$x \oplus y = xy' + x'y$ $\quad ad' = x \quad y = b'$ $\quad ad'(b')' + (ad')'b' = ad'b + (a'+d)b'$

Problem 1 (20 points)

1. (8 points) Using algebraic identities obtain a simplified sum of product for the following switching expression:
$E_1(a,b,c,d) = (ad' \oplus b')(c+d) + (a'+bc)'cd'$
Show each step of your work on a separate line and indicate which identity was used.

$(ad'b + (a'+d)b')(c+d) + (a'+bc)'cd'$ expression for XOR (and De Morgans)

$(ad'b + a'b' + db')(c+d) + (a\cdot(b'+c'))cd'$ De Morgans and distributivity

$(ad'bc + a'b'c + db'c + ad'db + a'b'd + ddb') + (ab' + ac')cd'$ distributivity

$(ad'bc + a'b'c + db'c + a'b'd + db') + (ab'cd' + ac'cd')$ distributivity

$= ad'bc + a'b'c + db'c + a'b'd + db' + ab'cd'$ commutativity

$= db'(c+1) + ad'bc + a'b'c + a'b'd + ab'cd'$

$= db'(1+a') + ad'bc + a'b'c + ab'cd'$

6

$= db' + ad'bc + a'b'c + ab'cd' = b'c(a' + ad') + db' + ad'bc$

$= b'c(a'+d') + db' + abcd' = a'b'c + b'cd' + b'd + abcd'$

$= b'cd'(a+1) + a'b'c + b'd$

$= b'cd' + a'b'c + b'd$ $= b'(cd'+d) + a'b'c$ $= b'c + b'd + a'b'c$

1.	$a+b$	$= b+a$	ab	$= ba$	Commutativity
2.	$a+(bc)$	$= (a+b)(a+c)$	$a(b+c)$	$= (ab)+(ac)$	Distributivity
3.	$a+(b+c)$	$= (a+b)+c$	$a(bc)$	$= (ab)c$	Associativity
		$= a+b+c$		$= abc$	
4.	$a+a$	$= a$	aa	$= a$	Idempotency
5.	$a+a'$	$= 1$	aa'	$= 0$	Complement
6.	$1+a$	$= 1$	$0a$	$= 0$	
7.	$0+a$	$= a$	$1a$	$= a$	Identity
8.	$(a')'$	$= a$			Involution
9.	$a+ab$	$= a$	$a(a+b)$	$= a$	Absorption
10.	$a+a'b$	$= a+b$	$a(a'+b)$	$= ab$	Simplification
11.	$(a+b)'$	$= a'b'$	$(ab)'$	$= a'+b'$	DeMorgan's Law

2. (4 points) Using a K-map, obtain minimal sum of products and product of sums. Compare the minimal SOP with the SOP in (1).

d (over columns 2–3), b (beside rows 2–3), a (beside rows 3–4), c (under columns 3–4)

0	1	1	1
0	0	0	0
0	0	1	1
0	1	1	1

$bcd' + a'bc + b'd$

$ad'bc + a'b'c + db'c$
$+ a'b'd + db' + ab'cd'$

2

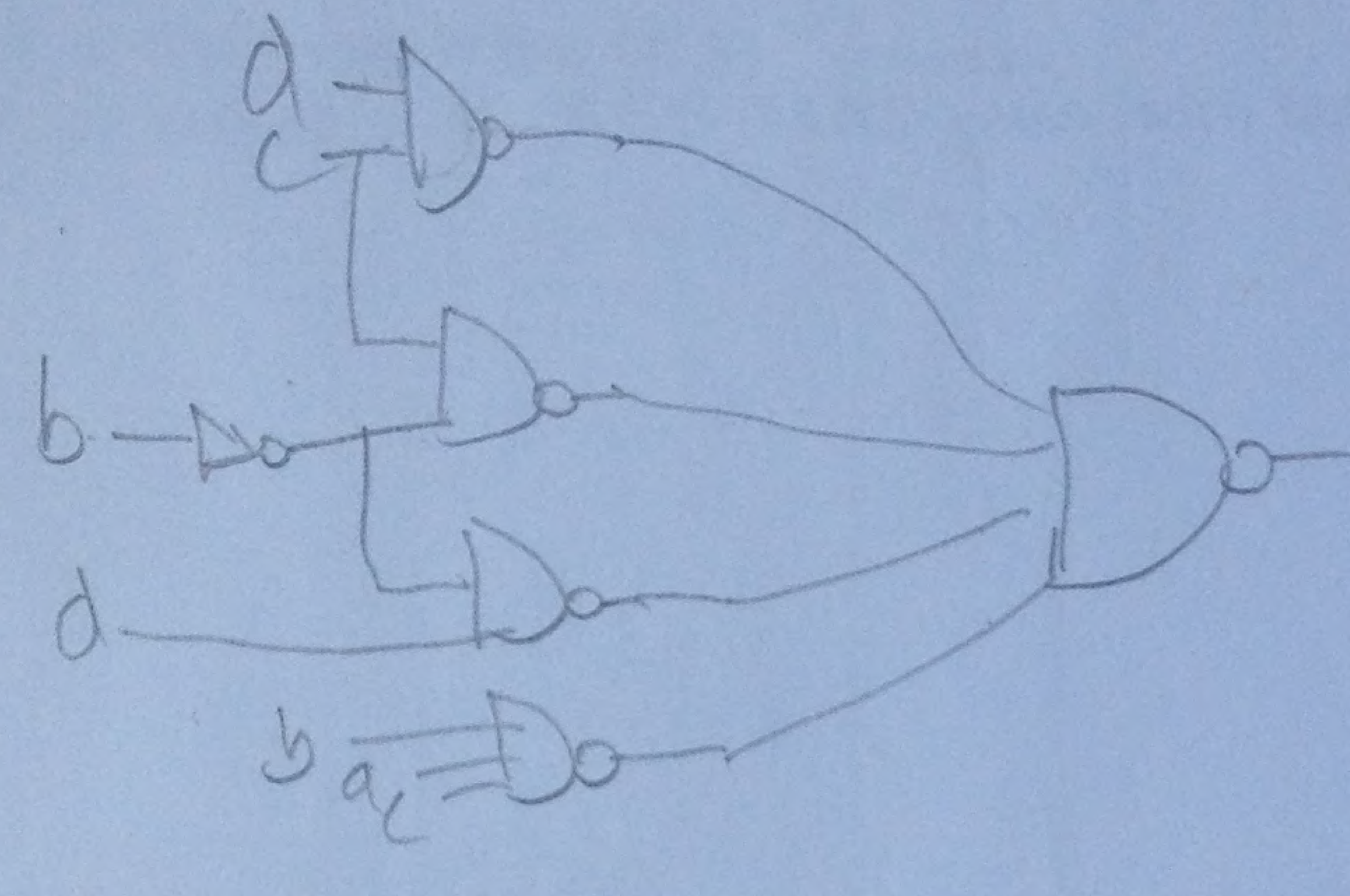
SOP: $dc + cb' + db' + bac$

POS: $(c+d)(b'+c)(b'+a+d)$

✓ X X

3. (8 points) Show implementation of min SOP and min POS expressions using NAND and NOR gates. Inverted inputs are not available, and no constant inputs are allowed. Compare the two networks with respect to the number of gates and the total number of inputs. (You are allowed to use NOT gates.)

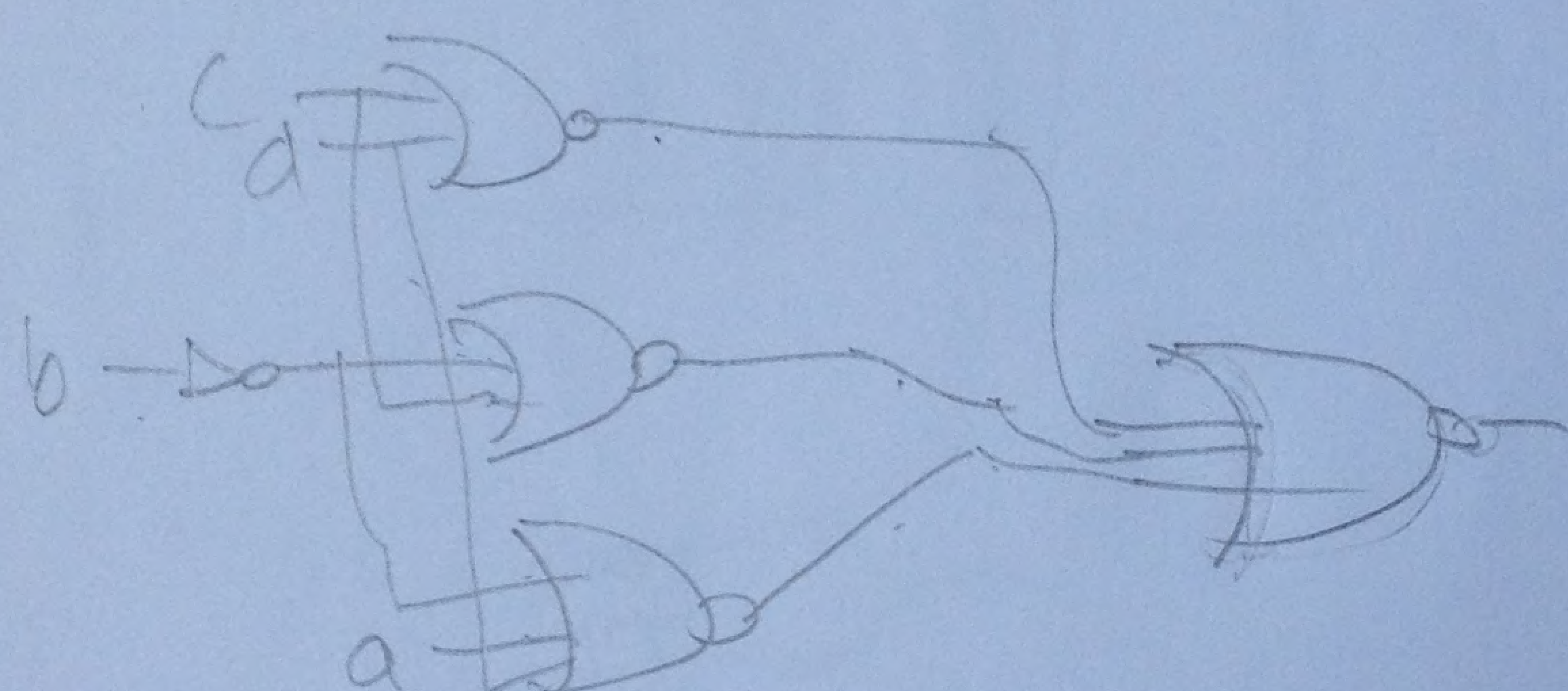
SOP!



6 gates 13 inputs

4

POS



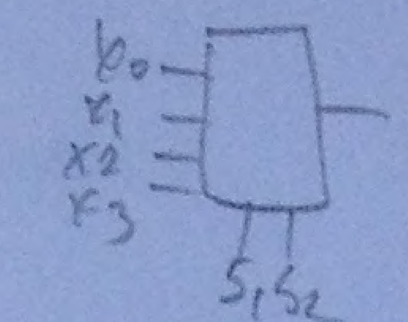
4 gates, 10 inputs

POS is better

Problem 2 (15 points) (17)

We want to design a gate network to implement a 4-input multiplexer module MUX. This module has four data inputs $\underline{x} = (x_3, x_2, x_1, x_0)$, two select inputs $\underline{s} = (s_1, s_0)$ and the output y, all in binary code. The output is connected to one of the data inputs determined by the select inputs. Formally, the MUX function is specified as

$$y = MUX(\underline{x}, \underline{s}) = x_i \; if \; s = i, \; i = 0, 1, 2, 3. \text{ For example, if } s = 2, \; y = x_2.$$



1. Show a sum of products expression for y. 4 transistors
2. Implement MUX module using CMOS NAND gates (with fanin as needed) and NOT gates. How many transistors are used? 2 transistors

1,

x_3	x_2	x_1	x_0	s_1	s_0	y
0	0	0	0	0	0	x_0
0	0	0	1	0	1	x_1
0	0	1	0	1	0	x_2
0	0	1	1	1	1	x_3
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

s=0 (00) y = x_0

s=1 (01) y = x_1

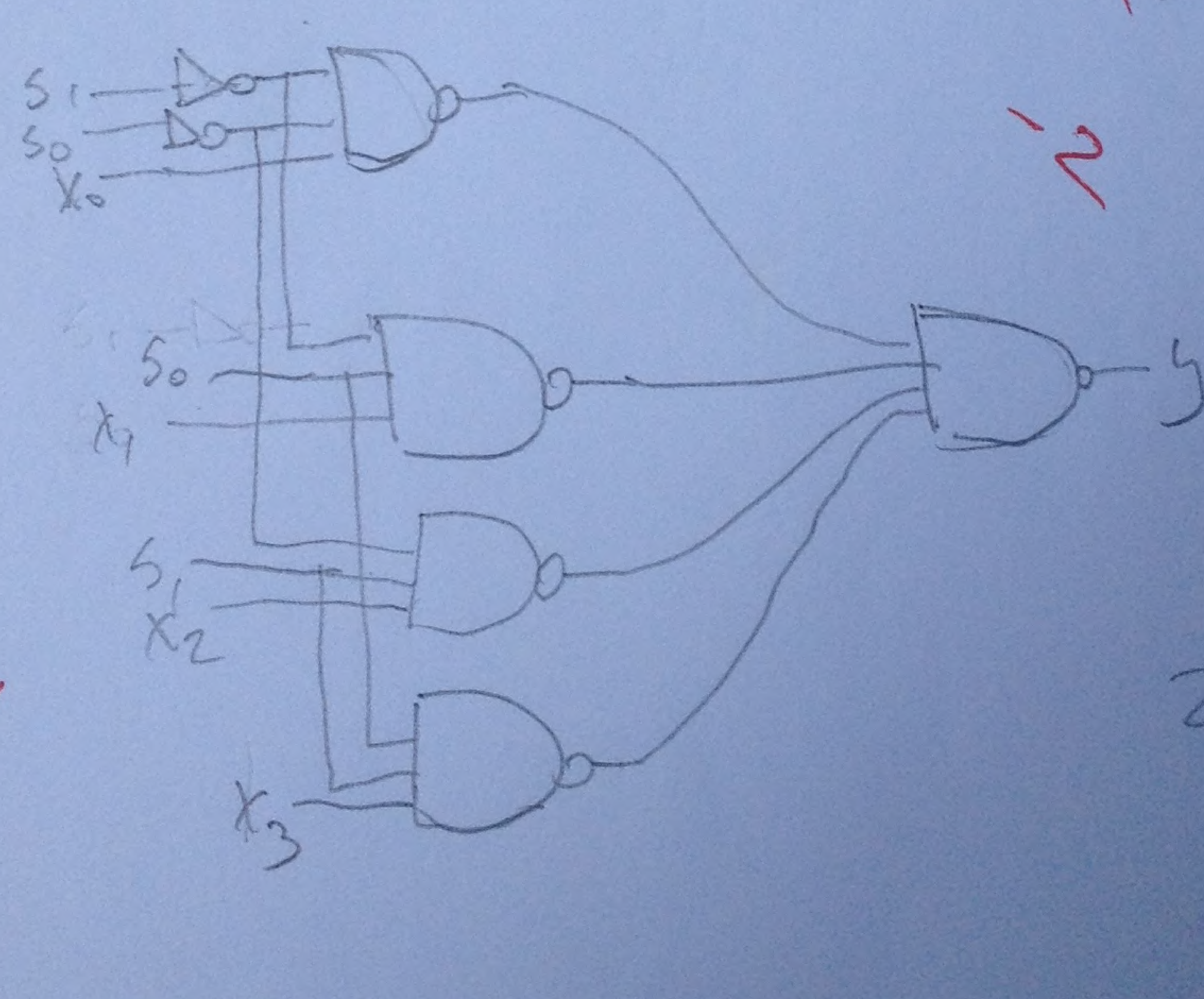
s=2 (10) y = x_2

y=3 (11) y = x_3

POS: $s_1' s_0' x_0 + s_1' s_0 x_1 + s_1 s_0' x_2 + s_1 s_2 x_3$ ✓

4 5 nand gates = 20 transistors

-2 2 Not gates = transistors



24 transistors needed

+4

Optional problem. (10 extra points) Implement MUX module using CMOS transmission gates TG, NOR and NOT gates. A transmission gate TG_i is controlled by signal C_i:

C_i	TG_i
0	*on*
1	*off*

Complete the following table defining the values of control variables C_i and the output y:

s_1	s_0	C_0	C_1	C_2	C_3	y
0	0	1	0	0	0	x_0
0	1	0	1	0	0	x_1
1	0	0	0	1	0	x_2
1	1	0	0	0	1	x_3

+1

Show switching expressions for C_i's.

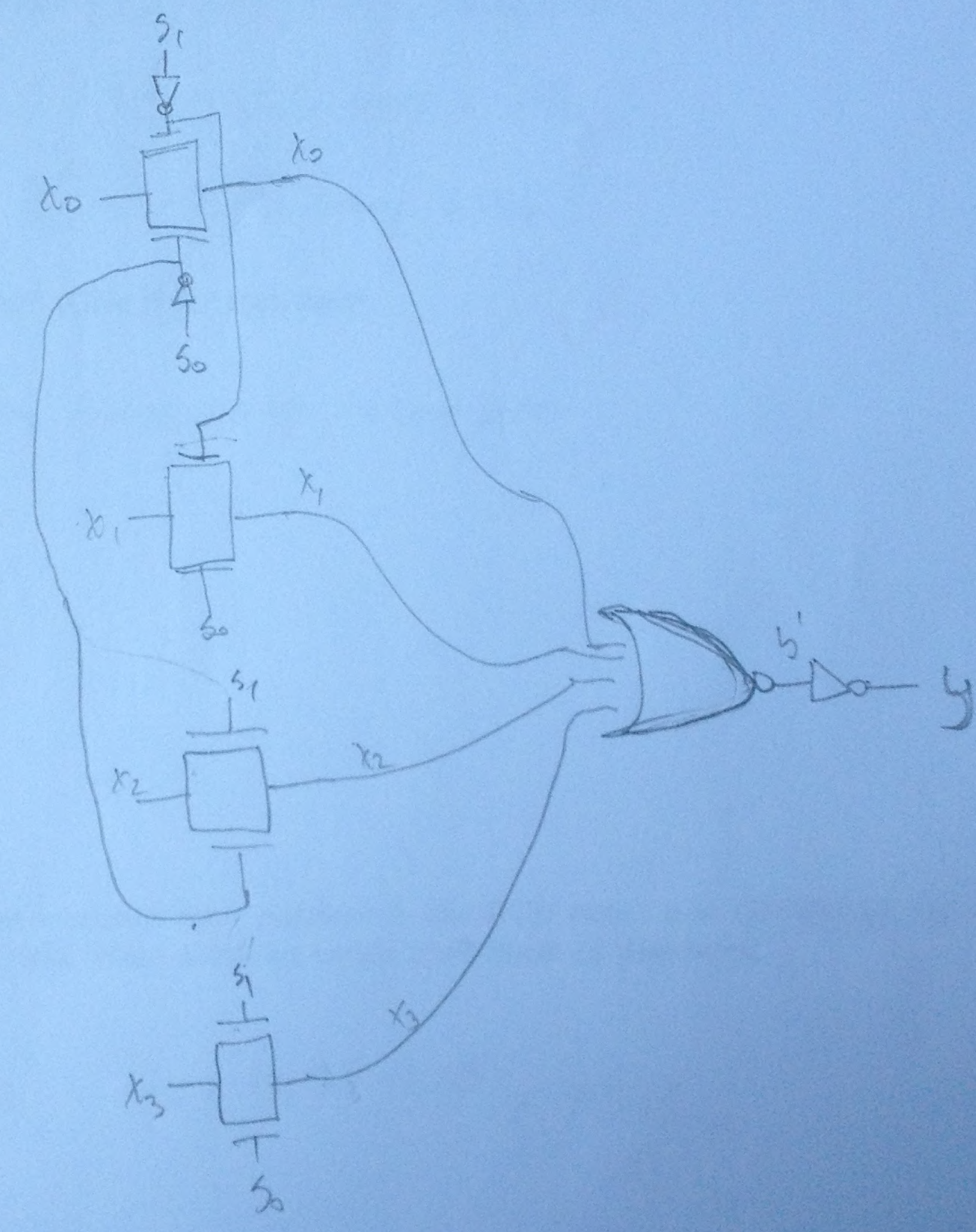
$C_0 = S_1' \cdot S_0'$ $C_1 = S_1' S_0$ $C_2 = S_1 S_0'$ $C_3 = S_1 S_0$

+1

Show the final network. Label all inputs and outputs (external and internal). How many transistors are needed in total?

not correct TG

+2



Problem 3 (10 points) 10

1. **(5 points)** A 8-bit vector represents a set of positive integers {0,...,N}. Which of the following coding alternatives

   (a) BCD 10011001 = 99, range = 100

   (b) 2421 code (a decimal code) 11111111 = 99 range = 100

   (c) Excess-3 code (a decimal code) 11001100 = 99 range = 100

   (d) Octal 11111111 = 377_8 = 255 range = 256

   (e) Binary 11111111 = $2^8 - 1$ = 255, range = 256

   provides the largest range? Why? (Give N for each case). ✓

   Octal and Binary since range is largest

2. **(5 points)**
   a and b are 12-bit vectors that represent their numbers in the BCD code. a = (100000110101) and the decimal of their sum $a + b$ is 1,800. What is the bit vector of b? Show all your work.

   8 3 5

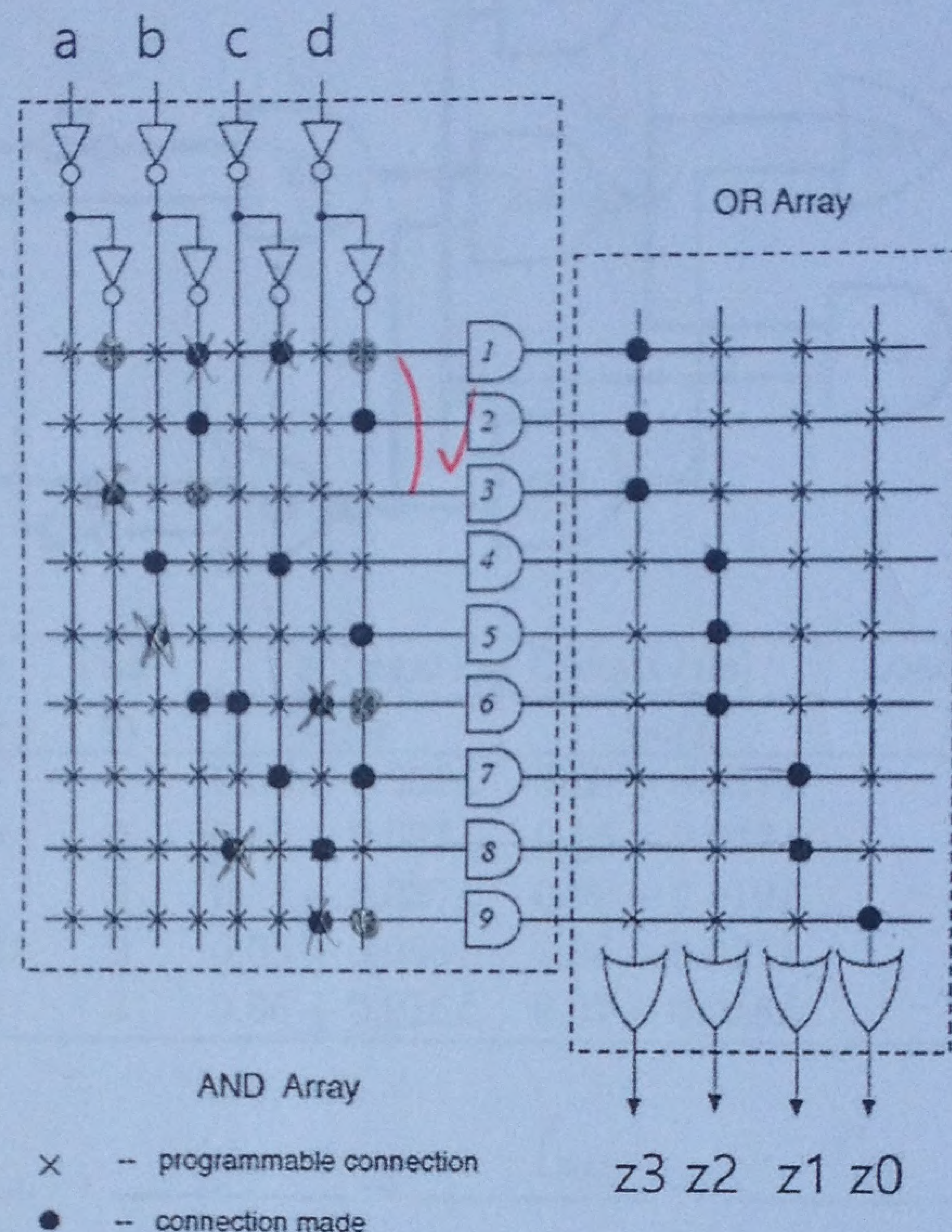
   $a = 835_{10}$ $b = 1800 - 835_{10} = 965_{10}$

   1001 0110 0101 = b ✓

Problem 4 (10 points) 10

We would like to verify that the PLA implementation shown here implements the following switching functions:

$$z3 = b + bd + ad$$
$$z2 = b'c + d + bc'd$$
$$z1 = cd + d'$$
$$z0 = d$$



1. **(6 points)** Analyze the PLA shown above and show the output expressions.

$Z_0 = d'$

$Z_1 = (d'c') + dc$

$Z_2 = (d'c'b) + (db') + (cb')$

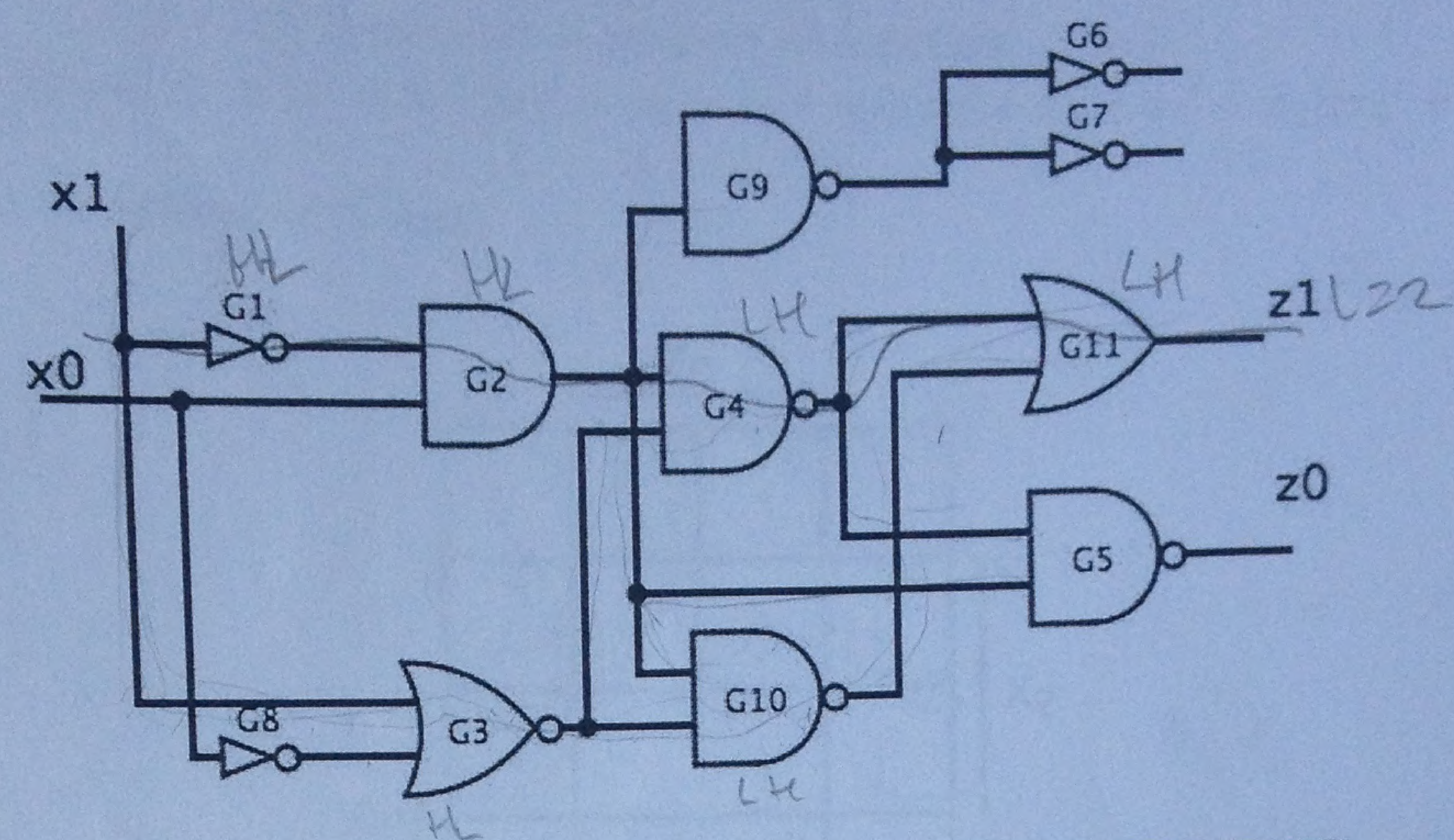
$Z_3 = (a) + (db) + (bc)$ ✓

2. **(4 points)** Is the PLA implementation correct? If not, find errors and show the correct implementation (cross out wrong connections and insert correct ones)

Not correct

Problem 5 (10 points)

Calculate the propagation delay $t_{pLH}(z1)$ when $x1$ changes. Assume that $z1$'s load value is 2. Fill in the blanks below with the appropriate values. You don't need to fill all the blanks.



Gate Type	Fan-in	Propagation Delays (ns) t_{pLH}	t_{pHL}	Load Factor I
NOT	1	$0.02 + 0.038L$	$0.05 + 0.017L$	1.0
AND	2	$0.15 + 0.037L$	$0.16 + 0.017L$	1.0
OR	2	$0.12 + 0.037L$	$0.20 + 0.019L$	1.0
NAND	2	$0.05 + 0.038L$	$0.08 + 0.027L$	1.0
NOR	2	$0.06 + 0.075L$	$0.07 + 0.016L$	1.0

Gate name:	G1 →	G2 →	G4 →	G11 →	→	
Gate type:	NOT-1 →	And2 →	NaNd2 →	OR2 →	→	
LH / HL:	HL →	HL →	LH →	LH →	→	
Output load L:	1 →	4 →	2 →	2 →	→	
Prop. Delay:	.05+.017(1) →	.16+.017(4) →	.05+.038(2) →	.12+.037(2) →	→	

✓

Problem 6 (25 points)

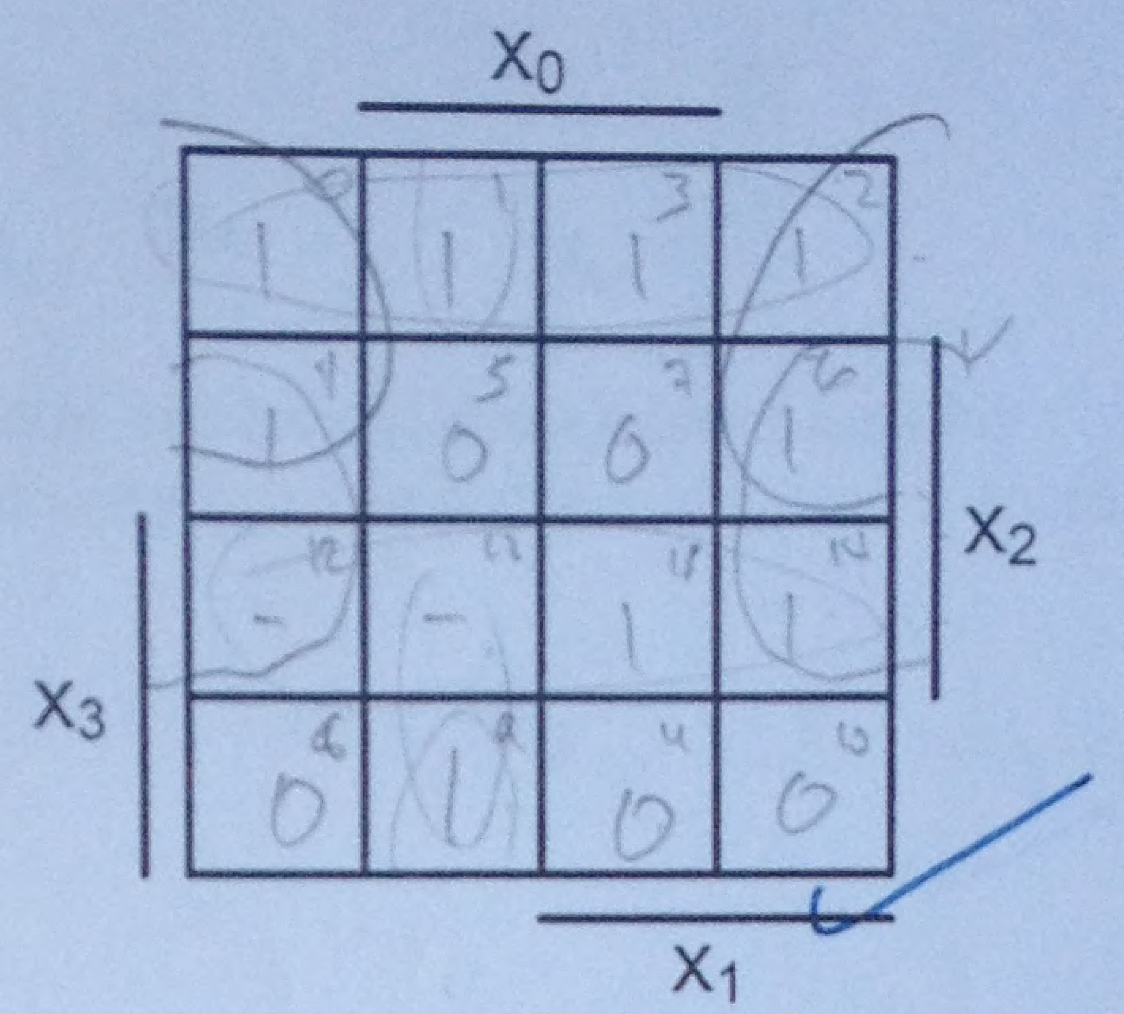
For the switching function $f(x_3, x_2, x_1, x_0)$, we are given the information below for the dc-set and zero-set.

dc-set = (12,13)

zero-set = zero-set of function

$$(x_3 + x_2' + x_1 + x_0')(x_3 + x_2' + x_1' + x_0')(x_3' + x_2 + x_1 + x_0)(x_3' + x_2 + x_1' + x_0)(x_3' + x_2 + x_1' + x_0')$$

1. (2 points) Fill out the following K-map.



2. (3 points) Which of the given expressions are prime implicants of the function given above? Circle all that apply. Do not circle implicants that are not prime.

(a) x_3x_1	(d) $x_3'x_0'$	(g) $x_3x_2x_1$	(j) $x_3x_1'x_0$
(b) $x_3'x_2'$	(e) x_2x_0'	(h) $x_3'x_2'x_1'$	(k) $x_3'x_2x_1'x_0$
(c) $x_3'x_1$	(f) x_3x_2	(i) $x_2'x_1'x_0$	(l) $x_3'x_2x_1x_0'$

3. (3 points) Write down the complete set of **essential** prime implicants.

4. (3 points) Write down the minimal sum of products expressions for f. If there are multiple forms of minimal sum of products expressions, you only need to write down one of them.

oints) Which of the given expressions are prime implicates of the function given above? Circle **all** that pply. Do not circle implicates that are not prime.

(a) $(x_3' + x_2')$	(d) $(x_3' + x_2 + x_1')$	(g) $(x_3 + x_1' + x_0)$	(j) $(x_3 + x_1' + x_0')$
(b) $(x_3' + x_1')$	(e) $(x_3' + x_1 + x_0')$	(h) $(x_2' + x_1 + x_0')$	(k) $(x_3 + x_2 + x_1 + x_0')$
(c) $(x_3' + x_2 + x_0)$	(f) $(x_3 + x_2' + x_0')$	(i) $(x_3' + x_1 + x_0)$	(l) $(x_3 + x_2' + x_1' + x_0)$

5

6. **(3 points)** Write down the complete set of **essential** prime implicates.

1

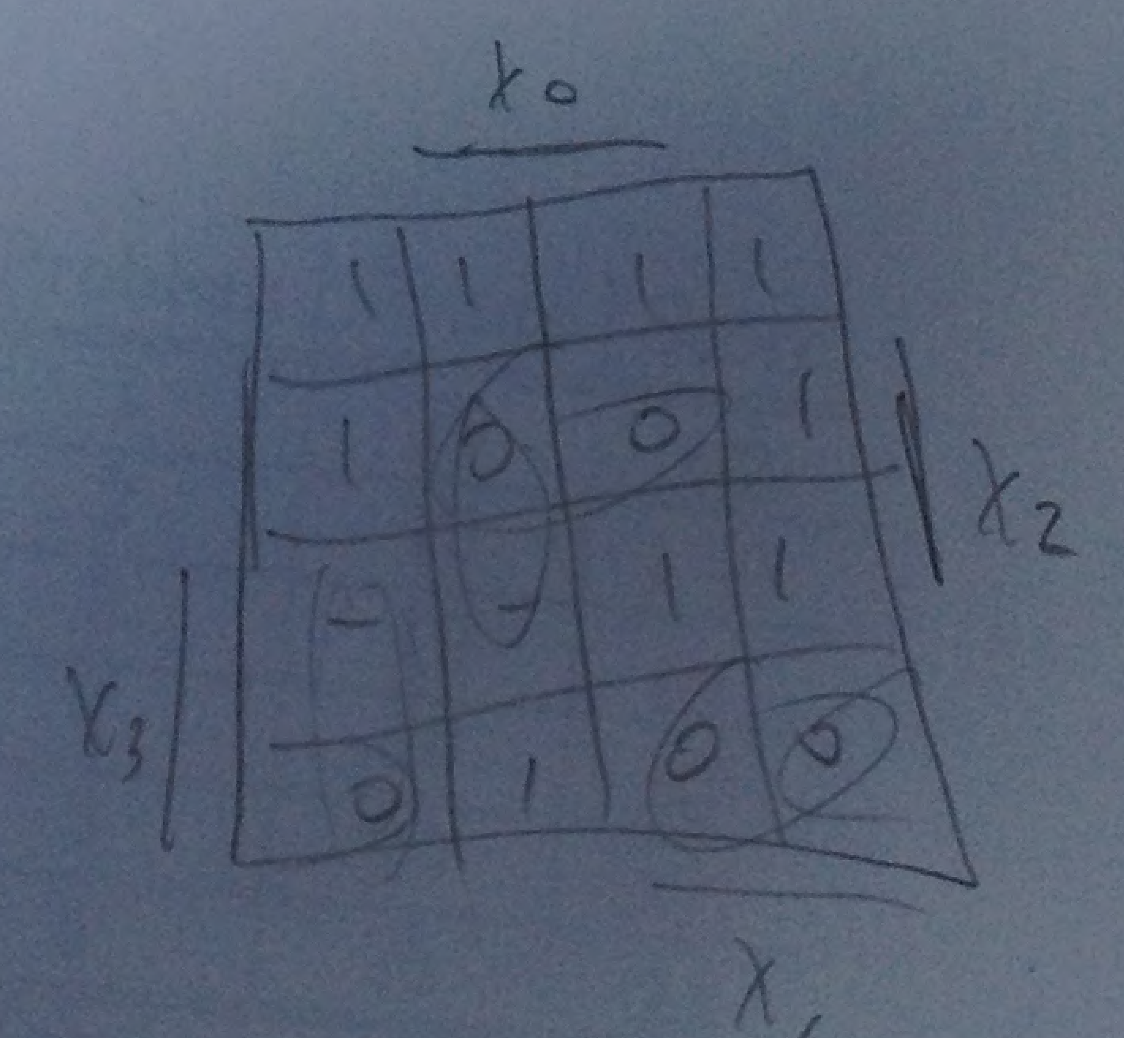
$(x_2' + x_3 + x_0'), (x_1' + x_3' + x_2), (x_3' + x_1 + x_0)$

X x_1' ✓ X

7. **(3 points)** Write down the minimal product of sums expressions for f. If there are multiple forms of minimal product of sums expressions, you only need to write down one of them.

2

$(x_2' + x_3 + x_1')(x_1' + x_3' + x_2)(x_3' + x_1 + x_0)$



PIs

$(x_0' + x_2' + x_3)$ ✓

$(x_0' + x_1 + x_2')$ ✓

$(x_3' + x_0 + x_1)$ ✓

$(x_3' + x_2 + x_0)$ ✓

$(x_1' + x_2 + x_3')$

1