# Digital Logic

Tutoriol:4 (Simplification & Combinational Logic) **Dead line: 2080/11/02**

1) State De Morgan's theorem and verify it using venn diagram.
2) Using K-map, simplify the following Boolean functions in SOP and POS form:
   i) F(A,B,C,D)=∑(5,7,9,12,13,14,15) with don't care d(A,B,C,D)=∑(3,6,8).
   ii) F(A,B,C,D)=∑(0,6,8,13,14) with don't care d(A,B,C,D)=∑(2,4,10).
   iii) F(A,B,C,D)=∑(2,3,7,10,11,14) with don't care d(A,B,C,D)=∑(1,5,15).
   iv) F(A,B,C,D)=∑(1,4,5,6,12,14,15) with don't care d(A,B,C,D)=∑(10,11).

   Implement i) & II) using NAND gate only and iii) & iv) using NOR gate only. Use 2-input gates only.

3) Simplify F(w,x,y,z)=∑(2,3,4,5,6,7,11,14,15) in POS form and implement with OR & AND gates.
4) Draw X-OR gate with minimum number of NAND gates. Also show that dual of X-OR is its complement.
5) Design a full subtractor circuit. Also construct it using two half subtractors.
6) Draw the truthtable of a full adder.Show that a full adder can be implemented using two half adders and an OR gate.
7) Design a combinational circuit with 4-input lines that represents a decimal digits in BCD and 4-output lines that generates the 9's complement of the input digits.
8) Design a combinational circuit that converts a decimal digit from 8 4 -2 -1 to the BCD.
9) Design a combinational circuit that converts 4-bit reflected code number to 4-bit binary number. Implement the circuit with X-OR gates.
10) Design a combinational circuit to generate odd parity of 3-bits. Also design a combinational circuit to check for odd parity. A logic-1 is required when the 4-bits donot constitute an odd parity.
11) Simplify the Boolean function:
    $F=A^{l}B^{l}D^{l}+A^{l}CD+A^{l}BC$ with don't care $D=A^{l}BC^{l}D+ACD+AB^{l}D^{l}$. Implement it using
    i) NAND gates only
    ii) NOR gates only {Use 2-i/p gates only for both the cases}
12) Show that $(A\odot B\odot C\odot D)=\sum(0,3,5,6,9,10,12,15)$.
13) Design a combinational logic circuit which converts BCD to seven segment decoder. Use the proper don't care conditions.