



भारतीय प्रौद्योगिकी संस्थान दिल्ली
INDIAN INSTITUTE OF TECHNOLOGY DELHI

मुख्य परीक्षा उत्तर पुस्तिका
MAJOR TEST ANSWER BOOK

नाम
Name ... Harsit Mawandia

अनुक्रमांक
Entry No. 2020CS10348

पाठ्यक्रम सं.
Course No. C0L215 ग्रुप संख्या
Group No.

पाठ्यक्रम शीर्षक
Course Title Digital Logic and System Design

दिनांक
Date 20-11-2021

प्रयोग किए गए अनुवर्ती पृष्ठों की संख्या
No. of continuation sheets used

प्रश्न सं. Q. No.	प्राप्त अंक Marks
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
12.	
कुल TOTAL	

पाठ्यक्रम निर्धारक के हस्ताक्षर और दिनांक
Signature of Course Co-ordinator and date

अनुचित साधनों का प्रयोग करने वाले छात्रों को निलम्बित/निष्कासित किया जा सकता है।

Students using unfair means are liable to be punished by Suspension/Expulsion

परीक्षा केन्द्र में सेलफोन, काम्युनिकेटर्स व पीडीए साधनों का प्रयोग करना सख्त मना है।

Use of cell-phones, Communicators & PDAs in the Examination Hall is Strictly prohibited.

सभी पृष्ठों पर लिखें। Write on all pages.

Q3

Quinton McCluskey for

$$Y = f(a, b, c, d, e) = \sum m(2, 7, 6, 11) + \sum d(7, 15, 23, 31)$$

0				
1	2	(2, 3), (2, 6)		
2	3, 6	(3, 7), (3, 11), (6, 7),		
3	7, 11	(7, 15), (7, 23), (11, 15)		
4	15, 23	(15, 31), (23, 31)		
5	31			

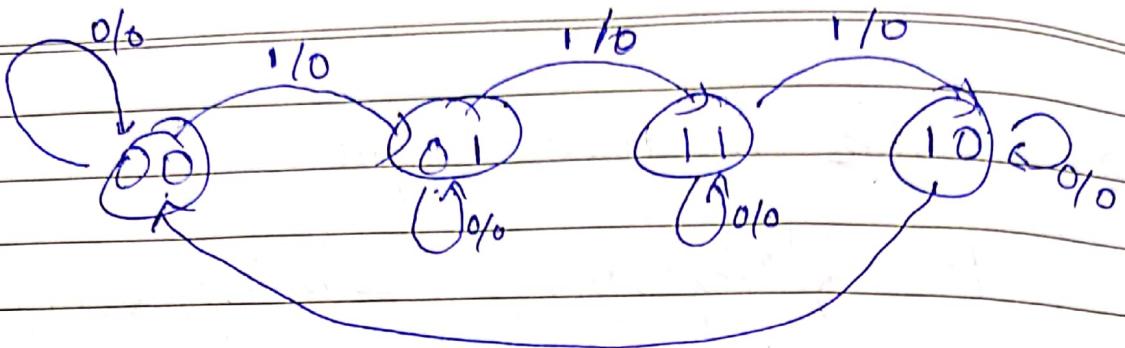
2	(2, 3)	(2, 6)	(3, 7, 11, 15)	(6, 7, 11, 15)	(7, 15, 23, 31)	(11, 15, 23, 31)
3						
6						
11						

$$f = (2, 3) + (6, 7, 11, 15)$$

$\hookrightarrow 00010 \quad 00011 \quad \hookrightarrow 001110 \quad 001111 \quad 011110 \quad 011111$

$y = a'b'c'd + a'd$

Q3)



		Present State	Next state ($y_2 y_1$)	y_1'	Output (z)
		00	00	1	0
		01	01	1	0
		11	11	0	0
		10	10	0	1

Now c is the clock

~~$y_2 y_1$~~

	00	01	11	10
y_1	0	0	1	1
	1	1	1	0

where $binary \quad w = y_2 y_1$

$$Y_1 = c y_2' + c' y_1 \leftarrow \text{For D flip flop}$$

$\text{For } T^{\ominus} = Y_1 \quad Y_1 = (c y_2' + c' y_1)'$

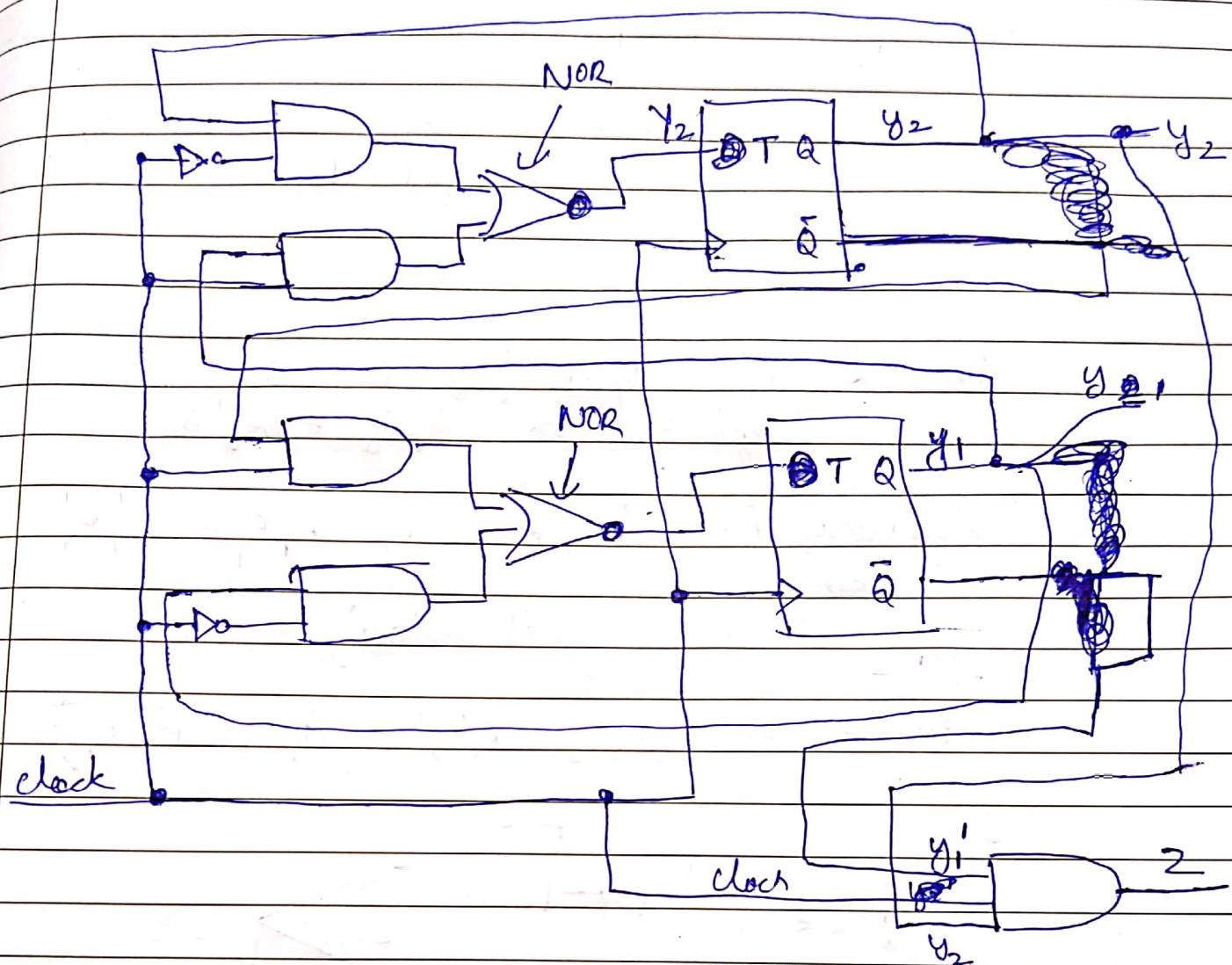
$c \cancel{y_2 y_1}$

	00	01	11	10
y_2	0	0	1	1
	1	0	1	0

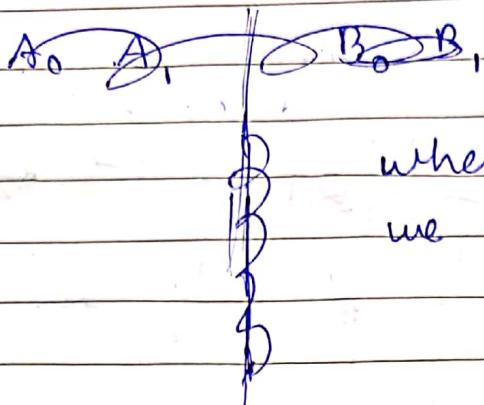
for D flip flop

$$Y_2 = (c' y_2 + c y_1)' \leftarrow \text{for T flip flop}$$

$$Z = c y_2 y_1' \quad (\text{By Observation})$$



Q4)



when $A_0 = B_0$ and $A_1 = B_1$,
we will get 1

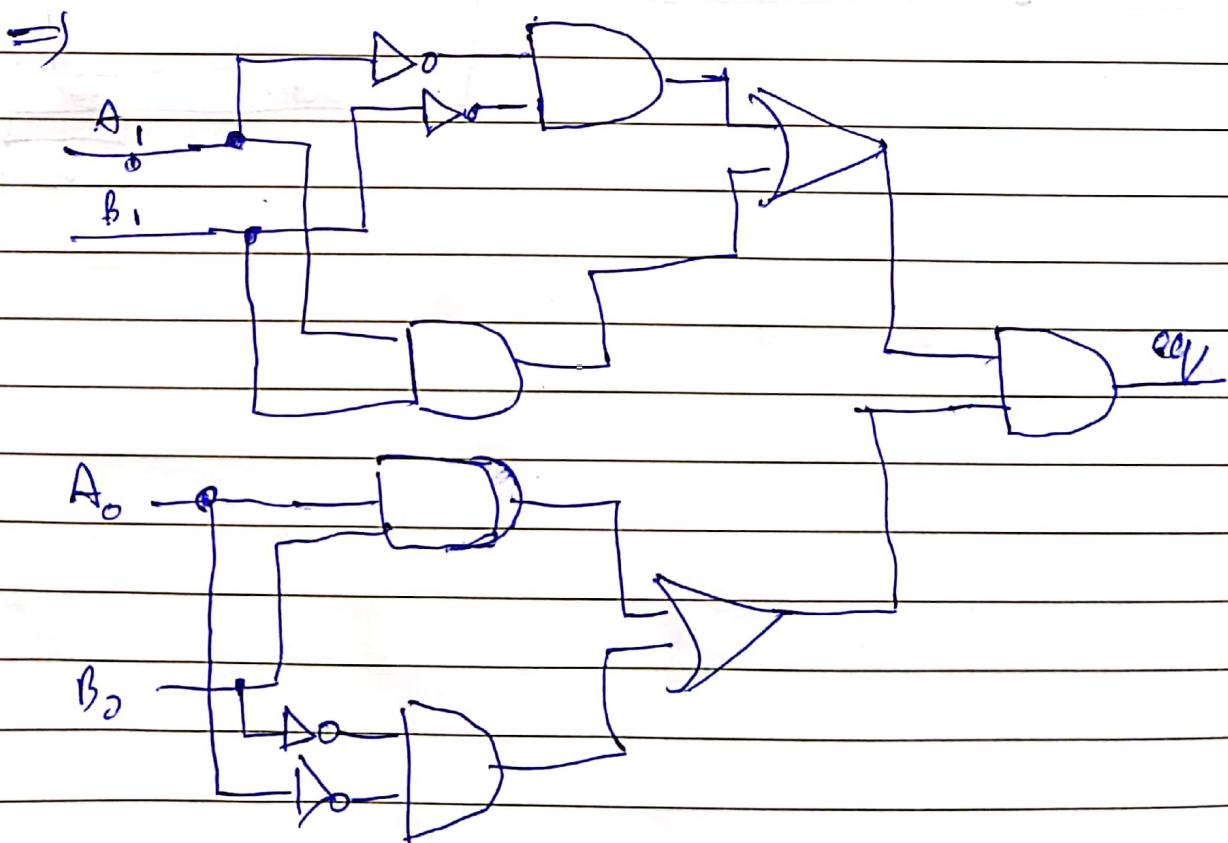
for $A_0 = B_0$ Output ($A_0 = B_0$)

$$\text{Logic} = A_0 B_0 + A_0' B_0'$$

Similarly for $(A_1 = B_1) \Rightarrow A_1 B_1 + A_1' B_1'$

Our final logic

$$= (A_0 B_0 + A_0' B_0') \cdot (A_1 B_1 + A_1' B_1')$$



(Q5)

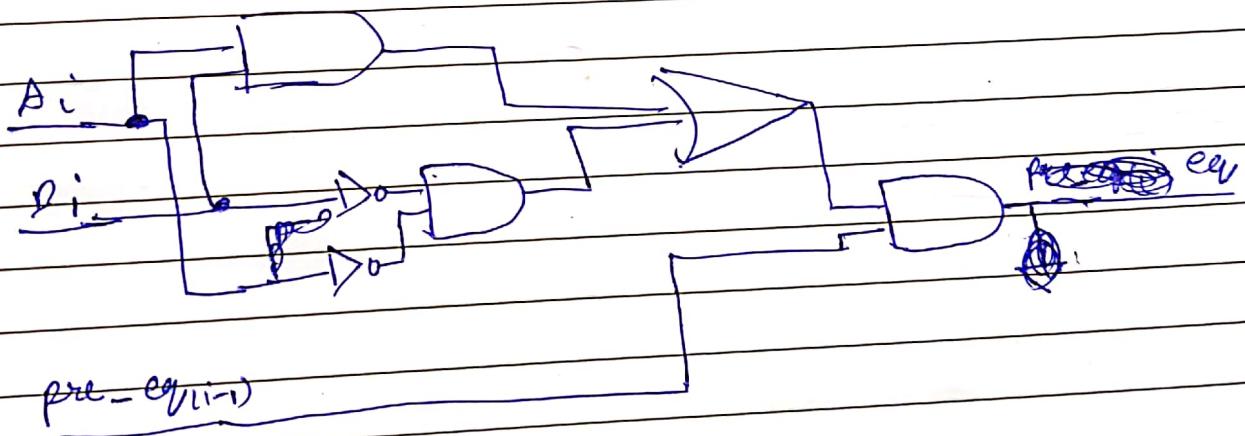
Similar to previous question
for $A_i = B_i$ logic = $(A_i B_i + A'_i B'_i)$

- For all previous outputs pre-eq_(i-1) is the output which tells if they are equal or not

So our final logic is

$$= (\text{pre-eq}_{(i-1)}) \cdot (A_i B_i + A'_i \cdot B'_i)$$

If ~~pre~~ our output = 1, then we set
pre-eq_(i) = 1 else we can say that
the strings are not equal



Q6 VHDL behavioral description of my module
CASACADE - EQ - COMP

entity CASACADE-EQ-COMP is
port (A, B, pre_eq : in bit;
eq : out bit);
end CASACADE-EQ-COMP;

architecture behaviour of CASACADE-EQ-COMP is
signal s: bit;

begin
 $s \leftarrow (A \text{ and } B) \text{ or } (\neg A \text{ and } \neg B);$
 $\text{eq} \leftarrow \text{pre_eq} \text{ and } s;$
end behaviour;

(Q7) v)

entity EQ-COMP8 is
port (a, b : in bit_vector (7 downto 0);
eqall : out bit);
end EQ-COMP8;

architecture structure of EQ-COMP8 is
component CASACADE-EQ-COMP

port (A, B, pre_eq : in bit; eq : out bit);
end component;

for all : CASACADE-EQ-COMP use entity ~~work~~
work.CASACADE-EQ-COMP (behaviour) b;
signal im : bit-vector (0 to 76);

begin

c0 : ~~CASACADE-EQ-COMP~~ port map (a(0), b(0), 1, im(0));
c1 to c6 : for i in 1 to 6 generate
c = ~~CASACADE-EQ-COMP~~ port map (a(i), b(i), im(i-1), im(i));
end generate;
c7 : ~~map~~ CASACADE-EQ-COMP port map (a(7), b(7), im(6), eqall);
end structure;

(P7) ii)

entity emailcounter is
 port (~~in~~ ~~out~~ ~~in~~ ~~out~~ ~~bit~~ ⁸¹⁹¹
~~vector~~ (~~8191~~ ~~1023~~ down to 0);
 count : ~~out~~ ~~in~~ ~~bit~~ ⁸¹⁹¹ integer);
 end email counter;

Architecture structure of emailcounter is
 component EQ-COMP8

port (a, b : in bit_vector (? down to 0);
 end EQ-COMP8;

for all: EQ-COMP8 use entity work.EQ-COMP8 (structure);

~~signal~~ signal count : integer = 0;

signal t : integer = 0;

signal a: bit; signal v: bit_vector(0 to 7);

begin

for i in ~~0~~ 0 to 1023 generate

~~EQ-COMP8~~ port map

for t in 0 to 7 generate {

~~v(t) ∈ x(ix8 + t);~~

}

c = EQ-COMP8 port map (v, bit_vector = <0, 0, 0, 0, 0, 0, 0, 1>);

begin if ~~(a = '1')~~ then

count = count + 1;

end if;

end generate;

end structure;

(8) We need three state machines: M₁, M₂ and M₃

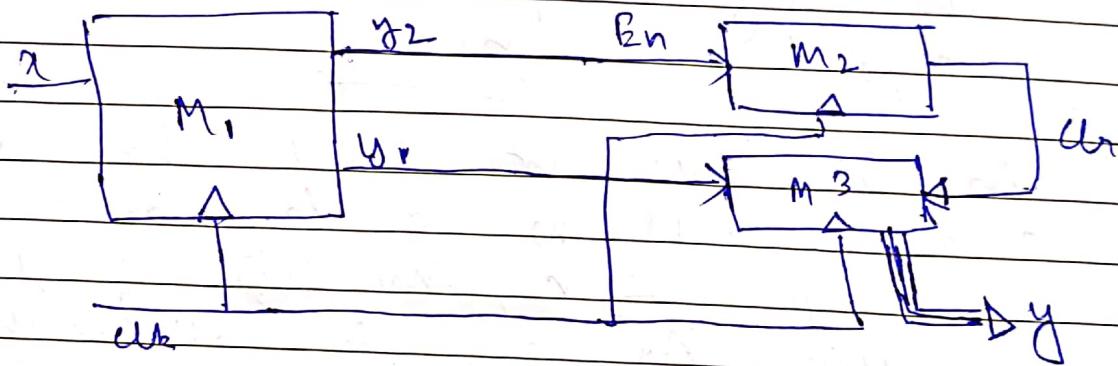
(using PLA-Comp 8)

- M₁: Pattern recognizer for "@" character
- M₂: 10-bit counter for counting 1024 characters
- M₃: 10-bit counter for counting no. of "@"

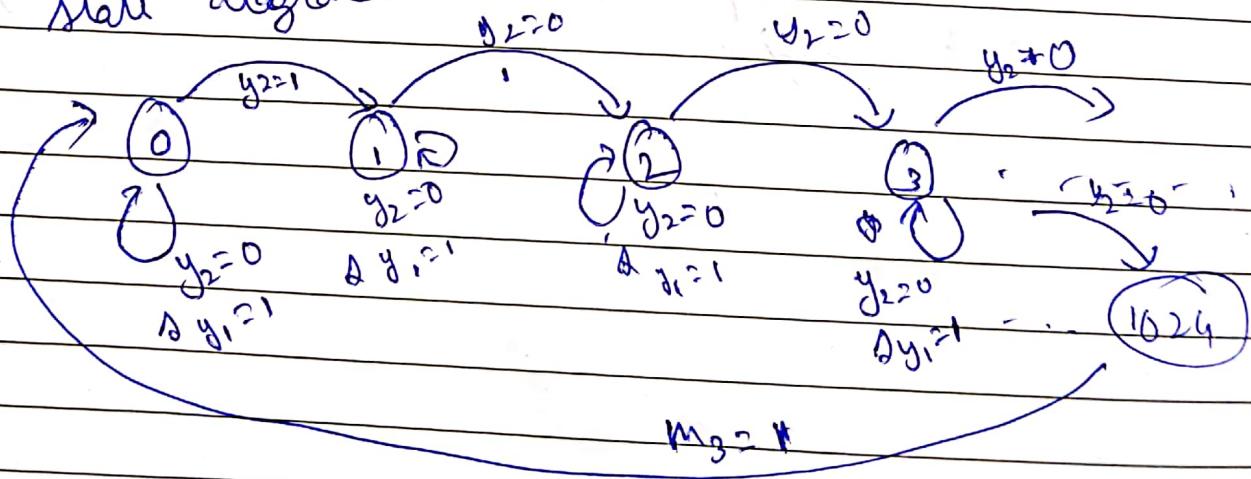
specification of M₁

$$y_1 = 1 \text{ if } \langle x(6-7 \dots x(r)) \rangle = "@" \text{ and } t \bmod 8 = 7$$

$$y_2 = 1 \text{ if } t \bmod 8 = 7$$



M₂ State diagram

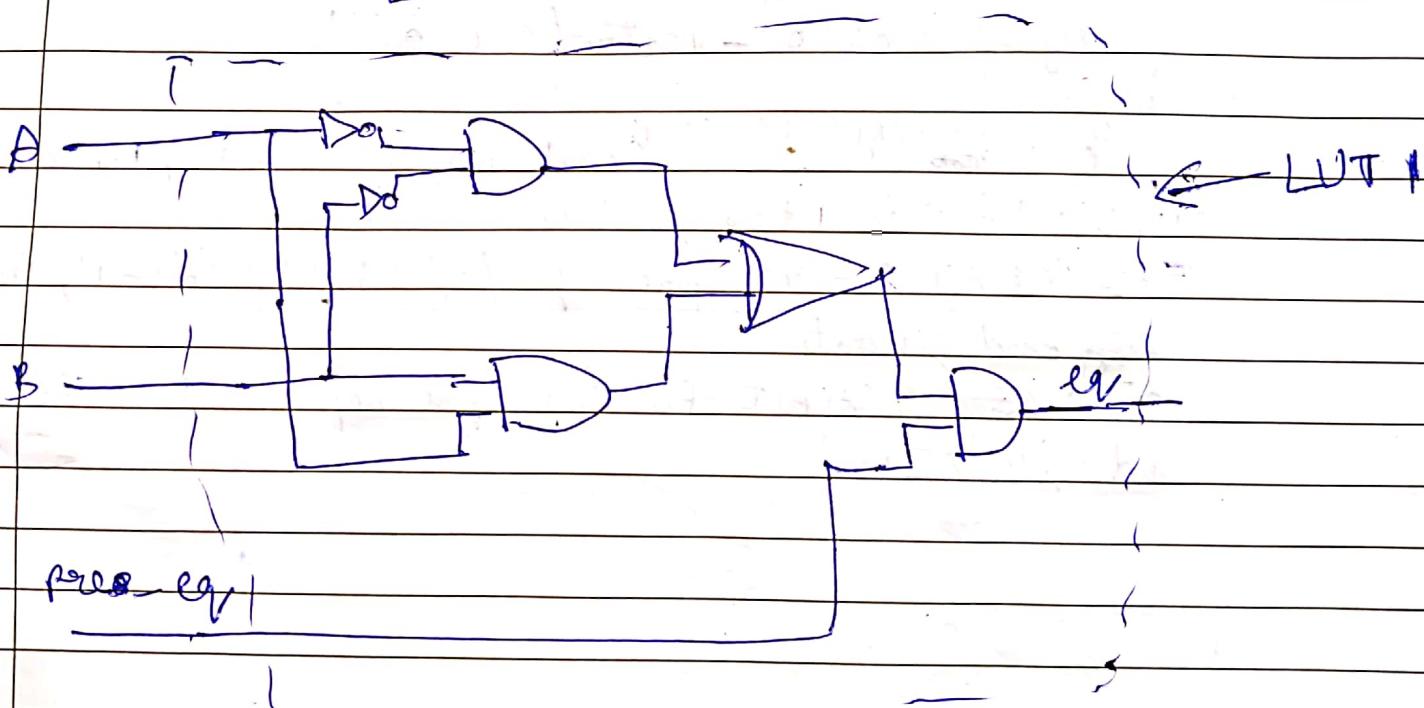


Q9) The total number of inputs for CASCADe - EQ - comb = 3 ($A, B, \text{pre-eq}$)

Therefore the number of function 3B-input LUTs required = ①

~~Logic to be implemented =~~

$$(\text{pre eq}) \cdot (A \cdot B + A' \cdot B')$$



pre_eq A B eq

0	x	x	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Q10)

FPGA is a device that contains ~~an LUT and the~~ LUT, say of ~~n input~~ n input variables. The number of functions that can be implemented by this n-input LUT = 2^{2^n} .

Now, the bit file of the FPGA describes which function out of these 2^{2^n} do we need using a series of bits. And therefore the size of the bit file only depends on 2^{2^n} which depends on n only.

Do it does not matter how complex our circuit is as long as the number of inputs $\leq n$, the size of bit file remains the same as all such function can be implemented in the bit file of that size.