<u>HOMEWORK I</u> DUE: Group1, 5 April 2017 and Group2, 6 April 2017

1. Express $(1965)_{10}$, $(367)_{10}$, $(29)_{10}$, in
   i. BCD
   ii. Excess-3
   iii. 2421

Decimal	1	9	6	5
BCD	**0001**	**1001**	**0110**	**0101**
Excess-3	**0011**	**1100**	**1001**	**1000**
2421	**0001**	**1111**	**1100**	**1011**

Decimal	3	6	7
BCD	**0011**	**0110**	**0111**
Excess-3	**0110**	**1001**	**1010**
2421	**0011**	**1100**	**1101**

Decimal	2	9
BCD	**0010**	**1001**
Excess-3	**0101**	**1100**
2421	**0010**	**1111**

2. Convert the following decimal numbers 77 and 38 to binary using the signed 2's complement representation and enough digits to accommodate numbers. Then perform the binary equivalent of (+77) + (-38), (-77) + (+38) and (-77) + (-38).

77	2 = 38	1	
38	2 = 19	0	
19	2 = 9	1	
9	2 = 4	1	
4	2 = 2	0	
2	2 = 1	0	
1	2 = 0	1	
77 = $(1001101)_2$			

38	2 = 19	0
19	2 = 9	1
9	2 = 4	1
4	2 = 2	0
2	2 = 1	0
1	2 = 0	1
38 = $(100110)_2$		

Therefore for sign 2's complement representation of

+77	**01001101**	-77	**10110011**
+38	**00100110**	-38	**11011010**

(+ 77)	01001101
+ (- 38)	+ 11011010
(+39)	**100100111** most significant bit is discarded.

(- 77)	10110011
+ (+ 38)	+ 00100110
(- 39)	**11011001** signed 2's complement of -39.

(- 77)	10110011
+ (- 38)	+ 11011010
(- 115)	**110001101** most significant bit is discarded. The rest is signed 2's complement of -115.

3. The following is a string of ASCII characters whose bit patterns have been converted into hexadecimal for compactness: 23 CE EF F2 F4 C8 43 79 D0 F2 75 D3. Of the 8 bits in each pair of digits, the leftmost is a parity bit. The remaining bits are the ASCII code.

   i. Convert to bit form and decode the ASCII.

Hex	Parity	ASCII		Hex	Parity	ASCII	
23	0	010 0011	**#**	**43**	0	100 0011	**C**
CE	1	100 1110	**N**	**79**	0	111 1001	**y**
EF	1	110 1111	**o**	**D0**	1	101 0000	**P**
F2	1	111 0010	**r**	**F2**	1	111 0010	**r**
F4	1	111 0100	**t**	**75**	0	111 0101	**u**
C8	1	100 1000	**H**	**D3**	1	101 0011	**S**

   ii. Determine the parity used: odd or even.

   Odd

4. Without simplification for a given Boolean function x' + xy + xz (y'+z) + (xz')'.

   i. Find the dual of the expression.

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

   ii. Find the complement of the expression.

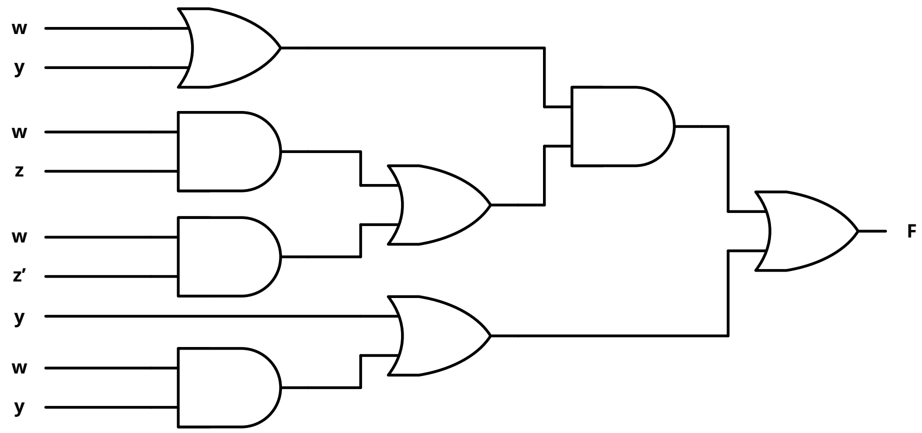
   = x (x'+y') (x'+z') + (yz') (x'+z)

5. Given the Boolean function F = (w+y) (wz + wz’) + wy + y

   i. Obtain the truth table of the function.

w	x	y	z	z’	w+y	wz	wz’	wz+wz’	(w+y)(wz+wz’)	wy	wy+y	F
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	0	0	0	0	0	1	1
0	0	1	1	0	1	0	0	0	0	0	1	1
0	1	0	0	1	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0	0	0	1	1
0	1	1	1	0	1	0	0	0	0	0	1	1
1	0	0	0	1	1	0	1	1	1	0	0	1
1	0	0	1	0	1	1	0	1	1	0	0	1
1	0	1	0	1	1	0	1	1	1	1	1	1
1	0	1	1	0	1	1	0	1	1	1	1	1
1	1	0	0	1	1	0	1	1	1	0	0	1
1	1	0	1	0	1	1	0	1	1	0	0	1
1	1	1	0	1	1	0	1	1	1	1	1	1
1	1	1	1	0	1	1	0	1	1	1	1	1

   ii. Draw the logic diagram using the Boolean expression.



iii. Simplify the function to a minimum number of literals using Boolean algebra.

F = (w+y) (wz + wz') + wy + y

= (w + y) w(z+z') + wy + y

= (w + y) w (1) + wy +y

= (w + y) w + wy +y

= (w.w + wy) + wy + y

= w + wy + wy + y

= w + wy + 1

= w (1+y) +y

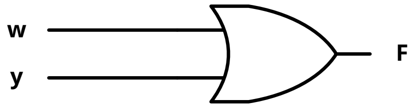
= w (1) + y

= w + y

iv. Obtain the truth table of the function from the simplified expression and show that it is the same as the one in part (i).

w	x	y	z	w+y	=	F (i)
0	0	0	0	0		0
0	0	0	1	0		0
0	0	1	0	1		1
0	0	1	1	1		1
0	1	0	0	0		0
0	1	0	1	0		0
0	1	1	0	1		1
0	1	1	1	1		1
1	0	0	0	1		1
1	0	0	1	1		1
1	0	1	0	1		1
1	0	1	1	1		1
1	1	0	0	1		1
1	1	0	1	1		1
1	1	1	0	1		1
1	1	1	1	1		1

v. Draw the logic diagram from the simplified expression and compare the total number of gates with the diagram of part (ii).



6. Express the following function in sum of products and product of sums.

F(A,B,C,D) = ABD′ + AB + BC′ + C′D′

STEP1:

ABD′ (C + C′) = **ABCD′** + **ABC′D′**

STEP2:

AB (C + C′) = ABC + ABC′

ABC (D + D′) + ABC′ (D+D′) = **ABCD** + **ABCD′** + **ABC′D** + **ABC′D′**

STEP3:

BC′ (A + A′) = ABC′ + A′BC′

ABC′ (D + D′) + A′BC′ (D + D′) = **ABC′D** + **ABC′D′** + **A′BC′D** + **A′BC′D′**

STEP4:

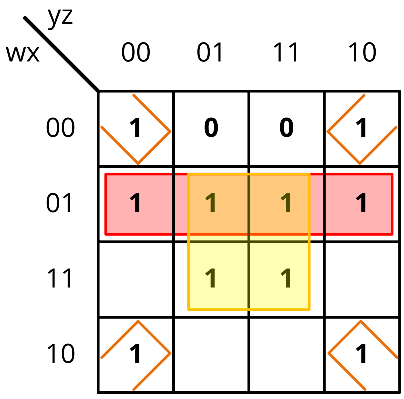
C′D′ (A + A′) = AC′D′ + A′C′D′

AC′D′ (B + B′) + A′C′D′ (B + B′) = **ABC′D′** + **AB′C′D′** + **A′BC′D′** + **A′B′C′D′**

F = ABCD′ + ABC′D′ + ABCD + ABC′D + A′BC′D + A′BC′D′+ AB′C′D′ + A′B′C′D′								
F =	**1110**	**1100**	**1111**	**1101**	**0101**	**0100**	**1000**	**0000**
F =	**m14**	**m12**	**m15**	**m13**	**m5**	**m4**	**m8**	**m0**
F(A,B,C,D) = ∑ (0,4,5,8,12,13,14,15)								
F(A,B,C,D) = ∏ (1,2,3,6,7,9,10,11)								

7. Simplify the following Boolean function using K-Map by first finding the essential prime implicants.

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



Essential Prime Implicants: F = x'z' + xz

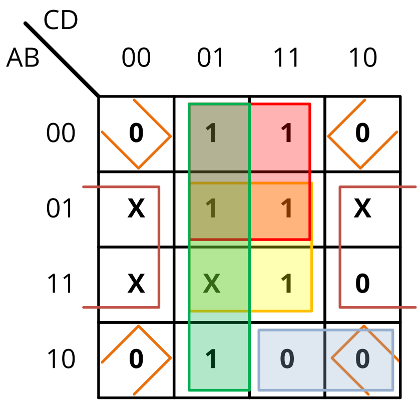
Prime Implicants: F = w'x

So F = x'z' + xz + w'x

8. Simplify the following Boolean function F, together with the don't care conditions d and then express the simplified function in sum of minterms and product of maxterms.

   $F(A,B,C,D) = \Sigma(1,3,5,7,9,15)$,

   $d(A,B,C,D) = \Sigma(4,6,12,13)$



F = C'D + A'D + BD (SOP)

F' = B'D' + BD' + AB'C

F = (B + D) (B' + D) + (A' + B + C')