1. Convert the following hexadecimal numbers to their decimal and binary equivalents:

(a) $(13AF)_{16}$ (b) $(25E6)_{16}$ (c) $(B4.C9)_{16}$ (d) $(45)_{16}$

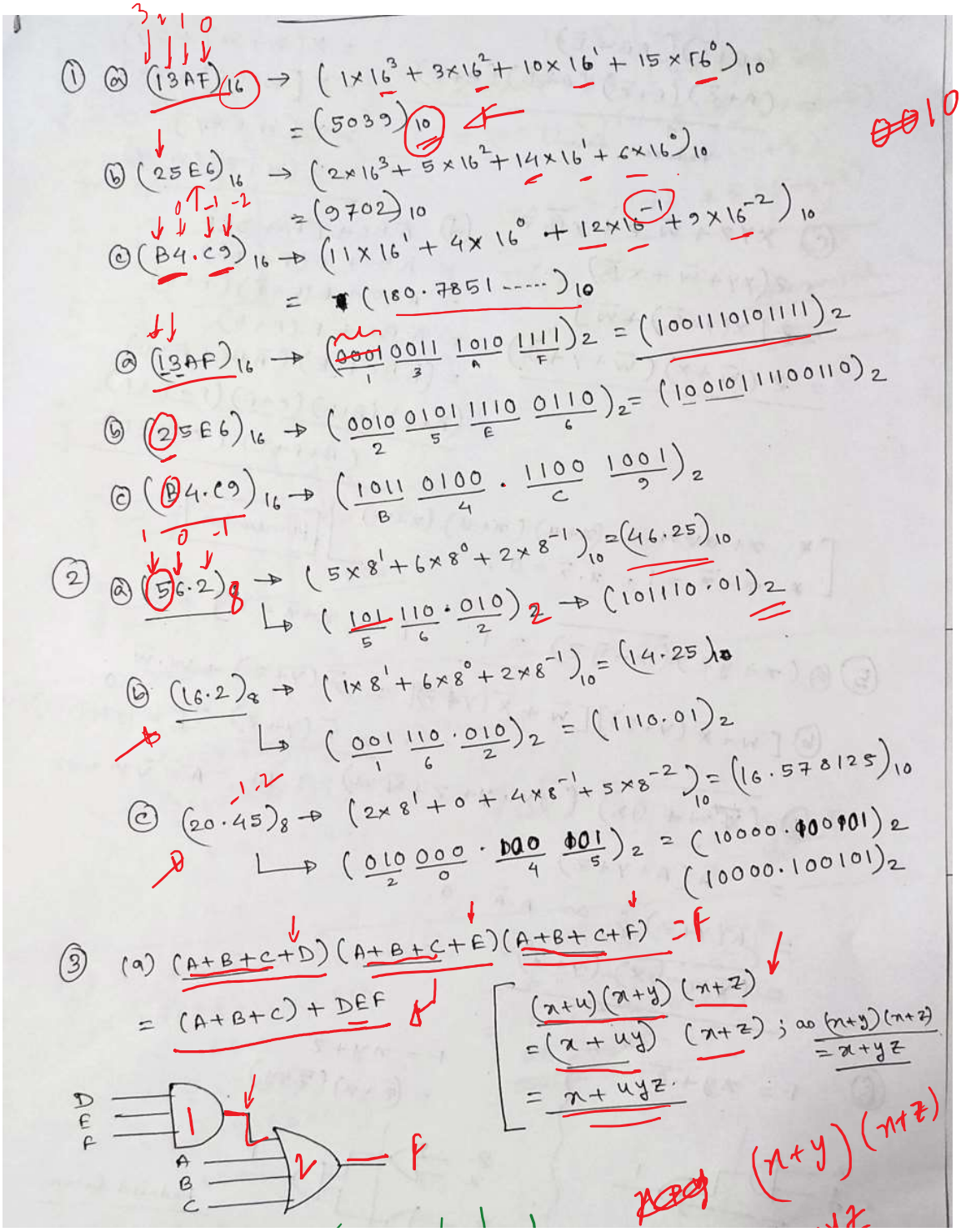
2. Convert the following octal number to their decimal and binary equivalents:

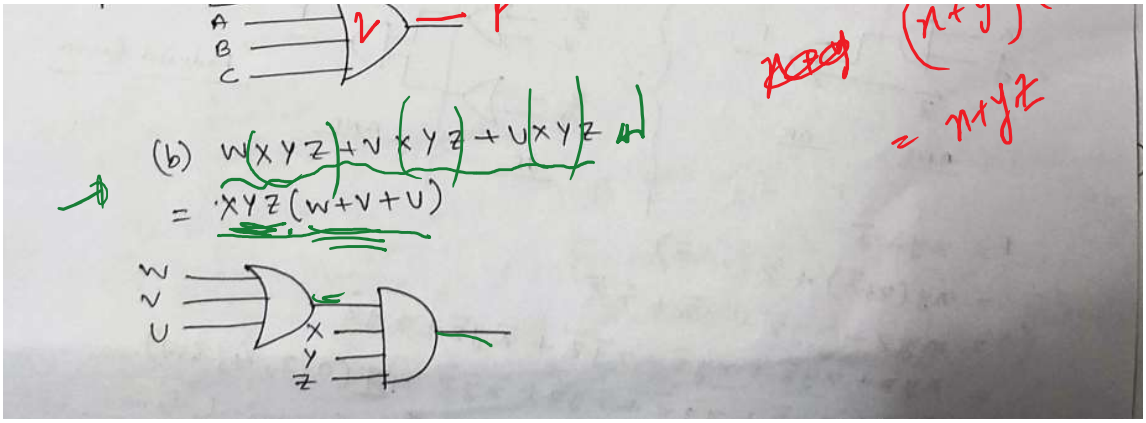
(a) $(56.2)_8$ (b) $(16.2)_8$ (c) $(20.45)_8$

3. Draw a combinational logic circuit that uses only one AND gate and one OR gate to realize each of the following functions:

(a) (A + B + C + D) (A + B + C + E) (A + B + C + F)

(b) WXYZ + VXYZ + UXYZ





4. Factor each of the following expressions to obtain a product of sums:

(a) AB + C'D'

(b) WX + WY'X + ZYX

(c) A'BC + EF + DEF'

(d) XYZ + W'Z + XQ'Z

5. Simplify each of the following expressions by applying one of the theorems.

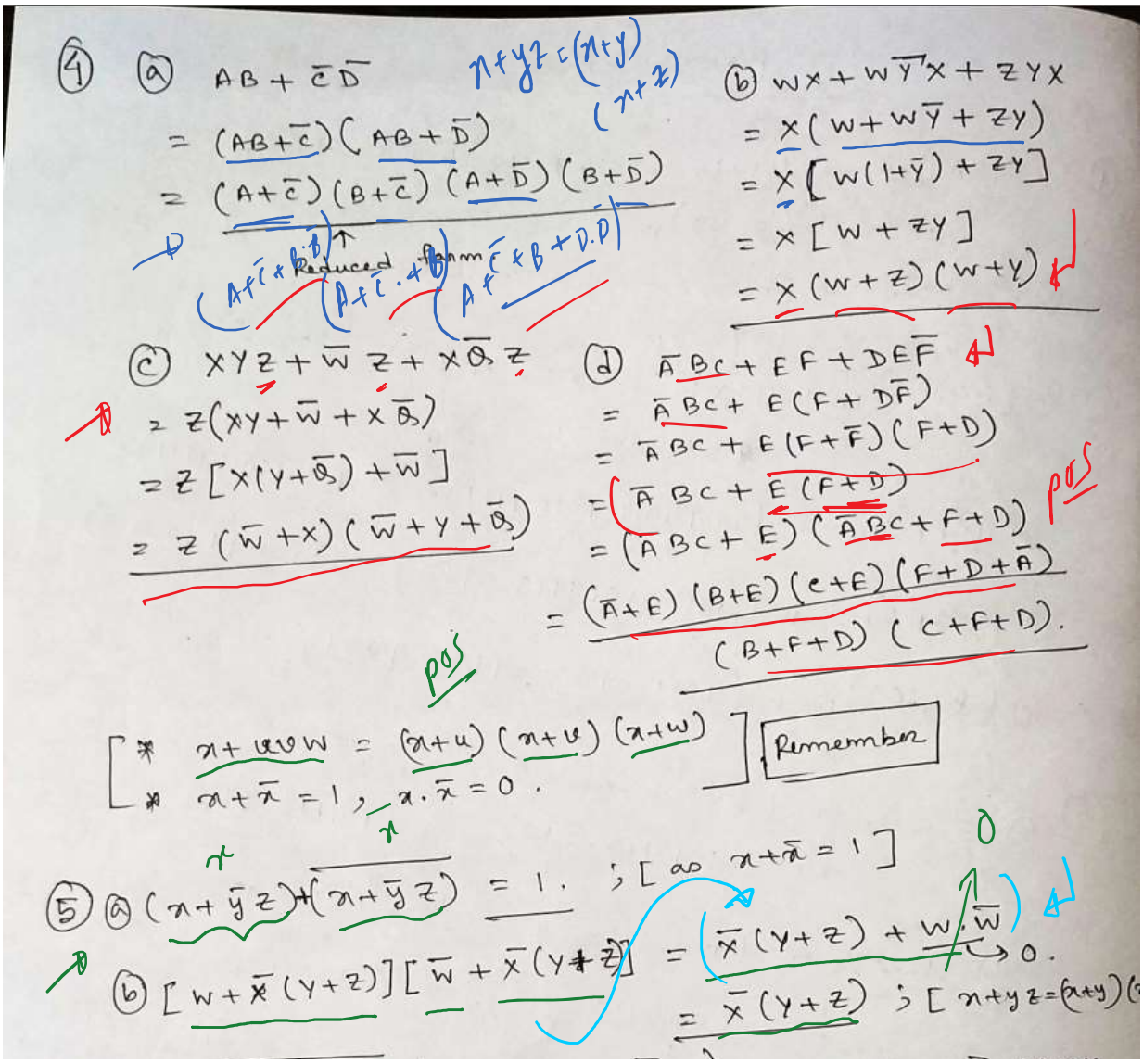
(a) (X + Y'Z) + (X + Y'Z)'

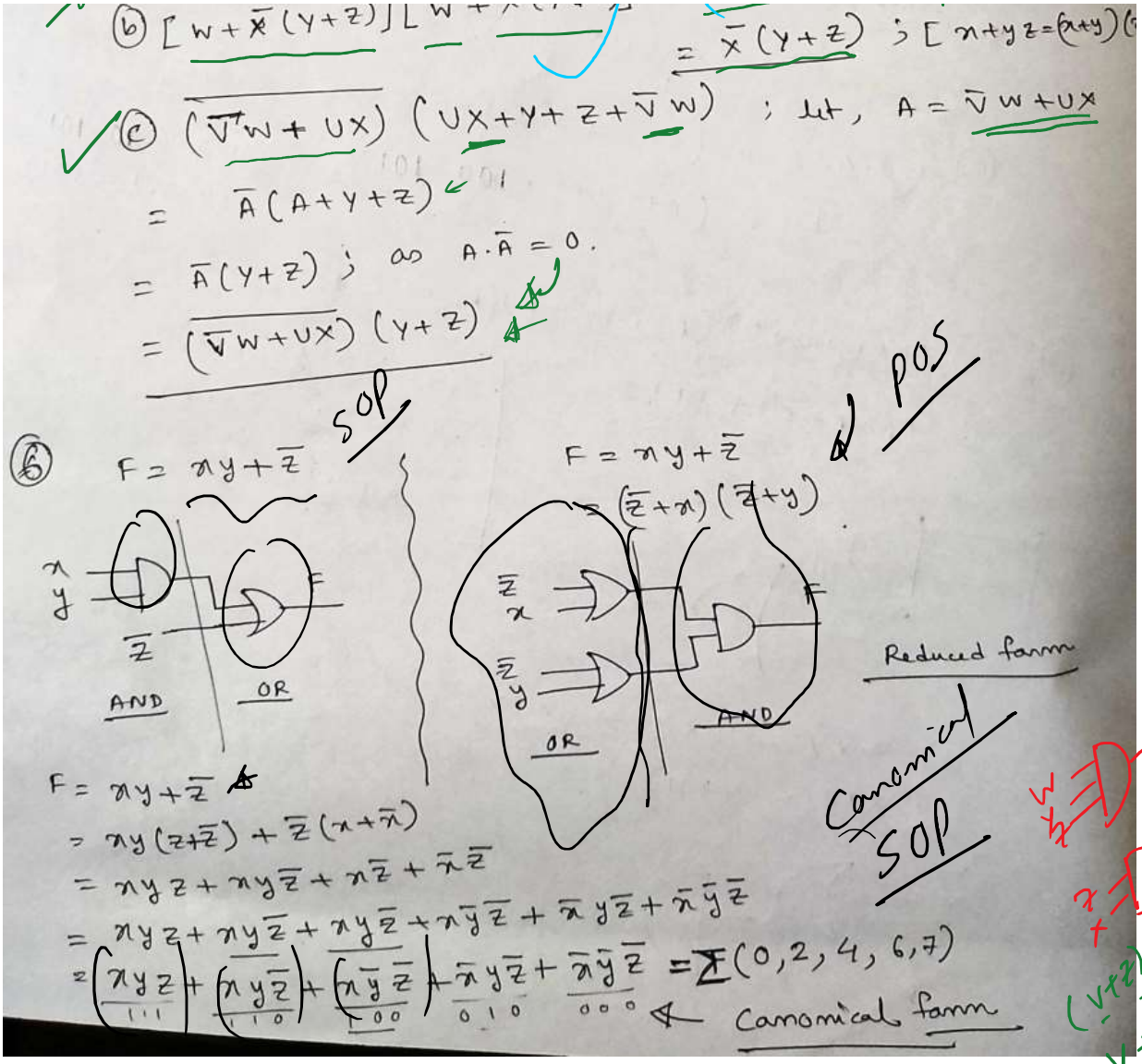
(b) [W + X'(Y + Z)][W' + X'(Y + Z)]

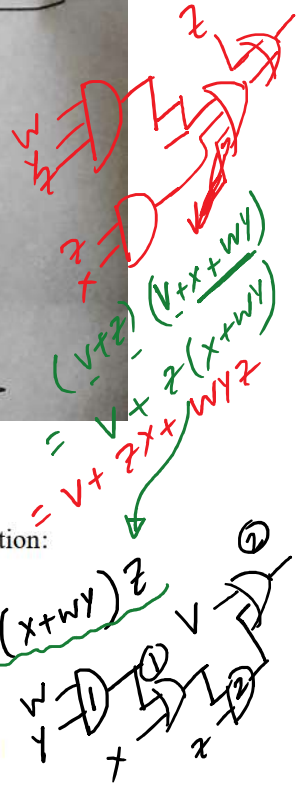
(c) (V'W + UX)'(UX + Y + Z + V'W)

6. Express the following function in POS (Product of Sums) and SOP (Sum of Products) forms:

F = xy + z'







2-OR 2-AND

7. Draw a circuit that uses two OR gates and two AND gates to realize the following function:

F = (V + W + X) (V + X + Y) (V + Z)

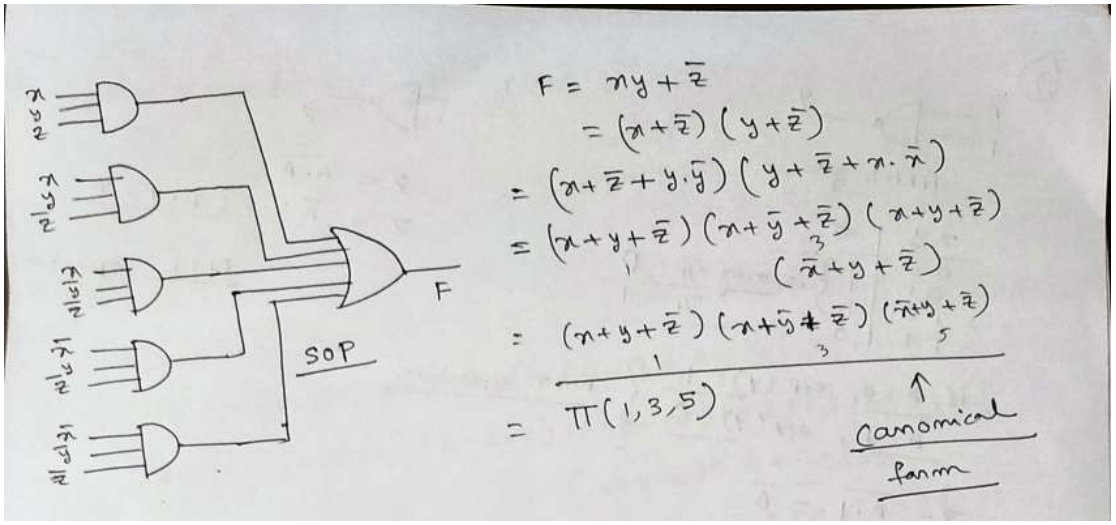
8. Simplify the following expressions:

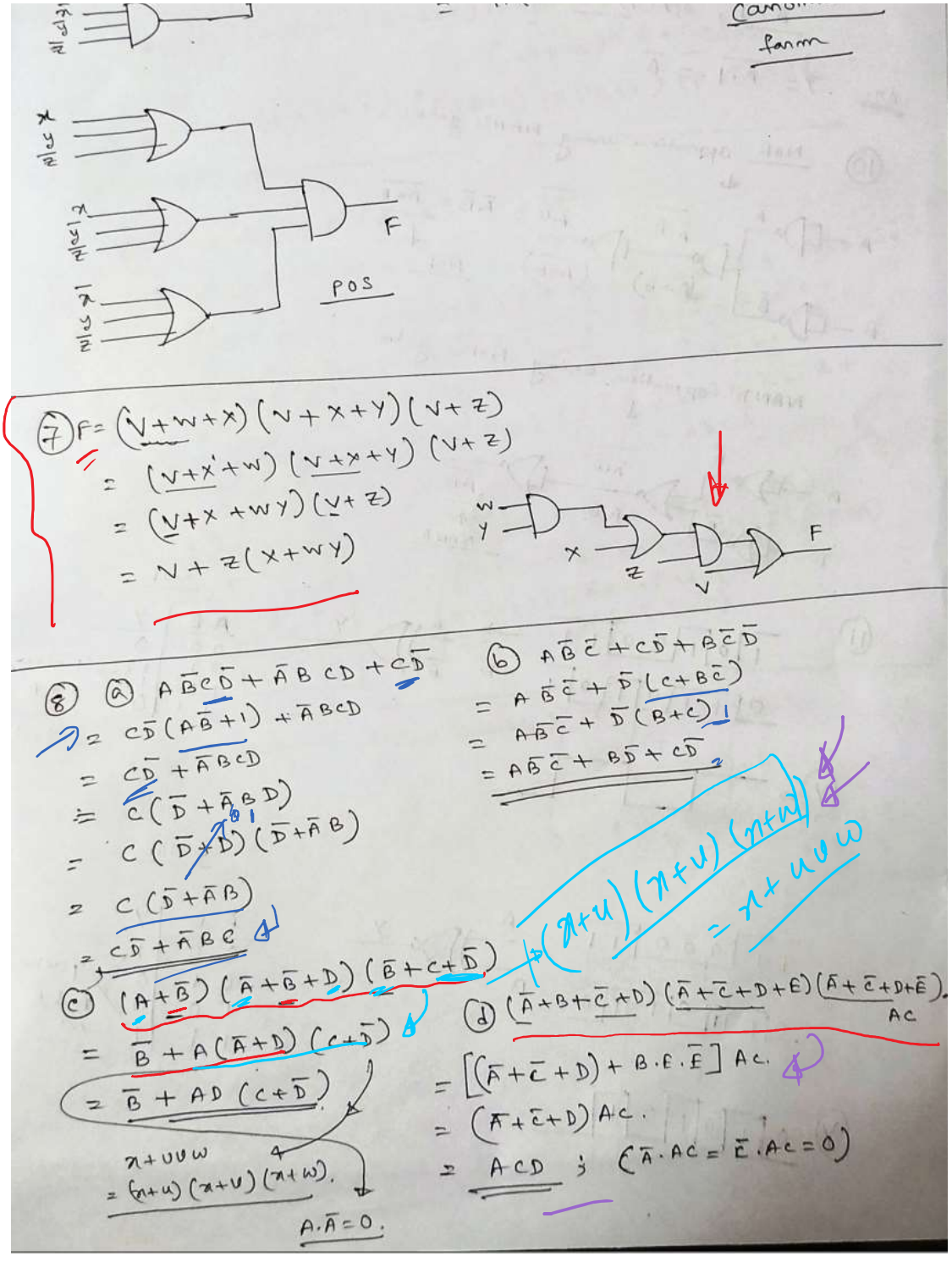
(a) AB′CD′ + A′BCD+CD′

(b) AB′C′+CD′+BC′D′

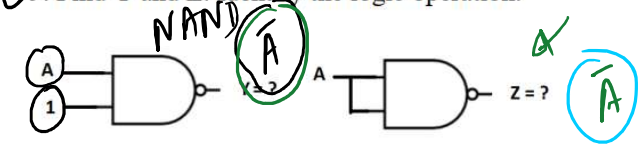
(c) (A+B′) (A′+B′+D) (B′+C+D′)

(d) (A′+B+C′+D) (A′+C′+D+E)(A′+C′+D+E′)

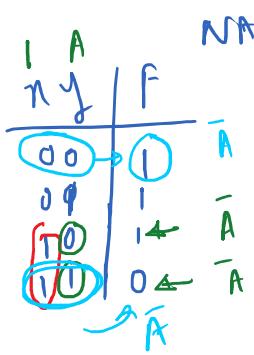




9. Find Y and Z. Identify the logic operation.



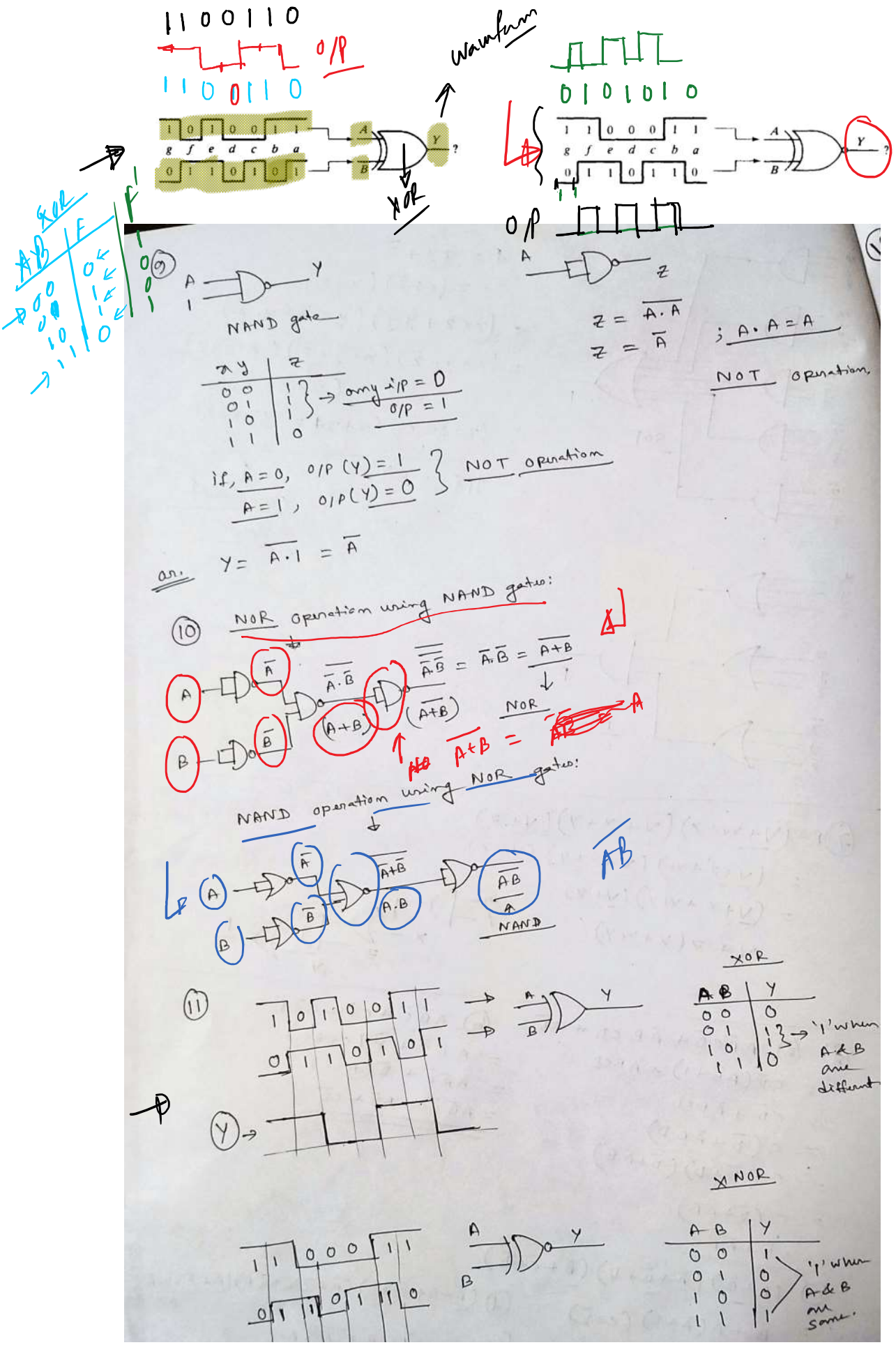
A·A = Ā

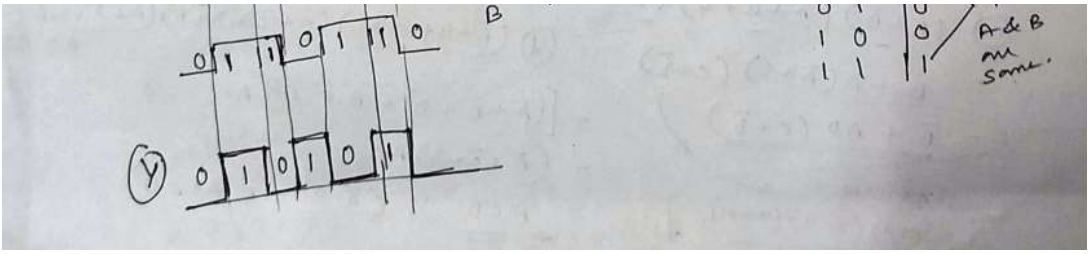


10. Implement NOR operation using only NAND gates.

Implement NAND operation using only NOR gates.

11. Draw the output waveform Y.


1 1 0 0 1 1 0
o/p
1 1 0 0 1 1 0
Waveform
0 1 0 1 0 1 0
XOR
A B | F
NAND gate
x y | z
any i/p = 0
o/p = 1
if, A = 0, o/p (Y) = 1
A = 1, o/p (Y) = 0
NOT operation
Z = A . A
; A . A = A
NOT operation
or.
Y = A . 1 = A
(10)
NOR operation using NAND gates:
NAND operation using NOR gates:
NAND
(11)
XOR
A B | Y
'1' when A & B are different
XNOR
A B | Y
'1' when A & B are same.



BCD to Excess-3 converter:

Decimal	BCD (A B C D)	Excess-3 code (W X Y Z)
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100
	1010	XXXX
	1011	
	1100	
	1101	
	1110	
	1111	

BCD → Binary coded digit.

Don't care combination

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

W = f(A, B, C, D)

W =

X = f(A, B, C, D)

Y =

Z =