Name: SOLUTIONS, ID: ____________

1. **[10 marks total.]**

(a) **[6 marks.]** Represent the following numbers using 6-bit 2's-complement format and perform the addition. Indicate in each case if an overflow occurs, and why.

(i) $(-12_{10}) + 13_{10}$

$+12_{10} = 001100_2$ → invert: 110011, + 1 → $-12_{10} = 110100_2$ (2's compl)

$+13_{10} = 001101_2$

$-12 = 110100$
$+13 = 001101$
ignore → (1)000001 ⇒ $000001 = +1_{10}$

$C_{IN} = C_{OUT}$ ⇒ NO OVERFLOW

(ii) $(-14_{10}) + (-32_{10})$

$+14_{10} = 001110_2$ → invert: 110001, + 1 → $-14_{10} = 110010_2$ (2's compl.)

$+32_{10} = 100000$ → invert: 011111, + 1 → $-32_{10} = 100000$ (2's)

$-14_{10} = 111010$
$+ -32_{10} = 100000$
ignore → (1)011010 ⇒ $011010 = +26_{10}$

$C_{IN} \neq C_{OUT}$ ⇒ OVERFLOW (must be −46)

(b) **[2 marks.]** Consider the number -37_8. Give the 6-bit sign-magnitude and 2's-complement binary representations of this number.

$37_8 = 011\ 111_2 = 2^4+2^3+2^2+2^1+2^0 = 31_{10}$

$-37_8 = -31_{10}$

SIGN-MAGNITUDE: $-31_{10} = 111111_2$ (sign bit: −, magnitude: 31)

2's-complement: $+31 = 011111$ → invert: 100000, + 1 → $-31 = 100001$ 2's compl.

(c) **[2 marks].** Add the decimal numbers 91 and 19 using BCD arithmetic.

$91_{10} = 1001\ 0001_{BCD}$ (9, 1)
$+\ 19_{10} = 0001\ 1001_{BCD}$ (1, 9)

= 1010 1010
+ correction (6) 0110 0110

= 1 0001 0000 BCD (1, 1, 0) $= 110_{10}$

2. **[8 marks total (4 marks each).]** Consider the following two logic expressions:

$$f = (x + y)(x' + z)$$
$$g = xz + x'y$$

Prove or disprove that f and g are equivalent in the following two ways.

(a) By expanding each expression to its canonical SOP form (f and g are equivalent if there canonical forms are identical).

$f = (x+y)(x'+z) = xx' + xz + x'y + yz = xz + x'y + yz =$
$= x(y'+y)z + x'y(z+z') + (x+x')yz = xy'z + xyz +$
$+ x'yz + x'yz' + xyz + x'yz = xy'z + xyz + x'yz + x'yz'$

$g = xz + x'y = x(y'+y)z + x'y(z+z') = xy'z + xyz + x'yz + x'yz'$

Canonical SOP for f and g are equal!

(b) By direct algebraic manipulation *without* using a K-map, a truth table, or your canonical forms from part (a).

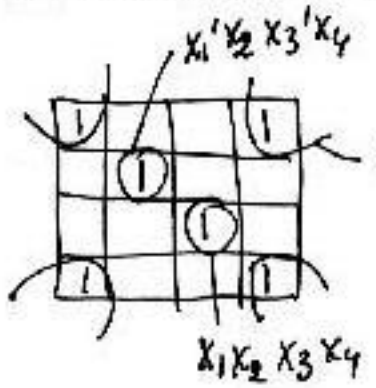
$f = (x+y)(x'+z) = xx' + xz + x'y + yz = xz + x'y + yz =$
$= xz + x'y + (x+x')yz = xz + x'y + xyz + x'yz =$
$= xz(1+y) + x'y(1+z) = xz + x'y$

(1) (1)

3. **[20 marks total.]** A function F of four variables x_1, x_2, x_3 and x_4 is given by the Karnaugh map:

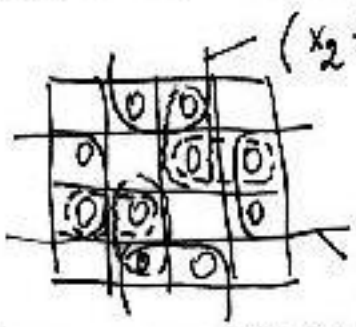
$x_1x_2 \backslash x_3x_4$	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	0	0	1	0
10	1	0	0	1

(a) **[2 marks.]** Find a minimal SOP algebraic expression using this map.



$$f = X_2'X_4' + X_1'X_2X_3'X_4 + X_1X_2X_3X_4$$

(b) **[2 marks.]** Find a minimal POS expression using this map.



$$f = (X_2+X_4')(X_2'+X_4) \times (X_1'+X_2'+X_3)(X_1+X_2'+X_3')$$

or

$$f = (X_2+X_4')(X_2'+X_4) \times (X_1'+X_3+X_4)(X_1+X_3'+X_4')$$

or

$$f = (X_2+X_4')(X_2'+X_4) \times (X_1'+X_2'+X_3)(X_1+X_3'+X_4')$$

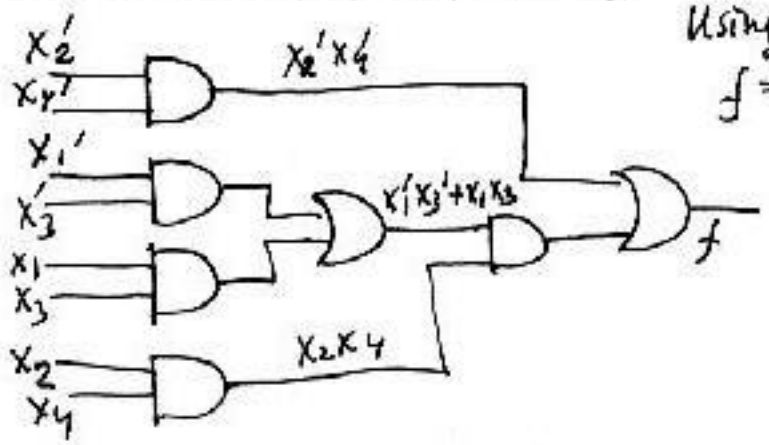
or

$$f = (X_2+X_4')(X_2'+X_4) \times (X_1+X_2'+X_3')(X_1'+X_3+X_4)$$

NOTE: (X_2+X_4'), $(X_2'+X_4)$ are essential prime implicants, $(X_1'+X_2'+X_3)$, $(X_1+X_2'+X_3')$, $(X_1'+X_3+X_4')$, $(X_1+X_3'+X_4')$ are not essential, and any pair of those that will cover central 0's, must be included in the solution.

Note: *For this and all remaining parts of this problem, you may assume that the variables and their complements are available to your circuit.*

(c) **[4 marks.]** Draw the logic diagram of an *AND-OR* circuit implementation corresponding to the simplest (i.e., lowest "cost") of your answers to parts (a) and (b) above. Use only 2-input gates in your drawing.



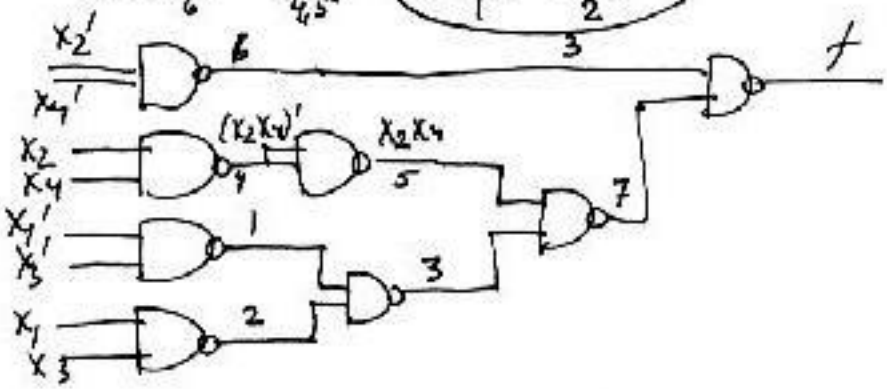
Using SOP,

$f = X_2'X_4' + X_2X_4(X_1'X_3' + X_1X_3)$

(d) **[4 marks.]** Using algebraic transformations and/or circuit manipulation, implement the circuit you drew in part (c) using 2-input NAND gates only (inverters are not available).

$$X_2'X_4' + X_2X_4(X_1'X_3' + X_1X_3) = \left[\left(X_2'X_4'\right)' \left(X_2X_4(X_1'X_3' + X_1X_3)\right)'\right]' =$$

$$= \left[\left(X_2'X_4'\right)' \cdot \left(\left(X_2X_4\right)'' \cdot \left[\left(X_1'X_3'\right)' \left(X_1X_3\right)'\right]'\right)'\right]'$$

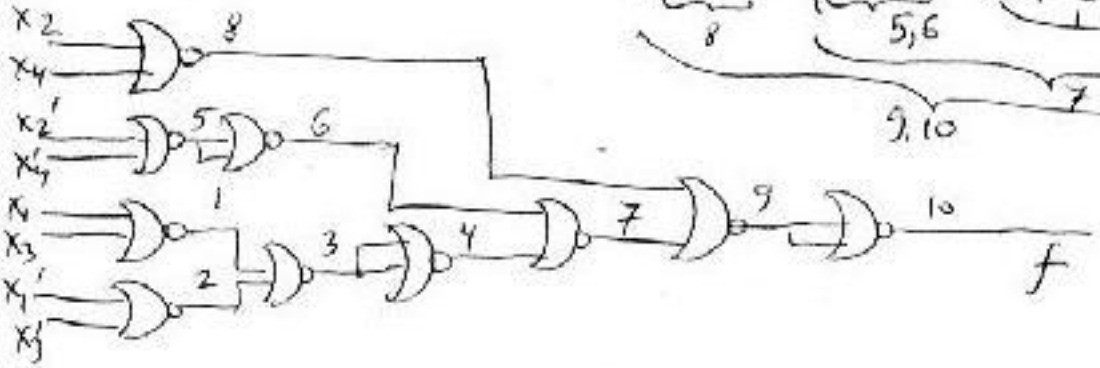


(e) **[4 marks.]** Using algebraic transformations and/or circuit manipulation, implement the circuit you drew in part (c) using 2-input NOR gates only (inverters are not available).

$$f = x_2' x_4' + x_2 x_4 (x_1' x_3' + x_1 x_3) = x_2' x_4' +$$

$$+ x_2 x_4 ((x_1 + x_3)' + (x_1' + x_3')') = (x_2 + x_4)' + (x_2' + x_4')' \times$$

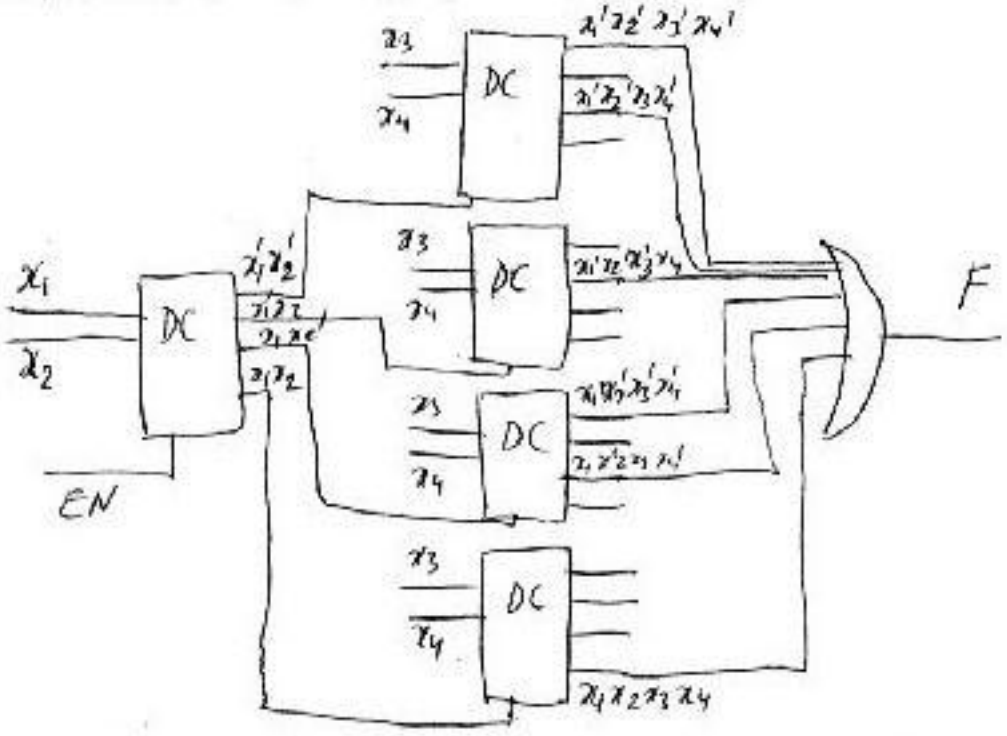
$$\times ((x_1 + x_3)' + (x_1' + x_3')') = [(x_2 + x_4)' + ((x_2' + x_4')'' + ((x_1 + x_3)' + (x_1' + x_3')')')']''$$



(f) **[4 marks.]** Implement the logic function F using 2-to-4 decoders. You may use any additional AND, OR, and NOT gates as necessary.

Canonical SOP: $f = x_1' x_2' x_3' x_4' + x_1' x_2' x_3 x_4' +$

$$+ x_1 x_2' x_3' x_4' + x_1 x_2' x_3 x_4' + x_1' x_2 x_3' x_4 + x_1 x_2 x_3 x_4$$

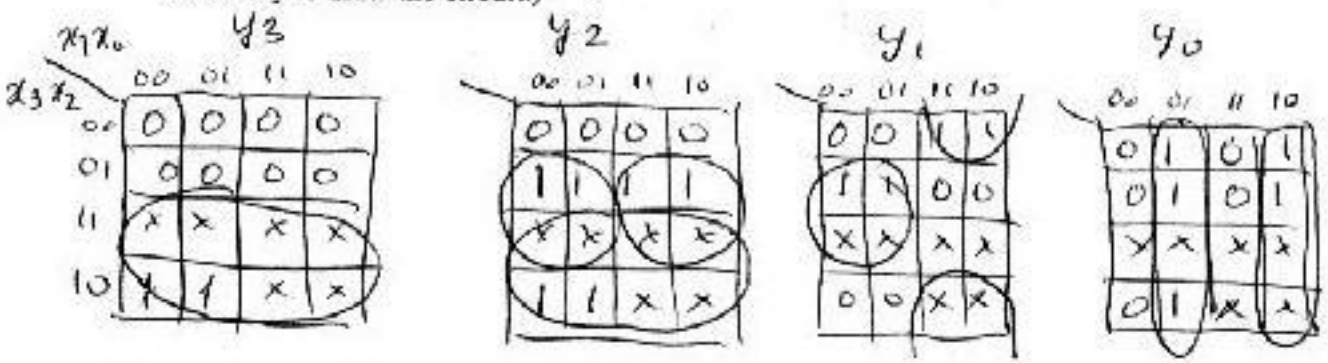


4. **[12 marks total.]** Design a circuit with four inputs x_3, x_2, x_1, x_0 (x_3 is the MSB) and four outputs y_3, y_2, y_1, y_0 (y_3 is the MSB) that implements a BCD-to-Gray code conversion.

   (a) **[4 marks.]** Create the truth table for the circuit's four outputs.

BCD				GRAY			
x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	0	1	0
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

   (b) **[8 marks.]** For your answer to part (a), give minimal SOP expressions for each output suitable for the simplest-possible multiple-output circuit implementation. Carefully indicate shared product terms (if any) that can be used. (It is not necessary to draw the circuit.)



One of possible answers:

$y_3 = x_3$

$y_2 = x_3 + x_2 x_1' + x_2' x_1$

$y_1 = x_2' x_1 + x_2 x_1'$

$y_0 = x_1' x_0 + x_1 x_0'$