Homework #1 (v190405)

(Deadline: 11:59PM PDT, Jan 23, 2020)

Name (Last, First): Arora, Gurbir

Student Id #: 105178554

INSTRUCTIONS

This homework is to be done individually. You may use any tools or refer to published papers or books, but may not seek help from any other person or consult solutions to prior exams or homeworks from this or other courses (including those outside UCLA). You're allowed to make use of tools such as Logisim, WolframAlpha (which has terrific support for boolean logic) etc.

You must submit all sheets in this file based on the procedure below. Because of the grading methodology, it is much easier if you print the document and answer your questions in the space provided in this problem set. It can be even easier if you answer in electronic form and then download the PDF. Answers written on sheets other than the provided space will not be looked at or graded. Please write clearly and neatly - if we cannot easily decipher what you have written, you will get zero credit.

SUBMISSION PROCEDURE: You need to submit your solution online at Gradescope (https://gradescope.com/). Please see the following guide from Gradescope for submitting homework. You'd need to upload a PDF and mark where each question is answered.
https://gradescope-static-assets.s3-us-west-2.amazonaws.com/help/submitting_hw_guide.pdf

Problem #1

Recall that a DNA is composed of 4 different nucleotides, G, C, A, and T (or U), grouped as 2 pairs (G with C, and A with T). The complementary pairing facilitates transcription of DNA into RNA. Three nucleotides in sequence are recognized as distinct amino acids. For instance, GGG correspond to glycine.

(a) Suppose that we chose a 2-bit binary representation for the nucleotides. We assign G to be nb[1:0] = {nb[1], nb[0]} = {0,0} = 2'b00 (note these different ways we represent this 2-bit signal) and T is represented as nb[1:0]=2'b01. What should one choose for C and A to facilitate representing the nucleotides after RNA transcription (which produces a "complement" strand).

(b) Special 3 nucleotide sequences are known as a "stop codon" that indicate the end of a gene. They are TAA, TAG, and TGA. The 3 nucleotide sequence when using the encoding in (a) is represented as cb[5:0] where for TAG, bits cb[5:4] are for T, cb[3:2] are for A, and cb[1:0] are for G. Write the logic equation for indicating the detection of a stop codon, S, as a function of cb[5:0] in fully-disjunctive normal form.

(c) Apply factoring and other Boolean properties to reduce the result in (b) into an expression using the least number of literals.

(d) Instead of binary representation, let us use one-hot representation for the nucleotides, whereby each nucleotide is expressed with 4 bits, coh[3:0] = 4'b0001 for G, 4'b0010 for C, 4'b0100 for A, and 4'b1000 for T. The codon is represented as a 12-bit combination of 3 nucleotides, coh[11:0]. Write a minimal sum-of-products, logic equation for the same indicator for a stop codon, S.

(e) Reduce the expression in part (d) and to one using the least number of literals.

(f) One-hot encoding is not difficult to "complement" between the nucleotide pairing, G-to-C, C-to-G, A-to-T, and T-to-A. Write the logical equations for cnoh[3:0], the "complemented" nucleotides, from the inputs noh[3:0].

Answer the question for all parts in the space below.

a) $G = nb[1:0] = 2'b00,\ C = \neg G = nb[1:0] = \boxed{2'b11}$

$T = nb[1:0] = 2'b01,\ A = \neg T = nb[1:0] = \boxed{2'b10}$

b) $\left(cb[5:4] \wedge cb[3:2] \wedge cb[3:2]\right) \vee \left(cb[5:4] \wedge cb[3:2] \wedge cb[1:0]\right)$

$\vee \left(cb[5:4] \wedge cb[1:0] \wedge cb[3:2]\right)$

TAG: 011000 TAA: 011010 TGA: 010010

(011010) V (011000) V (010010)

$$\left(!cb[5] \wedge cb[4] \wedge cb[3] \wedge !cb[2] \wedge cb[1] \wedge !cb[0]\right) \vee$$
$$\left(!cb[5] \wedge cb[4] \wedge cb[3] \wedge !cb[2] \wedge !cb[1] \wedge !cb[0]\right) \vee$$
$$\left(!cb[5] \wedge cb[4] \wedge !cb[3] \wedge !cb[2] \wedge cb[1] \wedge !cb[0]\right)$$

(IMPORTANT: Keep this page in submission even if left unused)

c) $A = cb[5],\ B = cb[4],\ C = cb[3],\ D = cb[2],\ E = cb[1],\ F = cb[0]$

$$(!A \cdot B \cdot C \cdot !D \cdot E \cdot F) + (!A \cdot B \cdot C \cdot !D \cdot !E \cdot !F) + (!A \cdot B \cdot !C \cdot !D \cdot E \cdot !F)$$

$$= (!A \cdot B \cdot !D) \cdot (C \cdot E \cdot F + C \cdot !E \cdot !F + !C E !F)$$

$$\boxed{\begin{array}{l} = (!cb[5] \land cb[4] \land !cb[2]) \land ((cb[3] \land cb[1] \land cb[0]) \lor (cb[3] \land !cb[1] \land !cb[0]) \\ \lor (!cb[3] \land !cb[1] \land cb[0])) \end{array}}$$

d)

$$\text{TAA: } \overset{ABCD}{1000}\ \overset{EFGH}{0100}\ \overset{IJKL}{0100} \qquad \text{TAG: } \overset{ABCD}{1000}\ \overset{EFGH}{0100}\ \overset{IJKL}{0001}$$

TGA: 1000 0001 0100

$A = coh[11],\ B = coh[10],\ C = coh[9],\ D = coh[8],\ E = coh[7],\ F = coh[6],\ G = coh[5],$
$H = coh[4],\ I = coh[3],\ J = coh[2],\ K = coh[1],\ L = coh[0]$

$$(A \land !B \land !C \land !D \land !E \land F \land !G \land !H \land !I \land J \land !K \land !L) \lor (A \land !B \land !C \land !D \land !E \land F \land !G \land !H \land !I \land !J \land !K \land L) \lor (A \land !B \land !C \land !D \land !E \land !F \land !G \land H \land !I \land J \land !K \land !L)$$

$$\Rightarrow (coh[11] \land coh[6] \land coh[2]) \lor (coh[11] \land coh[6] \land coh[0]) \lor (coh[11] \land coh[4] \land coh[2])$$

$$\boxed{= (A \cdot F \cdot J) + (A \cdot F \cdot L) + (A \cdot H \cdot J)}$$

e)

$$= A(F \cdot J + F \cdot L + H \cdot J)$$

$$= A \cdot F(J + L) + (A \cdot H \cdot J)$$

$$\boxed{= A(F(J+L) + (H \cdot J))}$$

f) A= noh[3], B= noh[2], C= noh[1], D= noh[0]: inputs

E= cnoh[3], F= cnoh[2], G= cnoh[1], H= cnoh[0]: outputs

input: G   4'b0001, so A=0, B=0, C=0, D=1

output: C

G: 4'b0001

C: 4'b0010

A: 4'b0100

T: 4'b1000

cnoh[3]=(!noh[3] ∧ noh[2])

cnoh[2]= (noh[3] ∧ ! noh[2])

cnoh[1]= (!noh[2] ∧ noh[0])

cnoh[0]= (!noh[2] ∧ noh[1])

$G\ \overset{3210}{(0001)} \rightarrow C\ \overset{3210}{(0010)}$ ✓

A (0100) → T (1000) ✓

C (0010) → G (0001) ✓

T (1000) → A (0100) ✓

Problem #2

Consider the Boolean function below.

$$y = \neg\big((a \vee \neg b) \wedge \neg c\big) \wedge \neg\big((a \wedge d) \vee (\neg b \wedge d)\big) \wedge e$$

(a) Use Boolean properties to rewrite the above function as a minimal product of sums.

(b) Use factoring and DeMorgan's theorem to rewrite and simplify the expression above into one with the least number of literals.

(c) Design and draw an implementation of your result in (b) using only 2-input NOR gates. You can assume that true and complemented versions of each literal are available as inputs (a and ~a can both be used as inputs without needing an INV).

(d) Using the design in (c) as the starting point, implement the function using only 2-input NAND gates (again, true and complemented inputs are available).

Answer the question for all parts in the space below.

a) $y = \neg\big((a \vee \neg b) \wedge \neg c\big) \wedge \neg\big((a \wedge d) \vee (\neg b \wedge d)\big) \wedge e$

$= \big(\neg(a \vee \neg b) \vee c\big) \wedge \big((\neg a \vee \neg d) \wedge (b \vee \neg d)\big) \wedge e$

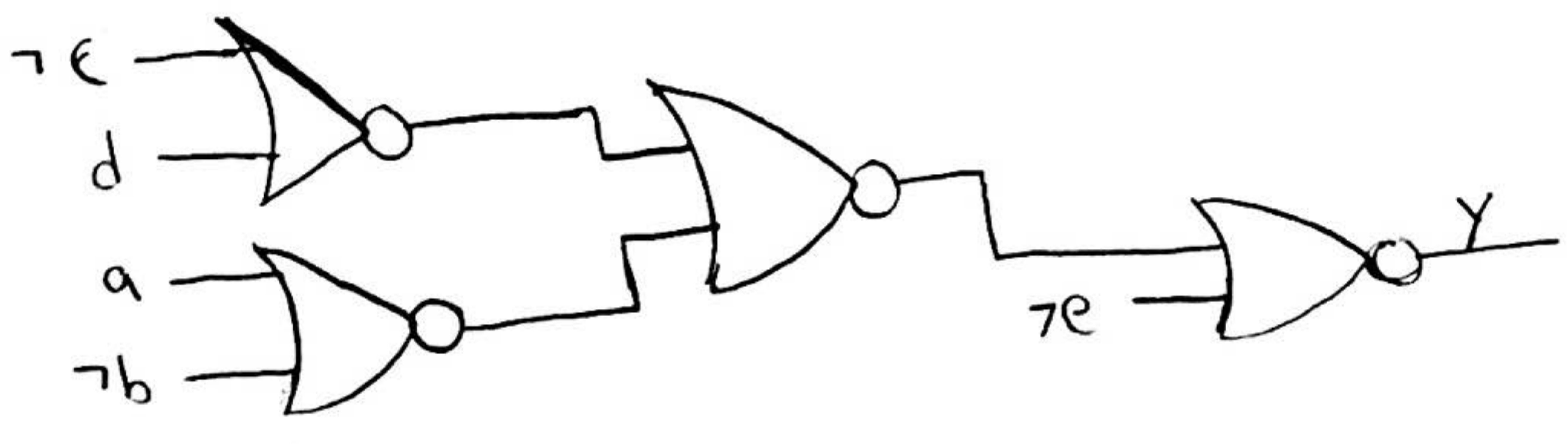
$= \big((\neg a \wedge b) \vee c\big) \wedge (\neg a \vee \neg d) \wedge (b \vee \neg d) \wedge e$

$\boxed{= (\neg a \vee c) \wedge (b \vee c) \wedge (\neg a \vee \neg d) \wedge (b \vee \neg d) \wedge e}$

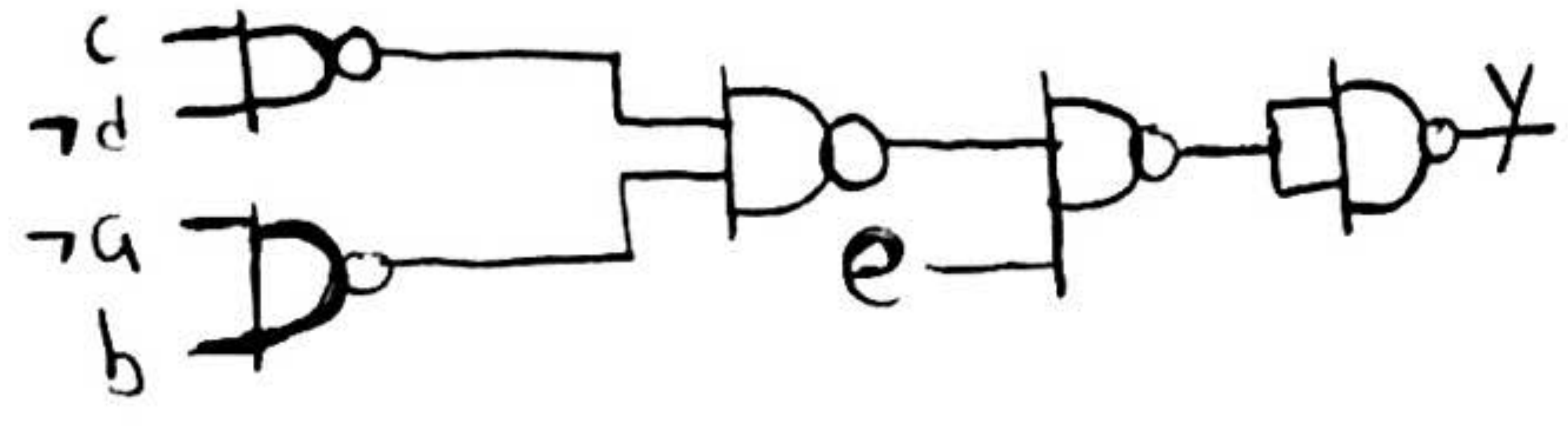
b) $= \big(\neg a \vee (c \wedge \neg d)\big) \wedge \big(b \vee (c \wedge \neg d)\big) \wedge e$ by distributive prop

$\boxed{= \big((c \wedge \neg d) \vee (\neg a \wedge b)\big) \wedge e}$

c)



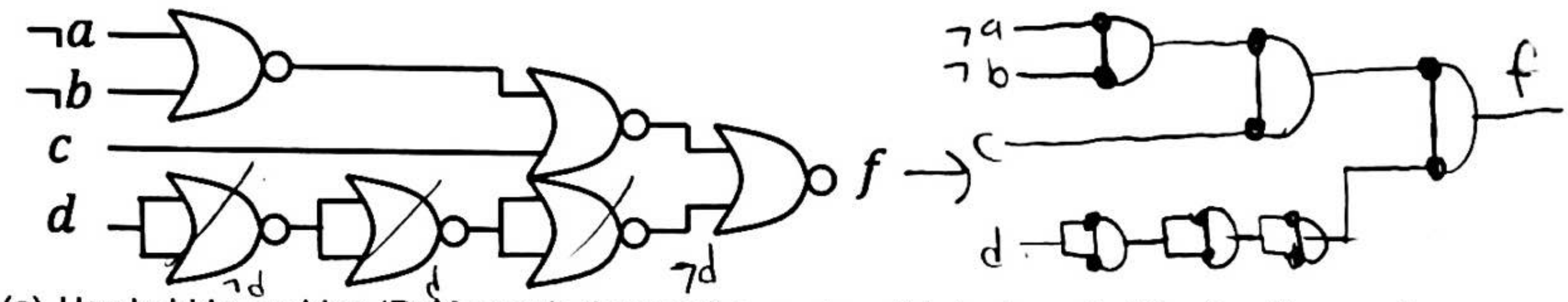
d)



(IMPORTANT: Keep this page in submission even if left unused)

Problem #3

The following logical function, f, is implemented using only 2-input NORs



(a) Use bubble-pushing (DeMorgan's theorem) to re-draw this logic so that the function can be easily expressed.

(b) Write the expression, based on (a), for the function, f.

Answer the question for all parts in the space below.

a) $f = \neg(\neg(\neg((\neg a \vee \neg b) \vee c)) \vee \neg(\neg(\neg(d \vee d) \vee d) \vee d))$

$\neg((a \wedge b) \wedge \neg c)$

$= \neg(\neg(\neg(\neg a \vee \neg b) \vee c) \vee \neg(\neg(\neg(d \vee d) \vee d) \vee d))$

$\neg((a \wedge b) \vee c)$

$= \neg(((\neg a \vee \neg b) \wedge \neg c) \vee \neg(\neg(\neg(d \vee d) \vee d) \vee d))$

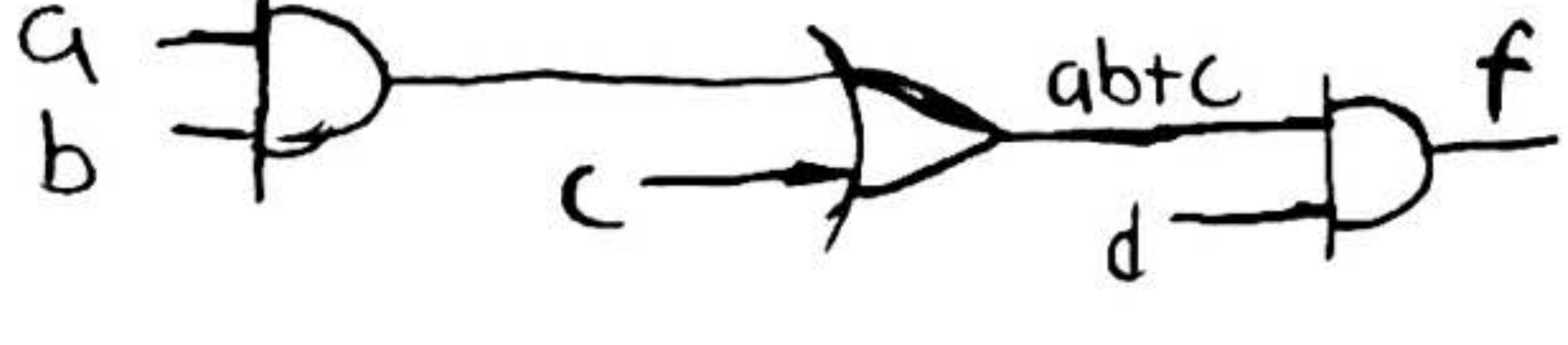
$\neg((\neg d \wedge \neg d) \vee d)$

$d \vee d \wedge \neg d) \vee d$

$= \neg((\neg a \vee \neg b) \wedge \neg c) \vee \neg d))$

$= ((a \wedge b) \vee c) \wedge d) = ((a \cdot b) + c) \cdot d)$

by looking at logic, we see that the right side simplifies to just ¬d prior to the final NOR gate

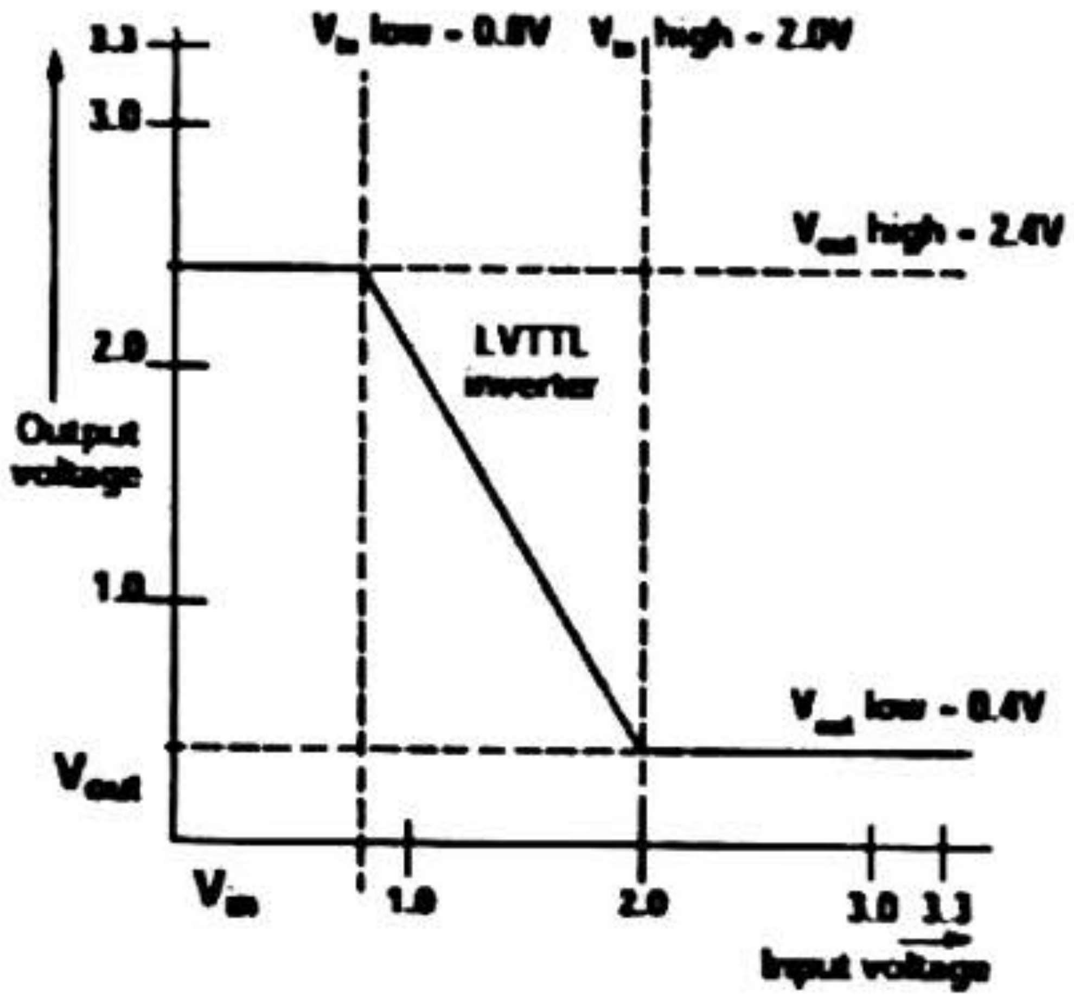


b) work shown above,

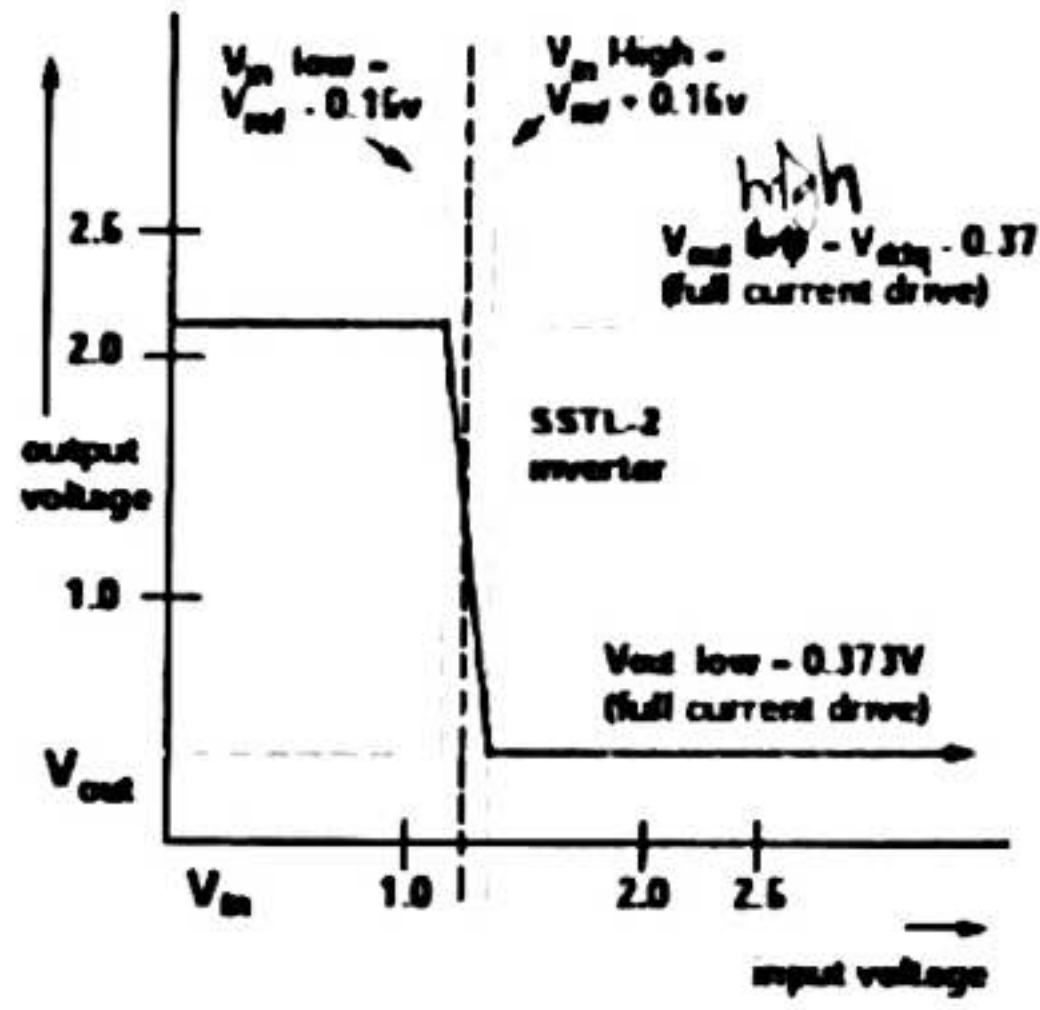
$f = ((a \wedge b) \vee c) \wedge d$

Problem #4

Two different types of logic are shown below with different voltage transfer characteristics: (i) low-voltage TTL (transistor-transistor logic), and (ii) series-stub termination logic. Both are used in memory DIMMs that you commonly find in compute servers and laptops.



(i) LVTTL



(ii) SSTL

(a) Determine the noise margin for LVTTL. The supply voltage is 3.3V.

(b) Determine the noise margin for SSTL. Vref is set to 1.25V. The supply voltage (Vddq) is 2.5V.

(c) Discuss the difference between these two different logic types. Despite needing a lower supply voltage, SSTL is better. What accounts for the noise margin differences? What are the advantages of using SSTL versus LVTTL?

Answer the question for all parts in the space below.

a) $V_{IL} = .8V$, $V_{IH} = 2.0V$, $V_{OH} = 2.4V$, $V_{OL} = 0.4V$

$V_{NH} = V_{OH} - V_{IH} (\geq 0) = 2.4V - 2.0V = \boxed{0.4V}$

$V_{NL} = V_{IL} - V_{OL} (\geq 0) = 0.8V - 0.4V = \boxed{0.4V}$

b) $V_{IL} = 1.25V - 0.15V$ $V_{IH} = 1.25V + 0.15V$ $V_{OH} = 2.5V - 0.373V$

$V_{OL} = 0.373V$

$V_{NH} = 2.127V - 1.40V = \boxed{0.727V}$

$V_{NL} = 1.10V - 0.373V = \boxed{0.727V}$

c) SSTL allows for a higher noise margin. The difference in noise margin is most likely due to the input high and low values being much closer than the LVTTL counterpart. TTL signifies that transistors perform both the logic function and the amplifying function. SSTL employs a reduced voltage swing on the inputs by specifying a reference voltage, which allows for a greater noise margin to be tolerated. SSTL is Known to be faster, which is an added bonus to the increased toleration of noise. The SSTL-2 inverter has a much steeper slope when compared to the LVTTL inverter. A sharper, slope usually indicates that the transfer curve is more ideal, meaning that noisy signals are determined faster. Furthermore, the graphs depict that the LVTTL has a lager ground between What is deemed as a "1" and "0", while the SSTL is set up to have a small "questionable" state zone and is predominantly 0 or 1.

Problem #5

D	C	B	A	Y	Y'
0	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	d	d
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	d	d
1	1	1	1	0	1

(a) Draw K-Map of the function corresponding to the truth table shown above. Identify on the K-Map the prime implicants of the function and identify which of the prime implicants (if any) are essential.

(b) Write the minimum cover as a sum-of-products Boolean function using the prime implicants found in (a)

(c) Draw the K-Map of the complement function, Y'. Identify on the K-Map the prime implicants of the function and identify which of the prime implicants (if any) are essential.

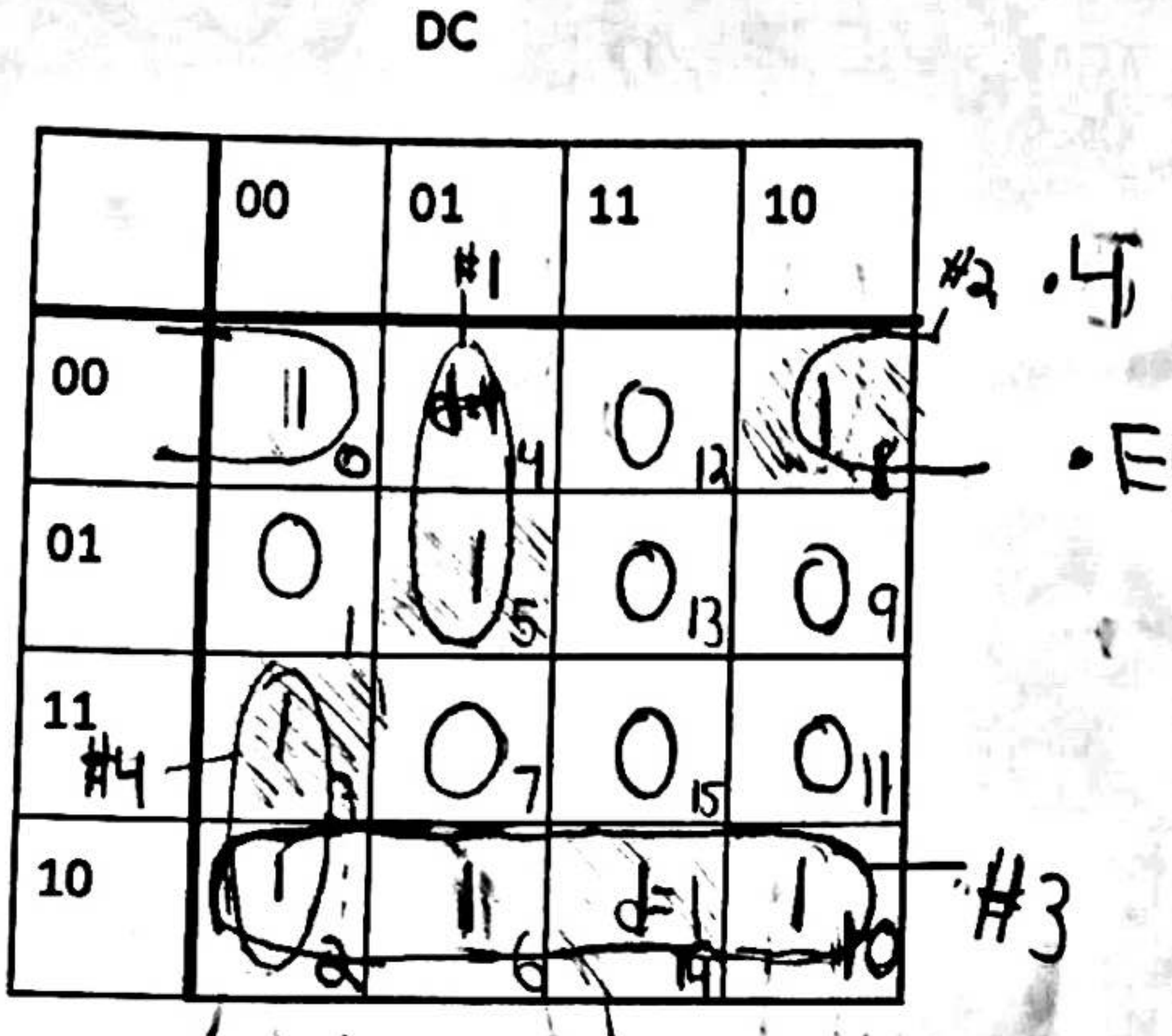
(d) Write the minimum cover as a sum-of-products Boolean function using the prime implicants found in (c).

(e) Write the complement function, Y', as a minimal product-of-sums.

(f) Draw the truth table of the dual of the function Y, Y^D.

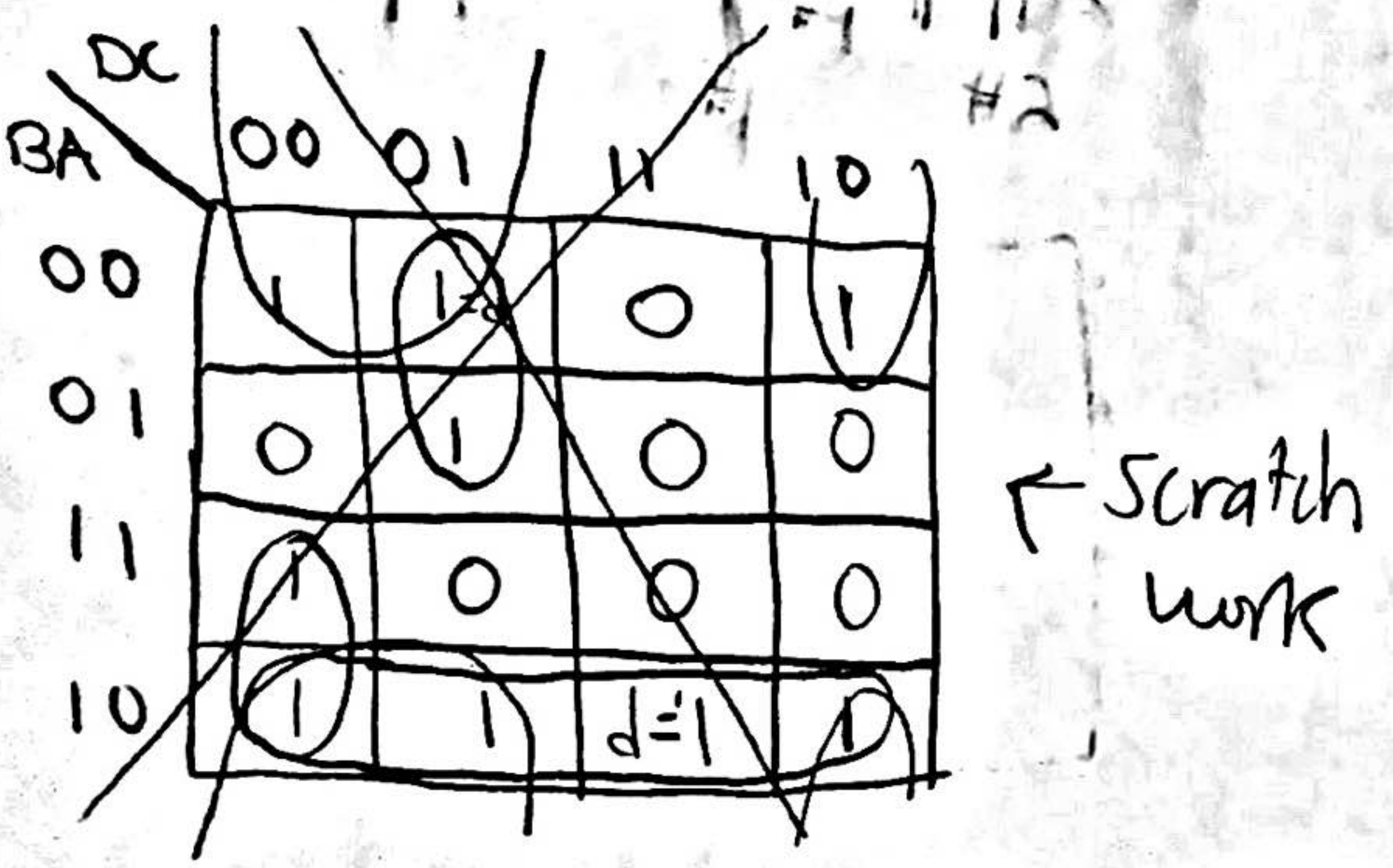
Answer the question for all parts in the space below.

(a)



BA \ DC	00	01	11	10
00	1_0	$d=1_4$	0_{12}	1_8
01	0_1	1_5	0_{13}	0_9
11	1_3	0_7	0_{15}	0_{11}
10	1_2	1_6	$d=1_{14}$	1_{10}

- 4 prime Implicants
- EPIs: cells 3,5,8,&14 are EPIs
  - 4 EPIs, shaded cells
    - all PIs are EPIs in this case

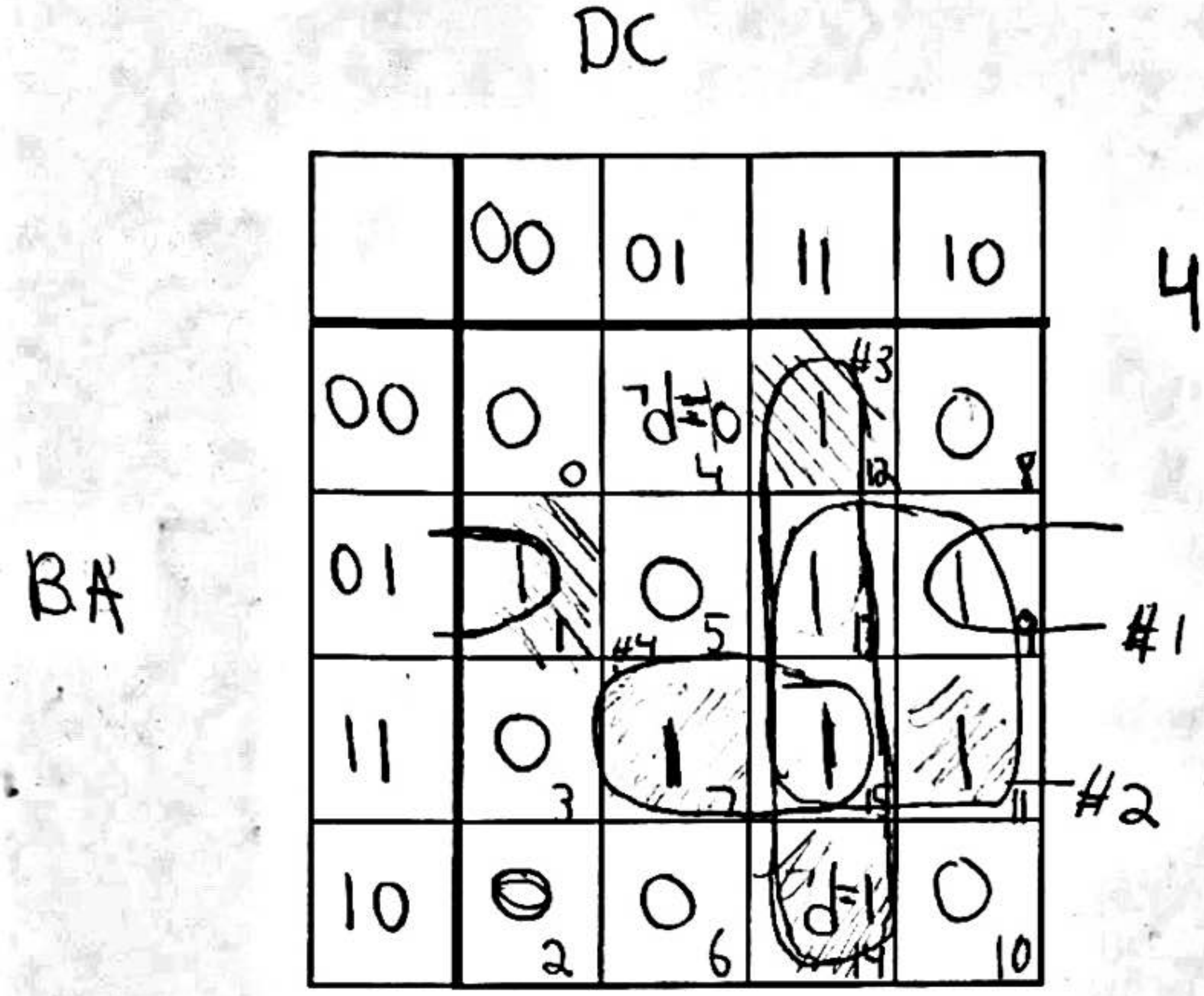


BA \ DC	00	01	11	10
00	1	1=d	0	1
01	0	1	0	0
11	1	0	0	0
10	1	1	d=1	1

← Scratch work

(b) $y = B\bar{A} + \bar{C}\bar{A}\bar{C} + \bar{B}\bar{D}C + B\bar{D}\bar{C}$

(c)



	00	01	11	10
00	0 0	d=0 4	1 12	0 8
01	1 1	0 5	1 13	1 9
11	0 3	1 7	1 15	1 11
10	0 2	0 6	d=1 14	0 10

4 Prime Implicants:

14 EPIs: cells 1, 7, 11, 12, & 14

So, all PIs are EPIs in this case

(d) $y = DC + A\bar{B}\bar{C} + ABC + DA\bar{C}$

(e)

#2

BA \ DC	00	01	11	10
00	0	d=0	1	0
01	1	0	1	1
11	0	1	1	1
10	0	0	d=0	0

#1

#4 #3

$$= (\bar{A}+\bar{B}+\bar{C})\cdot(\bar{B}+C+\bar{D})\cdot(B+\bar{D}+\bar{C})\cdot(B+\bar{A})$$

XX10 001X 010X X000

XX01 110X 101X X111

$$\boxed{= (\bar{B}+A)\cdot(D+C+\bar{B})\cdot(D+\bar{C}+B)\cdot(C+A)}$$

(f)

D	C	B	A	Y_D
0	0	0	0	1
0	0	0	1	0
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	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

$$Y_d = \underset{(1)}{(B+\bar{A})} \cdot \underset{(1)}{(\bar{C}+\bar{A})} \cdot \underset{(1)}{(\bar{B}+\bar{D}+C)} \cdot \underset{(1)}{(B+\bar{D}+\bar{C})}$$

(IMPORTANT: Keep this page in submission even if left unused)