



# Indian Institute of Information Technology Vadodara

End semester, Autumn/ Winter examination

B.Tech/ M.Tech/ Research student

(Strike off non applicable)

Course Code: ECE201 Course Name: Digital Logic Design Date: 07/01/2022

Candidate Name: Archit Agrawal Student ID: 2020851213

Number of Supplementary booklets:-- 1/2/3

## Read the instructions carefully

- 1 Listen to the instruction stated by invigilator carefully. It may be in addition to mentioned on answer sheet / question paper.
- 2 It is mandatory to present your ID card to the invigilator.
- 3 Answer new question in a new page.
- 4 Possession of books, notebook, data storage device, scanner, mobile phone is considered as malpractice in examination hall (scientific, non programmable calculator are permitted) unless specified by the course instructor.
- 5 Any type of communication or request for stationery items such as scale, pencil, eraser to other examines during exam will be treated as unfair means.
- 6 Don't write anything except your roll number on question paper unless specifically instructed.
- 7 At the end of exam, leave the examination hall quickly and quietly.

Question No.	Marks
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
12.	
13.	
14.	
Total	



## Pledge

I shall abide by rules and regulation of Institute. I affirm that I will not take any unauthorized help during exam.

Student's Signature Archit Agrawal

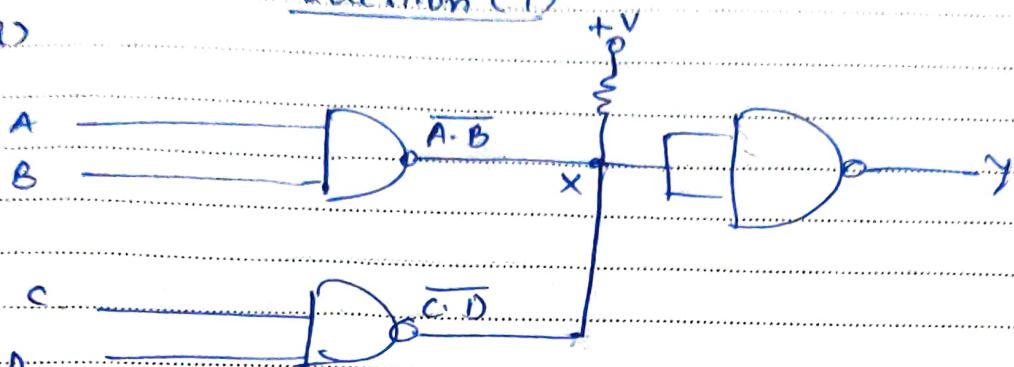
Information Verified

Invigilator's Signature Dibyendu Roy



(a)

Question (1)



A + point X,

$$x_0 = (\overline{A \cdot B})(\overline{C \cdot D}) \cdot 1$$

$$x_0 = (\overline{A \cdot B})(\overline{C \cdot D})$$

Now,

$$\cancel{x} = \cancel{(x_0 + x_0)}$$

$$y = \overline{(x_0 + x_0)}$$

$$y = \overline{x_0} + \overline{x_0}$$

$$y = \overline{x_0}$$

$$y = \overline{((\overline{A \cdot B}) + (\overline{C \cdot D}))}^B$$

$$\boxed{y = AB + CD} \quad \text{Ans}$$



Archit Agrawal  
202051213  
Date: 11/12/2020

### Question 1 (b)

$$F = P \otimes R \otimes S + P \otimes R' \otimes S + P \otimes R \otimes S' + P \otimes R' \otimes S' + P \otimes Q \otimes R \otimes T \\ + P \otimes Q \otimes R' \otimes T' + P \otimes Q \otimes R' \otimes T$$

$$F = P \otimes Q \otimes R (S + S') + P \otimes Q \otimes R' (S + S') + P \otimes Q \otimes S \otimes T (R + R') \\ + P \otimes Q \otimes R' \otimes S' \otimes T'$$

$$F = P \otimes Q \otimes R + P \otimes Q \otimes R' + P \otimes Q \otimes S \otimes T + P \otimes Q \otimes R' \otimes S' \otimes T' \quad (\because A + \bar{A} = 1)$$

$$F = P \otimes Q (R + R') + P \otimes Q \otimes S \otimes T + P \otimes Q \otimes R' \otimes S' \otimes T' \quad (\because A + \bar{A} = 1)$$

$$F = P \otimes Q (1 + S \otimes T + R' \otimes S' \otimes T')$$

$$\boxed{F = P \otimes Q} \quad (\because 1 + X = 1)$$

Question 2

$$(a) z(p, q) = \sum(0, 2)$$

	$p \backslash q$	0	1
0		1	0
1		1	0

For POS form, we will group 0's together.

$$\therefore z(p, q) = q'$$

∴ POS of  $z(p, q) = q'$

$$\text{Now, } z(p, q) = \sum(0, 2)$$

$$\therefore z'(p, q) = \sum(1, 3)$$

	$p \backslash q$	0	1
0		0	1
1		0	1

∴ POS of  $z'(p, q) = q$

	$p \backslash q$	0	1
0		0	1
1		0	1

and, SOP of  $z'(p, q) = q$

(b) To make a MOD-n binary counter, we need at least  $\log_2(n)$  flip flops.

For  $n = 32$ ,

$$\log_2 n = 5$$

5 flip flops are required to design a MOD 32 binary counter.

### Question 3

(a) The flip flop used here is a positive edge triggered D flip flop, which means that only at "rising edge of clock" flip flop will capture the input provided at P and give the output at Q.

The boolean expression for logic gate is

$$D = A \times + X' Q'$$

Now, In 1st clock period, the clock has rising edge at  $t=0$ , hence flip flop will change its state.

In the first clock  $X=1$ , ~~and~~ so,  $D=A$ .

Now, A logic line may have different levels at different clock periods, therefore we have to answer w.r.t. the  $i^{th}$  clock period where  $A_i$  is the logic level.



Archit Agrawal

202051213

### Question 3.

In the first clock period value of  $X$  is same as in the 6th clock, i.e. logic 1.  
So,  $X = 1$  and  $A = A_0$ .

$$\therefore Y = A_0$$

$\Rightarrow$  In 2nd clock period:

$X = 1$  (value in previous clock)

$$\therefore Y = A_1$$

$\Rightarrow$  In 3rd clock period

$X = 0$ , so  $D = 0' = A_1'$

$$\therefore Y = A_1'$$

$\Rightarrow$  In 4th clock period

$X = 1$ , so  $D = A_3$

$$\therefore Y = A_3$$

$\Rightarrow$  In 5th clock period

$X = 1$ , so  $D = A_4$

$$\therefore Y = A_4$$

Hence, output sequence is  $A_0 A_1 A_1' A_2 A_4$ .



## (b) XOR Gate:

### Design Code:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity xorg is

Port ( a : in STD\_LOGIC;

      b : in STD\_LOGIC;

      f : out STD\_LOGIC);

end xorg;

architecture behavior of xorg is

begin

    f <= a xor b;

end behavior;

### Testbench Code:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity xorg\_test is

end xorg\_test;

architecture behavior of xorg-test is

component xorg

Port ( a : in STD\_LOGIC;

      b : in STD\_LOGIC;

      f : out STD\_LOGIC);

end component;



Archit Agrawal  
202051213

signal a : std-logic := '0';  
signal b : std-logic := '0';

Signal f : std-logic;

Begin

unit: xorg port map (

a => a

b => b

f => f

);

start proc : process

begin

wait for 1ns;

a <= '0';

b <= '0';

wait for 1 ns;

a <= '1';

b <= '0';

wait for 1ns;

a <= '0';

b <= '1';

wait for 1ns;

a <= '1';

b <= '1';

wait;

end process;

end;

## XNOR Gate:

### Design Code:

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;
```

```
entity xnorg is  
Port ( a : in STD-LOGIC;  
      b : in STD-LOGIC;  
      f : out STD-LOGIC);
```

```
end xnorg;
```

architecture behavior of xnorg is

```
begin  
  if  $a \neq b$  then  
    f <= '1'  
  else  
    f <= '0';
```

```
end behavior;
```

### Testbench Code:

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;
```

```
entity xnorg_test is  
end xnorg_test;
```

architecture behavior of xnorg-test is

```
component xnorg
```

```
Port ( a : in STD-LOGIC;
```

```
      b : in STD-LOGIC;
```

```
      f : out STD-LOGIC);
```

```
end component;
```

Archit Agrawal  
202051213

signal a : std-logic := '0';  
signal b : std-logic := '0';  
signal f : std-logic;

begin

unit : xilinx part map (

a  $\Rightarrow$  a

b  $\Rightarrow$  b

f  $\Rightarrow$  f

) ;

stim-proc : process

begin

wait for 1ns;

a <= '0';

b <= '0';

wait for 1ns;

a <= '0';

b <= '1';

wait for 1ns;

a <= '1';

b <= '0';

wait for 1ns;

a <= '1';

b <= '1';

wait;

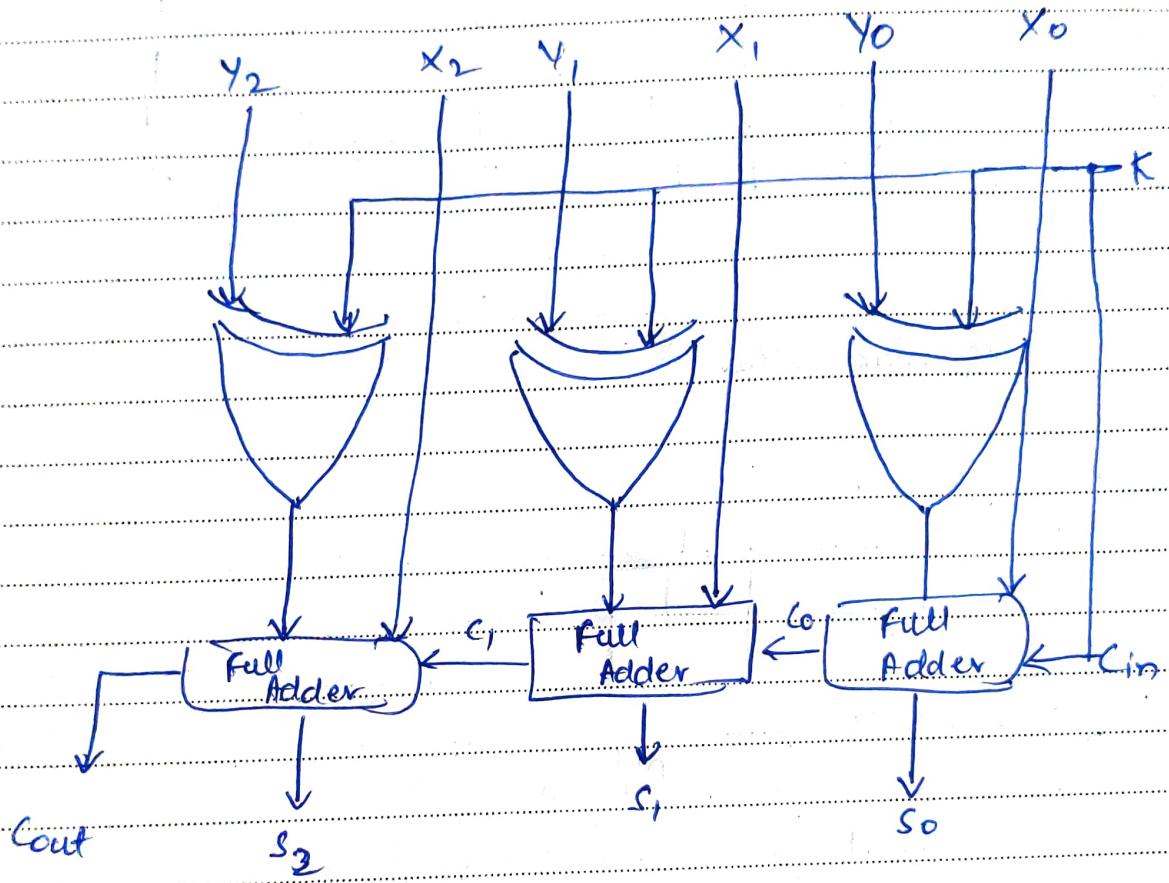
end process;

end;

Question 5

(a) Phone number is 8419037703.

i. 3-bit binary adder subtractor.



The control line  $K$  determines whether the operation being performed is either subtraction or addition. This is done by binary values 0 and 1.



Archit Agrawal

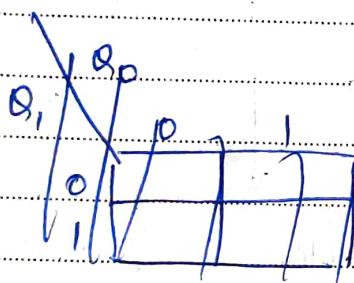
202051213

(b)

Using The excitation table for T flip-flop:

$Q_1$	$Q_0$	$Q_1'$	$Q_0'$	T <sub>1</sub>	T <sub>2</sub>
0	0	0	1	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	0	0

K-map for T<sub>1</sub>



Clearly,

$$T_1 = Q_1' Q_0$$

$$\text{and } T_2 = Q_1' + Q_0'$$