MODULE 3 : BOOLEAN ALGEBRA AND LOGIC CIRCUITS

Topics: Binary numbers, Number Base Conversion, Octal and Hexa Decimal Numbers, Complements, Basic definitions of Boolean Algebra, Basic Theorems and properties of Boolean Algebra, Boolean Functions, Canonical and Standard forms, Other logic operations, Digital logic gates.

Combinational logic :- Introduction, Design procedure, Adders - Halfadder, fulladder.

BINARY NUMBERS

In a digital Electronics system, the active devices are used to operate as switches, and have only 2 states i.e., ON and OFF.

The coefficients of the binary numbers have two possible values: 0 and 1.

Each coefficient a_j is multiplied by 2^j.

In general, a number expressed in base-r system has coefficients multiplied by powers of r:

$$\boxed{a_n \cdot r^n + a_{n-1} r^{n-1} + \cdots + a_2 \cdot r^2 + a_1 \cdot r + a_0 + a_{-1} \cdot r^{-1} + a_{-2} \cdot r^{-2} + \cdots + a_{-m} \cdot r^{-m}}$$

The coefficient a_j range from 0 to $r-1$.

- To distinguish between numbers of different bases, we Enclose the coefficients in parentheses and write a subscript Equal to base used.
- The Binary number system is base-2 or radix-2 number system.

Dr. Chaya.BM, Associate Professor, Dept of ECE

* The Subscript 10 Indicates that the number is in decimal or base-10 or radix-10 number system.
* In Binary number, Each digit is called bit.
* The left Most bit of a binary number is called MOST SIGNIFICANT BIT (MSB)
* The Right Most bit of a binary number is called LEAST SIGNIFICANT BIT (LSB)

Eg.. 1 0 0 1 1

MSB → Highest Weight

LSB → Least Weight

Convert the following Binary numbers to decimal Numbers.

(1) $(11010)_2 = 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$

$= 16 + 8 + 0 + 2 + 0$

$= \underline{\underline{(26)_{10}}}$

(2) $(11011.1101)_2 = 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2} + 0\times2^{-3} + 1\times2^{-4}$

↳ after decimal point

$= 16 + 8 + 0 + 2 + 1 + 0.5 + 0.25 + 0 + 0.0625$

$= \underline{\underline{(27.8125)_{10}}}$

(3) $(1011.11)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2}$

$= 8 + 0 + 2 + 1 + 0.5 + 0.25$

$= \underline{\underline{(11.75)_{10}}}$

(4) $(1011)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

$= 8 + 0 + 2 + 1$

$= (11)_{10}$

(5) $(11101.01)_2 = 1\times2^4 + 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2}$

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

$= (29.25)_{10}$

(6) $(111110101)_2 = 1\times2^8 + 1\times2^7 + 1\times2^6 + 1\times2^5 + 1\times2^4 + 0\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$

$= 256 + 128 + 64 + 32 + 16 + 0 + 4 + 0 + 1$

$= (501)_{10}$

(7) $(101010.101)_2 = 1\times2^5 + 0\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$

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

$= (42.625)_{10}$

(8) $(1101)_2 = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$

$= 8 + 4 + 1 = (13)_{10}$

(9) $(10001)_2 = 1\times2^4 + 0\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0$

$= 16 + 1 = (17)_{10}$

(10) $(10101)_2 = 1\times2^4 + 0\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$

$= 16 + 4 + 1$

$= (21)_{10}$

We observe that binary numbers take more digits to represent the decimal number.
For large numbers we have to deal with very large binary strings. So this fact gave rise to three new number systems.

(i) Octal Number Systems

(ii) Hexa Decimal number system.

(iii) Binary Coded Decimal number (BCD) System.

Binary number System:-

The binary number has a radix of 2, As $r = 2$, Only two digits are needed, and these are 0 and 1.

Decimal Number System:-

The decimal system has ten symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. In other words, it has a base of 10.

Octal Number System:-

Digital Systems operate only on binary numbers. Since binary numbers are often very long. two Shorthand notations. Octal and hexadecimal are used for representing large binary numbers.
Octal Systems use a base or radix of 8. It uses first Eight digits of decimal Numbers System. Thus it has digits from 0 to 7.

Hexa Decimal Number System.

The Hexadecimal numbering system has a base of 16. There are 16 symbols. The decimal digits 0 to 9 are used as the first ten digits as in the decimal system, followed by letters A, B, C, D, E and F, which represent values, 10, 11, 12, 13, 14 and 15 respectively.

Numbers with different Bases.

Dr. CBM, Dept of ECE

Decimal (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Convert Base-5 System to decimal System.

(a) $(4021.2)_5 = 4\times5^3 + 0\times5^2 + 2\times5^1 + 1\times5^0 + 2\times5^{-1}$

$= 500 + 0 + 10 + 1 + 0.4$

$= (511.4)_{10}$

(b) $(1234)_5 = 1\times5^3 + 2\times5^2 + 3\times5^1 + 4\times5^0$

$= 125 + 50 + 15 + 4$

$= (194)_{10}$

Convert the following Octal Numbers into decimal System.

(a) $(7034)_8 = 7\times8^3 + 0\times8^2 + 3\times8^1 + 4\times8^0$

$= 3584 + 0 + 24 + 4$

$= (3612)_{10}$

(b) $(625.36)_8 = 6\times8^2 + 2\times8^1 + 5\times8^0 + 3\times8^{-1} + 6\times8^{-2}$

$= 384 + 16 + 5 + 0.375 + 0.09375$

$= (405.46875)_{10}$

(c) $(1616.16)_8 = 1\times8^3 + 6\times8^2 + 1\times8^1 + 6\times8^0 + 1\times8^{-1} + 6\times8^{-2}$

$= (910.21875)_{10}$

(d) $(101010)_8 = 1\times8^5 + 0\times8^4 + 1\times8^3 + 0\times8^2 + 1\times8^1 + 0\times8^0$

$= 32768 + 512 + 8 + 0$

$= (33288)_{10}$

Convert the following hexadecimal numbers into decimal System.

A-10
B-11
C-12
D-13
E-14
F-15

① $(B65F)_{16} = 11\times16^3 + 6\times16^2 + 5\times16^1 + 15\times16^0$

$= 45056 + 1536 + 80 + 15$

$= (46687)_{10}$

② $(A38)_{16} = 10\times16^2 + 3\times16^1 + 8\times16^0$

$= (2616)_{10}$

③ $(834.41)_{16} = 8\times16^2 + 3\times16^1 + 4\times16^0 + 4\times16^{-1} + 1\times16^{-2}$

$= (2100.25390625)_{10}$

④ $(ABC.D)_{16} = A\times16^2 + B\times16^1 + C\times16^0 + D\times16^{-1}$

$= 10\times16^2 + 11\times16^1 + 12\times1 + \underset{13}{D}\times16^{-1}$

$= (2748.8125)_{10}$

⑤ $(1110.01)_{16} = 1\times16^3 + 1\times16^2 + 1\times16^1 + 0\times16^0 + 0\times16^{-1} + 1\times16^{-2}$

$= (4368.00390625)_{10}$

Number Base Conversions.

The human beings use decimal number system while computer uses binary number system. Therefore it is necessary to Convert decimal number system into its Equivalent binary.

(i) Binary to Octal number Conversion

(ii) Binary to hexadecimal number Conversion

(iii) Octal to binary Conversion

(iv) Hexadecimal to binary Conversion

(v) Octal to decimal Conversion

(vi) Decimal to Octal Conversion

(vii) Hexadecimal to decimal Conversion

(viii) Decimal to Hexadecimal

(ix) Octal to Hexadecimal Conversion

(x) Hexadecimal to Octal Conversion.

(i) Binary to Octal Conversion.

Step 1:- Make group of 3-bits starting from LSB for Integer part and MSB for fractional part, by adding 0's at the End if required.

Step 2: Write Equivalent Octal number for Each group of 3 bits.

Problems on Binary to Octal Conversion:-

1. $(10101101.0111)_2$

$$\underbrace{10}_{3}\ \underbrace{101}_{2}\ \underbrace{101}_{1} \,.\, \underbrace{011}_{1}\ \underbrace{1}$$

append 0   append 0

$\therefore$

$\underline{010}\ \underline{101}\ \underline{101} . \underline{011}\ \underline{100}$

2 5 5 . 3 4

$\therefore (10101101.0111)_2 = (255.34)_8$

2. $(1101011110\,1\,1101.101110)_2$

$= 110\ 101\ 111\ 011\ 101 . 101\ 110$

$= (65735.56)_8$

3. $(11011111010.0111011)_2$

$\Rightarrow \underline{11}\ \underline{011}\ \underline{111}\ \underline{010} . \underline{011}\ \underline{101}\ \underline{1}$

$\Rightarrow \underline{011}\ \underline{011}\ \underline{111}\ \underline{010} . \underline{011}\ \underline{101}\ \underline{100}$

$\Rightarrow (3372.354)_8$

4. $(110101110)_2$

$= \underline{110}\ \underline{101}\ \underline{110}$

$= \underline{001}\ \underline{101}\ \underline{011}\ \underline{110}$

$= (1536)_8$

5. $(11011011110)_2$

$= \underline{11}\ \underline{011}\ \underline{011}\ \underline{110}$

$= 011\ 011\ 011\ 110$

$= (3336)_8$

6. $(11101101110.11101)_2$

$= 11101101110.11101$

$= \underline{011}\ \underline{101}\ \underline{101}\ \underline{110}.\underline{111}\ \underline{010}$

$= (3556.72)_8$

7. $(0.111101011101)_2$

$= 0.\underline{111}\ \underline{101}\ \underline{011}\ \underline{101}$

$= 0.\underline{111}\ \underline{101}\ \underline{011}\ \underline{010}$

$= (0.7532)_8$

8. $(101010110.100011)_2$

$= 101010110.100011$

$= (526.43)_8$

9. $(101010111100)_2$

$= \underline{101}\ \underline{010}\ \underline{111}\ \underline{100}$

$= (5374)_8$

10. $(1101101110.1001101)_2$

$= 1101101110.1001101$

$= \underline{001}\ \underline{101}\ \underline{101}\ \underline{110}.\underline{100}\ \underline{110}\ \underline{100}$

$= (1556.464)_8$

(ii) Binary to Hexadecimal Number Conversion.

We know that base for Hexadecimal number is 16 and base for binary number is 2.
The base for hexadecimal number is 4^{th} power of base for binary number i.e., $2^4 = 16$.
Therefore, by grouping 4 digits of binary numbers and Converting each group digit to its hexadecimal Equivalent, we can convert binary $\longrightarrow$ its Hexadecimal Equivalent.

I Convert binary number to hexadecimal Equivalent.

1. $(110110111 0.1001101)_2$

$\underbrace{1101101110}_{\text{Integer part}} . \underbrace{1001101}_{\text{fractional part}}$ → Make group of 4 bits

1101101110 . 1001101

Adding 0 to make a group of 4 bits

Adding 0s to make a group of 4 bits

∴ 0011 0110 1110 . 1001 1010

3 6 E . 9A

$= (36E.9A)_{16}$

2. $(110111101.01)_2$

= 110111101.01

= 0001 1011 1101 . 0100

$= (1BD.4)_{16}$

3. $(110111101011101)_2$

= 110 1111 0101 1101

= 0110 1111 0101 1101

$= (6F5D)_{16}$

Dr. CBM, Associate Professor, ECE

Dr. CBM, Associate Professor

4. $(0.1101010\ 111011)_2$

$= 0.\underbrace{1101}_{D}\underbrace{0101}_{5}\ \underbrace{1101}_{D}\ \underbrace{1}\longrightarrow$ append 0's

$= 0.\underbrace{1101}\ \underbrace{0101}\ \underbrace{1101}\ \underbrace{1000}$

$= (0.D5D8)_{16}$

5. $(110101101)_2$

$= \leftarrow 1\ \underbrace{1010}\ \underbrace{1101}$

$= \underbrace{0001}\ \underbrace{1010}\ \underbrace{1101}$

$= (1AD)_{16}$

6. $(111111111111)_2$

$= \underbrace{1111}_{F}\ \underbrace{1111}_{F}\ \underbrace{1111}_{F}$

$= (FFF)_{16}$

7. $(0001\ 0010\ 0100\ 1000\ 1001\ 1010\ 1101\ 1111)_2$

$= \underbrace{0001}_{1}\ \underbrace{0010}_{2}\ \underbrace{0100}_{5}\ \underbrace{1000}_{8}\ \underbrace{1001}_{9}\ \underbrace{1010}_{A}\ \underbrace{1101}_{D}\ \underbrace{1111}_{F}$

$= (125\,89ADF)_{16}$

8. $(1011011110.1100101001 1)_2$

$= 10\ \underbrace{1101}\ \underbrace{1110}.\underbrace{1100}\ \underbrace{1010}\ \underbrace{011}$

Appending 0's

$= 0010\ 1101\ 1110 . 1100\ 1010\ 0110$

$= (2DE.CA6)_{16}$

(iii) Octal to Binary Conversion:

To Convert a given Octal number to a Binary, just replace each Octal digit by its 3-bit binary Equivalent. All we have then to remember is the 3-bit binary Equivalents of the basic digits of the Octal number systems.

Convert Octal to its Binary Equivalent number.

1. Convert $(27633)_8$ into binary

Soln:-

$$\begin{array}{ccccc} 2 & 7 & 6 & 3 & 3 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 010 & 111 & 110 & 011 & 011 \end{array} = (010\ 111\ 110\ 011011)_2$$

2. $(725.25)_8$

$\Rightarrow (111\ 010101.\ 010\ 101)_2$

3. Convert $(7463.245)_8$ to Binary

$\Rightarrow 7\ 4\ 6\ 3.\ 2\ 4\ 5$

$\Rightarrow (111\ 100\ 110\ 011.\ 010\ 100\ 101)_2$

4. $(4352)_8$.

$\Rightarrow \underset{4}{100}\ \underset{3}{011}\ \underset{5}{101}\ \underset{2}{010}$

$= (100011101010)_2$

5. $(75.543)_8 =$ 111, 101, 101, 100, 011

$\Rightarrow (111101.101100011)_2$

6. $(673.124)_8 = (\underbrace{110}_{6}\ \underbrace{111}_{7}\ \underbrace{011}_{3}.\ \underbrace{001}_{1}\ \underbrace{010}_{2}\ \underbrace{100}_{4})_2$

Dr. Chaya. B.M, Dept of ECE

(iv) Hexadecimal to Binary Conversion.

To Convert a given Hexadecimal number to a binary, just replace Each hexadecimal digit by its 4-bit binary Equivalent. All we have to remember is the four-bit binary Equivalents of the basic digits of the hexadecimal number System.

Convert the following hexadecimal numbers into Binary.

(i) $(8BE6.7A)_{16}$

$$\underbrace{1000}_{8}\ \underbrace{1011}_{B}\ \underbrace{1110}_{E}\ \underbrace{0110}_{6}\ .\ \underbrace{0111}_{7}\ \underbrace{1010}_{A}$$

$\therefore (8BE6.7A)_{16} = (1000101111100110.01111010)_2$

(ii) $(306.D)_{16} = (\underbrace{0011}_{3}\ \underbrace{0000}_{0}\ \underbrace{0110}_{6}\ .\ \underbrace{1101}_{D})_2$

(iii) $(8A9.B4)_{16} = \underbrace{1000}_{8}\ \underbrace{1010}_{A}\ \underbrace{1001}_{9}.\ \underbrace{1011}_{B}\ \underbrace{0100}_{4}$ → Remove any leading Zeroes.

$= (100010101001.101101)_2$

(iv) $(2F9A)_{16}$ to binary system.

$$\begin{array}{cccc} 2 & F & 9 & A \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 0010 & 1111 & 1001 & 1010 \end{array} \Rightarrow (0010111110011010)_2$$

(v) $(ABCD.F)_{16} = (1010\ 1011\ 1100\ 1101\ .\ 1111)_2$

(v) Octal to decimal Conversion.

An Octal number can be converted to decimal Equivalent by multiplying Each Octal digit by its positional weightage.

I Convert Octal to its decimal Equivalent number.

1. $(6327.4051)_8$

$= 6\times8^3 + 3\times8^2 + 2\times8^1 + 7\times8^0 + 4\times8^{-1} + 0\times8^{-2} + 5\times8^{-3} + 1\times8^{-4}$

$= 3072 + 192 + 16 + 7 + \frac{4}{8} + 0 + \frac{5}{512} + \frac{1}{4096}$

$= (3287.5100098)_{10}$

2. Decimal to Binary Conversion

① Convert $(734)_{10}$ into Binary

2	734
2	367 − 0
2	183 − 1
2	91 − 1
2	45 − 1
2	22 − 1
2	11 − 0
2	5 − 1
2	2 − 1
2	1 − 0
	0 − 1

$\therefore (734)_{10} = (1011011110)_2$

② Convert $(0.705)_{10}$ into binary

Solution:-

$0.705\times2 = 1.41$
$0.41\times2 = 0.82$
$0.82\times2 = 1.64$
$0.64\times2 = 1.28$
$0.28\times2 = 0.56$
$0.56\times2 = 1.12$
$0.12\times2 = 0.24$
$0.24\times2 = 0.48$
$0.48\times2 = 0.96$
$0.96\times2 = 1.92$

$\therefore (0.705)_{10} = (0.1011010001)_2$

③ Convert $(1593.875)_{10}$ into binary

ⓐ

2	1593
2	796 − 1
2	398 − 0
2	199 − 0
2	99 − 1
2	49 − 1
2	24 − 1
2	12 − 0
2	6 − 0
2	3 − 0
	1 − 1

$(1593)_{10} = (11000111001)_2$

ⓑ 0.87

$0.87\times2 = 1.75$
$0.75\times2 = 1.50$
$0.50\times2 = 1.00$

$\therefore (0.875)_{10} = (0.111)_2$

$\therefore (1593.875) = (11000111001.111)_2$

Dr CBM, Dept. of ECE, SVIT

(vi) Decimal to Octal Conversion.

The Conversion decimal to Octal is similar to that of decimal to binary Conversion. Again, the Integer and fractional parts of the decimal number are treated seperately, and then the results are combined to Obtain the desired octal number.

Integer part Conversion : Repeated division method

To Convert the given decimal Integer number to Octal, Successively divide the given number by 8 till the quotient is 0. The last remainder is the MSD. The remainders read from bottom to top give the Equivalent octal Integer number.

Fractional part Conversion: Repeated multiplication Method.

To Convert the given decimal fraction to Octal, Sucessively multiply the decimal fraction and subsequent decimal fractions by 8 till the product is 0 or till the required accuracy is obtained. The first ~~...~~ Integer from the ~~...~~ top is the MSD. Thus, the Integer read from top to bottom give the Equivalent Octal fraction.

(i) Perform the following $(2003)_{10} = (?)_8$

Soln

8	2003
8	250 - 3
8	31 - 2
8	3 - 7
	0 - 3

$\therefore [2003]_{10} = [3723]_8$

(ii) Convert $(0.705)_{10}$ into Octal

Solution

$0.705 \times 8 = 5.64$
$0.64 \times 8 = 5.12$
$0.12 \times 8 = 0.96$
$0.96 \times 8 = 7.68$
$0.68 \times 8 = 5.44$

$\therefore (0.705)_{10} = (0.55075\ldots)_8$

(iii) Convert $(11582.875)_{10}$ into Octal.

Solution:-

$$\begin{array}{r|l} 8 & 11582 \\ \hline 8 & 1447 - 6 \\ \hline 8 & 180 - 7 \\ \hline 8 & 22 - 4 \\ \hline 8 & 2 - 6 \\ \hline & 0 - 2 \end{array}$$

Remainders

$\therefore (11582)_{10} = (26476)_8$

$0.875 \times 8 = 7.00$ $\quad \therefore (0.875)_{10} = (0.7)_8$

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

(iv) Convert 658.825 decimal into Octal

$$\begin{array}{r|l} 8 & 658 \\ \hline 8 & 82 - 2 \\ \hline 8 & 10 - 2 \\ \hline & 1 - 2 \end{array}$$

$.825 \times 8 = 6.6 \longrightarrow 6$

$0.6 \times 8 = 4.8 \longrightarrow 4$

$0.8 \times 8 = 6.4 \longrightarrow 6$

$\therefore (0.825)_{10} = (0.646)_8$

$\therefore (658)_{10} = (1222)_8$

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

(v) Convert $(378.93)_{10}$ to Octal.

ⓐ 378 : Sucessive division

$$\begin{array}{r|l} 8 & 378 \\ \hline 8 & 47 - 2 \\ \hline 8 & 5 - 7 \\ \hline & 0 - 5 \end{array}$$

$= (572)_8$

ⓑ $(0.93)_{10}$ to Octal

$0.93 \times 8 = 7.44$

$0.44 \times 8 = 3.52$

$0.53 \times 8 = 4.16$

$0.16 \times 8 = 1.28$

$= (0.7341)_8$

$\therefore (378.93)_{10} = (572.7341)_8$

(vi) Decimal to Hexadecimal

In the Conversion decimal, the Integer and fractional part of the decimal numbers are treated seperately, and then the results are combined to Obtain the desired hexadecimal number.

Integer part : Repeated division Method

To Convert the given decimal Integer number to hexadecimal, Succesively divide the given number by 16, till the quotient is 0. The last remainder is the MSD.
The remainders read from bottom to top give the Equivalent hexadecimal Integer number.

Fractional Part Conversion: Repeat Multiplication Method.

To Convert the given decimal fraction to hexadecimal, Succesively multiply the decimal fraction and Subsequent decimal fractions by 16 till the product is 0 or till the required accuracy is Obtained. The first Integer from the top is MSD. Thus, the Integers read from top to bottom give the Equivalent hexadecimal fraction.

Convert $(675.625)_{10}$ into hexadecimal.

Soln (a) for Integer part

$$\begin{array}{r|l} 16 & 675 \\ 16 & 42 - 3 \\ & 2 - 10 \rightarrow A \end{array} \uparrow$$

$\therefore (675)_{10} = (2A3)_{16}$

(b) for fractional part

$0.625 \times 16 = 10 = A$

$(0.625)_{10} = (0.A)_{16}$

$\therefore (675.625)_{10} = (2A3.A)_{16}$

Dr. CBM, Associate Prof, SVIT

2. $(48350)_{10} = (?)_{16}$

Soln:- (a)

16	48350	
16	3021 – 14 – E	LSD
16	188 – 13 – D	
16	11 – 12 – C	
	0 – 11 – B	MSD

(b)

$\therefore (48350)_{10} = (BCDE)_{16}$

3. $(532.65)_{10} = (?)_{16}$

Soln:- (a)

16	532
16	33 – 4
	2 – 1

$\therefore (532)_{10} = (214)_{16}$

(b)

$0.65 \times 16 =$	10.4	A	MSD
$0.4 \times 16 =$	6.4	6	LSD

$(0.65)_{10} = (0.A6)_{16}$

$\therefore (532.65)_{10} = (214.A6)_{16}$

4. Convert $(8899)_{10}$ into hexadecimal.

Solution:-

16	8899
16	556 – 3
16	34 – C
	2 – 2

$\therefore (8899)_{10} = (22C3)_{16}$

5. Perform $(894867)_{10} = (?)_{16}$

16	894867
16	55929 – 3
16	3495 – 9
16	218 – 7
16	13 – A
	0 – 13 – D

$\therefore (894867)_{10} = (DA793)_{16}$

6. Convert $(7084.95)_{10}$ into hexadecimal

Solution: (a)

$$\begin{array}{r|l} 16 & 7084 \\ 16 & 442 - C \\ 16 & 27 - A \\ 16 & 1 - B \\ & 0 - 1 \end{array}$$

(b) $0.95 \times 16 = 15.2 = F.2$

$0.2 \times 16 = 3.2 = 3.2$

$0.2 \times 16 = 3.2 = 3.2$

$(0.95)_{10} = (0.F33)_{16}$

$\therefore (7084)_{10} = (1BAC)_{16}$

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

7. Convert $(0.368)_{10}$ into hexadecimal

Solution: $0.368 \times 16 = 5.888 = 5.888$

$0.888 \times 16 = 14.208 = E.208$

$0.208 \times 16 = 3.328 = 3.328$

$0.328 \times 16 = 5.248 = 5.248$

$0.248 \times 16 = 3.968 = 3.968$

$\therefore (0.368)_{10} = (0.5E353...)_{16}$

8. Convert $(4477.85)_{10}$ into hexadecimal

Solution:

$$\begin{array}{r|l} 16 & 4477 \\ 16 & 279 - D \\ 16 & 17 - 7 \\ 16 & 1 - 1 \\ & 0 - 1 \end{array}$$

$(4477)_{10} = (117D)_{16}$

$0.85 \times 16 = 13.6 = D.6$

$0.6 \times 16 = 9.6 = 9.6$

$0.6 \times 16 = 9.6 = 9.6$

$(0.85)_{10} = (0.D99)_{16}$

$\therefore (4477.85)_{10} = (117D.D99)_{16}$

Dr. CBm, Associate Professor. Dept of ECE

(VIII) Hexadecimal to decimal Conversion

A hexadecimal number can be converted to binary Equivalent by multiplying Each hexadecimal digit by its positional weightage.

Eg1: ① Convert $(5C7)_{16}$ into decimal

$(5C7)_{16} = 5\times16^2 + C\times16^1 + 7\times16^0$

$= 5\times16^2 + 12\times16^1 + 7\times1$

$= 1280 + 159 + 7 = (1479)_{10}$

② Convert $(AC8)_{16}$ into decimal.

$AC8 = A\times16^2 + C\times16^1 + 8\times16^0$

A-10 $= 10\times16^2 + 12\times16^1 + 8\times1$

C-12

$= 2560 + 192 + 8$

$= (2760)_{10}$

(ix) Octal to Hexadecimal Conversion.

To Convert an Octal number to hexadecimal, first Convert the given Octal number to Binary and then binary number to hexadecimal.

1. Convert $(3576)_8$ to hexadecimal.

3	5	7	6	octal
011	101	111	110	Binary

To Convert Binary to hexadecimal

$$\underbrace{0111}_{7}\ \underbrace{0111}_{7}\ \underbrace{1110}_{E} = (77E)_{16}$$

2. $(273)_8$ to hexadecimal

2	7	3

010 · 111011 ← Binary

To Convert Binary to hexadecimal

$$0\ \underbrace{1011}\ \underbrace{1011} = (0BB)_{16}$$

Dr. CBm, Associate Professor,
Dept of ECE, SVIT

(x) Hexadecimal to Octal Conversion

To Convert a hexadecimal number to Octal, the Easiest way is to first Convert the given hexadecimal number to binary and then binary number to Octal.

1. Convert $(3576)_{16}$ to Octal.

3	5	7	6	Hexadecimal
0011	0101	0111	0110	Binary

$\underbrace{001}\underbrace{101}\underbrace{010}\underbrace{111}\underbrace{0110}$ — written as 0 0 1 1 0 1 0 1 0 1 1 1 0 1 1 0

$= (1\ 2\ 5\ 6\ 6)_8$

2. $(ABCD)_{16} = (?)_8$

A	B	C	D	
1010	1011	1100	1101	→ Binary

00 1010101111001101 ⇒ $(125715)_8$

Complements

Complements are used in digital Computer for Simplifying the subtraction Operation and for logical manipulations.

There are two types of Complements for Each base-r system:-

(i) r's Complement

(ii) (r-1)'s Complement

* When the values of base is Substituted, the two types receive the names 2's and 1's Complement for binary numbers, or 10's and 9's Complement for decimal numbers.

i.e., Binary Number System → 1's Compliment
→ 2's Compliment

Decimal Number System → 9's compliment
→ 10's Compliment

Note :. Binary Addition.

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 10 (0 → carry 1)

$$\begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array} \rightarrow 2$$

1's Complement Representation

The 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 Representation

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

2's complement = 1's complement + 1

Problems

Find 1's Complement of the following

(a) $(11100)_2 \Rightarrow$ 1's complement is
00011 → [Convert 0's → 1's & 1's → 0's]

(b) $(11010100)_2 \Rightarrow$ 00101011

(c) $(15)_{10} = (1111)_2$
1's Complement of 1111 is 0000

$$\begin{array}{r|l} 2 & 15 \\ 2 & 7-1 \\ 2 & 3-1 \\ & 1-1 \end{array}$$

(d) $(11100)_2 \Rightarrow$ 1's complement is 00011

(e) $(28)_{10} \Rightarrow$ Step 1:- Convert to Binary

$$\begin{array}{r|l} 2 & 28 \\ 2 & 14-0 \\ 2 & 7-0 \\ 2 & 3-1 \\ & 1-1 \end{array}$$

$(28)_{10} \Rightarrow (11100)_2$
1's complement of $(11100)_2$ is
$(00011)_2$

(a) Find 2's Complement of $(11000100)_2$

11000100 Dr. CBM, Dept of ECE, SVIT

1's complement is 00111011 $\qquad 1+1=10$

Add +1 $\qquad$ +1

00111100 → 2's Complement

(b) Find 2's complement of

(i) $(101011)_2$

1st Complement of $(101011)_2$ is 010100

add 1 → +1

010101 → 2's complement

(ii) $(111001)_2$

1's complement + 1 => 2's complement.

⇒ 000110

+1

$(000111)_2$ ← 2's Complement

(iii) $(101011)_2$

2's complement = 1's complement + 1

= 010100 + 1

= $(010101)_2$

010100
+1
010101

(iv) $(0111000)_2$

1's complement is 1000111

+1

$(1001000)_2$ → 2's complement

Binary subtraction using 1's Complement Method

To perform A − B, following steps are followed:

1. Take 1's Complement of B (subtrahend)
2. Result ← A (minuend) + 1's complement of B
3. If carry is generated then result is positive, Add carry to the result to get final result.
4. If carry is not generated result is negative and in the 1's complement form.

— Dr. CBM, Dept of ECE, SVIT

Problems:-

(i) Perform $(5)_{10} - (3)_{10}$

$$\begin{array}{r} 5 \\ -3 \\ \hline 2 \end{array}$$

In Binary $(0101)_2 - (0011)_2$

Step 1:- 1's complement of 0011 → 1100

Step 2:- Add 0101 + 1100

$$\begin{array}{r} 0101 \\ +\ 1100 \\ \hline \boxed{1}\,0001 \end{array}$$

Carry generate

Add to get final result

$$\begin{array}{r} 0001 \\ +\ 1 \\ \hline 0010 \end{array}$$

∴ Ans = $(0010)_2$

⇓

Binary Equivalent of $(2)_{10}$

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

$(28)_{10}$ = 11100

$(19)_{10}$ = 10011

2	28	
2	14	0
2	7	0
2	3	1
	1	1

2	19	
2	9	1
2	4	1
2	2	0
	1	0

1's complement of $(19)_{10}$ = i.e., 10011 ⇓ is 01100

add (11100) to 01100

$$\begin{array}{r} 11100 \\ +\ 01100 \\ \hline \boxed{1}\,01000 \end{array}$$

Carry generated

$$\begin{array}{r} 01000 \\ +\ 1 \\ \hline 01001 \end{array} = (9)_{10}$$

(iii) Perform $(15)_{10} - (28)_{10}$ using 1's complement

$(15)_{10} = (01111)_2$

$(28)_{10} = (11100)_2$

$$\begin{array}{r} (15)_{10} \\ -(28)_{10} \\ \hline -(13)_{10} \end{array}$$

1's complement of $(28)_{10}$ = 00011

add $(15)_{10}$ = 01111

$$\begin{array}{r} 00011 \\ 01111 \\ \hline 10010 \\ \hline \end{array}$$

no carry generated

Again 1's complement of result $\underline{\underline{01101}} \Rightarrow -(13)_{10}$

(iv) M = 1010100, N = 1000100 → Perform M−N using 1's complement.

1's complement of N = 0111011

$$\begin{array}{r} 0111011 \\ + \; M = 1010100 \\ \hline \boxed{1}\; 0001111 \\ +1 \\ \hline 0010000 \end{array}$$

Carry

∴ answer = 10000

(v) M = 1000100

N = 1010100, Perform M−N using 1's complement

Soln:- 1's complement of N = 0101011

$$\begin{array}{r} 0101011 \\ + \; M = 1000100 \\ \hline 1101111 \end{array}$$

no carry

∴ answer = − (1's complement of 1101111)

= − 0010000

= $\underline{\underline{-10000}}$

Dr. CBM, Dept of ECE, SVIT.

Binary Subtraction using 2's Complement.

To perform subtraction of 2 binary numbers in 2's Complement Method. for A − B, follow the steps.

Step 1:- Take 2's Complement of B

Step 2:- Result ← A + 2's complement of B

Step 3:- If carry is generated then result is positive, In this case, carry is Ignored.

Step 4:- If carry is not generated then the result is negative and in the 2's Complement form.

Problem:-

1) Perform $(5)_{10} - (3)_{10}$ Dr. CBm, Dept of ECE, SVIT.

$$\downarrow$$

$$(0101)_2 - (0011)_2$$

2's complement of 0011 is

$$\begin{array}{r} 1100 \\ +1 \\ \hline 1101 \end{array}$$

Add, 1101 + 0101

$$\Rightarrow \begin{array}{r} 1101 \\ +\,0101 \\ \hline \boxed{1}\,0010 \end{array}$$

Carry generated → Ignore

∴ Answer $= (0010)_2 = \underline{\underline{(2)_{10}}}$

2) Perform $(3)_{10} - (5)_{10}$

$$(0011)_2 - (0101)_2$$

2's complement of $(5)_{10} = 0101$ is

$$\begin{array}{r} 1010 \\ +1 \\ \hline (1011)_2 \end{array}$$

Add, 0011 + 1011 ⇒

$$\begin{array}{r} 0011 \\ +\,1011 \\ \hline 1110 \end{array}$$

no carry (−ve sign)

$$\begin{array}{r} 0001 \\ +1 \\ \hline 10 \end{array}$$

Take again 2's complement of 1110 ⇒ $0001 + 1 = -(0010)_2$

$= -(2)_{10}$

③ Given M = 1010100, N = 1000100, Perform M − N using 2's complement.

Soln:- 2's complement of N = 0111011
(1000100)

$$\begin{array}{r} 0111011 \\ +\ \ \ \ \ \ 1 \\ \hline 0111100 \end{array}$$

$$\begin{array}{rr} M \rightarrow & 1010100 \\ + & 0111100 \\ \hline \boxed{1} & 0010000 \end{array}$$

discard carry

∴ answer = 10000

④ M = 1000100
N = 1010100

Perform M − N using 2's complement.

Soln:- 2's complement of N ⇒ 0101011

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

$$\begin{array}{rr} M & 1000100 \\ + & 0101100 \\ \hline & 1110000 \end{array}$$

no carry

∴ again perform 2's complement of 1110000 to get final result.

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

∴ answer = −10000

(−ve sign because no carry).

5. Subtract [4 − 9]

4 - 0100

9 - 1001

2's complement of 9 → 0110 + 1

⇒ 0111

$$\begin{array}{r} 0100 \\ +\ 0111 \\ \hline 1011 \end{array}$$

No carry, ∴ result is in 2's complement form ⇒ Negative sign

$$\begin{array}{r} 0100 \\ +\ 1 \\ \hline 0101 \end{array}$$

$\therefore (0100)_2 - (1001)_2 = -(0101)_2 = -(5)_{10}$

6. Perform Subtraction using 2's complement method.

1101 − 1010

1010 ← 2nd Number

$$\begin{array}{r} 0101 \\ +\ 1 \\ \hline 0110 \end{array}$$

$$\begin{array}{r} 1101 \\ +\ 0110 \\ \hline \boxed{1}\ 0011 \end{array}$$

1101 ← 2's complement of 2nd number

0110 ← 1st number

→ carry generated → Ignore

∴ answer = $(0011)_2$

Decimal Number System $\rightarrow$ 9's Complement
$\rightarrow$ 10's Complement.

To find 10's Complement,

General representation is $\boxed{\begin{array}{ll} r^n - N, & \text{for } N \neq 0 \\ 0, & \text{for } N = 0 \end{array}}$

$r \rightarrow$ base

$n \rightarrow$ no of digits in Integer part.

$N \rightarrow$ Given number.

To find 9's Complement,

Dr. CBM, Associate Professor
Dept. of ECE, SVIT.

$\boxed{r^n - r^{-m} - N}$

where $r^n \rightarrow r \rightarrow$ base
$\rightarrow n \rightarrow$ no of digits in Integer part

In $r^{-m} \rightarrow +m \rightarrow$ is no of digits in fraction part

$N \rightarrow$ Given Number.

Problem:-

(a) Find 10's of following $(52520)_{10}$

WKT $r^n - N$ $\qquad n \rightarrow$ no of digits in number = 5

$\Rightarrow 10^5 - 52520$

$\Rightarrow 100000 - 52520$

$\Rightarrow \underline{\underline{47480}}$

(b) Find 10's of the following

(i) $(0.3267)_{10}$

$\rightarrow$ No Integer part

formula is $r^n - N$

$n = 0 \rightarrow$ no numbers in Integer part $\qquad [10^0 = 1]$

$(10^0 - 0.3267) = (1 - 0.3267) = \underline{0.6733}$

Dr Chaya. B M, Associate Professor ECE, SVIT

(ii) $(25.639)_{10}$

Note that There are 2 digits in Integer part

$\therefore n = 2$

$r^n - N$

$= 10^2 - 25.639$

$\Rightarrow 100 - 25.639$

$= \underline{\underline{74.361}}$

(iii) 13579

$r^n - N$

$n = 5$, $\therefore 10^5 - 13579$

$= 100000 - 13579$

$\Rightarrow 86421$

(iv) 900900

$r^n - N$

$n = 6$ $\quad \therefore = 10^6 - 900900$

$N = 900900$ $\quad = 1000000 - 900900$

$= \underline{\underline{099100}}$

(v) 0000

$r^n - N$

$n = 4 \quad = 10^4 - 0000$

0000

The Complement of 0000 is 0000

(vii) 0.1035

$r^n - N$

$n = 0$

$10^0 - 0.1035$

$1 - 0.1035 = \underline{\underline{9.8965}}$

(vi) 8374.59

$r^n - N \quad n = 4$

$10^4 - 8374.59$

1625.41

(viii) 17850.6584

$r^n - N$, $n = 5$

$10^5 - N$

$= 100000 - 17850.6584$

$= \underline{\underline{82149.3416}}$

Find the 9's Complement of the following decimal numbers.

① 52590

formula $r^n - r^{-m} - N$

$n \to$ Integer part (number of digits)

$m \to$ no of digits in fractional part

$n=5 \quad m=0$

$\therefore r^5 - 10^{-0} - 52590$

$10^5 - 1 - 52590 \Rightarrow 47409$

② $(0.3267)_{10}$

$n=0 \quad m=4$

$r^n - r^{-m} - N$

$= 10^0 - 10^{-4} - 0.3267$

$= 1 - 0.0001 - 0.3267$

$\Rightarrow \underline{\underline{0.6732}}$

③ $(25.639)_{10}$

$n=2, \quad m=3$

$r^{+n} - r^{-m} - N$

$= 10^2 - 10^{-3} - 25.639$

$= 74.36$

④ 13579

$n=5, \quad m=0$

$r^n - r^{-m} - N$

$10^5 - 10^0 - 13579$

$= 86420$

⑤ 8374.59

$r^n - r^{-m} - N$

$n=4$

$m=2$

$10^4 - 10^{-2} - 8374.59$

$= \underline{\underline{1625.4}}$

⑥ 0.1035

$r^n - r^{-m} - N$

$n=0 \quad m=4$

$\Rightarrow 10^0 - 10^{-4} - 0.1035$

$\Rightarrow 1 - 0.0001 - 0.1035$

$\Rightarrow \underline{\underline{0.8964}}$

⑦ 17850.6584

$r^n - r^{-m} - N$

$n=5 \quad m=4$

$10^5 - 10^{-4} - 17850.6584$

$\Rightarrow \underline{\underline{82149.3415}}$

Binary Subtraction Using 10's Complement.

Perfom the following using 10's Complement.

ⓐ $(487)_{10} - (354)_{10}$

Minuend - $(487)_{10}$

Subtrahend - $(354)_{10}$

Step 1:- Find 10's complement of Subtrahend,

$\therefore r^n - N$

$n = 3 \quad = 10^3 - 354$

$= 646$

Step 2 → add Minuend and Subtrahend

$$\begin{array}{r} 487 \\ + \ 646 \\ \hline \boxed{1}\,133 \end{array}$$

Carry generated → Discard

$\therefore$ Answer $= (133)_{10}$

ⓑ $(8437)_{10} - (39)_{10}$

M - $(8437)_{10}$

S - $(0039)_{10}$

$r^n - N \Rightarrow$ 10's complement of subtrahend.

$10^4 - 0039 \Rightarrow (9961)_{10}$

$$\begin{array}{r} 8431 \\ + \ 9961 \\ \hline \boxed{1}\,8398 \end{array}$$

→ discard carry

$\therefore$ Ans $= (8398)_{10}$

③ $(5)_{10} - (3)_{10}$

10's complement of 3, $r^n - N$

$\Rightarrow 10^1 - 3$

$= \underline{\underline{7}}$

$$\begin{array}{r} 5 \\ +\ 7 \\ \hline \boxed{1}\ 2 \end{array}$$

↳ discard carry

$\therefore$ Answer is $\underline{\underline{(2)_{10}}}$

④ $(309)_{10} - (1447)_{10}$

$M \rightarrow (309)_{10}$

$S \rightarrow (1447)_{10}$

10's complement of Subtrahend is $r^n - N$

$10^4 - 1447$

$10000 - 1447 = (8553)_{10}$

$$\begin{array}{r} 309 \\ +\ 8553 \\ \hline 8862 \end{array}$$

↳ No carry, So again take 10's complement of $(8862)_{10}$

{result is negative}

$r^n - N$

$10^4 - 8862$

$10000 - 8862 \Rightarrow 1138$

Answer $= \underline{\underline{-(1138)_{10}}}$

⑤ Subtract 72532 − 03250 using 10's Complement

M − 72532

S − 03250 ⟶ 10's complement is $r^n - N \Rightarrow 10^5 - 03250$

$= \underline{\underline{96750}}$

$$\begin{array}{r} 72532 \\ +\ 96750 \\ \hline \boxed{1}\ 69282 \end{array}$$

↳ Carry Exists − Discard it

$\therefore$ Answer $= \underline{\underline{(69282)_{10}}}$

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

M → 03250

N → 72532 → 10's complement is $10^5 - 72532$

$= \underline{27468}$

Step 2:- Add M + 27468

$\therefore$ 3250

27468

30718

↳ no carry → so again take 10's complement of $(30718)_{10}$

↓

result is negative

$r^n - N$

$= 10^5 - 30718$

$= 69282$

$\therefore$ Answer = $-(69282)_{10}$

Binary Subtraction using 9's Complement

1. Perform 9's Complement for (525.20)

WKT, $r^n - r^{-m} - N$

$n \to 3 \quad m = 2$

$10^3 - 10^{-2} - 525.20$

$1000 - 0.01 - 525.20 \Rightarrow \underline{474.79}$

2. Subtract $(487)_{10} - (354)_{10}$ using 9's complement.

```
  4 8 7
- 3 5 4
```

Step 1:- 9's complement of $(354)_{10}$ is $r^m - r^{-m} - N$

$n = 3 \quad m = 0$

$\therefore 10^3 - 10^{-0} - 354$

$1000 - 1 - 354 = 645$

```
   4 8 7
 + 6 4 5
 -------
 [1] 1 3 2
  └─> + 1
 -------
 (1 3 3)10
```

carry generated → add it to the obtained result → To get FINAL result

3. $(8437)_{10} - (39)_{10}$

```
  8 4 3 7
- 0 0 3 9  <— 9'scomplement
```

$r^n - r^m - N$

$10^4 - 10^0 - 0039$

$10000 - 1 - 0039 \Rightarrow 9960$

$$\begin{array}{r} 8\,4\,3\,7 \\ +\,9\,9\,6\,0 \\ \hline 1\;8\,3\,9\,7 \end{array}$$

carry generated

$$\therefore \begin{array}{r} 8397 \\ +\quad 1 \\ \hline (8398)_{10} \end{array}$$

← add carry to Existing result

← Final result

4. $(5)_{10} - (3)_{10}$

9's complement of 3 → $r^n - r^{-m} - N$

$10^1 - 10^0 - 3$

$10 - 1 - 3 = 6$

$$\begin{array}{r} 5 \\ +\,6 \\ \hline \boxed{1}\,1 \end{array}$$

carry

$$\begin{array}{r} 1 \\ +\,1 \\ \hline (2)_{10} \end{array}$$

$\therefore$ Answer is $(2)_{10}$

5. $(309)_{10} - (1447)_{10}$

M = 309

N = 1447

9's complement of 1447

$r^n - r^{-m} - N$

$10^4 - 10^0 - 1447$

8552

$$\begin{array}{r} 8552 \\ +\ 309 \\ \hline 8861 \end{array}$$

→ no carry generated → negative result

also, Perform 9's complement of the Existing result to get final result.

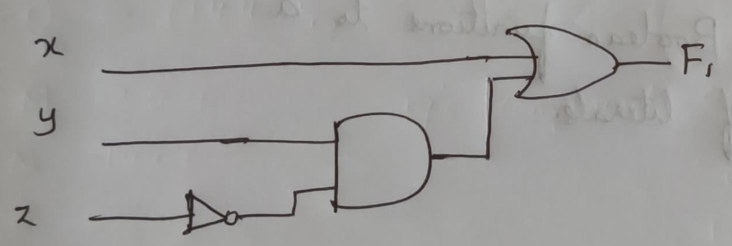
$8861 \Rightarrow 10^4 - 1 - 8861$

$= -(1138)_{10}$

Realize using Truth table and Implement with logic gates

1. $F_1 = x + yz'$

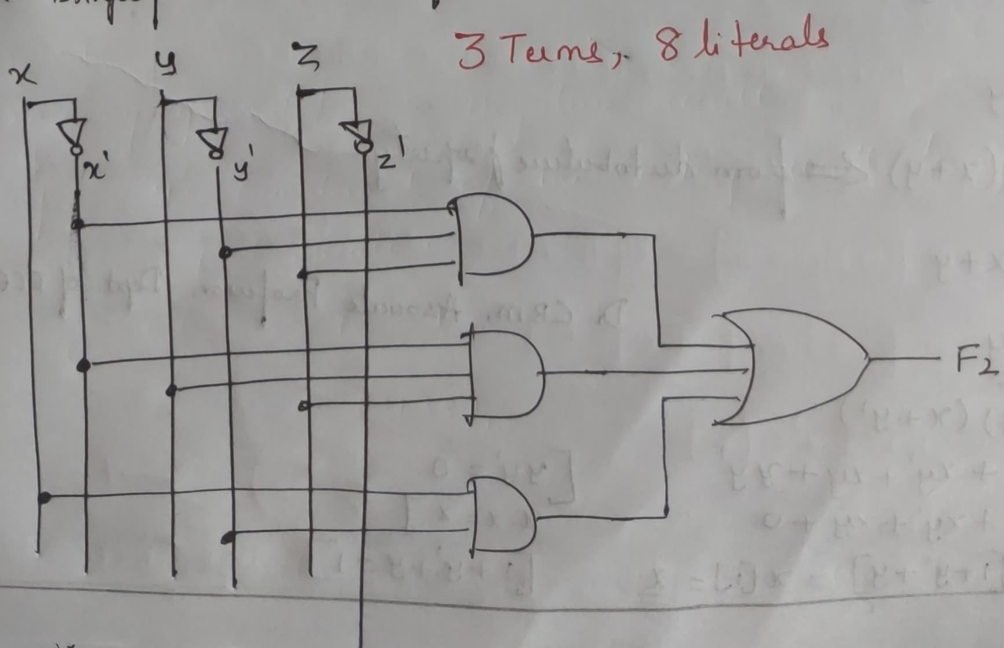
x	y	z	z'	yz'	x+yz'
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	1	1
1	1	1	0	0	1



2. $F_2 = x'y'z + x'yz + xy'$

Before Simplification, No of Gates used is more

3 Terms, 8 literals



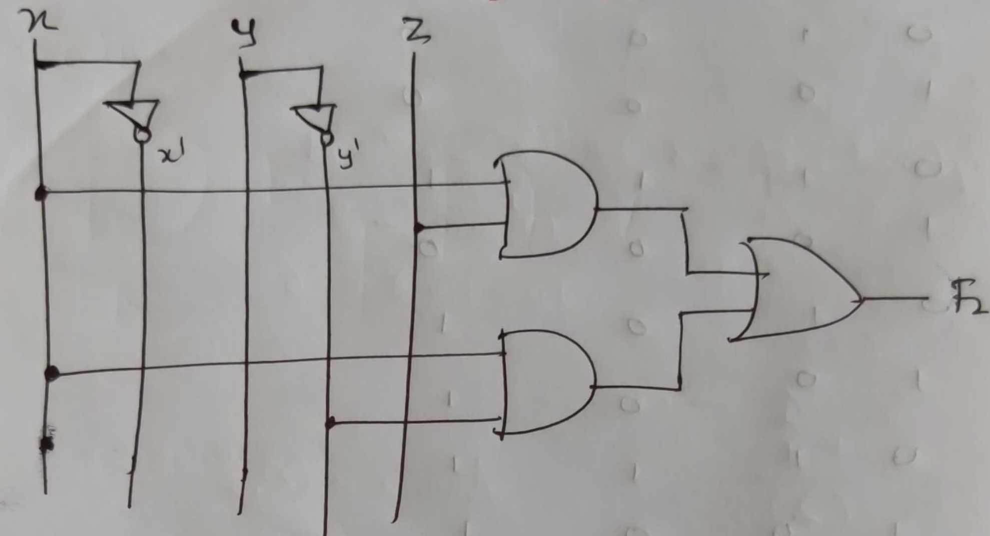
$F_2 = x'y'z + x'yz + xy'$ [After simplification, The Number of gates used is less)

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

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

$F_2 = x'z + xy'$ $\Rightarrow$ 2 Terms, 4 literals

[$x'z, xy'$] [x', z, x, y']



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

(a) $x(x'+y)$

$xx' + xy$

$0 + xy = xy$

(b) $x + x'y$

$(x+x')(x+y)$ $\Leftarrow$ from distributive property

$1 \cdot x+y$

$= \underline{\underline{x+y}}$

Dr. CBM, Associate Professor, Dept. of ECE, SVIT

(c) $(x+y)(x+y')$

$x.x + xy' + xy + yy'$

$x + xy' + xy + 0$

$x[1+y'+y] = x[1] = \underline{\underline{x}}$

[$yy' = 0$

$x.x = x$]

[$1+y'+y = 1$]

(d) $xy + x'z + yz \, (1)$

$xy + x'z + yz(x + x')$ $\qquad 1 \rightarrow x + x'$

$xy + x'z + xyz + x'yz$

$\underbrace{xy + xyz} + \underbrace{x'z + x'yz}$ $\qquad \left[\because \begin{array}{l} 1 + z = 1 \\ 1 + y = 1 \end{array}\right]$

$xy[1+z] + x'z[1+y]$

$\underline{\underline{xy + x'z}}$

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

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

$(\bar{B}\,\underbrace{\bar{A}\bar{A}}_{0} + \bar{A}\bar{B}\bar{C})\ \bar{B}+C$

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

$\bar{A}\bar{B}\bar{C}\bar{B} + \bar{A}\bar{B}\,\underbrace{\bar{C}C}_{0}$

$\underline{\underline{\bar{A}\bar{B}\bar{C}}}$

(f) $Y = A[\overline{ABC} + A\bar{B}C]$ $\qquad \overline{AB} = \bar{A} + \bar{B} \rightarrow$ Demorgan's law

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

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

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

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

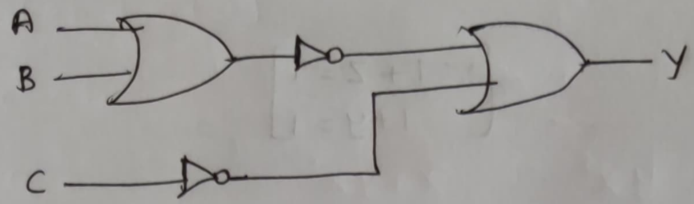
$= 0 + A\bar{C} + A\bar{B}$

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

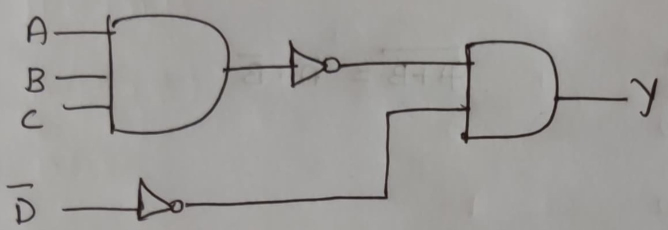
Dr. Chaya BM, Associate Professor, Dept. of ECE, SVIT

Draw the logic circuit for, $Y = A+B+C$ using basic gates

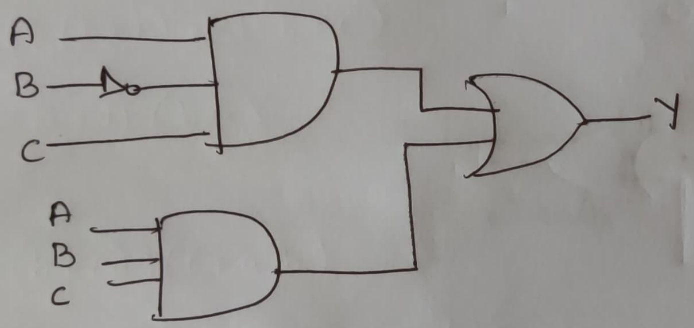
1. $Y = \overline{A+B} + \overline{C}$



2. $Y = (\overline{ABC})\overline{D}$



3. $Y = A\overline{B}C + ABC$



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

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

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

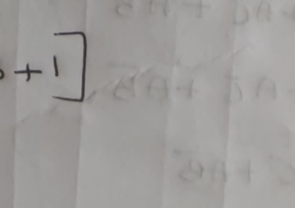
$= \overline{A}B + \overline{A}C + AB + B + C + AC + BC + C$

$= B[\overline{A} + A + 1 + C] + C[\overline{A} + 1 + A + B + 1]$

$= B[1+1+C] + C[1]$

$= B[1] + C[1]$

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



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

1. $xyz + x'y + xyz'$

$xy[z + z'] + x'y$

$xy + x'y$

$y\underbrace{[x + x']}_{(1)}$

$\underline{\underline{y}}$

2. $xz + x'zy$

$z(x + x'y) \longrightarrow$ Distributive law

$z[x + x'][x + y]$

$z[1][x + y]$

$\underline{\underline{z[x + y]}}$

3. $y(wz' + wz) + xy$

$= yw(z' + z) + xy$

$= yw(1) + xy$

$= wy + xy$

$= \underline{\underline{y(w + x)}}$

4. $a'b + a'bc' + a'bcd + a'bc'd'e$

$a'b[1 + c' + cd + c'd'e]$

$\underline{\underline{a'b}}$

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

$wx.x + wx.w + wy'x + wwy' + wxx' + wxy'$

$wx + wx + wy'x + wy' + w(0) + wxy'$

$wx[1 + 1 + y' + y'] + wy'$

$wx(1) + wy'$

$\underline{\underline{w(x + y')}}$

6. $a'b + a'bc' + a'bcd + a'bc'd'e$

$a'b[1 + c' + cd + c'd'e]$

$\underline{\underline{a'b}}$

Dr. CBM, Associate Professor, Dept. of ECE, SVIT

7. $abc\,[ab + c'(bc + ac)]$

$abc\,[ab + bcc' + acc']$

$abc\,[ab + 0 + 0]$

$abc \cdot ab$

$\underline{\underline{abc}}$

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

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

$y = A(1)$

$y = A$

9. $y = ab + ac + bd + cd$

$y = a(b+c) + d(b+c)$

$y = (a+d)(b+c)$

10. $y = (b+ca)(c+a'b)$

$= bc + b \cdot ba' + c \cdot c \cdot a + a'bca$

$= bc + a'b + ac + a' \cdot abc$

$= bc + a'b + ac + 0$

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

11. $y = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + a\bar{b}\bar{c}\bar{d} + a\bar{b}\bar{c}d$

$= \bar{b}\bar{c}\,[\bar{a}\bar{d} + \bar{a}d + a\bar{d} + ad]$

$= \bar{b}\bar{c}\,[\bar{a}[\bar{d}+d) + a[\bar{d}+d)]$

$= \bar{b}\bar{c}\,[\bar{a} \cdot 1 + a \cdot 1]$

$= \bar{b}\bar{c}\,[\bar{a} + a]$

$= \bar{b}\bar{c}\,[1]$

$= \bar{b}\bar{c}$

Dr. Chaya. B m, Associate Professor,
Dept. of ECE, SVIT

12. $ab + \overline{ac} + a\bar{b}c\,(ab + c)$

$ab + \overline{ac} + a\bar{b}cab + a\bar{b}cc$ WKT, $\overline{xy} = \bar{x} + \bar{y}$

$ab + \bar{a} + \bar{c} + a \cdot \bar{b}\, b \cdot c + a\bar{b}c$

$ab + \bar{a} + \bar{c} + a \cdot 0 \cdot c + a\bar{b}c$

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

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

$a(1)[b + c] + \bar{a} + \bar{c}$

$ab + ac + \bar{a} + \bar{c}$

$\bar{a} + ab + ac + \bar{c}$

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

$1(\bar{a} + b) + (a + \bar{c})(1)$

$\bar{a} + b + a + \bar{c}$

$1 + b + \bar{c} = \underline{\underline{1}}$

13. Show that, $A\bar{B}C + B + B\bar{D} + AB\bar{D} + \bar{A}C = B + C$

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

$A\bar{B}C + B[1 + \bar{D} + AB] + \bar{A}C$

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

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

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

$C[1]\,[\bar{A} + \bar{B}] + B$

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

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

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

$C[\underbrace{\bar{A} + 1}_{1}] + B$

$= B + C$

Dr. CBM, Dept of ECE, SVIT

14. $A + AC$

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

$= \underline{\underline{A}}$

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

$A[1 + BC + \bar{C}] + \bar{A}B$

$A[1] + \bar{A}B$

$A + \bar{A}B \longrightarrow$ Distributive law

$\Downarrow$

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

$(1)\,(A+B)$

$\underline{\underline{A+B}}$

16. $A + \bar{A}B + AB\bar{C}$

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

$A[1] + \bar{A}B$

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

$1 \cdot [A+B] = A+B$

17. $\bar{A}C + \overline{AC}$

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

$\bar{A}[C+1] + \bar{C}$

$\bar{A}[1] + \bar{C}$

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

18. $\overline{\overline{A+B} + \bar{C}}$

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

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

$\underline{\underline{(A+B)C}}$

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

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

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

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

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

$= \underline{1}$

20. $AB + \bar{A} + \overline{AB}$

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

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

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

$\underbrace{\bar{A}+\bar{A}} + \underbrace{B+\bar{B}}_{1}$

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

Sri Sai Vidya Vikas Shikshana Samithi ®
SAI VIDYA INSTITUTE OF TECHNOLOGY
Approved by AICTE, New Delhi, Affiliated to VTU, Recognized by Govt. of Karnataka.Accredited by NBA, New Delhi (CSE, ECE, ISE, MECH & CIVIL), NAAC – "A" Grade
DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
RAJANUKUNTE, BENGALURU – 560 064, KARNATAKA
Phone: 080-28468191/96/97/98 * E-mail: hodece@saividya.ac.in * URL www.saividya.ac.in

Course Name: Introduction to Electronics Engineering	Course Code: 22ESC143	AY:2022-2023
Course Coordinator: Dr. Chaya B M	Scheme: 2022	Batch: 2022

Boolean Algebra and Logic Gates

Axiomatic Definition of Boolean Algebra

- 1854: George Boole developed an algebraic system now called *Boolean algebra.*
- 1904: E. V. Huntington formulated a set of postulates that formally define the Boolean algebra
- 1938: C. E. Shannon introduced a two-valued Boolean algebra called switching algebra that represented the properties of bistable electrical switching circuits

- Two binary operators, + and •, (Huntington) postulates:
  1. (a) The structure is closed with respect to the operator +.
     (b) The structure is closed with respect to the operator •.
  2. (a) The element 0 is an identity element with respect to +; that is, $x + 0 = 0 + x = x$.
     (b) The element 1 is an identity element with respect to •; that is, $x \cdot 1 = 1 \cdot x = x$.
  3. (a) The structure is commutative with respect to +; that is, $x + y = y + x$.
     (b) The structure is commutative with respect to • ; that is, $x \cdot y = y \cdot x$.
  4. (a) The operator • is distributive over +; that is, $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$.
     (b) The operator + is distributive over •; that is, $x + (y \cdot z) = (x + y) \cdot (x + z)$.

5. For every element $x \in B$, there exists an element $x \in B$ (called the complement of x) such that (a) $x + x = 1$ and (b) $x \cdot x = 0$.
6. There exist at least two elements $x, y \in B$ such that $x \neq y$.

- Comparing Boolean algebra with arithmetic and ordinary algebra

1. Huntington postulates do not include the associative law. However, this law holds for Boolean algebra and can be derived (for both operators) from the other postulates.
2. The distributive law of + over • (i.e., $x + (y \cdot z) = (x + y) \cdot (x + z)$) is valid for Boolean algebra, but not for ordinary algebra.
3. Boolean algebra does not have additive or multiplicative inverses; therefore, there are no subtraction or division operations.
4. Postulate 5 defines an operator called the complement that is not available in ordinary algebra.
5. Ordinary algebra deals with the real numbers, which constitute an infinite set of elements. Boolean algebra deals with the as yet undefined set of elements, B, but in the two-valued Boolean algebra defined next (and of interest in our subsequent use of that algebra), B is defined as a set with only two elements, 0 and 1.

- Distributive laws:
  - $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
  - $x + (y \cdot z) = (x+y) \cdot (x+z)$

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

$y + z$	$x \cdot (y + z)$
0	0
1	0
1	0
1	0
0	0
1	1
1	1
1	1

$x \cdot y$	$x \cdot z$	$(x \cdot y) + (x \cdot z)$
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	1	1
1	0	1
1	1	1

y • z	x+(y • z)
0	0
0	0
0	0
1	1
0	1
0	1
0	1
1	1

x+y	x+z	(x+y) • (x+z)
0	0	0
0	1	0
1	0	0
1	1	1
1	1	1
1	1	1
1	1	1
1	1	1

Two Valued Boolean Algebra

- B = {0,1}
- The rules of operations

AND				OR				NOT	
x	y	$x \cdot y$		x	y	$x+y$		x	x'
0	0	0		0	0	0		0	1
0	1	0		0	1	1		1	0
1	0	0		1	0	1			
1	1	1		1	1	1			

- Closure: the result of each operation is either 1 or 0 and 1, 0 $\in B$.
- Identity elements: 0 for + and 1 for •
- The commutative laws are obvious from the symmetry of the binary operator tables.

- Complement
  - $x+x'=1$: $0+0'=0+1=1$; $1+1'=1+0=1$
  - $x \cdot x'=0$: $0 \cdot 0'=0 \cdot 1=0$; $1 \cdot 1'=1 \cdot 0=0$
- Has two distinct elements 1 and 0, with $0 \neq 1$
- We have just established a two-valued Boolean algebra:
  - a set of two elements
  - + : OR operation; • : AND operation
  - a complement operator: NOT operation
  - Binary logic is a two-valued Boolean algebra
  - also called "switching algebra" by engineers

Basic Theorems and Properties of Boolean Algebra

- Duality
  - the binary operators are interchanged; AND $\Leftrightarrow$ OR
  - the identity elements are interchanged; 1 $\Leftrightarrow$ 0

Table 2.1
Postulates and Theorems of Boolean Algebra

Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$
Theorem 3, involution		$(x')' = x$		
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$

Proof of Theorems:

- Theorem 1(a): $x+x=x$

$x+x = (x+x) \cdot 1$	by postulate: 2(b)
$= (x+x)(x+x')$	5(a)
$= x+xx'$	4(b)
$= x+0$	5(b)
$= x$	2(a)

- Theorem 1(b): $x \cdot x = x$

$x \cdot x = xx + 0$	by postulate: 2(a)
$= xx + xx'$	5(b)
$= x(x + x')$	4(a)
$= x \cdot 1$	5(a)
$= x$	2(b)

- Theorem 1(b) is the dual of theorem 1(a)

- Theorem 2(a): $x + 1 = 1$

$x + 1 = 1 \bullet (x + 1)$ by postulate: 2(b)

$= (x + x')(x + 1)$ 5(a)

$= x + x' \bullet 1$ 4(b)

$= x + x'$ 2(b)

$= 1$ 5(a)

- Theorem 2(b): $x \bullet 0 = 0$ by duality
- Theorem 3: $(x')' = x$
  - Postulate 5 defines the complement of x, $x + x' = 1$ and $x \bullet x' = 0$
  - The complement of x' is x is also $(x')'$

- Theorem 6(a): $x + xy = x$

$x + xy = x \bullet 1 + xy$ by postulate: 2(b)

$= x (1 + y)$ 4(a)

$= x \bullet 1$ 2(a)

$= x$ 2(b)

- Theorem 6(b): $x (x + y) = x$ by duality

- By means of truth table

x	y	xy	$x + xy$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

DeMorgan's Theorem

– $(x+y)' = x'y'$

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

– $(x\,y)' = x' + y'$

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

Operator Precedence:

- The operator precedence for evaluating Boolean expressions is
  1. parentheses
  2. NOT
  3. AND
  4. OR
- Examples
  - $x\,y' + z$
  - $(x\,y + z)'$

Boolean Functions:

- A Boolean function is an algebraic expression consists of
  - binary variables
  - binary operators OR and AND
  - unary operator NOT
  - parentheses
- A Boolean function expresses the logical relationship between binary variables and is evaluated by determining the binary value of the expression for all possible values of the variables.
- Examples
  - $F_1 = x + y z' \rightarrow$ $F_1 = 1$ if $x = 1$ or if $y = 0$ and $z = 1$, others $F_1 = 0$.
  - $F_2 = x'y'z + x'yz + xy' \rightarrow$
    $F_2 = 1$ if $(x = 0, y = 0, z = 1)$ or $(x = 0, y = 1, z = 1)$ or $(x = 1, y = 0)$, others $F_2 = 0$.

Truth Table

- Boolean function can be represented in a truth table.
- Truth table has 2^n rows where n is the number of variables in the function.
- The binary combinations for the truth table are obtained from the binary numbers by counting from 0 through $2^n - 1$.

Table 2.2
Truth Tables for F_1 and F_2

x	y	z	F_1	F_2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

Implementation of F_1 with logic gates

$F_1 = x + y z'$

Equivalent Logics

- Boolean function can be represented in truth table only in one way.
- In algebraic form, it can be expressed in a variety of ways, all of which have equivalent logic.
- Using Boolean algebra, it is possible to obtain a simpler expression for the same function with less number of gates and inputs to the gate.
- Designers work on reducing the complexity and number of gates to significantly reduce the circuit cost.

$$F_2 = x'y'z + x'yz + xy'$$
$$= x'z(y' + y) + xy'$$
$$= x'z + xy'$$

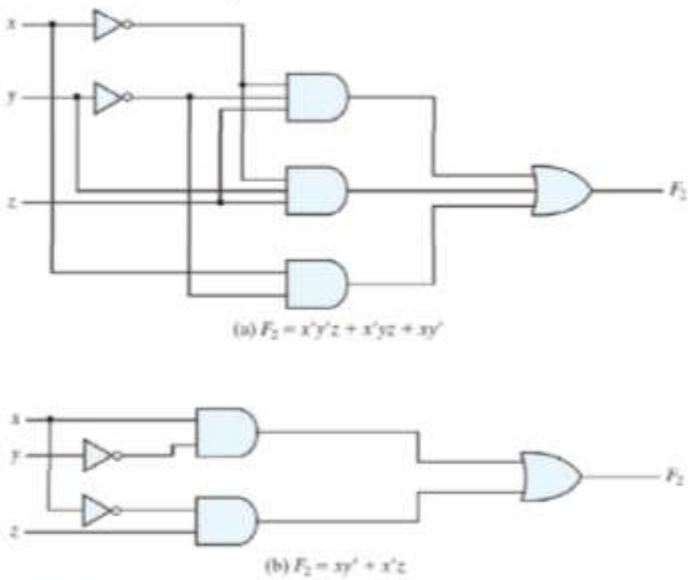


FIGURE 2.2
Implementation of Boolean function F_2 with gates

Algebraic Manipulation

- To minimize Boolean expressions
  - literal: a complemented or un-complemented variable (an input to a gate)
  - term: an implementation with a gate
  - The minimization of the number of literals and the number of terms => a circuit with less equipment

$$F_2 = x'y'z + x'yz + xy' \quad \rightarrow \text{3 terms, 8 literals}$$
$$= x'z(y' + y) + xy'$$
$$= x'z + xy' \quad \rightarrow \text{2 terms, 4 literals}$$

- Functions of up to five variables can be simplified by the map method described in the next chapter.
- For complex Boolean functions and many different outputs, designers of digital circuits use computer minimization programs that are capable of producing optimal circuits with millions of logic gates.

Minimization of Boolean Function

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

1. $x(x' + y) = xx' + xy = 0 + xy = xy.$
2. $x + x'y = (x + x')(x + y) = 1(x + y) = x + y.$
3. $(x + y)(x + y') = x + xy + xy' + yy' = x(1 + y + y') = x.$
4. $xy + x'z + yz = xy + x'z + yz(x + x')$
   $= xy + x'z + xyz + x'yz$
   $= xy(1 + z) + x'z(1 + y)$
   $= xy + x'z.$
5. $(x + y)(x' + z)(y + z) = (x + y)(x' + z)$, by duality from function 4.

Complement of a Function

- F' is obtained from an interchange of 0's for 1's and 1's for 0's in the value of F
- The complement of a function may be derived using DeMorgan's theorem.
- Three-variable DeMorgan's theorem:

$(A + B + C)' = (A + X)'$	let $B + C = X$
$= A'X'$	by DeMorgan's
$= A'(B + C)'$	$X = B + C$
$= A'(B'C')$	by DeMorgan's
$= A'B'C'$	associative

- Generalized form
  - $(A + B + C + ... + F)' = A'B'C' ... F'$
  - $(ABC ... F)' = A' + B' + C' + ... + F'$

Find the complement of the functions $F_1 = x'yz' + x'y'z$ and $F_2 = x(y'z' + yz)$. By applying DeMorgan's theorems as many times as necessary, the complements are obtained as follows:

$$F_1' = (x'yz' + x'y'z)' = (x'yz')'(x'y'z)' = (x + y' + z)(x + y + z')$$
$$F_2' = [x(y'z' + yz)]' = x' + (y'z' + yz)' = x' + (y'z')'(yz)'$$
$$= x' + (y + z)(y' + z')$$
$$= x' + yz' + y'z$$

Find the complement of the functions F_1 and F_2 of Example 2.2 by taking their duals and complementing each literal.

1. $F_1 = x'yz' + x'y'z.$
   The dual of F_1 is $(x' + y + z')(x' + y' + z).$
   Complement each literal: $(x + y' + z)(x + y + z') = F_1'.$
2. $F_2 = x(y'z' + yz).$
   The dual of F_2 is $x + (y' + z')(y + z).$
   Complement each literal: $x' + (y + z)(y' + z') = F_2'.$

Minterms and Maxterms

- A minterm (standard product): an AND term consists of all literals in their normal form or in their complement form
- For example, two binary variables x and y, has 4 minterms
  - $xy, xy', x'y, x'y'$
- n variables can be combined to form 2^n minterms (m_j, $j = 0 \sim 2^n-1$)
- A maxterm (standard sum): an OR term; 2^n maxterms (M_j, $j = 0 \sim 2^n-1$)
- Each maxterm is the complement of its corresponding minterm, and vice versa.

			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

Canonical Form: Sum of Minterms

- An Boolean function can be expressed by
  - a truth table
  - sum of minterms ➔ $f = \Sigma m_j$
  - product of maxterms ➔ $f = \Pi M_j$
  - $f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$
  - $f_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$

Table 2.4
Functions of Three Variables

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

Canonical Form: Product of Maxterms

- The complement of a Boolean function
  - the minterms that produce a 0
  - $f_1' = m_0 + m_2 + m_3 + m_5 + m_6 = x'y'z' + x'yz' + x'yz + xy'z + xyz'$
  - $f_1 = (f_1')' = (x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$
  - $= M_0 M_2 M_3 M_5 M_6$
  - $f_2 = (x + y + z)(x + y + z')(x + y' + z)(x' + y + z)$
    $= M_0 M_1 M_2 M_4$
- Canonical form: any Boolean function expressed as a sum of minterms or a product of maxterms

Minterm Expansion

- **EXAMPLE 2.4:** Express the Boolean function $F=A+B'C$ as a sum of minterms.
  - F = A + B'C = A (B + B') + B'C = AB + AB' + B'C
  - = AB(C + C') + AB'(C + C') + (A + A')B'C
  - = ABC + ABC' + AB'C + AB'C' + A'B'C
  - = A'B'C + AB'C' + AB'C + ABC' + ABC
  - $= m_1 + m_4 + m_5 + m_6 + m_7$
  - F(A,B,C) = Σ (1, 4, 5, 6, 7)
  - or, built the truth table first

Table 2.5
Truth Table for $F = A + B'C$

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

Maxterm Expansion

EXAMPLE 2.5: Express the Boolean function $F = xy + x'z$ as a product of maxterms.

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

- $x' + y = x' + y + zz' = (x' + y + z)(x' + y + z')$
- $x + z = x + z + yy' = (x + y + z)(x + y' + z)$
- $y + z = y + z + xx' = (x + y + z)(x' + y + z)$

- $F = (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z') = M_0 M_2 M_4 M_5$
- $F(x,y,z) = \Pi(0,2,4,5)$

- check this result with truth table

Canonical Form Conversion

- Conversion between Canonical Forms
  - $F(A,B,C) = \Sigma(1,4,5,6,7) \rightarrow F'(A,B,C) = \Sigma(0,2,3) = m_0 + m_1 + m_2$
  - By DeMorgan's theorem

    $F = (m_0 + m_1 + m_2)' = m'_0 \bullet m'_2 \bullet m'_3$

    $= M_0 M_2 M_3 = \Pi(0, 2, 3)$
  - $m_j' = M_j$
- sum of minterms ⇔ product of maxterms
  - interchange the symbols Σ and Π and list those numbers missing from the original form
    - Σ of 1's ⇔ Π of 0's

Conversion Example

- $F = xy + x'z$
- $F(x, y, z) = \Sigma(1, 3, 6, 7)$
- $F(x, y, z) = \Pi\ (0, 2, 4, 5)$

Table 2.6
Truth Table for $F = xy + x'z$

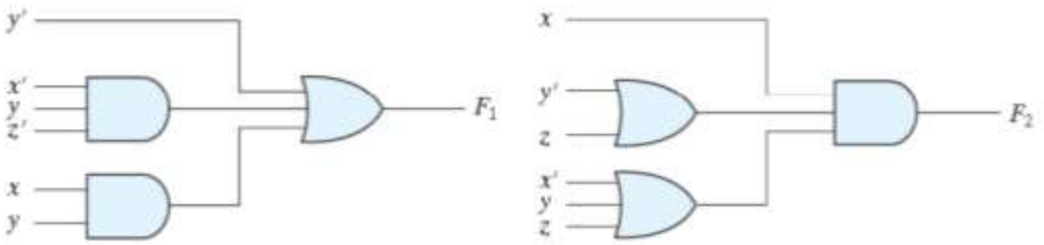
x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Minterms

Maxterms

Standard Forms

- Canonical forms are baseline expression and seldom used, they are not minimum
- Two standard forms are used usually
  - sum of products $F_1 = y' + xy + x'yz'$
  - product of sums $F_2 = x(y' + z)(x' + y + z')$

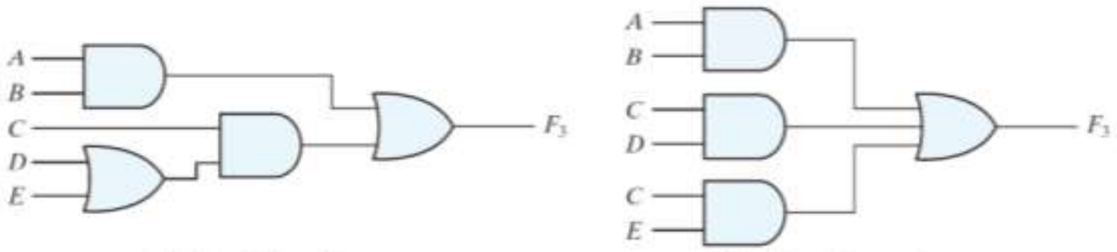


(a) Sum of Products (b) Product of Sums

- This circuit configuration is referred to as a *two-level* implementation.
- In general, a two-level implementation is preferred because it produces the least amount of delay through the gates when the signal propagates from the inputs to the output. However, the number of inputs to a given gate might not be practical.

Non Standard Forms

- $F_3 = AB + C(D + E)$
  $= AB + C(D + E) = AB + CD + CE$



(a) $AB + C(D + E)$ (b) $AB + CD + CE$

FIGURE 2.4
Three- and two-level implementation

- Which kind of gate will have the least delay (high switching speed)?
- The delay through a gate is largely dependent on the circuit design and technology, as well as manufacturing process used. (taught in VLSI design)

Other Logic Operations

- 2^n rows in the truth table of n binary variables
- 2^{2^n} functions for n binary variables (each row may either be 0 or 1)
- 16 (2^{2^2})functions of two binary variables

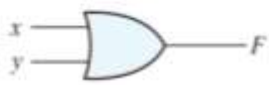
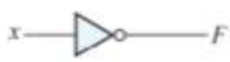
Table 2.7
Truth Tables for the 16 Functions of Two Binary Variables

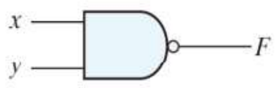
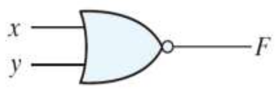
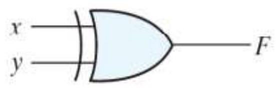
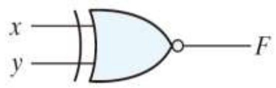
x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Table 2.8
Boolean Expressions for the 16 Functions of Two Variables

Boolean Functions	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
$F_9 = xy + x'y'$	$(x \oplus y)'$	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

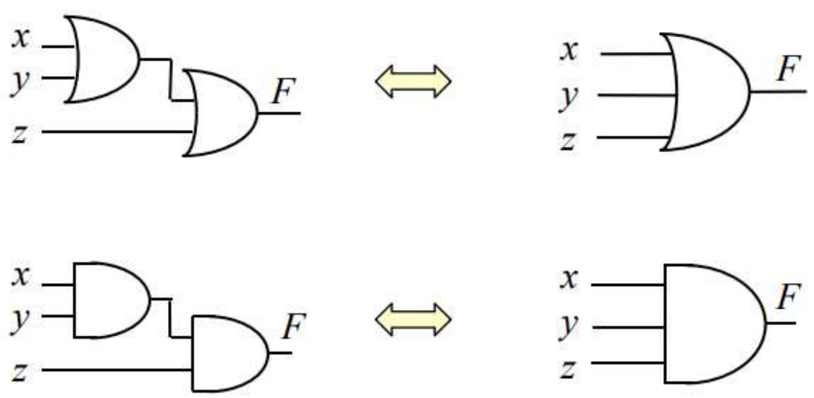
Digital Logic Gates

Name	Graphic symbol	Algebraic function	Truth table
AND		$F = x \cdot y$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1

Name	Graphic symbol	Algebraic function	Truth table
NAND		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

Extension to Multiple Inputs

- A gate can be extended to multiple inputs
  - if its binary operation is commutative and associative
- AND and OR are commutative and associative
  - commutative: $x + y = y + x$, $xy = yx$
  - associative: $(x + y) + z = x + (y + z) = x + y + z$, $(x\,y)z = x(y\,z) = x\,y\,z$



Multiple –input NOR/NAND

- NAND and NOR are commutative but not associative => they are not extendable

  $(x \downarrow y) \downarrow z = [(x + y)' + z]' = (x + y) z' = xz' + yz'$

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

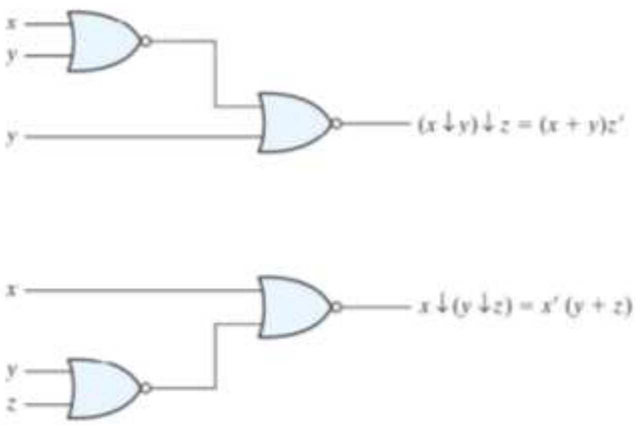


FIGURE 2.6
Demonstrating the nonassociativity of the NOR operator: $(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$

- Multiple-input NOR = a complement of OR gate $(x \downarrow y \downarrow z) = (x + y + z)'$
- Multiple-input NAND = a complement of AND $(x \uparrow y \uparrow z) = (x y z)'$
- The cascaded NAND operations = sum of products
- The cascaded NOR operations = product of sums

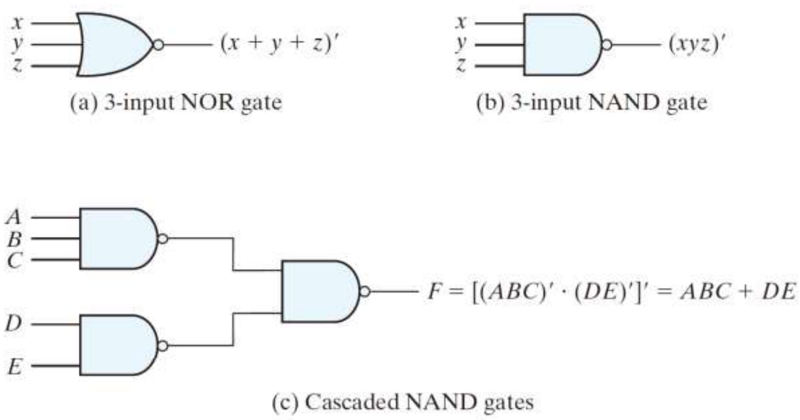
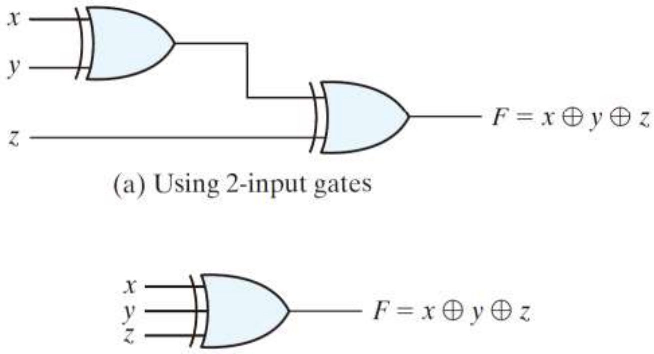


FIGURE 2.7
Multiple-input and cascaded NOR and NAND gates

DeMorgan's theorems are useful here.

Multiple –input XOR/XNOR

- The XOR and XNOR gates are commutative and associative
- Multiple-input XOR gates are uncommon (this is not true anymore!)
- XOR(XNOR) is an odd(even) function: it is equal to 1 if the inputs variables have an odd(even) number of 1's



(a) Using 2-input gates

(b) 3-input gate

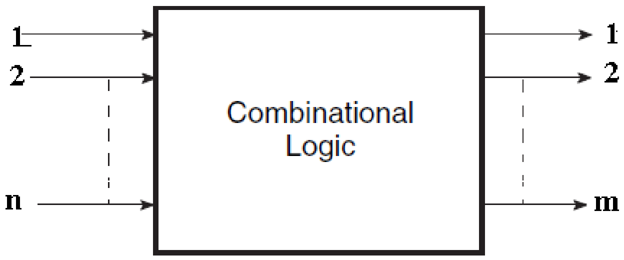
x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(c) Truth table

FIGURE 2.8
Three-input exclusive-OR gate

Combinational Logic:

A combinational Logic consists of input variables, logic gates and output variables. The logic gates accepts signals from the inputs and generate signals to the outputs. This process transforms binary information from the given input data to the required output data.



A block diagram of a combinational circuit is shown in the figure above. The n input binary variables come from an eternal source, the m output variables go to an external destination.

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. A combinational circuit can be described by m Boolean functions, one for each output variable. Each output function is expressed in terms of the n input variables.

Design Procedure:

The design of combinational circuit starts from the verbal outline of a problem and ends in a logic circuit diagram.

The 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 are defined.
5. The simplified Boolean function for each output is obtained.
6. The logic diagram is drawn.

Adders

Digital computers perform a variety of information processing tasks. Among the basic functions encountered are the various arithmetic operation.
The most common basic arithmetic operation is the addition of two binary digits.

Half Adder:

The half adder circuit has two inputs: A and B, which add two input digits and generates a carry and a sum.
The Half Adder adds two binary digits where the input bits are termed as augend and addend and the result will be two outputs one is the sum and the other is carry. To perform the sum operation, XOR is applied to both the inputs, and AND gate is applied to both inputs to produce carry.

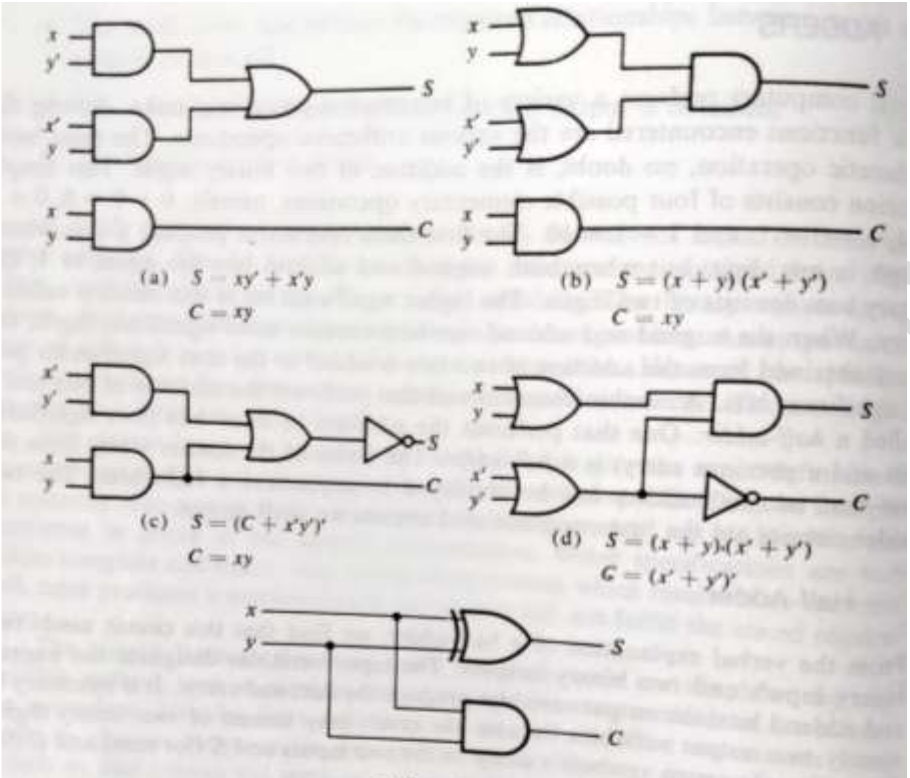
The truth table is shown below:

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

The simplified sum of products expressions are:

$$S = x'y + xy'$$
$$C = xy$$

Various implementation of a half adder:



(a) $S = xy' + x'y$, $C = xy$
(b) $S = (x + y)(x' + y')$, $C = xy$
(c) $S = (C + x'y')'$, $C = xy$
(d) $S = (x + y)(x' + y')$, $C = (x' + y')'$
(e) $S = x \oplus y$, $C = xy$

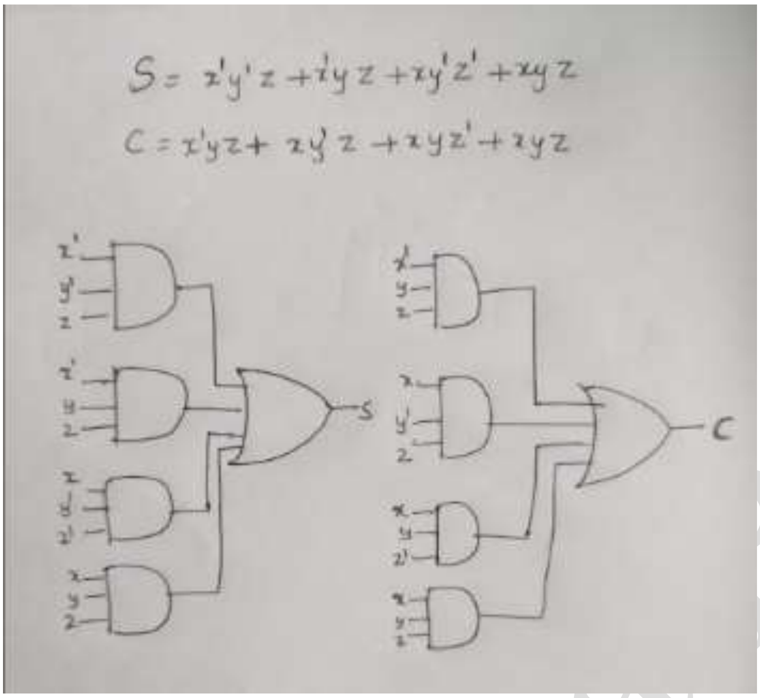
Full Adder:

A full adder is a combinational circuit that forms the arithmetic sum of the three input bits. It consists of three inputs and two outputs.
The Truth Table of the full adder is as follows:

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

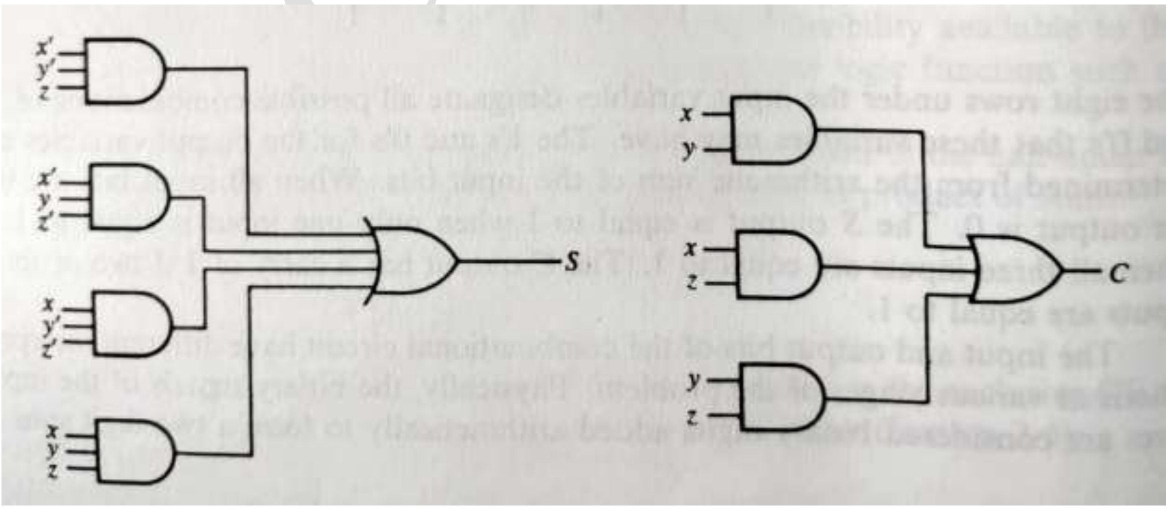
The implementation uses the following Boolean expressions:

First Form:

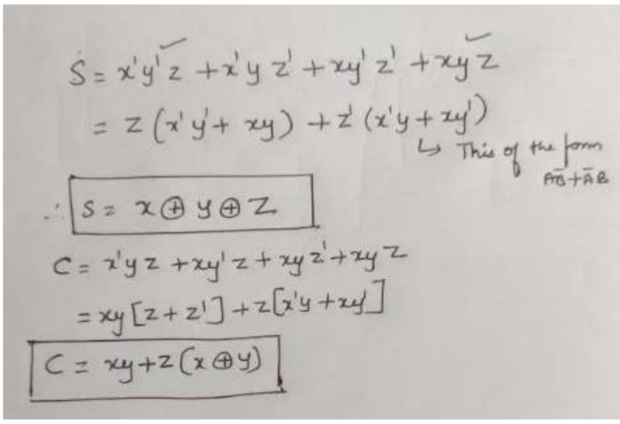


Second Form: The full adder Expression after reduction can also be written as:

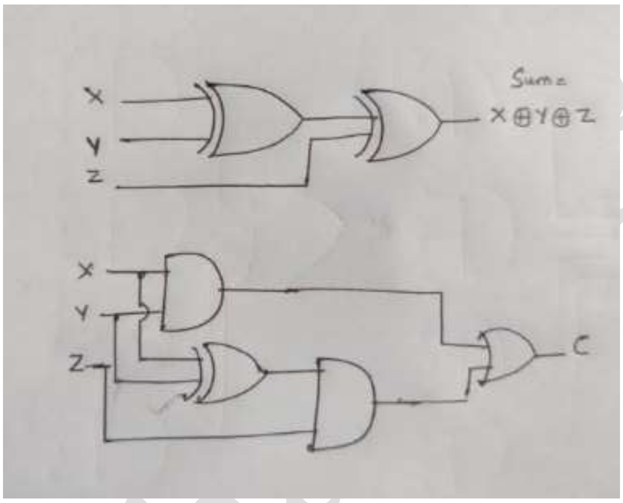
$$S = x'y'z + x'yz' + xy'z' + xyz$$
$$C = xy + xz + yz$$



Third Form: The full adder Expression can also be written as: (After reduction)

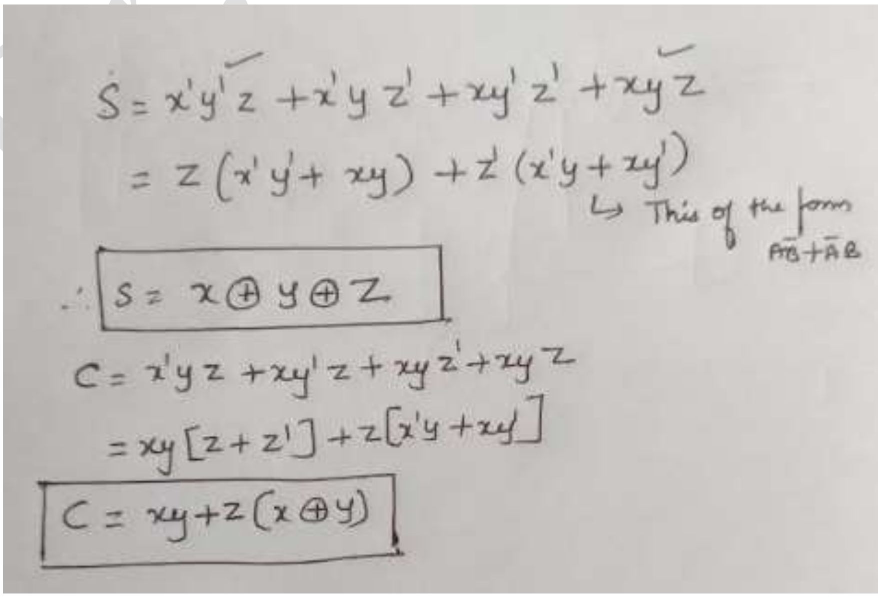


$$S = x'y'z + x'yz' + xy'z' + xyz$$
$$= z(x'y' + xy) + z'(x'y + xy')$$
↳ This of the form $A\bar{B}+\bar{A}B$
$$\therefore \boxed{S = x \oplus y \oplus z}$$
$$C = x'yz + xy'z + xyz' + xyz$$
$$= xy[z+z'] + z[x'y + xy']$$
$$\boxed{C = xy + z(x \oplus y)}$$

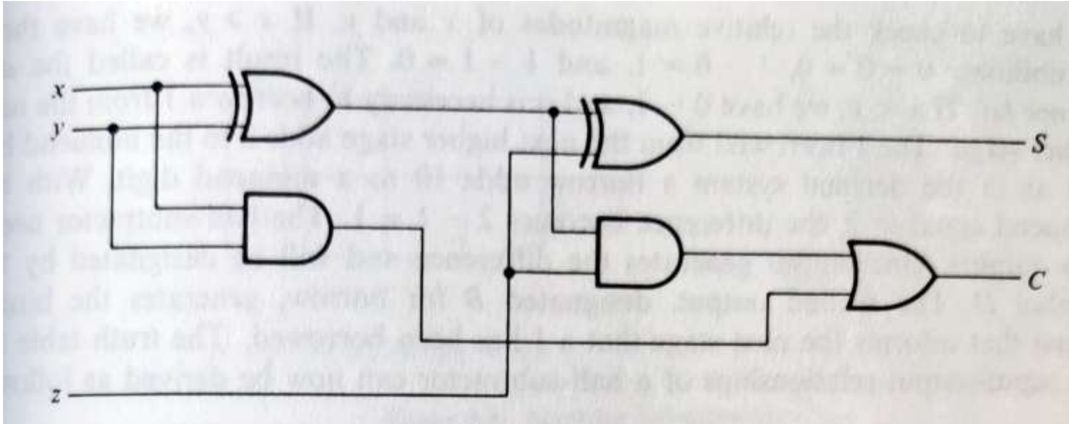


Design Full adder circuit using two Half adder and one OR gate.

Solution: Considering the Full adder expressions:



$$S = x'y'z + x'yz' + xy'z' + xyz$$
$$= z(x'y' + xy) + z'(x'y + xy')$$
↳ This of the form $A\bar{B}+\bar{A}B$
$$\therefore \boxed{S = x \oplus y \oplus z}$$
$$C = x'yz + xy'z + xyz' + xyz$$
$$= xy[z+z'] + z[x'y + xy']$$
$$\boxed{C = xy + z(x \oplus y)}$$



A full adder can be implemented with two half adders and one OR gate as shown in the figure. The S from the second half adder is the exclusive OR-gate of z and the output of the first half adder giving:

$$\begin{aligned} S &= z \oplus (x \oplus y) \\ &= z'(xy' + x'y) + z(xy' + x'y)' \\ &= z'(xy' + x'y) + z(xy + x'y') \\ &= xy'z' + x'yz' + xyz + x'y'z \end{aligned}$$

and the carry output is:

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