POSTAL Study Package 2019

Electronics Engineering

Conventional Practice Sets

Digital Circuits

Contents

MADE EASY Publications

1 CHAPTER Number Systems and Codes

Q.1 Convert the following Octal numbers into its equivalent Hexadecimal numbers

(a) 134 (b) 67

(c) 1527.5 (d) 4753

Solution:

The most convenient method for converting the Octal numbers into its equivalent Hexadecimal is that firstly we have to change octal to the binary (in 3-bits) and then we arrange a group of 4-bits from right to its equivalent Hexadecimal numbers.

(a) $(134)_8 \xrightarrow{\text{Binary}} (001011100)_2$

Now $\underbrace{0000}_{0}\,\underbrace{0101}_{5}\,\underbrace{1100}_{C} = (5C)_{16}$

(b) $(67)_8 \xrightarrow{\text{Binary}} (110111)_2$

Now, $\underbrace{0011}_{3}\,\underbrace{0111}_{7} = (37)_{16}$

(c) $(1527.5)_8 \xrightarrow{\text{Binary}} (001101010111.101)_2$

Now, $\underbrace{0011}_{3}\,\underbrace{0101}_{5}\,\underbrace{0111}_{7}\,.\,\underbrace{1010}_{10}$

In Hexform, $(357.A)_{16}$

(d) $(4753)_8 \xrightarrow{\text{Binary}} (100111101011)_2$

Now, $\underbrace{1001}_{9}\,\underbrace{1110}_{E}\,\underbrace{1011}_{B} \xrightarrow{\text{Hexa}} (9EB)_{16}$

Q.2 List out the rules for the BCD (Binary Coded Decimal) addition with corresponding examples?

Solution:

BCD numbers are class of binary encodings of decimal numbers where each decimal digit is represented by four bits (4 bit code). This is also called 8421 weighted code. Valid BCD numbers are from decimal digit 0-9. BCD is a numerical code. In arithmetic operation, addition is the most important operation. The rule for addition of two BCD numbers are as following :

(i) Add the two numbers using the rules for binary addition.

(ii) In BCD addition if any invalid number (i.e. a 4-bit stream greater than 9) is present then $(0110)_2$ is added to the corresponding group to make it valid.

Example:

Add 56 and 57

56 →	0101	0110
67 →	0110	0111
BCD sum =	1011 (not a valid BCD no.)	1101 (not a valid BCD no.)

So, add 6 → 0110 0110

⇓ ⇓

Now,

$$\text{sum} = \underbrace{1}_{1}\ \underbrace{0010}_{2}\ \underbrace{0011}_{3}$$

(Carry)

$$\therefore \quad 56 + 67 \xrightarrow{\text{BCD SUM}} 123$$

(iii) If any carry is generated from only particular group then we add $(0110)_2$ to the corresponding group.

Example:

Add 74 and 94

$$74 \rightarrow 0111 \quad 0100$$
$$94 \rightarrow 1001 \quad 0100$$

Carry

$$\text{BCD sum} = 1\ \underbrace{0000}_{\text{Due to carry add }(0110)_2}\ \underbrace{1000}_{\text{Valid BCD}}$$

$$\begin{array}{ccc} 1 & 0000 & 1000 \\ & 0110 & \\ \hline \end{array}$$

Now, $$\text{BCD sum} = \underbrace{1}_{1}\ \underbrace{0110}_{6}\ \underbrace{1000}_{8}$$

$$\therefore \quad 74 + 94 \xrightarrow[\text{SUM}]{\text{BCD}} 168$$

Q.3 Convert the following:

(a) Octal 756 to Decimal

(b) Hexadecimal 3B2 to Decimal

(c) Long Binary number 1001001101010001 to Octal and to Hexadecimal

(d) Decimal 675.625 to Hexadecimal

Solution:

(a) Octal to decimal conversion:

$$(756)_8 = 7 \times 8^2 + 5 \times 8^1 + 6 \times 8^0$$
$$= 7 \times 64 + 5 \times 8^1 + 6 \times 1$$
$$= (494)_{10}$$

(b) Hexa to decimal:

$$(3\,B\,2)_{16} = 3 \times 16^2 + B \times 16^1 + 2 \times 16^0$$
$$= 3 \times 256 + 11 \times 16 + 2$$
$$= (946)_{10}$$

(c) Binary number can be converted into equivalent octal by making groups of 3-bits starting from LSB and moving towards MSB for integer part and then replace each group of 3-bit by its octal representation.

So,

MSB side $$\underset{1}{1}\ \underbrace{001}_{1}\ \underbrace{001}_{1}\ \underbrace{101}_{5}\ \underbrace{010}_{2}\ \underbrace{001}_{1}$$ LSB side

$$= (111521)_8$$

Now for Hexadecimal conversion we have to group 4-bit from LSB to MSB as shown.

2 CHAPTER Boolean Algebra, Logic Gates and Minimization Techniques

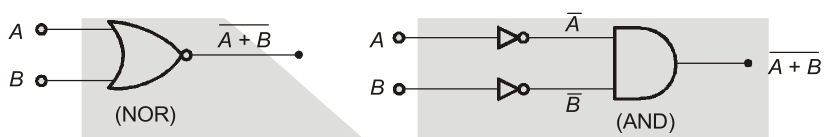
Q.1 Explain De-Morgan's first theorem using circuits of NOR gate as well as AND gate with inverted inputs. Give the relevant table using NAND gate and OR gate with inverted inputs, explain De-Morgain's second theorem.

Solution:

Two of the most important theorems of Boolean algebra were contributed by DeMorgan. These theorems are extremely useful in simplifying expressions in which a product or sum of variables is inverted.

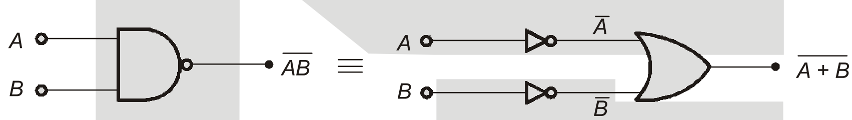
- 1st theorem of De-Morgan's stated that when the OR sum of two variables is inverted, it is same as inverting each variable individually and then ANDing these inverted variables.

$$\overline{A+B} = \overline{A}.\overline{B} \quad \text{...(i)}$$



- 2nd theorem of De-Morgan's stated that when the AND product of two variables is inverted, it is same as inverting each variable individually and then ORing these inverted variables.

$$\overline{A\cdot B} = \overline{A}+\overline{B} \quad \text{...(ii)}$$



⇒ These two above theorems can be verified by the relevant truth table wihch is given below:

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

C_1 C_2 C_3 ---------------------- C_7 C_8 C_9 C_{10}

From this Truth table, it is clear that column no. 7 and 8 are equal so, our theorem (i) is proved.

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

Similarly, from column number 9 and 10, we have,

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

These two theorems can be extended up to *n*-variable as,

$$\overline{A+B+C+\text{..........}} = \overline{A}\cdot\overline{B}\cdot\overline{C}\text{..........}$$

and $$\overline{A\cdot B\cdot C\cdot\text{..........}} = \overline{A}+\overline{B}+\overline{C}+\text{...........}$$

Q.2 Design a circuit using NAND gates only that has one control line and three data lines. When the control line is HIGH, the circuit is to detect when one of the data lines has a '1' on it. No more than one data line will ever have a '1' on it. When the control line is low, the circuit will output a '0', regardless of what is on the data lines?

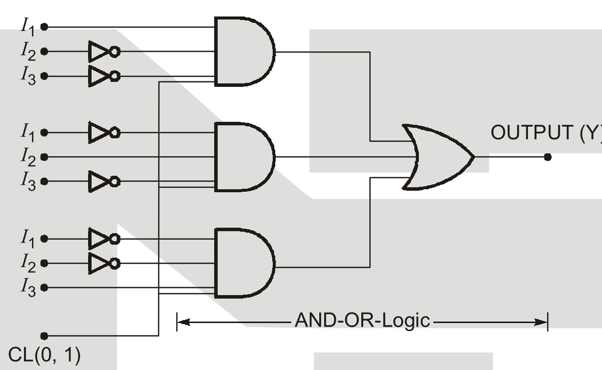
Solution:

Let us consider three data lines as I_1, I_2 and I_3 and control lines as "CL". We know that AND-OR logic is equivalent to NAND-NAND logic. Since we have to design a circuit using NAND gates only so we may proceed by AND-OR circuit.

According to the given condition, there are three combinations of data lines.

	I_1	I_2	I_3
1st case	1	0	0
2nd case	0	1	0
3rd case	0	0	1

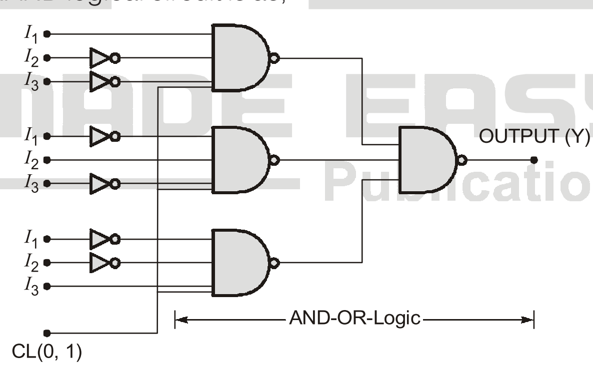
$\Rightarrow$ *when* $CL = 1$



When CL = 1, circuit is to detect

CL = 0, output (Y) = 0

Now equivalent NAND-NAND logical circuit is as,



Q.3 (a) The sum of all minterms of a Boolean function of n variables is 1. Prove the statement for $n = 3$.

(b) Show that a positive logic AND gate is a negative logic OR gate, and vice-versa.

Solution:

(a)
$$F(A, B, C) = A'B'C' + A'B'C + A'BC' + A'BC + AB'C' + AB'C + ABC' + ABC$$
$$= A'(B'C' + B'C + BC' + BC) + A(B'C' + B'C + BC' + BC)$$
$$= (A' + A)(B'C' + BC' + B'C + BC)$$
$$= B'(C' + C) + B(C' + C) = B' + B = 1$$

For $n = 3$, i.e. 3 variables the sum of all minterms is 1.

3 CHAPTER Combinational Digital Circuits

Q.1 What do you mean by combinational digital circuit? For designing a combinational circuit what are the main procedure involved?

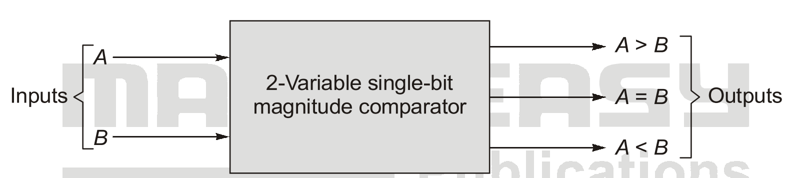
Solution:

- A combinational circuit consists of input variables, logic gates and output variables, where outputs at any time are determined from only the present combination of inputs. And their logic gates react to the values of the signals at their inputs and produce the value of output signal, transforming binary information from the given input data to the required output data. Also we can say a combinational digital circuit performs an operation that can be specified logically by a set of boolean functions.
- The design of combinational circuit has some procedure that are given in the following steps:
  (i) From the specification of the circuit, determine the required number of inputs and outputs and assign a symbol to each.
  (ii) Derive the truth table that defines the required relationship between inputs and outputs.
  (iii) Obtain the simplified boolean function for each output as a function of the input variables.
  (iv) Finally, draw the logic gate diagram and verify the correctness of the design.

Q.2 Develop the logical expression for a comparator involving two variables *A* and *B*. Also design a combinational circuit with appropriate gates.

Solution:

The comparison of two numbers is an operation that determines whether one number is greater than, less than or equal to the other number. *A* "magnitude comparator" is a combinational circuit that compares two numbers A and B and outcome of the comparison determines the three binary variables that indicate whether $A > B$, $A = B$, or $A < B$.



Truth table:

A	B	$A > B$	$A = B$	$A < B$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

For output, $A > B \Rightarrow A\bar{B}$

For output, $A = B \Rightarrow \bar{A}\bar{B} + AB = A \odot B$