Name: SOLUTION, *ID:* _____

1. **[8 marks (2 marks each).]** Consider the following 8-bit binary codes stored in a register in your computer's CPU:

   (i) 0001 0101
   (ii) 1001 0111

   (a) Assuming these codes are in sign-magnitude format, give their values in decimal.

   $0001\_0101 = +0001\ 0101 = +(2^4+2^2+2^0) = +21$

   $1001\_0111 = -0001\ 0111 = -(2^4+2^2+2^1+2^0) = -23$

   (b) Assuming these are unsigned binary codes, convert them to unsigned octal numbers.

   $0001\ 0101_2 = 25_8$

   $1001\ 0111_2 = 227_8$

   (c) Assuming these are unsigned binary codes, convert them to unsigned decimal numbers.

   $0001\ 0101 = 2^4+2^2+2^0 = 21$

   $1001\ 0111 = 2^7+2^4+2^2+2^1+2^0 = 128+16+4+2+1 = 151$

   (d) For your answers to part (c), find the four-digit 10's-complement of each.

   $0001\ 0101 \Rightarrow 10^4 - 21 = 9979_{10's}$

   $1001\ 0111 \Rightarrow 10^4 - 151 = 9849_{10's}$

2. **[4 marks.]** Assume that the binary codes given in Problem 1 are now in 8-bit 2's-complement binary format. Perform 8-bit binary addition of these numbers, and express your answer in decimal. Explain whether or not overflow occurs.

```
  0001 0101 = 21
+ 1001 0111 = -105
-----------
  1010 1100 = -84
```

```
  1001 0111 (2's)
     ⇓
  0110 1000
+         1
-----------
  0110 1001 = 105
     ⇓
  1001 0111 ⇒ -105

  1010 1100 (2's)
     ⇓
  0101 0011
+         1
-----------
  0101 0100 = 84
     ⇓
  1010 1100 = -84
```

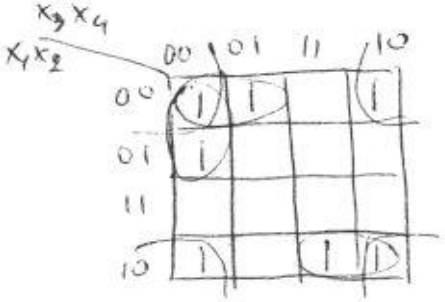
3. **[4 marks.]** Assume that the binary codes given in Problem 1 are now in BCD format. Perform BCD addition of these numbers.

```
            0001 0101 = 1 5 (BCD)
+           1001 0111 = 9 7 (BCD)
            ---------
            1010 1100 ⇒ NOT A BCD ⇒ must add +6 for correction
+ correct   0110 0110
            ---------
          1 0001 0010 = 1 1 2 (BCD)
```

4. **[18 marks total.]** A Boolean function F of four variables x_1, x_2, x_3 and x_4, is given by the Karnaugh map:

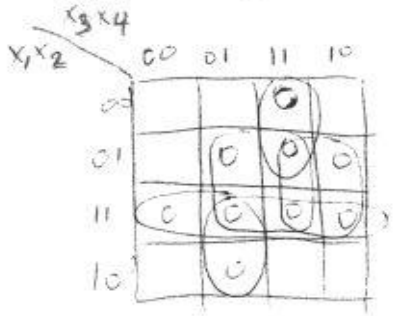
$x_1x_2 \backslash x_3x_4$				
	1	1	0	1
	1	0	0	0
	0	0	0	0
	1	0	1	1

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



$SOP = X_2' X_4' + X_1' X_3' X_4' + X_1' X_2' X_3' + X_1 X_2' X_3$

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

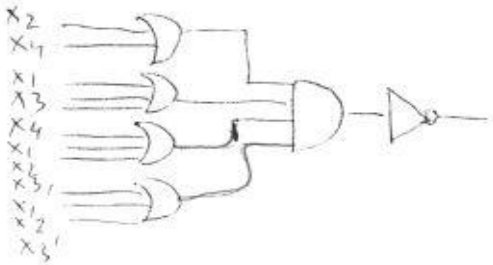


$POS = (X_2' + X_4')(X_2' + X_3')$
$\cdot (X_1' + X_2')(X_1 + X_3' + X_4')(X_1' + X_3 + X_4')$

(c) **[2 marks.]** Draw the logic diagram of an *OR-AND-INVERT* circuit implementation. OR and AND gates with three or more inputs are allowed. Inverters are available. *(Note: for this and all remaining parts of this problem, you may assume that the variables and their complements are available to your circuit.)*

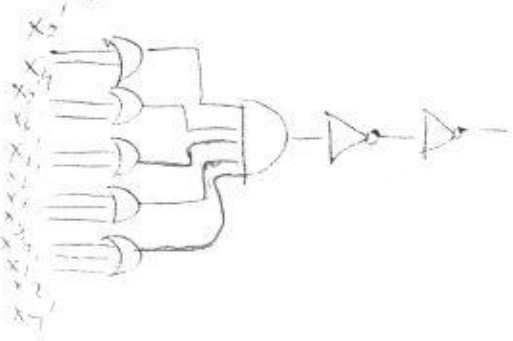
SOLUTION 1:

$$X_2' X_4' + X_1' X_3' X_4' + X_1' X_2' X_3' + X_1 X_2' X_3 = [(X_2 + X_4)(X_1 + X_3 + X_4) \cdot (X_1 + X_2 + X_3)(X_1' + X_2 + X_3')]'$$



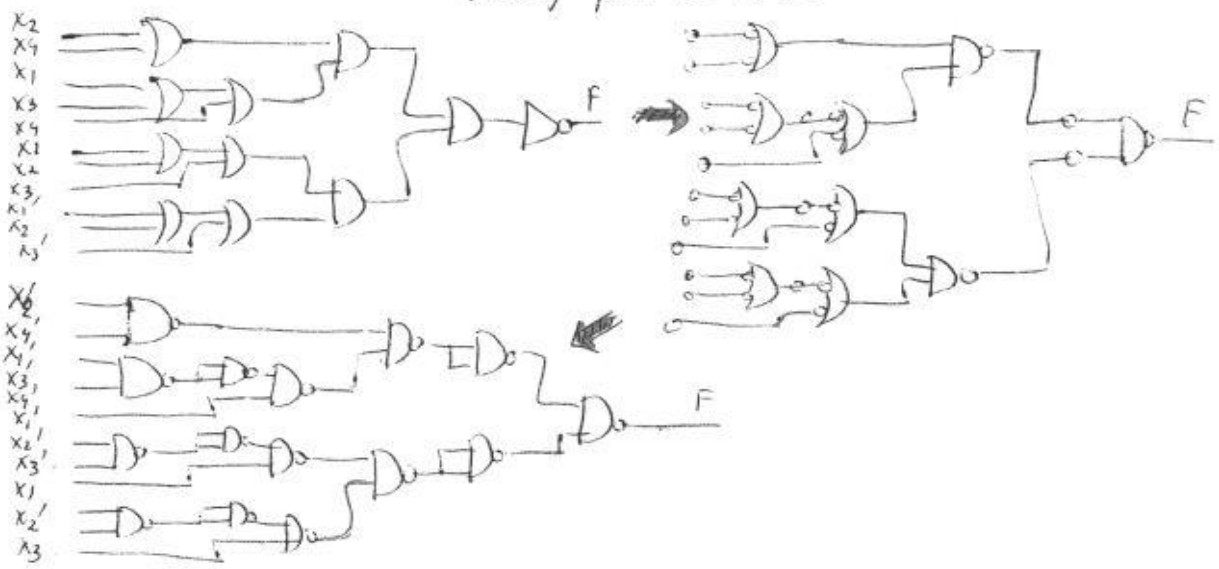
SOLUTION 2:

$$(X_2' + X_4')(X_2' + X_3')(X_1' + X_2') \cdot (X_1 + X_3' + X_4')(X_1' + X_3 + X_4')$$



(d) **[4 marks.]** Using logic-symbol transformations and circuit manipulation, implement whatever circuit you give in part (c) using 2-input NAND gates only. Inverters are *not* available.

one of possible solutions:



(e) **[5 marks.]** Implement the logic function F using a 4-to-1 multiplexer. You may use any AND, OR, and NOT gates as necessary. *(Hint: you may use either truth table decomposition or Shannon's expansion theorem.)*

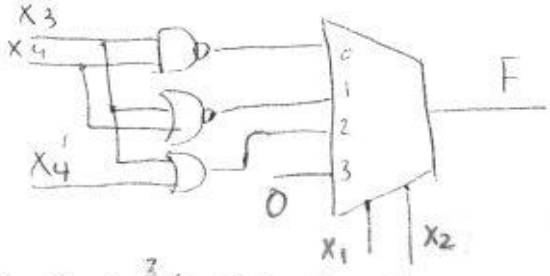
X_1	X_2	X_3	X_4	F	
0	0	0	0	1	$(X_3X_4)' = X_3' + X_4'$
0	0	0	1	1	
0	0	1	0	1	
0	0	1	1	0	
0	1	0	0	1	$(X_3 + X_4)' = X_3'X_4'$
0	1	0	1	0	
0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	1	$X_3 + X_4'$
1	0	0	1	0	
1	0	1	0	1	
1	0	1	1	1	
1	1	0	0	0	0
1	1	0	1	0	
1	1	1	0	0	
1	1	1	1	0	

We've chosen X_1, X_2 as select inputs

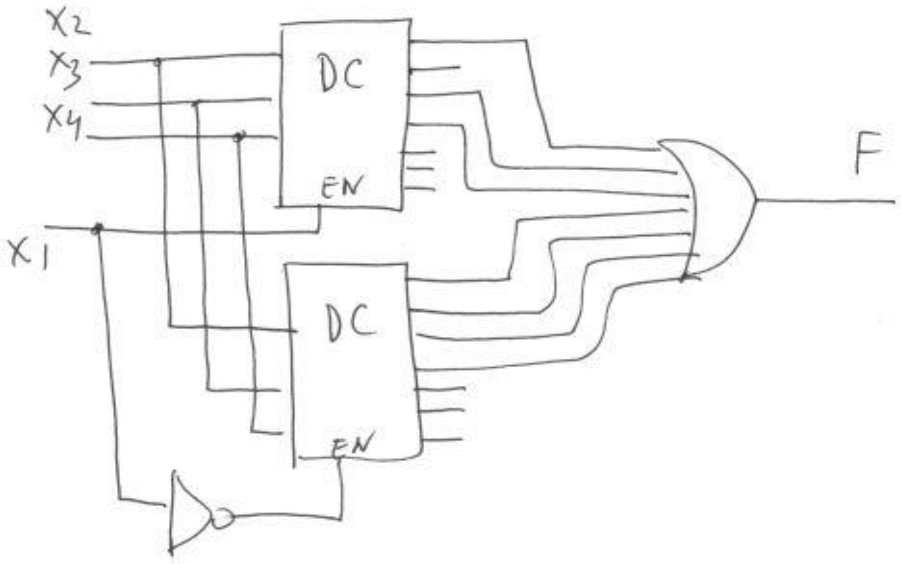
$$F = X_2'X_4' + X_1'X_3'X_4' + X_1'X_2'X_3' + X_1X_2'X_3 = X_1'X_2' f_{\substack{X_1=0\\X_2=0}} + X_1'X_2 f_{\substack{X_1=0\\X_2=1}} + X_1X_2' f_{\substack{X_1=1\\X_2=0}} + X_1X_2 f_{\substack{X_1=1\\X_2=1}} =$$

$$= X_1'X_2'(X_4' + X_3'X_4' + X_3') + X_1'X_2(X_3'X_4') + X_1X_2'(X_4' + X_3) + X_1X_2(0)$$

($X_3' + X_4'$)



(f) **[3 marks.]** Implement the logic function F using 3-to-8 decoders. You may use any AND, OR, and NOT gates as necessary.



5. **[6 marks.]** Design a circuit with four inputs x_1, x_2, x_3, x_4 and a single output F. The output should be $F = 1$ in either of the following cases:

   - $x_1 = 0$ and we have an *odd* function of x_2, x_3, x_4;
   - $x_2 = 1$ and we have an *even* function of x_1, x_3, x_4.

   The only logic gates available are AND and XOR gates. They may have three or more inputs. Inverters are *not* available.