

SUBJECT: COMPUTER SCIENCE
CLASS 11Th**UNIT 3****NUMBER SYSTEM****MARKS: 05****Q. Define number system. Explain types of number systems.**

It is a set of rules and symbols, used to represent a number. It also defines a set of operations that can be performed on the numbers. There are two types of number systems.

1. **Non Positional number system:** In this system if we change position of a symbol its value does not get changed.
2. **Positional number system:** In this system if we change position of a symbol its value also changes.

Example: Decimal number system

Q. Explain following number systems.

1. **Decimal number system:** It has a base of 10. It uses 10 digits as 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 to form all the numbers of this number system.

As a general case, a decimal number is represented by two groups of digits separated by the symbol "." called the decimal point (or radix point). The portion of the number to the right of the decimal point is called the fractional part, and the portion to the left of the decimal point is called the integral part.

Example: 985_{10} , 7023.75_{10} etc.

2. **Binary number system:** It has a base of 2. It uses only 2 digits as 0 and 1 to form all the numbers of this number system.

Computer doesn't make use of decimal number system. Internally it makes use binary number system. It is the fundamental base of all the computers.

Example: 1011_2 , 11000_2 etc.

3. **Octal number system:** It has a base of 8. It uses 8 digits as 0, 1, 2, 3, 4, 5, 6 and 7 to form all the numbers of this number system.

The octal number system was used widely in the Electronics and Computer Industry. Although digital electronics is based on gates with only two states and is therefore fundamentally binary but binary numbers can quickly become long and hard to record without errors. Their octal equivalents are much shorter and easier to remember.

Example: 532_8 , 661_8 etc.

4. **Hexadecimal number system:** It has a base of 16. It uses TEN digits as 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and six symbols as A, B, C, D, E, F to form all the numbers of this number system. The hexadecimal system is useful because it can represent every byte (8 bits) as two consecutive hexadecimal digits. It is easier for humans to read hexadecimal numbers than binary numbers.

Example: $5A3_{16}$, $F97_{16}$ etc.

Q. Number System conversions.

The conversion between different number systems is to be performed mathematically, so all the methods and examples related to number system conversion will be shown in the class.

Q. Conversion from binary number system to decimal number system.

Q. Conversion from octal number system to decimal number system.

Q. Conversion from hexadecimal number system to decimal number system.

Q. Conversion from decimal number system to binary number system.

Q. Conversion from decimal number system to octal number system.

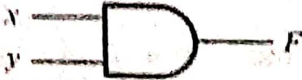

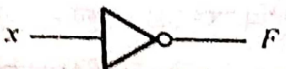
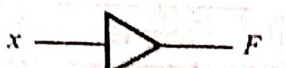
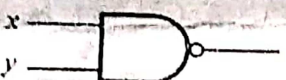



Q. Conversion from decimal number system to hexadecimal number system.

Q. Conversion from binary number system to octal number system using shortcut method.

Q. Conversion from binary number system to octal number system using shortcut method.

Q. Conversion from octal number system to binary number system using shortcut method.

Q. Conversion from hexadecimal number system to binary number system using shortcut method.

Name	Graphic symbol	Algebraic function	Truth table															
AND		$F = xy$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	0	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = x + y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	1
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Inverter		$F = x'$	<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	x	F	0	1	1	0									
x	F																	
0	1																	
1	0																	
Buffer		$F = x$	<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	x	F	0	0	1	1									
x	F																	
0	0																	
1	1																	
NAND		$F = (xy)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	1	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = (x + y)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	0
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= x \odot y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																