Faculty of Engineering

DEPARTMENT of ELECTRICAL AND ELECTRONIC ENGINEERING

EENG (INFE) 115	Instructor:
Introduction to Logic Design	G. YEMİŞÇİOĞLU
Midterm EXAMINATION	Duration: 90 minutes
April 8, 2017	Number of Questions: 4

Good Luck

STUDENT'S	
NUMBER	
NAME	SOLUTIONS
SURNAME	
GROUP NO.	

Question	Achieved	Points
1		25
2		25
3		25
4		25
TOTAL		100

Read the following instructions carefully:

1. Calculators are not allowed.
2. Switch off mobile phones and do not borrow any stationery from your friends.
3. In your solutions, show all details you claim credit for.

<u>Question 1:</u>

a) Convert $(82.6875)_{10}$ to binary. Show your steps. (**5 pts.**)

82	2 =	41	0
41	2 =	20	1
20	2 =	10	0
10	2 =	5	0
5	2 =	2	1
2	2 =	1	0
1	2 =	0	1

$82 = (1010010)_2$

0.6875	x 2	=	1.375	1
0.375	x 2	=	0.75	0
0.75	x 2	=	1.50	1
0.50	x 2	=	1.00	1

$0.6875 = (1011)_2$

$(1010010.1011)_2$

b) Convert $(1011011.0101)_2$ to decimal. Show your steps. (**5 pts.**)

1	0	1	1	0	1	1	.	0	1	0	1
2^6	2^5	2^4	2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}	2^{-4}
1x64	0	1x16	1x8	0	1x2	1	.	0	1x0.25	0	0.0625
64	+ 0	+ 16	+ 8	+ 0	+ 2	+1	.	+ 0	+ 0.25	+ 0	+ 0.0625
						91	.	3125			

c) Convert $(6473.2510)_8$ to binary and to hexadecimal. Show your steps. (**5 pts.**)

6	4	7	3	.	2	5	1	0	oct
110	100	111	011	.	010	101	001	000	bin
1 1 0 1	0 0 1 1	1 0 1 1		.	0 1 0 1	0 1 0 0	1 0 0 0		
D	3	B		.	5	4	8		hex

d) Convert +63 and +42 to binary, using the signed-2's complement representation and enough digits to accommodate the numbers. Then perform the binary equivalent of (+63) + (-42), (-63) + (+42), and (-63) + (-42). (**10 pts.**)

63	2 =	31	1
31	2 =	15	1
15	2 =	7	1
7	2 =	3	1
3	2 =	1	1
1	2 =	0	1

42	2 =	21		0
21	2 =	10		1
10	2 =	5		0
5	2 =	2		1
2	2 =	1		0
1	2 =	0		1

$63 = (111111)_2$

$42 = (101010)_2$

Therefore for sign 2's complement representation of

+63	00111111	-63	11000001
+42	00101010	-42	11010110

(+ 63)	00111111
+ (- 42)	+ 11010110
(+21)	100010101 most significant bit is discarded.

(- 63)	11000001
+ (+ 42)	+ 00101010
(- 21)	11101011 signed 2's complement of -21.

(- 63)	11000001
+ (- 42)	+ 11010110
(- 21)	110010111 most significant bit is discarded. The rest is signed 2's complement of -105.

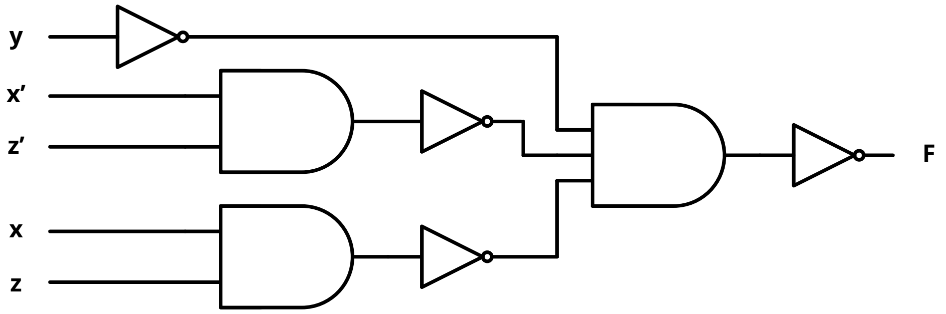
Question 2:

a) Simplify the following Boolean function ***F*(x,y,z) = x'y'z' + x'y + xyz' +xz** to a minimum number of literals using **algebraic manipulation** and implement

= x'y'z' + x'y + xyz' +xz
= x' (y'z' + y) + x (yz' + z)
= x' [(y + y') (y + z')] + x [(y + z) (z + z')]
= x' [(1) (y + z')] + x [(y + z) (1)]
= x' (y + z') + x (y + z)
= x'y + x'z' + xy +xz
= x'y + xy + x'z' + xz
= y (x'+x) + x'z' + xz
= y (1) + x'z' +xz
=y + x'z' + xz

i. With **AND** and **inverter gates** (7.5 pts.)

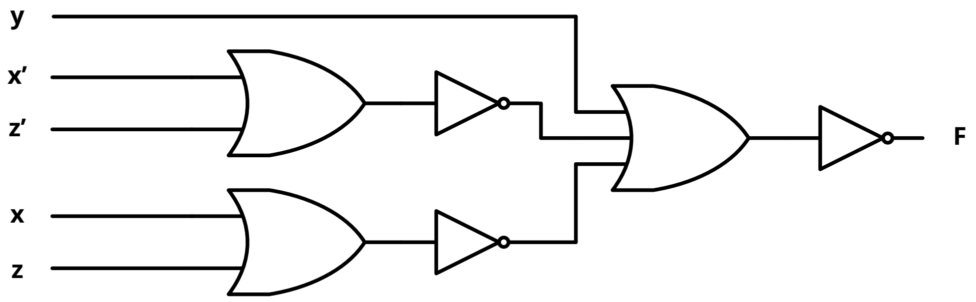
[(y)' (x'z')' (xz)')]' implementation with AND and inverters
((y)')' + ((x'z')')' + ((xz)')'
y + x'z' + xz



ii. With **OR** and **inverter gates** (7.5 pts.)

(15 pts.)

y + (x+z)' + (x'+z')' implementation with AND and inverters
y + x'z' + xz



b) The function ***F*(A,B,C,D) = A'BCD' + AB'C'+ BCD' +ACD**,

i. Construct a truth table for the function given above.

A	B	C	D	A'	B'	C'	D'	A'BCD'	AB'C'	BCD'	ACD	F	Min	Max
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	0	0		1
0	0	1	0	1	1	0	1	0	0	0	0	0		2
0	0	1	1	1	1	0	0	0	0	0	0	0		3
0	1	0	0	1	0	1	1	0	0	0	0	0		4
0	1	0	1	1	0	1	0	0	0	0	0	0		5
0	1	1	0	1	0	0	1	1	0	1	0	1	6	
0	1	1	1	1	0	0	0	0	0	0	0	0		7
1	0	0	0	0	1	1	1	0	1	0	0	1	8	
1	0	0	1	0	1	1	0	0	1	0	0	1	9	
1	0	1	0	0	1	0	1	0	0	0	0	0		10
1	0	1	1	0	1	0	0	0	0	0	1	1	11	
1	1	0	0	0	0	1	1	0	0	0	0	0		12
1	1	0	1	0	0	1	0	0	0	0	0	0		13
1	1	1	0	0	0	0	1	0	0	1	0	1	14	
1	1	1	1	0	0	0	0	0	0	0	1	1	15	

ii. Use the truth table of (i) to write the function *F* in sum of minterms.

F(A,B,C,D) = $\sum$(6,8,9,11,14,15)

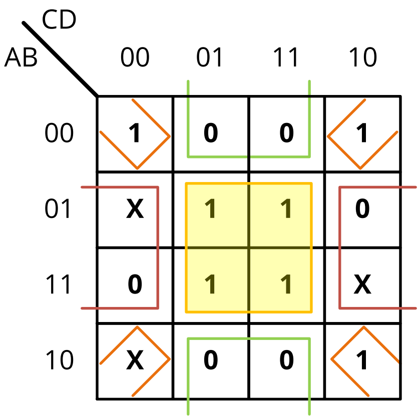
iii. Use the truth table of (i) to write the function *F* in product of maxterms.

F(A,B,C,D) = $\prod$(0,1,2,3,4,5,7,10,12,13)

(10 pts.)

Question 3:

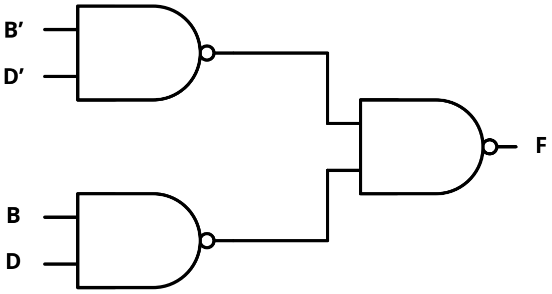
For a given Boolean function **F(A,B,C,D) = ∑(0,2,5,7,10,13,15)** which has don't care conditions **d(A,B,C,D) = ∑(4,8,14).**



a) Determine the sum of products (SOP).

F = B'D' + BD

b) Implement *F* with only **NAND** gates.

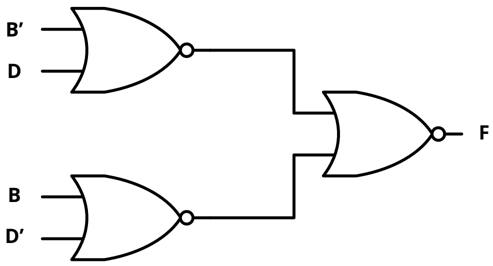


c) Determine the product of sums (POS).

F' = BD' + B'D

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

d) Implement *F* with only **NOR** gates.



(25 pts.)

<u>Question 4:</u>

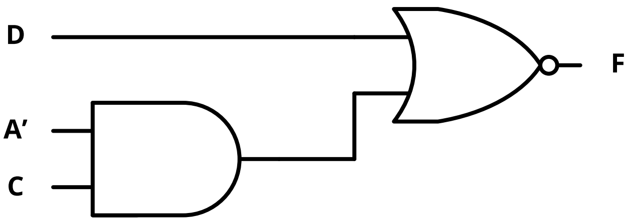
Simplify and then implement the Boolean function

$F(A,B,C,D) = \prod(1,2,3,5,6,7,9,11,13,15)$ using two-level forms of logic by using

AB \ CD	00	01	11	10
00	1	0	0	0
01	1	0	0	0
11	1	0	0	1
10	1	0	0	1

a) AND – NOR (SOP)

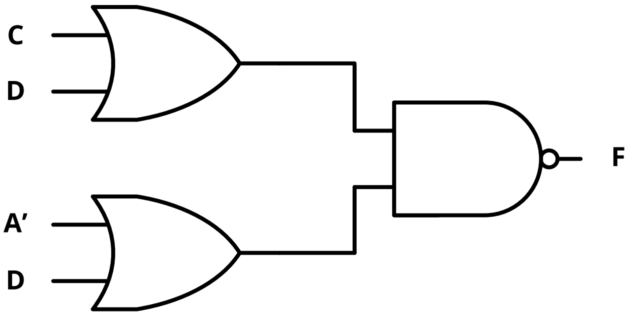
F = (D + A'C)'



b) OR – NAND (POS)

F' = (C'D' + AD')

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



(25 pts.)