Tenson Jose
SOE

Module1

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

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

and its decimal equivalent is

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

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

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

and its equivalent is,

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

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

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

and its equivalent is

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

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

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

and its equivalent is

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

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

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	110		D
14	1110		E
15	1111		F

* Decimal to Binary Conversion.

① $(398.75)_{10} = (?)_2$.

↳ Fractional pa

↳ I nteger part.

Integer part.

$$
\begin{array}{r|l}
2 & 398 \\
2 & 199 - 0 \\
2 & 99 - 1 \\
2 & 49 - 1 \\
2 & 24 - 1 \\
2 & 12 - 0 \\
2 & 6 - 0 \\
2 & 3 - 0 \\
 & 1 - 1
\end{array}
\uparrow
$$

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

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

2	172
2	86 – 0
2	43 – 0
2	21 – 1
2	10 – 1
2	5 – 0
2	2 – 1
	1 – 0

$0.625 \times 2 = 1.25$
$0.25 \times 2 = 0.50$
$0.50 \times 2 = 1.00$

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

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

2	725
2	362 – 1
2	181 – 1
2	90 – 1
2	45 – 0
2	22 – 1
2	11 – 0
2	5 – 1
2	2 – 1
	1 – 0

$0.25 \times 2 = 0.50$
$0.50 \times 2 = 1.00$

$(725.25)_{10} = (1011010111.01)$

* **Decimal to Octal Conversion.**

① $(658.825)_{10} = (?)_8$

8	658
8	82 – 2
8	10 – 2
	1 – 2

$0.825 \times 8 = 6.6$
$0.6 \times 8 = 4.8$
$0.8 \times 8 = 6.4$
⋮

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

(2) $(2003)_{10} = (?)_8$

$$\begin{array}{r|l} 8 & 2003 \\ \hline 8 & 250 - 3 \\ \hline 8 & 31 - 2 \\ \hline & 3 - 7 \end{array}$$

$(2003)_{10} = \underline{\underline{(3723)_8}}$

(3) $(11582.875)_{10} = (?)_8$

$$\begin{array}{r|l} 8 & 11582 \\ \hline 8 & 1447 - 6 \\ \hline 8 & 180 - 7 \\ \hline 8 & 22 - 4 \\ \hline & 2 - 6 \end{array}$$

$0.875 \times 8 = 7.00$

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

(4) $(934.705)_{10} = (?)_8$

$$\begin{array}{r|l} 8 & 934 \\ \hline 8 & 116 - 6 \\ \hline 8 & 14 - 4 \\ \hline & 1 - 6 \end{array}$$

$0.705 \times 8 = 5.64$

$0.64 \times 8 = 5.12$

$0.12 \times 8 = 0.96$

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

* Decimal to Hexadecimal Conversion:

(1) $(7084.95)_{10} = (?)_{16}$

$$\begin{array}{r|l} 16 & 7084 \\ \hline 16 & 442 - C \\ \hline 16 & 27 - A \\ \hline & 1 - B \end{array}$$

$0.95 \times 16 = 15.2$

$0.2 \times 16 = 3.2$

$0.2 \times 16 = 3.2$

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

(2) $(4475.85)_{10} = (?)_{16}$

$$\begin{array}{r|l} 16 & 4475 \\ \hline 16 & 279 - D \\ \hline 16 & 17 - 7 \\ \hline & 1 - 1 \end{array}$$

$0.8 \times 16 = 13.6$

$0.6 \times 16 = 9.6$

$0.6 \times 16 = 9.6$

$\underline{\underline{(117D.D99\ldots)_{16}}}$

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

$16 \,|\, 894867$
$16 \,|\, 55929 - 3$
$16 \,|\, 3495 - 9$
$16 \,|\, 218 - 7$
$13 - A$

$0.368 \times 16 = 5.888$
$0.88 \times 16 = 14.208$
$0.208 \times 16 = 3.328$
$\vdots$

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

* Binary to Decimal Conversion:

① $(1\,1\,0\,1\,.01)_2 = (\ ?\ )_{10}$

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

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

$(11101\,01)_2 = (29.25)_{10}$

② $(101010.101)_2 = (\ ?\ )_{10}$

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

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

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

* Octal to decimal Conversion:

① $(327.56)_8 = (\ ?\ )_{10}$

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

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

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

② $(107.70)_8 = (\ ?\ )_{10}$

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

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

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

Hexadecimal to Decimal Conversion:

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

$$= (A\times16^{2})+(B\times16^{1})+(C\times16^{0}) + (9\times16^{-1}) +(0\times16^{-2})$$

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

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

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

$$=(3\times16^{1}) +(A\times16^{0}) +(2\times16^{-1}) +(F\times16^{-2})$$

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

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

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
- 1's complement
- 2's Complement.

Decimal Number system
- 9's complement
- 10's complement.

Complement of Binary Numbers:

<u>1's Complement</u>: 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 ⟶ 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.

(i) Perform $(28)_{10} - (19)_{10}$ using representation.

Soln:

$(28)_{10} = (011100)_2$ — A

$(19)_{10} = (010011)_2$ — B

Take 1^s complement of B

$$\begin{array}{rr} & 1\,0\,1\,1\,0\,0 \\ \text{Add A} + & 0\,1\,1\,1\,0\,0 \\ \hline & 1\;0\,0\,1\,0\,0\,0 \\ \text{Carry} & +1 \\ \hline & (0\,0\,1\,0\,0\,1)_2 \end{array}$$

$= (9)_1$

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

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

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

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

$$
\begin{array}{lr}
\text{1's comp of B} = & \overset{1\,1}{10110000} \\
\text{Add A} \quad + & 0111000 \\
\hline
& 11101000
\end{array}
$$

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

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

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

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

$A = (101011)_2$

$B = (10000)_2$

$$
\begin{array}{lr}
\text{1's comp of B} = & 101111 \\
\text{Add A} \quad + & 101011 \\
\hline
\text{Sign bit } 1 & 011010 \\
\text{Carry} & +1 \\
\hline
& (011011)_2
\end{array}
$$

$(011011)_2 = (27)_{10}$.

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

$A = 1011.10$

$B = 1000.01$

$$
\begin{array}{lr}
\text{1's comp of B} = & 00111.10 \\
\text{Add A} = & 1011.10 \\
\hline
& 10011.00 \\
\text{Carry} \longrightarrow & +1 \\
\hline
& (0011.01)_2
\end{array}
$$

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

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 $\longrightarrow$ 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

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

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

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

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

$$\begin{array}{rr} & 010000 \\ \text{Add A} + & 100111 \\ \hline & 110111 \end{array} \rightarrow \text{No carry.}$$

Take 2's comp of result $\rightarrow$ 001000

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

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

(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 1 0

$$
\begin{array}{r}
0111110 \\
+1 \\
\hline
0111111 \\
\text{Add A} + 1001110 \\
\hline
1\,0001101
\end{array}
$$

Carry ignored.

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

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

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

$(101011)_2 - (11001)_2$.

$A = (101011)_2$

$B = (111001)_2$

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

$$
\begin{array}{r}
000110 \\
+1 \\
\hline
000111 \\
\text{Add A} + 101011 \\
\hline
110010
\end{array}
$$

No carry, Take 2's comp.

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

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

(4) $(1000100)_2 - (1010100)_2$.

$A = (1000100)_2$

$B = (1010100)_2$

Take 2's comp of B = 0101011

$$
\begin{array}{r}
0101011 \\
+\ 1 \\
\hline
0101100 \\
\text{Add A} \quad +\ 1000000 \\
\hline
1110000
\end{array}
$$

No carry, Take 2's comp

$$
\begin{array}{r}
0001111 \\
+\ 1 \\
\hline
(-0010000)_2 = (-16)_{10}.
\end{array}
$$

Subtraction of Decimal Numbers:

To perform Subtraction of Decimal Numbers in 9's complement.

→ The 9's complement of a decimal number may be obtained by subtracting each digit of that number from 9.

Ex: $(354)_{10}$ → 9's complement is

$$
\begin{array}{r}
999 \\
-354 \\
\hline
(645)_{10}
\end{array}
\rightarrow \text{9's complement.}
$$

Operation A−B is performed using following steps.

Step 1: Find the 9's complement of B.

Step 2: Add 9's complement of B + A ⟶ 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 9'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 →

$$\begin{array}{r} 9\ 9\ 9 \\ -\ 3\ 5\ 4 \\ \hline (6\ 4\ 5)_{10} \end{array} \rightarrow \text{9's comp of B}$$

$$\begin{array}{r} \text{Add A} + \quad 4\ 8\ 7 \\ \hline ①\ 1\ 3\ 2 \\ +1 \\ \hline (1\ 3\ 3)_{10} \end{array}$$

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

② $(309)_{10} - (1447)_{10}$

$A = (309)_{10}$

$B = (1447)_{10}$

Take 9's comp of B →

$$\begin{array}{r} 9\ 9\ 9\ 9 \\ -\ 1\ 4\ 4\ 7 \\ \hline 8\ 5\ 5\ 2 \end{array} \rightarrow \text{9's comp of B}$$

$$\begin{array}{r} \text{Add A} + \quad 3\ 0\ 9 \\ \hline 8\ 8\ 6\ 1 \end{array}$$

No carry. Take 9's comp of result.

$$\begin{array}{r} 9\ 9\ 9\ 9 \\ -\ 8\ 8\ 6\ 1 \\ \hline (-1\ 1\ 3\ 8)_{10} \end{array}$$

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

③ $(357.434)_{10} - (1048.05)_{10}$.

$A = (357.434)_{10}$

$B = (1048.05)_{10}$

Take 9's comp of B =

$$
\begin{array}{r}
9999.99 \\
1048.05 \\
\hline
8951.94\;9 \rightarrow \text{signed}
\end{array}
$$

Add A +

$$
\begin{array}{r}
8951.949 \\
+\;357.434 \\
\hline
9309.383
\end{array}
$$

No carry, Take 9's comp of result

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

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

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 $(354)_{10}$

9's comp →

$$
\begin{array}{r}
999 \\
354 \\
\hline
645 \\
+\;1 \\
\hline
(646)_{10}
\end{array}
$$

Add 1

$(646)_{10}$ → 10's 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 → 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's complement of result.

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

$A = (8437)_{10}$

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

Take 10's complement of B = 9999

$$\begin{array}{rr} & 9999 \\ - & 39 \\ \hline & 9960 \\ \text{Add1} + & 1 \\ \hline & 9961 \rightarrow \text{10's complement} \\ \text{Add A} + & 8437 \\ \hline & (1)8398)_{10}. \end{array}$$

Discard carry.

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

2. $(357.434)_{10} - (1048.05)_{10}$.

$A = (357.434)_{10}$

$B = (1048.050)_{10}$.

Take 10's Complement of B = 9999.999

$$\begin{array}{rr} & 9999.999 \\ - & 1048.050 \\ \hline & 8951.949 \\ \text{Add 1} + & 1 \\ \hline & 8951.950 \\ \text{Add A} + & 357.434 \\ \hline & 9309.384 \end{array}$$

No carry, Take 10's 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^s complement of B =

$$\begin{array}{r} 9\;9\;9\;9 \\ -\;3\;2\;5\;0 \\ \hline 6\;7\;4\;9 \\ +\;\;\;\;\;\;\;\;1 \\ \hline 9\;6\;7\;5\;0 \end{array}$$

Add A

$$\begin{array}{r} +\;7\;2\;5\;3\;2 \\ \hline 1\;6\;9\;2\;8\;2 \end{array}$$

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

80hu

X = 1010100

Y = 1000011

<u>using 1's complement</u>

a) X − Y

Take 1's of Y = 0 1 1 1 1 0 0

Add X = 1 0 1 0 1 0 0

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

$X - Y = (0010001)_2$.

b) Y − X

Take 1's comp of X = 0101011

Add Y = 1000011

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

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 = 0 1 1 1 1 1 0 0

$$
\begin{array}{rcccccccc}
 & & & & & & & & +1 \\
\hline
 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
\text{Add } X \;+ & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
\hline
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{array}
$$

Ignore carry,

$X - Y = (0010001)_2$

b) $Y - X$

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

$$
\begin{array}{rccccccc}
 & & & & & & & +1 \\
\hline
 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
\text{Add } Y \;+ & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
\hline
 & 1 & 1 & 0 & 1 & 1 & 1 & 1
\end{array}
$$

No carry, Take 2^s comp of result

$$
\begin{array}{ccccccc}
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 & & & & & & +1 \\
\hline
\end{array}
$$

$(-0\,0\,1\,0\,0\,0\,1)_2$

$Y - X = (-0010001)_2$

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 (·)

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

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

Truth-Table :

A	B	Y=A·B
0	0	0
0	1	0
1	0	0
1	1	1

ii) OR - Addition Symbol (+)

$\boxed{Y = A + B}$

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

Truth-Table :

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

iii) NOT - complement (¯/)

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

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

Truth-Table :

A	Y
0	1
1	0

Duality theorem :

→ It states that every Boolean algebraic expression remains valid if the operator (+ and ·) 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 :

Sl.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 redundance 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.B	B.A
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

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

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

A	B	A+B	B+A
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

Hence $A + B = B + A$

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	0	0
1	1	1	1	1	1	1

Hence $A(BC) = (A \cdot B)C$

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 $= A + AB$

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

$= A(1)$

$= A$

$= RHS.$

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

LHS $= A(A+B)$

$= AA + AB$

$= A + AB$ $\quad \because 1+A=1$

$= A(1+B)$

$= A$

$= RHS$

De-Morgan's Theorem:

It states that,

i) $\overline{AB} = \overline{A} + \overline{B}$ ii) $\overline{A+B} = \overline{A} \cdot \overline{B}$ 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) $\overline{A+B+C+D} = \overline{A} \cdot \overline{B}\,\overline{C} \cdot \overline{D}$

* 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	$\overline{AB}$	$\bar{A}$	$\bar{B}$	$\bar{A}+\bar{B}$
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

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

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

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

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

Simplification of Boolean functions.

* Simplify the following Boolean expressions.

i) $A + AC$

Soln: $A + AC$

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

$= A\cdot 1$

$= A$

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

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

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

$= A + \bar{A}B$

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

$= 1\cdot(A+B)$

$= A+B$

iii) $\bar{A}C + \overline{AC}$

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

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

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

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

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

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

$= (A+B)\cdot C$

v) $\overline{AB + \overline{AB} + A}$

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

$= \overline{AB} \cdot AB\bar{A} = 0$

vi) $AB + \bar{A} + \overline{AB}$

$= AB + \bar{A} + \bar{A} + \bar{B}$ $\{$De-Morgan$\}$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$= 0 + 0.$ $\quad B\cdot\bar{B} = 0$

$= 0 = RHS.$

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

LHS $= AB + A + AB$

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

$= A(1+B)$

$= A$

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

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

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

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

$= \bar{Y}$

Prove the following identities:

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

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

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

$= (A+B) = RHS.$

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

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

$= A\bar{A} + AC + \bar{A}B + BC\cdot 1$

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

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

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

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

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

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

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

$= AB+CD$

$= RHS.$

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

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

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

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

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

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

$= X\bar{X}Y$

$= 0 +$

$= 0$

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

2) $AB\bar{C} + A\bar{B}C + ABC + \bar{A}BC$

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

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

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

$= A[(B+\bar{B})(B+C)] + \bar{A}BC \qquad A+BC=(A+B)(A+C)$

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

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

$= C(A+B) + AB$

$= AB + BC + CA$

3) $F = B[(A+B')(B+C)]$

$= B[AB + AC + B'B + B'C] \qquad BB' = 0$

$= ABB + ABC + BBB' + BB'C \qquad BB = B$

$= AB + ABC + 0 + 0$

$= AB[1+C] \qquad 1+C=1$

$F = AB$

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

$= \bar{B}\bar{C}[\bar{A}\bar{D} + \bar{A}D + A\bar{D} + AD]$

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

$= \bar{B}\bar{C}[\bar{A}+A] \qquad \Rightarrow \bar{A}+A=1$

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

5) $(wx + wy')(x+w) + wx(x'+y')$

$= wx + wy'x + wxw + wy'w + wxx' + wxy'$

$= wx + wxy + wx + wy' + 0$

$= wx + wxy' + wy'$

$= wx + wy'(x+1) = wx + wy' = w(x+y')$

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

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

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

$= \bar{A} + \bar{B} + \bar{C} + A\bar{B} + A\bar{C} + \bar{A}\bar{C} + \bar{B}\bar{C}$ $\qquad \overline{A+B} = \bar{A}\cdot\bar{B}$

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

$= \bar{A} + \bar{B} + \bar{C} + A\bar{B}$ $\qquad 1+A = 1$

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

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

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

$= A[\bar{A} + \bar{B} + \bar{C} + A\bar{B}C]$ $\qquad A\bar{A} = 0$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$= C(1+A^-) + B$

$= B + C$

$= RHS.$

9) $xyz + x'y + xyz'$

$= xy(z+z') + x'y$ $\{ z+z' = 1$

$= xy + x'y$

$= y(x+x')$ $\{ x+x' = 1$

$= y$

10) $xz + x'zy$

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

$= z(x+x')(x+y)$ $\{ x+x' = 1$

$= z(x+y)$

11) $y(wz' + wz) + xy$

$= wyz' + wyz + xy$

$= wy(z+z') + xy$ $(z+z') = 1$

$= wy + xy$

$= y(x+w)$

12) $x'y' + xy + x'y$

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

$= x' + xy.$

13) $(x+y)(x+y') = x + xy' + xy + yy'$ $yy' = 0$

$= x(1+y') + xy + 0$ $1+y' = 1$

$= x + xy$ $1+y = 1$

$= x(1+y) = x.$

2i) Simplify the Boolean expression and draw its logic circuit or implement using basic gates.

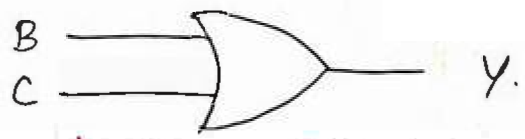
$Y = (\bar{A}+B+C)(A+B+C)$

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

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

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

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



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:

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

$(f_1)' = (x'yz' + x'y'z)'$ Apply De-Morgan's theorem

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

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

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

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

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

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

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

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

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

14) $x'y + xy' + xy + x'y'$

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

$= x' + x$

$= 1$

$y+y'=1$

$x+x'=1$

15) $x' + xy + xz' + xy'z' = x' + xy + xz'(1+y')$

$= x' + xy + xz'$

$1+y'=1$

16) $xy' + y'z' + x'z' =$

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

$= xy' + x'z$

{consensus theorem}

$y'z'$ – neglected

17) $xy + x'z + yz$

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

$= xy + xyz + x'yz$

$= xy(1+z) + x'z(1+y)$

$= xy + x'z$

Add $(x+x')$

$1+y=1$

$(1+z)=1$

18) $abc + a'b + abc'$

$= ab(c+c') + a'b$

$= ab + a'b$

$= b(a+a')$

$= b$

19) $x'yz + xz$

$= z(x'y+x)$

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

$= z(x+y)$

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

$= x'x'y' + x'y'y'$

$= x'y' + x'y'$

$= x'y'$ //

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

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

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

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

* Find the complement of the functions f_1 and f_2 by taking their duals and complementing each literal.

i) $\phi_1 = x'yz' + x'y'z$ ii) $f_2 = x(y'z' + yz)$

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

The dual of f_1 is $(x'+y+z')(x'+y'+z)$

Complement of each literal is

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

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

The dual of f_2 is $x + (y'+z')(y+z)$

Complement of each literal is

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

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.

* Minterms 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

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

$$f_1 = m_1 + m_4 + m_7$$

minterms of function f_2 is

$$f_2 = x'yz + xy'z + = xyz' + m_5 + M_6 + m_7$$

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

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

$$f_1 = M_0 . M_2 . M_3 . M_5 . M_6 .$$

Maxterm of function f_2 is

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

$$f_2 = M_0 . M_1 . M_2 . M_4$$

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, $F = A + B'C \longrightarrow (1)$

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

This is still missing one variable,

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

$$A = ABC + ABC' + AB'C + AB'C' \longrightarrow (2)$$

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

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

Sub (2) & (3) in (1).

$$(1) \Rightarrow F = ABC + ABC' + AB'C + AB'C' + AB'C + A'B'C$$

$A + A = A$

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

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

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

Truth Table: $\boxed{Y = A + B'C}$

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:

$$F = xy + x'z \longrightarrow (1)$$

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

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

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

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

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

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

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

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

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 DeMorgan's theorem,

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

$$F = M_0 \, M_2 \, M_3$$

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

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

(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 $x.y$ and $x+y$ respectively.
* It stated that there are 2^{2n} possible functions

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
$F_{11} = x + y'$	$x \subset y$	Implication	If y then x
$F_{12} = x'$	x'	Complement	Not x
$F_{13} = x' + y$	$x \supset y$	Implication	If x then y
$F_{14} = (xy)'$	$x \uparrow y$	NAND	Not-AND
$F_{15} = 1$		Identity	Binary Constant-1

Digital logic gates:

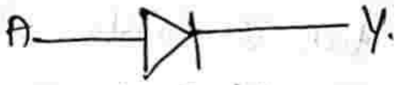
A logic gate is a digital device that performs a Boolean function, a logical operation performed on one or more binary inputs that produces a single binary output.

There are different types of logic gates listed below.

(1) **Buffer Gate:**

The output follows the input.

Symbol:



Logic Expression:

$\boxed{Y = A}$

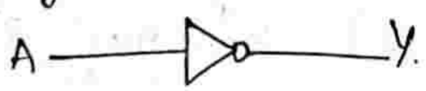
Truth Table.

A	Y
0	0
1	1

(2) **NOT Gate:**

The output is the complement of input.

Symbol:



Logic Expression:

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

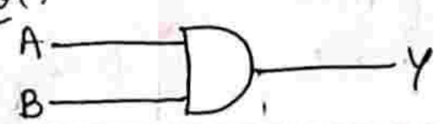
Truth Table:

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:

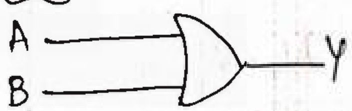
$\boxed{Y = A \cdot B}$

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:

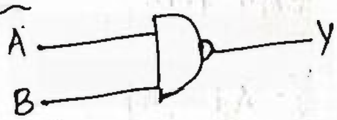
$$\boxed{Y = A + B}$$

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:



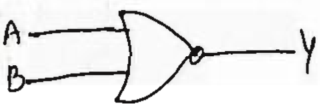
Logic Expression:

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

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.



Logic Expression:

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

Truth Table

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

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

Symbol:

Logic Expression

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

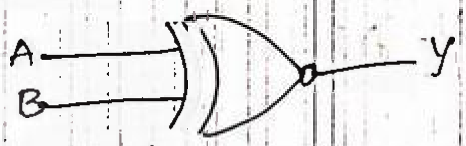
$\boxed{Y = A \oplus B}$

Truth Table

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

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

Symbol:



Logic Expression:

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

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

Truth Table

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

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

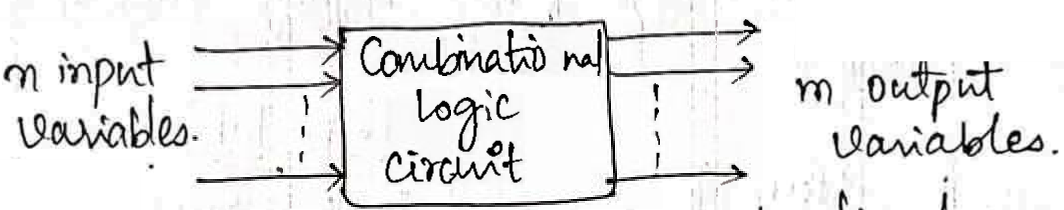


fig: Block diagram of Combinational circuit.

* 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.

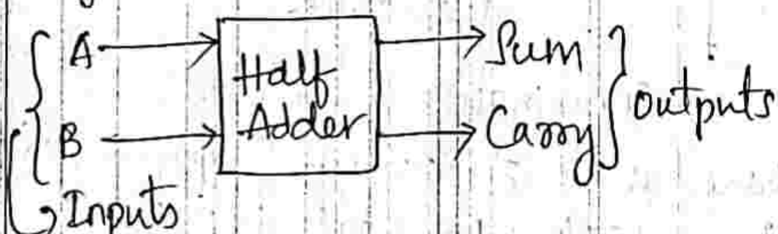
ADDERS:

* 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 Adder:

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

Symbol:



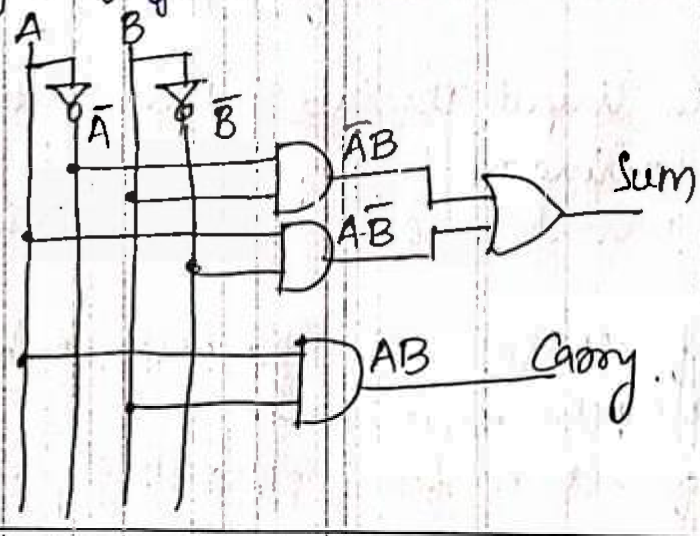
Truth Table:

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

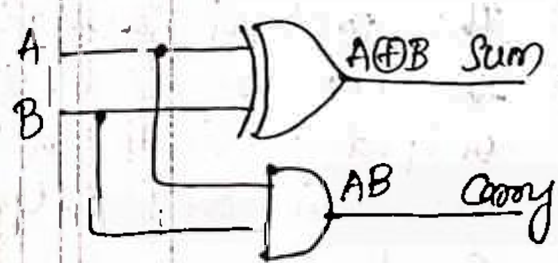
Boolean Expression:

$$Sum = \bar{A}B + A\bar{B} = A \oplus B$$
$$Carry = AB$$

logic diagram:



(or)



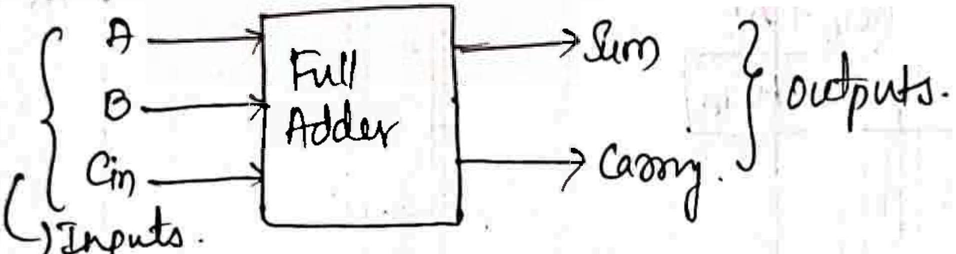
Drawbacks:

* 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	Cin	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:

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

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

Implementation using basic gates:

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

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

WKT $A+A=A$

Add $ABC_{in} + ABC_{in}$

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

Rearrange the terms.

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

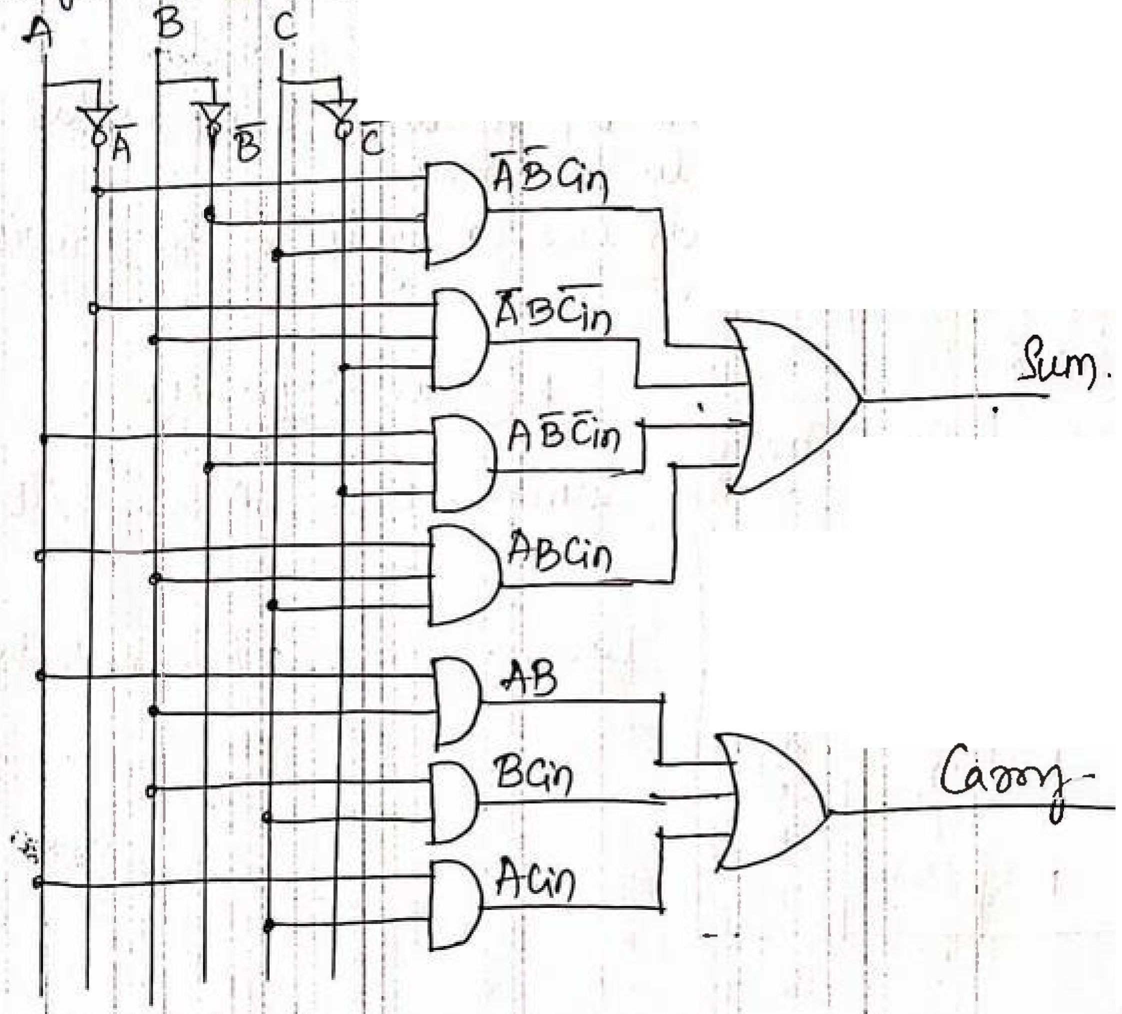
$$= BC_{in}(\bar{A}+A) + AC_{in}(\bar{B}+B) + AB(\bar{C}_{in}+C_{in})$$

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

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

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

Logic diagram:



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

Boolean Expression.

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

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

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

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

let $A \oplus B = X$

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

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

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

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

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

$$= C_{in}(\bar{A}B + A\bar{B}) + AB(\bar{C}_{in} + C_{in}) \qquad \bar{A} + A = 1$$

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

logic diagram: