Paper 7, TDC Part-3
Discussion of some questions of 2018
Lecture - 3

By:

Mayank Mausam

Assistant Professor (Guest Faculty)

Department of Electronics

L.S. College, BRA Bihar University,

Muzaffarpur, Bihar

Discussion of 2018 Questions

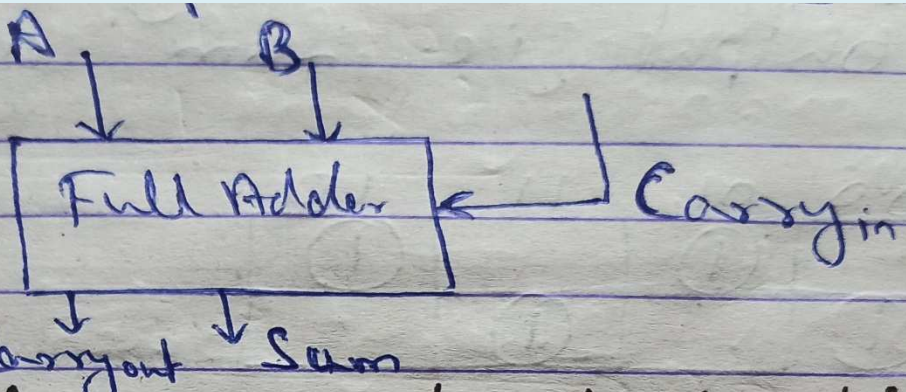
Q8 Explain the operation of Full Adder with truth table and realize them using NAND gates.

Soln A full adder is a logic circuits that allows addition of carry from the lower order bits when multibit addition is performed.

Therefore the full adder circuits have 3-inputs lines (two for binary inputs and one for carry bit from the previous addition) and two-output lines are "Sum and Carry".

Discussion of 2018 Questions

Carry".



Block diagram of full Adder ckt.

As a full adder has 3-inputs lines so we will have 8 possible operation i.e. 8 possible input condition and for each possible input condition we will have two output. The M.S.B of the output is referred as "Carry" and LSB represents "Sum". The truth table for the operation of Full adder is given in

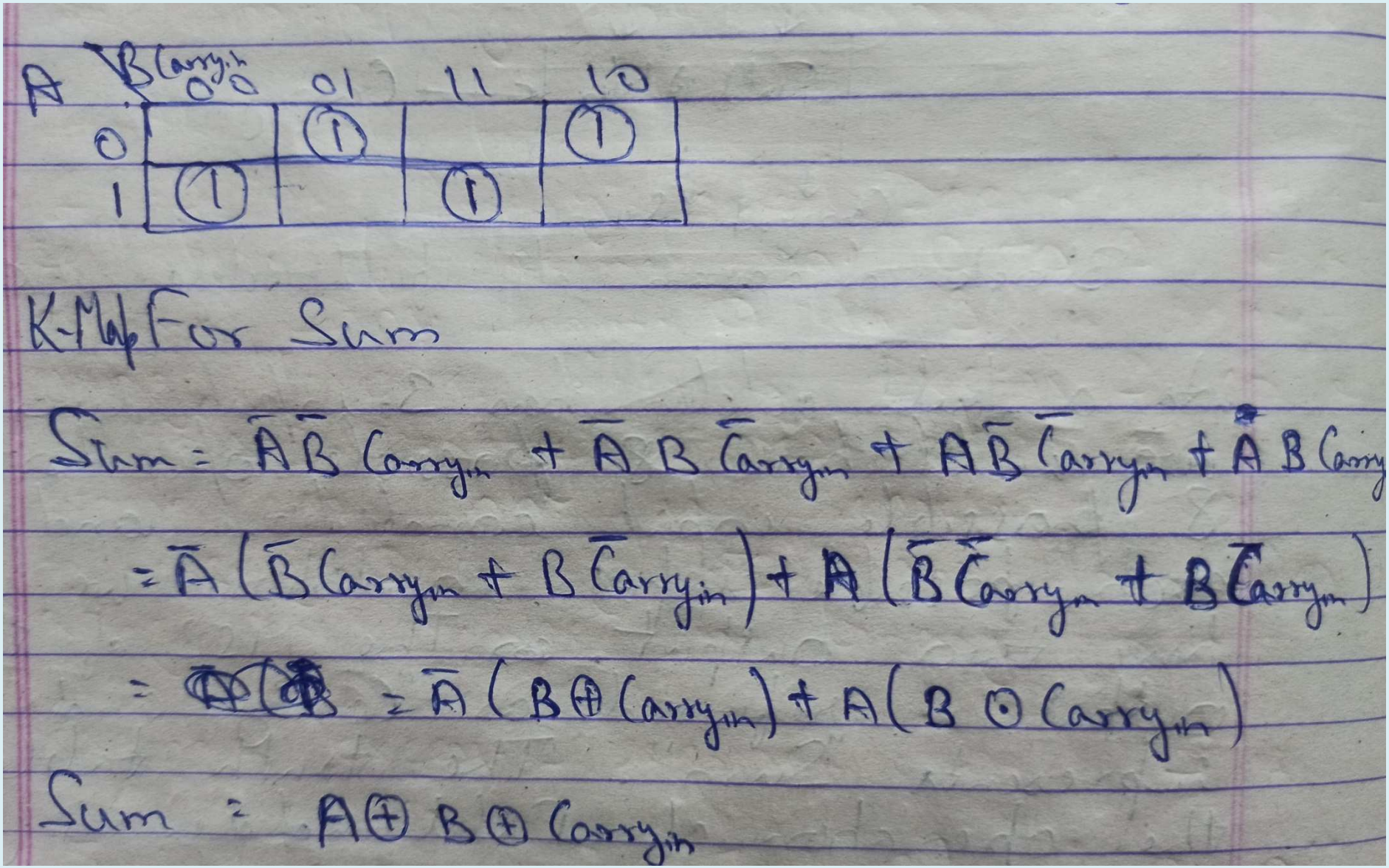
REDMI NOTE 8 PRO
AI QUAD CAMERA

Discussion of 2018 Questions

Inputs			Outputs	
A	B	Carry in	Carry out	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

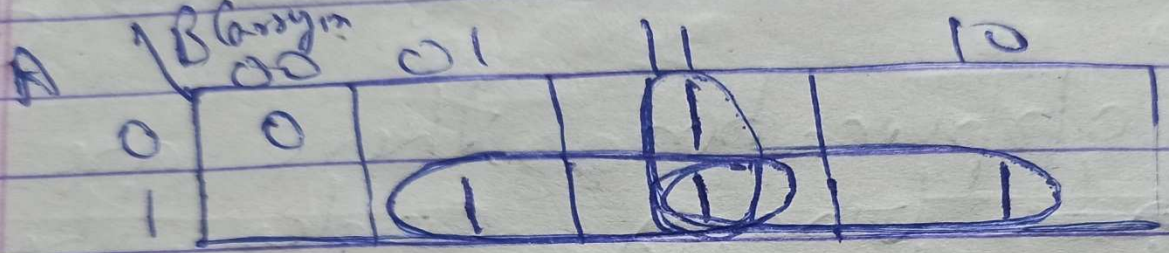
The boolean expression for Sum and Carry out can be derived using 8-cells K-map for each output i.e. Sum and Carry out

Discussion of 2018 Questions



Discussion of 2018 Questions

K-Map for Carryout:



K-Map for Carryout

$Carry_{out} = A\,Carry_{in} + B\,Carry_{in} + AB$

for Realizing using NAND gates only we write,

$Carry_{out} = AB + A\bar{B}\,Carry_{in} + \bar{A}B\,Carry_{in}$

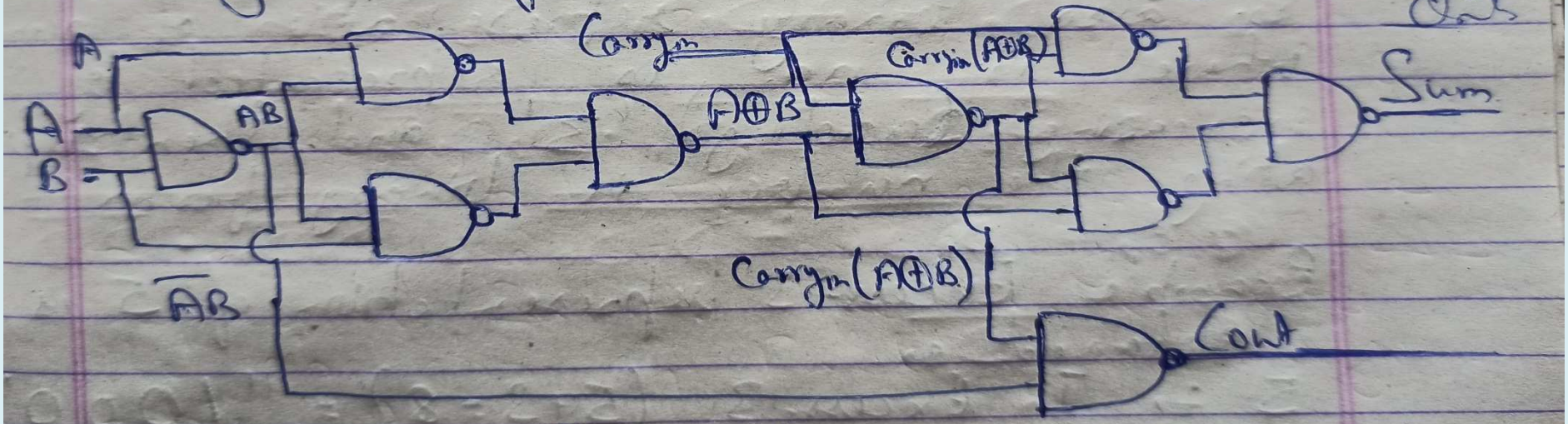
Discussion of 2018 Questions

$$= AB + Carry_{in}(A \oplus B)$$

$$Carry_{Out} = \overline{AB + Carry_{in}(A \oplus B)}$$

$$= \overline{\overline{AB} \cdot \overline{Carry_{in}(A \oplus B)}}$$

Realization of Full Adder Using NAND gate



Realization of Full Adder using NAND gates

Discussion of 2018 Questions

Q(9) Design BCD to Excess-3 code converter using minimum number of NAND gates.

Ans:- In digital application, there are different types of ~~code~~ binary codes that are used in different logic ckt. To interact one application with other application with different types of binary code we need code converter ckt.

The process of code conversion can be done with the help of combinational circuits. Combinational ckts can be design by obtaining the expression using K-Map and mapping the truth table on K-Map

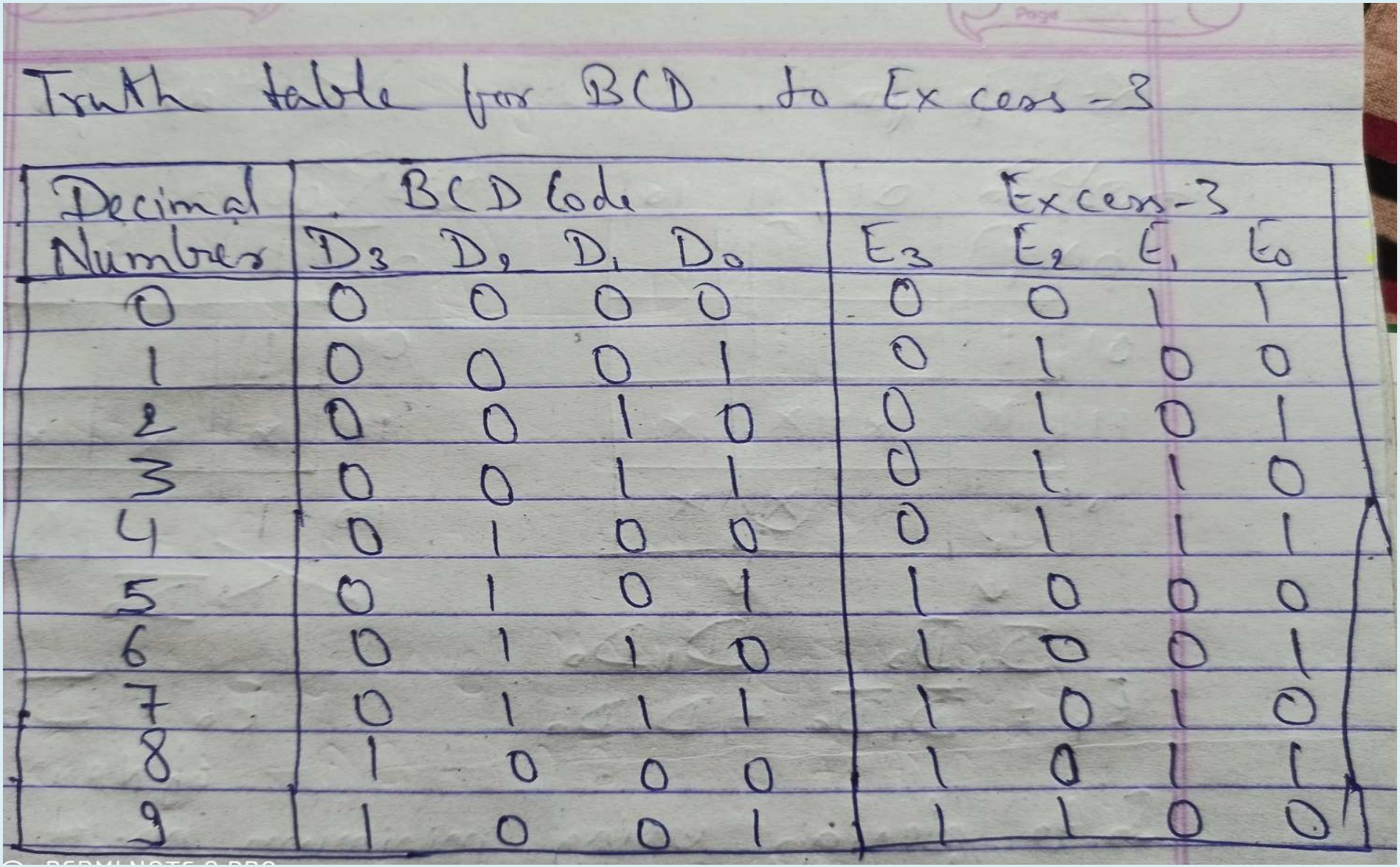
Discussion of 2018 Questions

BCD is Binary Coded Decimal representation of decimal no. 0 to 9, where decimal nos. from 0 to 9 is represented using 4-bits. Whereas Excess-3 code is also binary representation of numbers where is no. is represented by increasing (adding) '3' and then representing the number in binary form

Example Excess-3 representation of 5 is obtained as.

5 :→ In Excess-3 (5+3=8) = "1000"

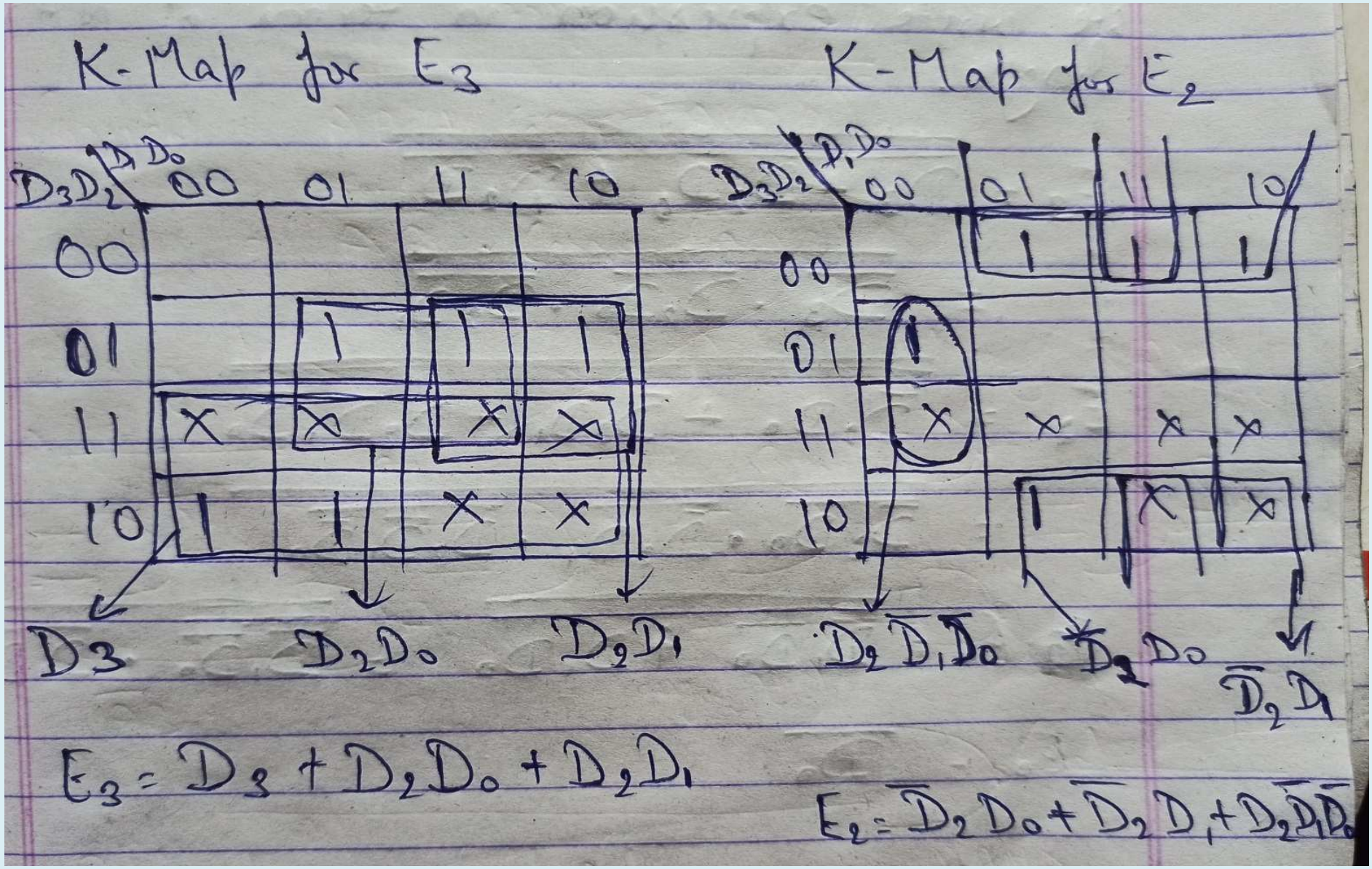
Discussion of 2018 Questions



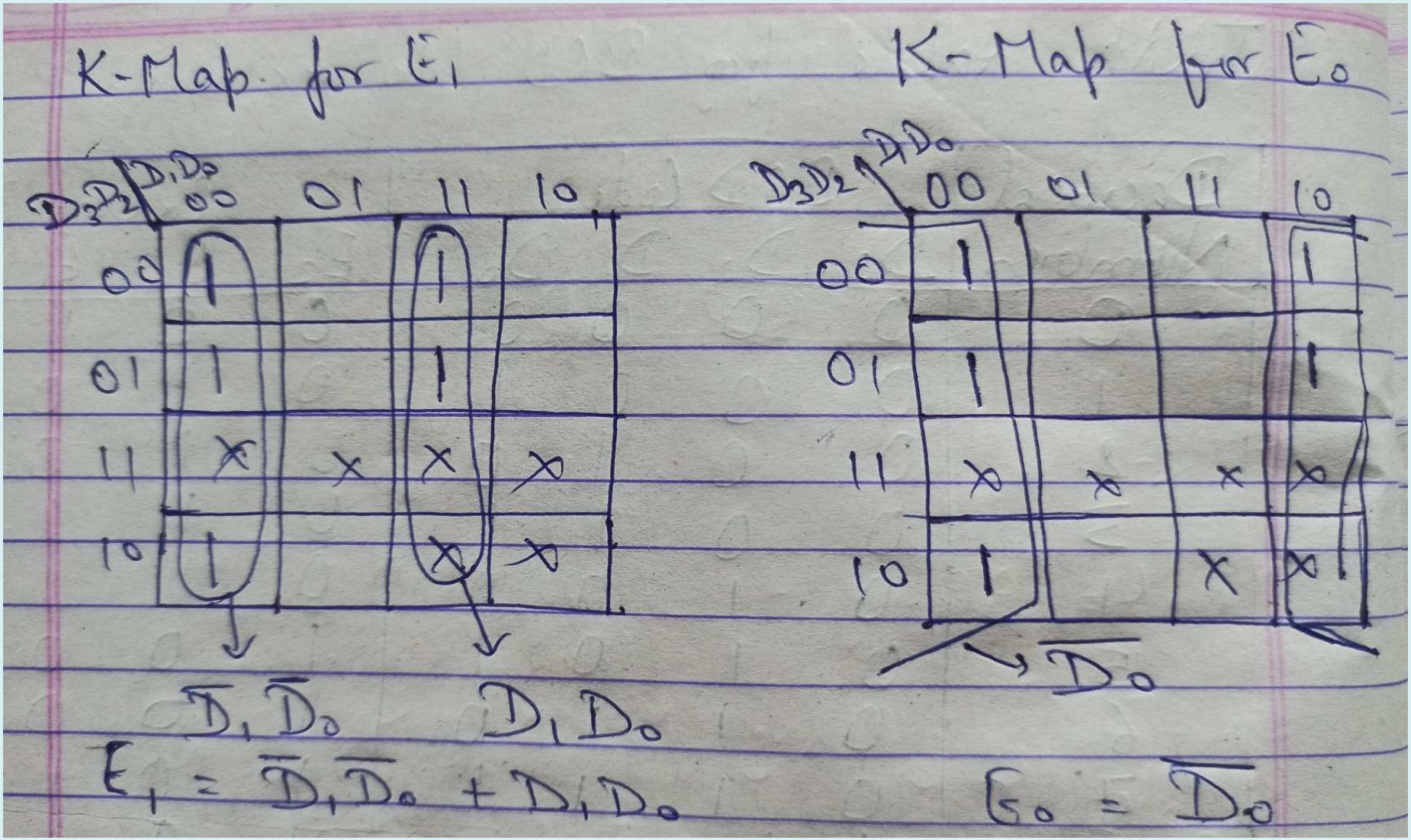
Truth table for BCD to Excess-3

Decimal Numbers	BCD Code				Excess-3			
	D_3	D_2	D_1	D_0	E_3	E_2	E_1	E_0
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

Discussion of 2018 Questions



Discussion of 2018 Questions



Discussion of 2018 Questions

So to design using minimum no. of gates of NAND we write expression in terms of NAND logic.

$$E_3 = \overline{\overline{D_3} + \overline{D_2 D_0} + \overline{D_2 D_1}}$$

$$= \overline{\overline{D_3} \cdot \overline{D_2 D_0} \cdot \overline{D_2 D_1}}$$

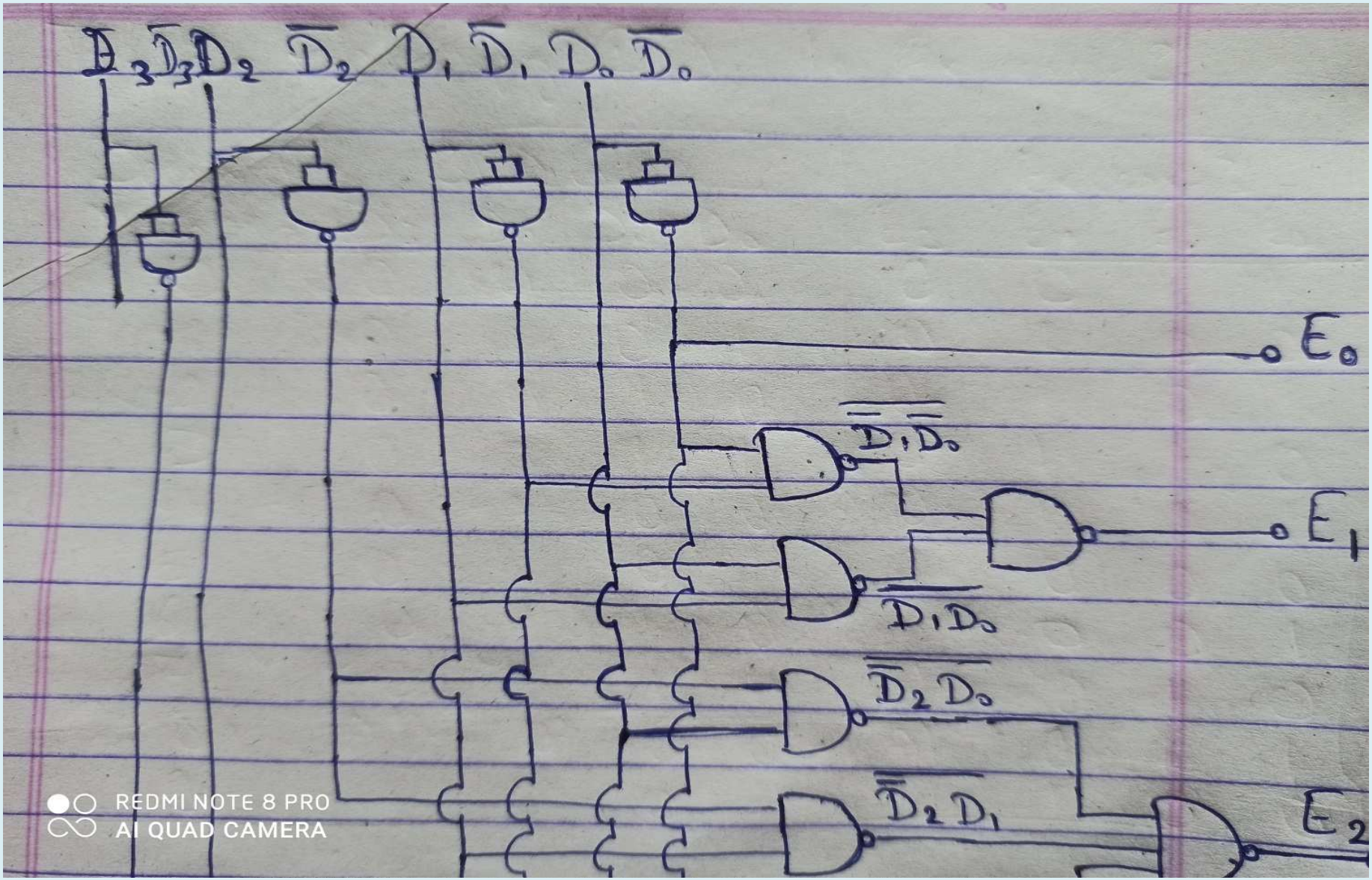
$$E_2 = \overline{\overline{D_2} D_0 + \overline{D_2} D_1 + D_2 \overline{D_1} \overline{D_0}}$$

$$= \overline{\overline{\overline{D_2} D_0} \cdot \overline{\overline{D_2} D_1} \cdot \overline{D_2 \overline{D_1} \overline{D_0}}}$$

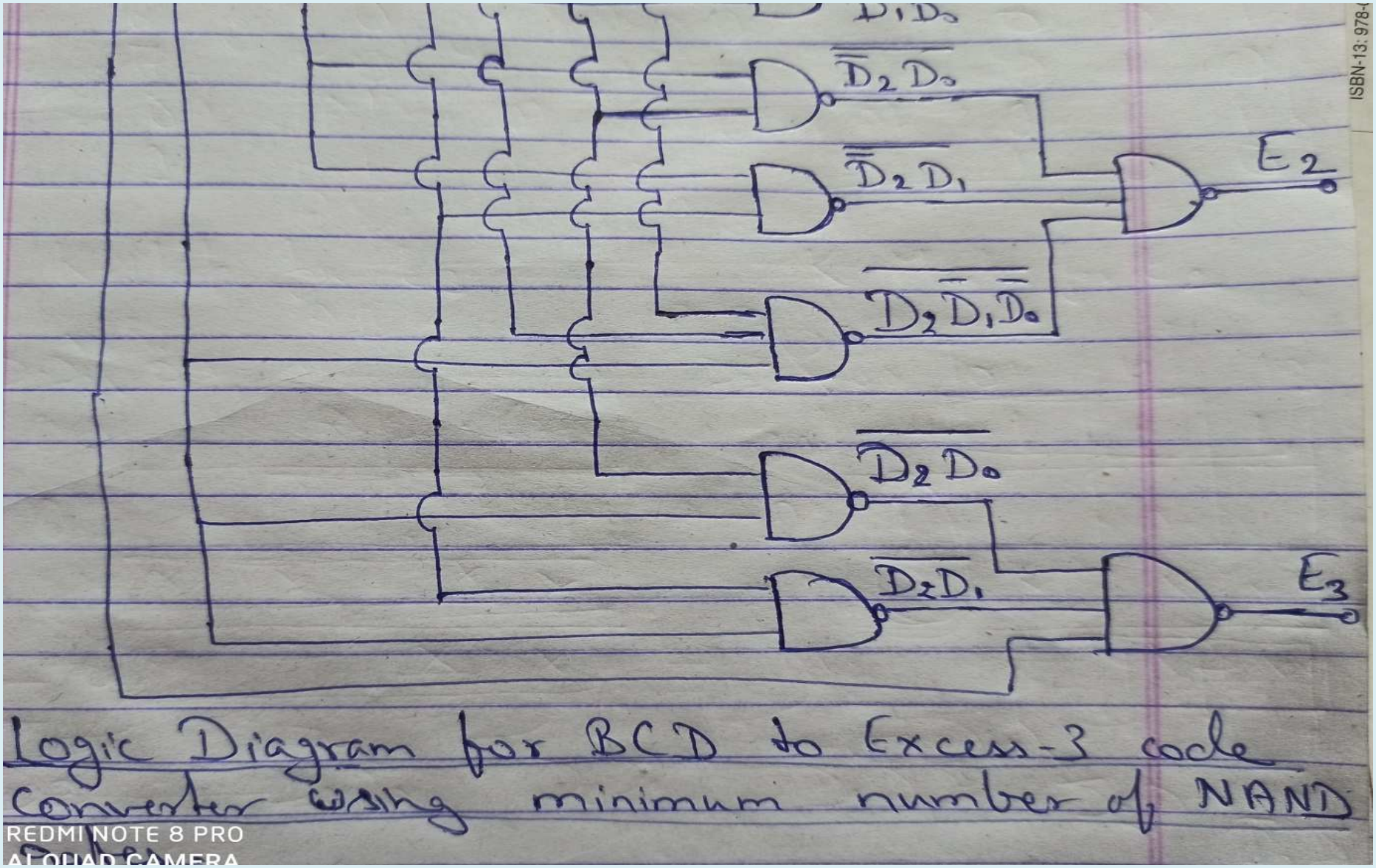
$$E_1 = \overline{\overline{D_1} \overline{D_0} + D_1 D_0} = \overline{\overline{\overline{D_1} \overline{D_0}} \cdot \overline{D_1 D_0}}$$

$$E_0 = \overline{D_0}$$

Discussion of 2018 Questions

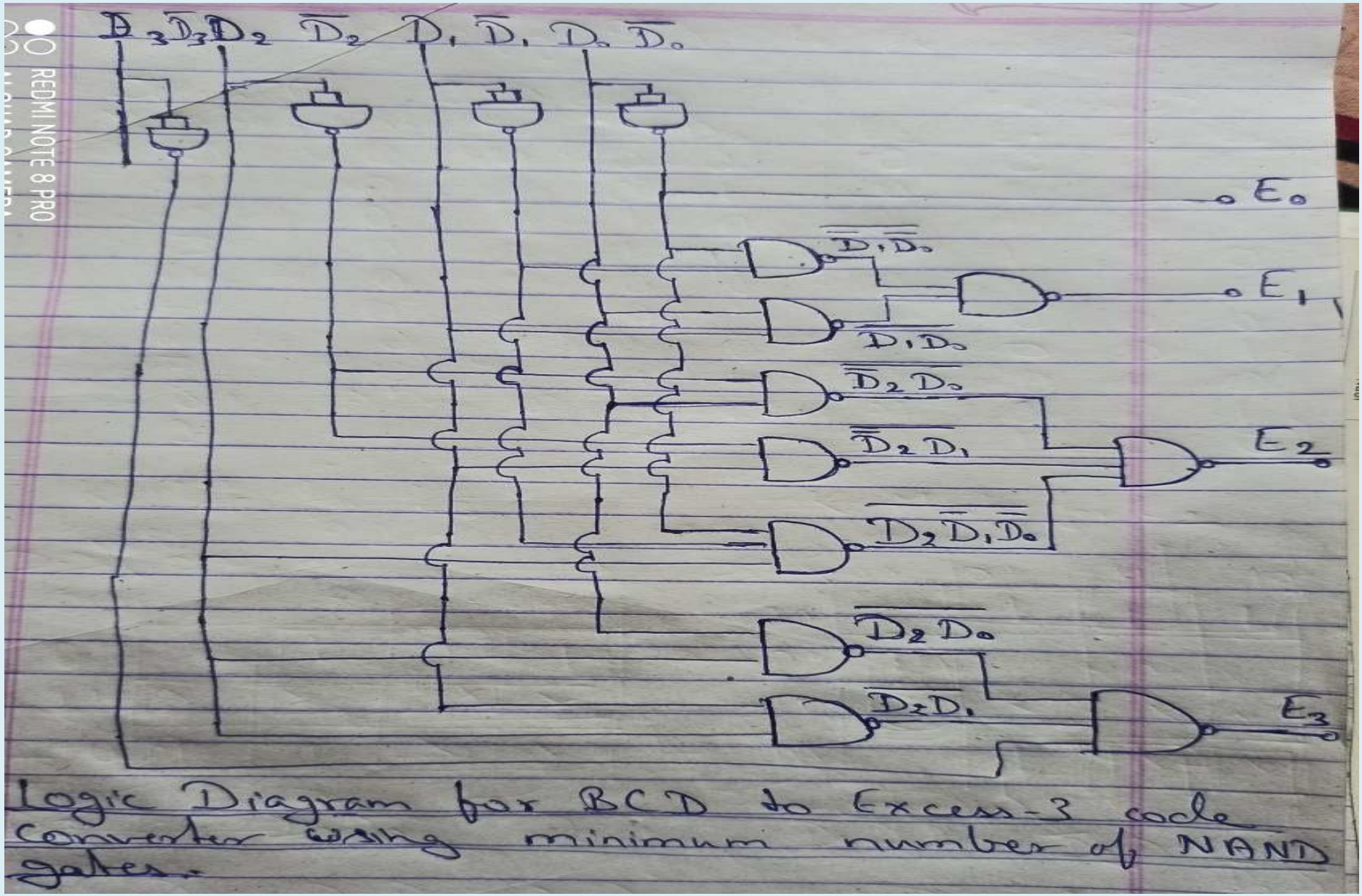


Discussion of 2018 Questions



Logic Diagram for BCD to Excess-3 code converter using minimum number of NAND

Discussion of 2018 Questions



Logic Diagram for BCD to Excess-3 code converter using minimum number of NAND gates.

Combinational Logic Design

Refer book- Modern Digital Electronics by RP Jain.

Thank You