Name: Solutions *ID:* _____

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

(a) **[6 marks.]** Represent the following numbers using 7-bit 2's-complement format and perform the addition. Indicate in each case if an overflow occurs, and why. Convert your answer to signed decimal notation.

i. $48_{10} + 23_{10}$

$48 = 32+16$, so $48_{10} = 0110000_2$

$23 = 16+4+2+1$ so $23_{10} = 0010111_2$

$$\begin{array}{r} {\scriptstyle (0\,1)\,1} \\ 0110000 \\ 0010111 \\ \hline 1000111 \end{array}$$

different! so overflow

↑ also negative, so overflow

$1000111 \rightarrow 0111000 + 1 = 0111001$ (57)

result = −57

ii. $14_{10} + (-33_{10})$

$14_{10} = 8+4+2 = 0001110_2$

$33_{10} = 32+1 = 0100001_2 \rightarrow 1011110 + 1 = 1011111$ (−33)

$$\begin{array}{r} {\scriptstyle (0\,0)\,1} \\ 0001110 \\ 1011111 \\ \hline 1101101 \end{array}$$

same! so no overflow

$\rightarrow 0010010 + 1 = 0010011$ (19)

result = −19

(b) **[2 marks.]** Represent each of the following numbers in 7-bit sign-magnitude and 2's-complement binary formats.

i. 33_8

$33_8 = 0011011_2$ same in both representations

ii. $(-3E)_{16}$

$3E_{16} = 0111110_2$

sign-magnitude = 1111110

two's-complement = $1000001 + 1 = 1000010$

(c) **[2 marks].** Add the decimal numbers 96 and 87 using BCD arithmetic.

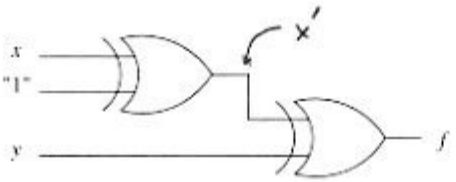
96 = 1001 0110
1000 0111
1 0001 1101
0110 ← correction
1 0010 0011
correction → 0110
1 1000 0011 ← 183_{BCD}

2. **[8 marks total (4 marks each).]** Perform algebraic manipulation for each of the following tasks:

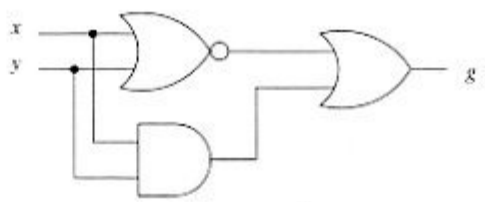
(a) Expand the function $f = a + b + c'$ to its canonical form.

$f = a(b+b')(c+c') + b(a+a')(c+c') + c'(a+a')(b+b')$
$= (ab + ab')(c+c') + (ab + a'b)(c+c') + (ac' + a'c')(b+b')$
$= \underline{abc} + \underline{abc'} + \underline{ab'c} + \underline{ab'c'} + \underline{abc} + \underline{abc'} + \underline{a'bc} + \underline{a'bc'} + \underline{abc'} + \underline{ab'c'} + \underline{a'bc'} + a'b'c'$ (duplicated terms)
$= abc + abc' + ab'c + ab'c' + a'bc + a'bc' + a'b'c'$
$m_7 \quad m_6 \quad m_5 \quad m_4 \quad m_3 \quad m_2 \quad m_0$

(b) Prove or disprove algebraically that the following circuits are functionally equal:



$f = x' \oplus y$
$= x'y' + xy$



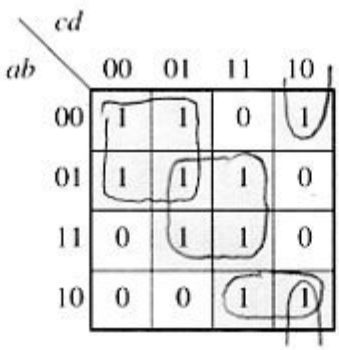
$g = (x+y)' + xy$
$= x'y' + xy$

same!

3. **[6 marks total.]** Karnaugh maps are given identically for a function $F(a, b, c, d)$ in parts (a) and (b) below:

   (a) **[3 marks.]** Find a minimal SOP algebraic expression using this map.

   - If there is more than one minimal sum, give any *one* of them.
   - Give the "cost" associated with this function (give the number of gate-inputs as well as the number of gates, assuming any number of inputs is allowed on each gate).



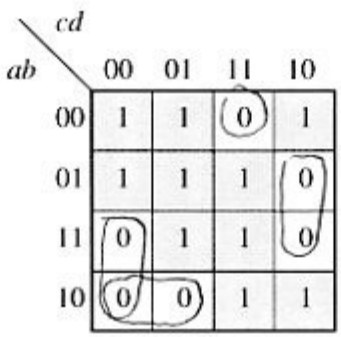
SOP: multiple correct answers

One of them is

$F = a'c' + bd + ab'c + b'cd'$

This corresponds to 14 gate inputs and 5 gates

   (b) **[3 marks.]** Similarly, find any *one* minimal POS expression using this map, and give the "cost" associated with this function.



POS: also multiple correct answers

$F' = ac'd + ab'c' + bcd' + a'b'cd$

So $F = (a'+c+d')(a'+b+c)(b'+c'+d)(a+b+c'+d')$

This corresponds to 17 gate inputs and 5 gates

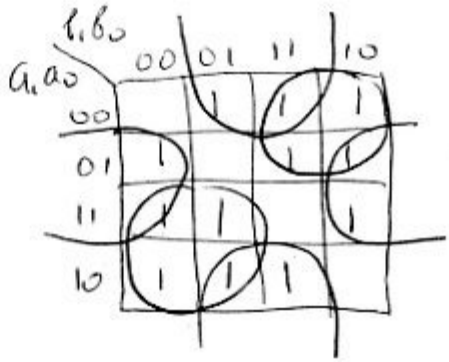
Note: *When drawing circuits in the questions below, you may assume that the inputs are available in both complemented and uncomplemented form.*

---

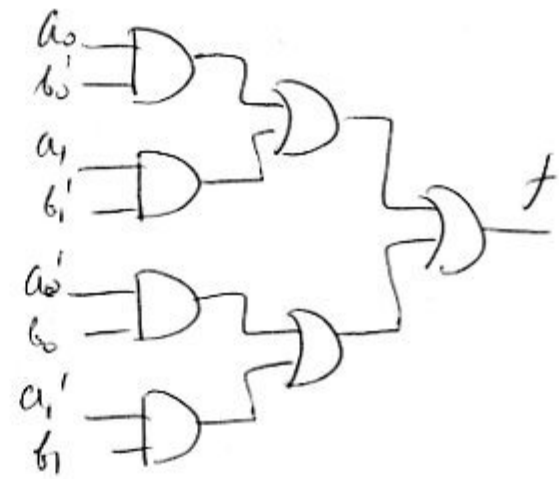
4. **[18 marks total.]** Consider a two-bit magnitude comparator circuit such as the one designed in Lab #2. It has two two-bit inputs $A = a_1a_0$ and $B = b_1b_0$ (where a_1 and b_1 are the most-significant bits). The output f should be $f = 1$ when $A \neq B$; otherwise $f = 0$.

   (a) **[4 marks.]** Find a minimal SOP expression for f. Draw the logic diagram of an *AND-OR* circuit implementation of f using only 2-input gates. Use whatever answer you obtain here to answer parts (b)-(d) below.

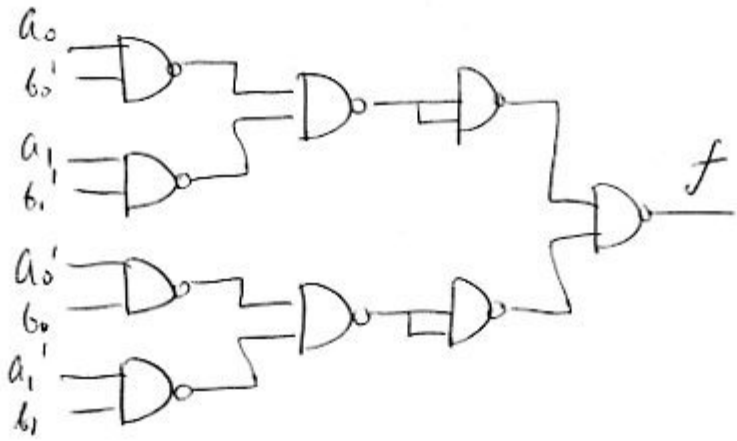
a_1	a_0	b_1	b_0	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0



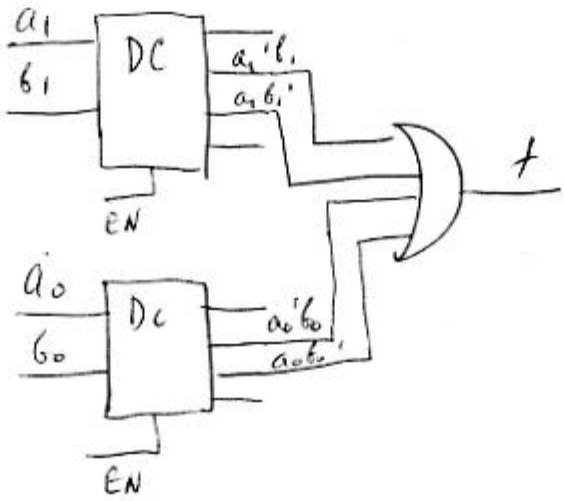
$$f = a_0 b_0' + a_1 b_1' + a_0' b_0 + a_1' b_1$$



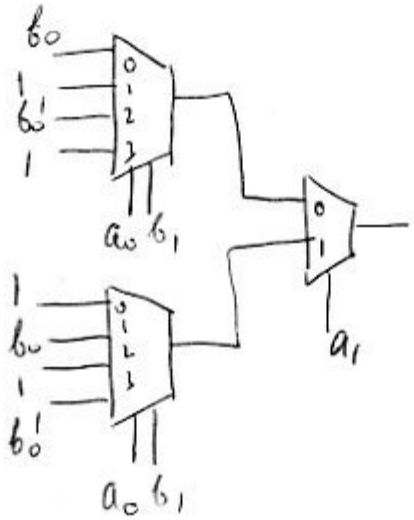
(b) **[4 marks.]** Using algebraic transformations and/or circuit manipulation, implement the circuit you drew in part (a) using 2-input NAND gates only (inverters are not available).



(c) **[4 marks.]** Implement the logic function f using 2-to-4 decoders. You may use any additional AND, OR, and NOT gates as necessary. Each decoder has an ENABLE input.



(d) **[6 marks.]** Implement the logic function f using two 4-to-1 muliplexers and one 2-to-1 multiplexer. You may use any additional AND, OR, and NOT gates as necessary.



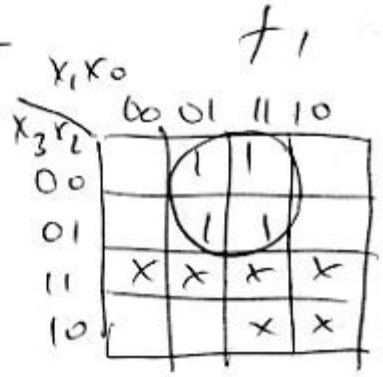
$a_1 a_0 b_1 b_0$	f	
0000	0	$f = b_0$ || $a_1 = 0$, $a_0 b_1 = 00$
0001	1	
0010	1	$f = 1$ || $a_1 = 0$, $a_0 b_1 = 01$
0011		
0100	1	$f = b_0'$
0101	0	
0110	1	$f = 1$
0111	1	
1000	1	$f = 1$
1001	1	
1010	0	$f = b_0$
1011	1	
1100	1	$f = 1$
1101	1	
1110	1	$f = b_0'$
1111	0	

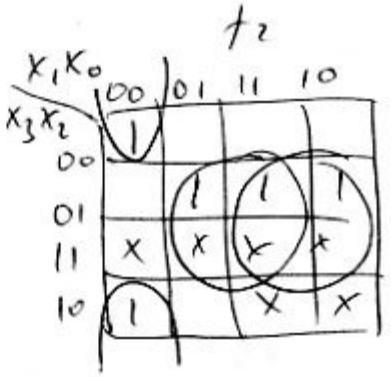
etc.

5. **[8 marks.]** Design a circuit with four inputs x_3, x_2, x_1, x_0 (x_3 is the MSB) and two outputs f_1 and f_2 that implements a BCD code detector. The function f_1 should be $f_1 = 1$ if the BCD code is 1, 3, 5 or 7; otherwise $f_1 = 0$. The function f_2 should be $f_2 = 1$ if the code is 0, 5, 6, 7 or 8; otherwise $f_2 = 0$.

   Give minimal SOP expressions for each output suitable for the simplest-possible multiple-output circuit implementation. Use don't-cares for the invalid BCD codes. Carefully indicate shared product terms (if any) that can be used. (It is not necessary to draw the circuit.)

x_3	x_2	x_1	x_0	f_1	f_2
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	1	0	0
1	0	1	0	X	X
1	0	1	1	X	X
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X





$f_1 = x_3' x_0$

$f_2 = x_2' x_1' x_0' + x_2 x_0 + x_2 x_1$