)$$

$$\text{let } A \oplus B = X$$

$$= C_{in} \cdot \bar{X} + \bar{C}_{in} \cdot X \rightarrow \text{xor gate}$$

$$= C_{in} \oplus X$$

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

$$\boxed{\text{Sum} = A \oplus B \oplus C_{in}}$$

$$\text{Carry} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

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

$$\bar{A} + A = 1$$

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

logic diagram:

