HACETTEPE UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING
BBM231 LOGIC DESIGN

Homework 2 (For all sections)

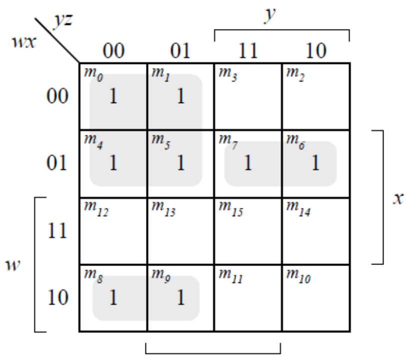
Assigned : **5.11.2018**
Due : **12.11.2018**

Hand in your homework solutions in class.

QUESTIONS:

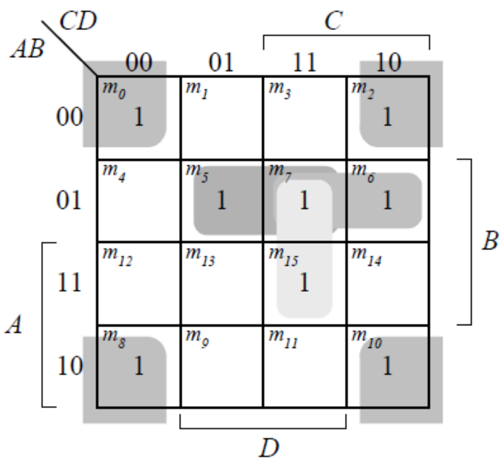
Q1. Simplify the following Boolean functions using four-variable maps:

a) $F(w,x,y,z) = \sum(0,1,4,5,6,7,8,9)$



(c) $F = w'y' + wx'y' + w'xy$

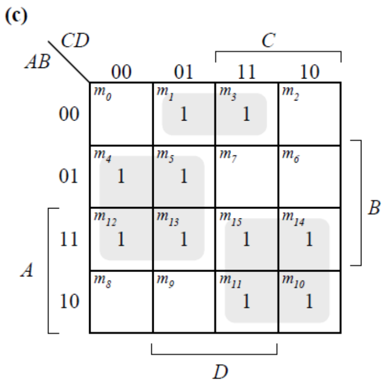
b) $F(A,B,C,D) = A'B'C'D' + A'CD' + AB'D' + ABCD + A'BD$



(c) $F = B'D' + BCD + A'BD + A'BC$

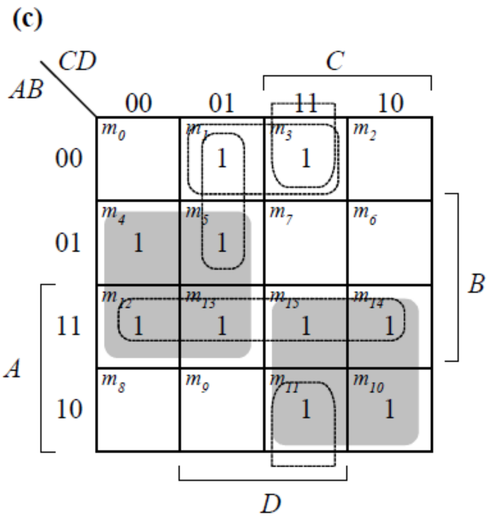
Q2. Find all the prime implicants for the Boolean functions and determine which are essential:

a) $F(A,B,C,D) = \sum(1,3,4,5,10,11,12,13,14,15)$



Essential: BC', AC, $A'B'D$
$F = BC' + AC + A'B'D$

b) $F(w,x,y,z) = \sum(0,2,4,5,6,7,8,10,13,15)$



Essential: BC', AC
Non-essential: AB, $A'B'D$, $B'CD$, $A'C'D$
$F = BC' + AC + A'B'D$

Q3. Given $h = f \oplus g$,

$f(w,x,y,z)= \sum(2,3,5,6,7,12,13,15)$,

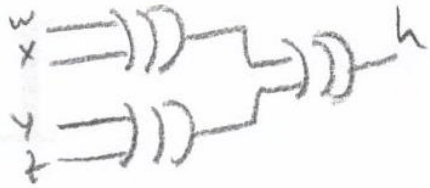
$g(w,x,y,z)=\sum(1,3,4,5,6,8,11,12,14,15)$,

find the minimal expression for $h(w,x,y,z)$ and draw its circuit using minimum number of two-input-gates. You can use AND, OR, NOT, XOR gates with two-inputs.

	w	x	y	z	f	g	h
0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	1
2	0	0	1	0	1	0	1
3	0	0	1	1	1	1	0
4	0	1	0	0	0	1	1
5	0	1	0	1	1	1	0
6	0	1	1	0	1	1	0
7	0	1	1	1	1	0	1
8	1	0	0	0	0	1	1
9	1	0	0	1	0	0	0
10	1	0	1	0	0	0	0
11	1	0	1	1	0	1	1
12	1	1	0	0	1	1	0
13	1	1	0	1	1	0	1
14	1	1	1	0	0	1	1
15	1	1	1	1	1	1	0

wx \ yz	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

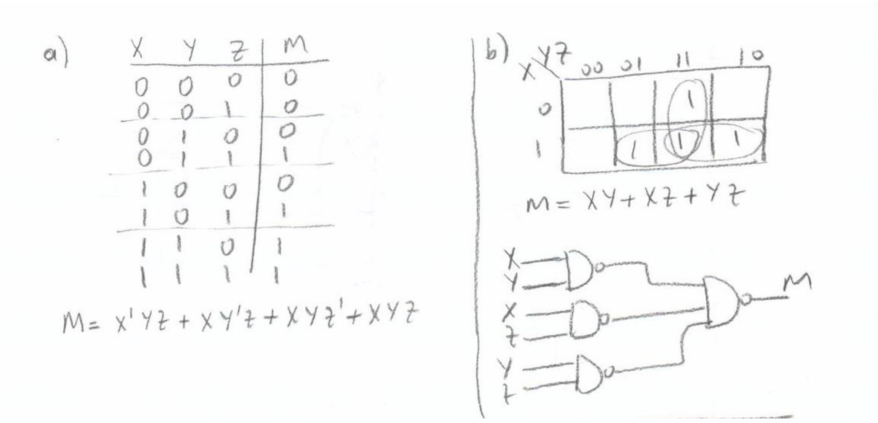
$h = w \oplus x \oplus y \oplus z$



Q4. A majority function has an output value of one if there are more 1s than 0s on its inputs.

a. Express three input majority function in sum of minterms form after filling the truth table.

b. Implement the **optimized circuit** with only NAND gates.

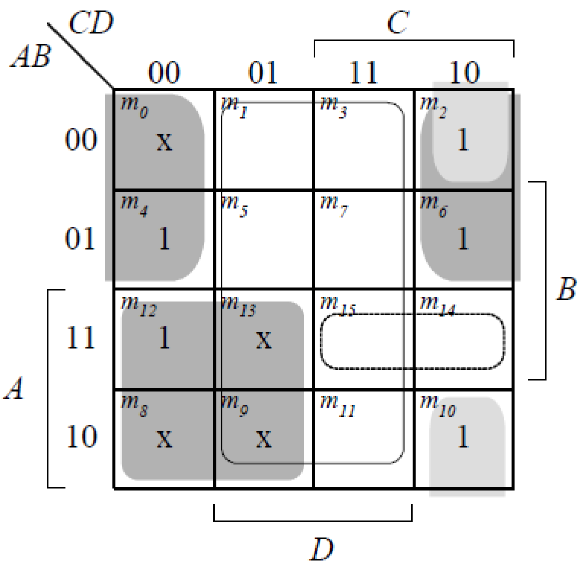


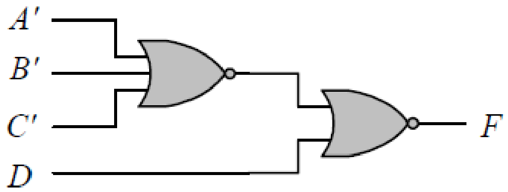
Q5. Implement the following Boolean function F, together with the don't-care conditions *d,* using no more than two NOR Gates.

$F(A,B,C,D) = \sum(2,4,6,10,12)$

$d(A,B,C,D) = \sum(0,8,9,13)$

Assume that both the normal and complement inputs are available.





$F = AC' + A'D' + B'CD'$

$F' = D + ABC$

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