NAME :- Bilal Sayid
Program :- B-Tech
Semester :- 3rd.
Roll No :- 20BCS057
UNique Paper Code :- CEN-304
Paper Title :- Digital Logic Theory.
Date :- 01/01/2022
Time of Exam :- 10:00 A.M.
Pages :-

1)a) Given hexadecimal number → 7DE
To convert into decimal, we multiply digits with corresponding powers of 16.

$D_{16} \Rightarrow 13_{10}$ $E_{16} \Rightarrow 14_{10}$

$7DE_{16} = 7 \times 16^2 + 13 \times 16^1 + 14 \times 16^0$

$= 1792 + 208 + 14$

$= 2014_{10}$

Given octal number => 2014
To convert into decimal, we multiply digits with powers of 8.

$7014_8 = 7 \times 8^3 + 0 \times 8^2 + 1 \times 8^1 + 4 \times 8^0$

$= 3584 + 0 + 8 + 4$

$= 3596_{10}$

20BCS057

→ ADD -1010 and -0110 in 5bit register. Perform 1111 x 1010.

To add -1010 and -0110 in 5 bit register. the representation of negative numbers is done using 2's complement.

∴ 2' complement of 01010 ⇒ 10101
$(10)_{10}$ + 1
10110
$= (-10)_{10}$

2's complement of 00110 $(6)_{10}$
⇒ 11001
+ 1
11010 $= (-6)_{10}$

Adding the 2 -ve no.s

(-10) 10110
+(-6) + 10110
10000

finding magnitude of 10000

⇒ 10000 → flip bits → 01111
→ 1
10000 ⇒ 16

∴ Add -1010 and -0110 ⇒ 10000 (-16)
(-10) (-6)

20BCS5057

Performing 1111 x 1010

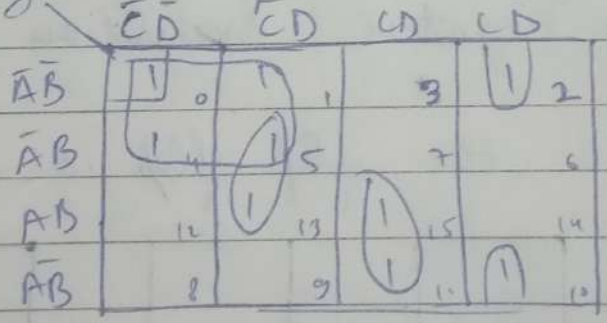
```
    1111
    1010
  ------
    0000
   1111x
  0000xx
 1111xxxx
--------
10010110   Ans.
```

$(10010110)_2 = (150)_{10}$

1) b)

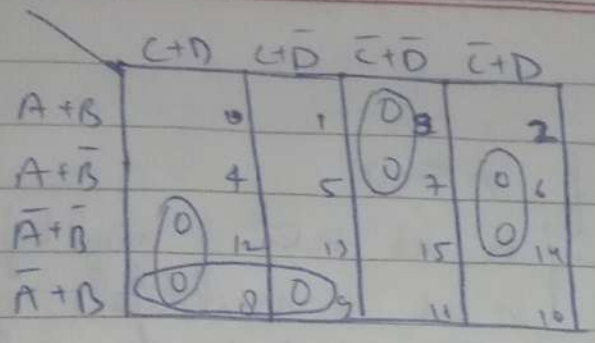
i) $F(A,B,C,D) = \Sigma m(0,1,2,4,5,10,11,13,15)$.

using k-map



$F_{(SOP)} = \bar{A}\bar{C} + B\bar{C}D + ACD + \bar{B}C\bar{D}$

$F(A,B,C,D) = \Pi m(3,6,7,8,9,12,14)$

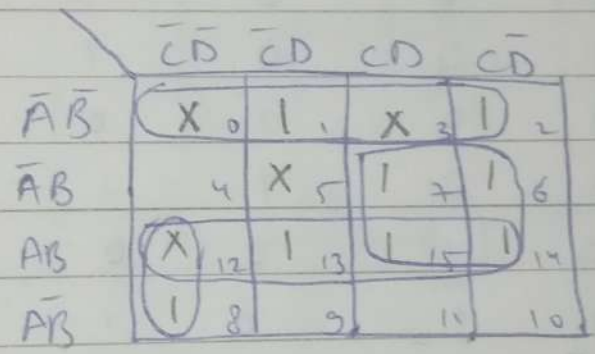


$F_{(POS)} = (\bar{A} + C + D) \cdot (\bar{A} + B + C) \cdot (A + \bar{C} + \bar{D}) \cdot (\bar{B} + \bar{C} + D)$

2)

a) $F(A, B, C, D) = m(1, 2, 6, 7, 8, 13, 14, 15) + d(0, 3, 5, 12)$.

Using k-map =>



$F_{(SOP)} = \bar{A}\bar{B} + AB + BC + A\bar{C}\bar{D}$

Don't Care conditions

Don't care condition are those input conditions whose output is not our concern. i.e., "we don't care" for those outputs. So we assume those outputs to be 0 or 1 according to our calculation need. It helps in reducing expressions in k-maps.

20BCS5057

2)

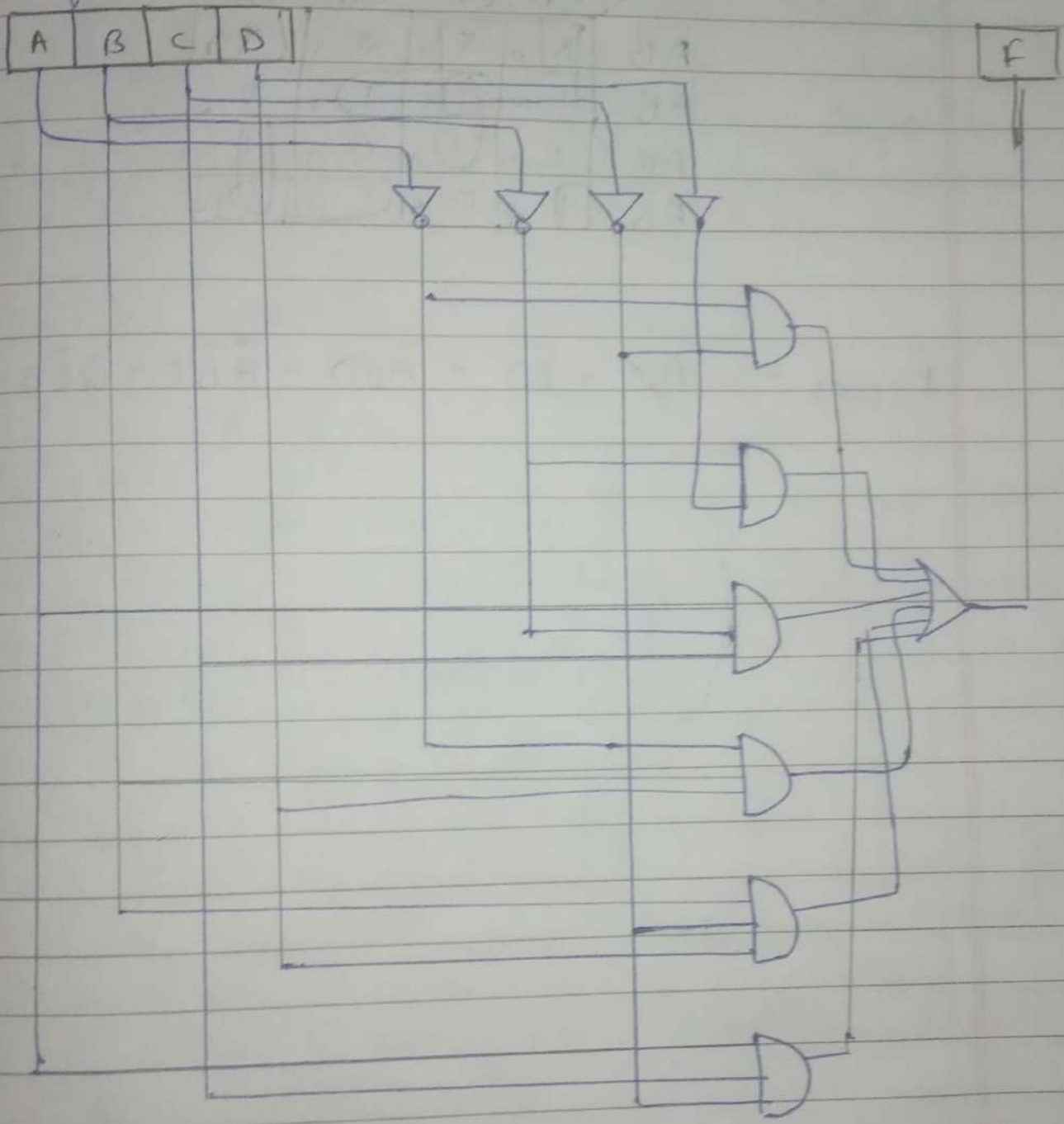
b) $F = x$, if $n \leq 2$
1 , if n is not a multiple of 3
0 , if n is a multiple of 3
(n is the 4 bit input binary number).

Truth table for the following function is as below.

A	B	C	D	n	F
0	0	0	0	0	X
0	0	0	1	1	X
0	0	1	0	2	X
0	0	1	1	3	0
0	1	0	0	4	1
0	1	0	1	5	1
0	1	1	0	6	0
0	1	1	1	7	1
1	0	0	0	8	1
1	0	0	1	9	0
1	0	1	0	10	1
1	0	1	1	11	1
1	1	0	0	12	0
1	1	0	1	13	1
1	1	1	0	14	1
1	1	1	1	15	0

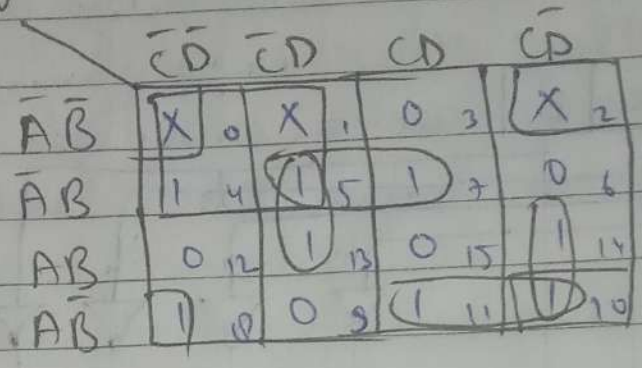
Page No. 6
Date

Logical diagram for the given function is as follows:-



20BCS057

k- map for the given function is as follows:-



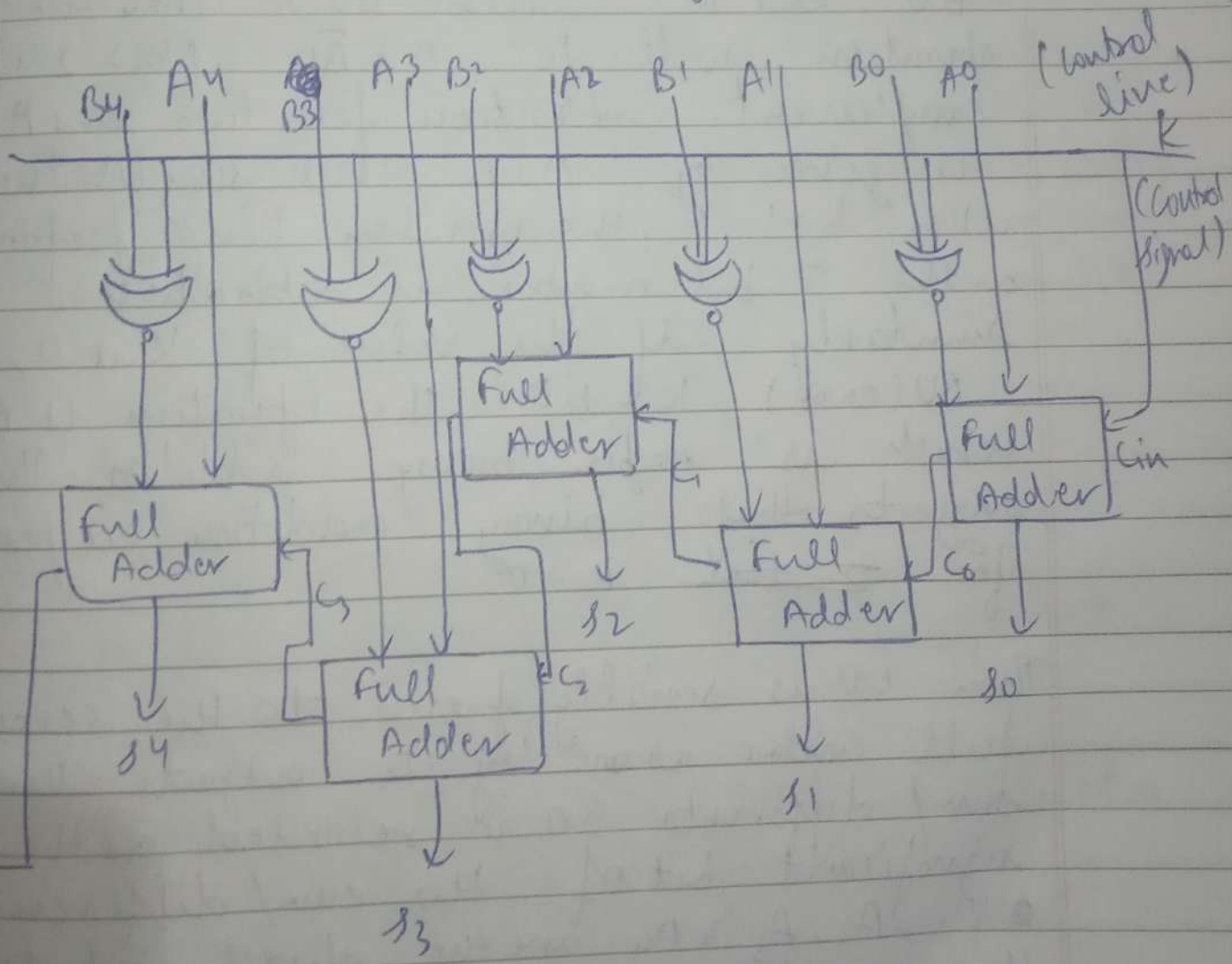
$$F_{(SOP)} = \bar{A}\bar{C} + \bar{B}\bar{D} + \bar{A}BD + \bar{A}\bar{B}C + B\bar{C}D + AC\bar{D}$$

Page No. 8
Date:

3)

a) A Binary Adder-Subtractor is one which is capable of both addition and subtraction of binary numbers in one circuit itself. The operation being performed depends upon the binary value the control signal holds.

A0 A1 A2 A3 A4 for A
B0 B1 B2 B3 B4 for B



20BCS057

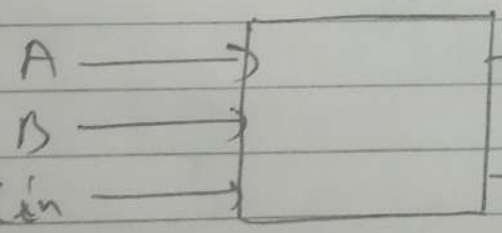
As shown in the diagram, the first full adder has control line directly at its input (input carry Cin) the input A0 (least significant bit of A) is directly input in the full adder. The 3rd input is the exor of B0 & k.
The two outputs produced are sum/difference (S0) and carry (C0).

If the value of k is 1, the output of B0, exor k will be $\bar{B0}$ & Then the operation will be $A + \bar{B0}$. Now 2's complement subtraction for two nos A & B is given by $A + \bar{B}$. This suggests that when $k = 1$, the operation being performed on the 5 bit number is subtraction.
Similarly, if the value of K is 0, B0 (exor) k & B0. The operation is A+B, which is simple binary addition. Thus suggests that binary addition is performed for ~~5 bit~~ $k = 0$.

Then C0 is serially passed to the second full adder as one of its outputs. The sum/ differents S0 is recorded as the least significant bit of the sum/ difference.
A1, A2, A3 & A4 are the direct inputs to the second, third, fourth & fifth full adders.

Then, the 3rd inputs are B_1, B_2, B_3 & B_4, exored with k to the 2nd, 3rd, 4th & 5th full adder respectively. The carry c_1, c_2, c_3 are passed serially to full adderss as inputs. C_4 becomes the total carry to the sum/ difference. $S1$, $S2$, $S3$, $S4$ are recorded to force the result with SO.

3) b) Full adder is the adder which adds 3 inputs and produces two outputs. The first two inputs are A & B and the 3rd input is an input carry C-in the output carry is designed as C-out and the normal output is designated as S which is SUM.



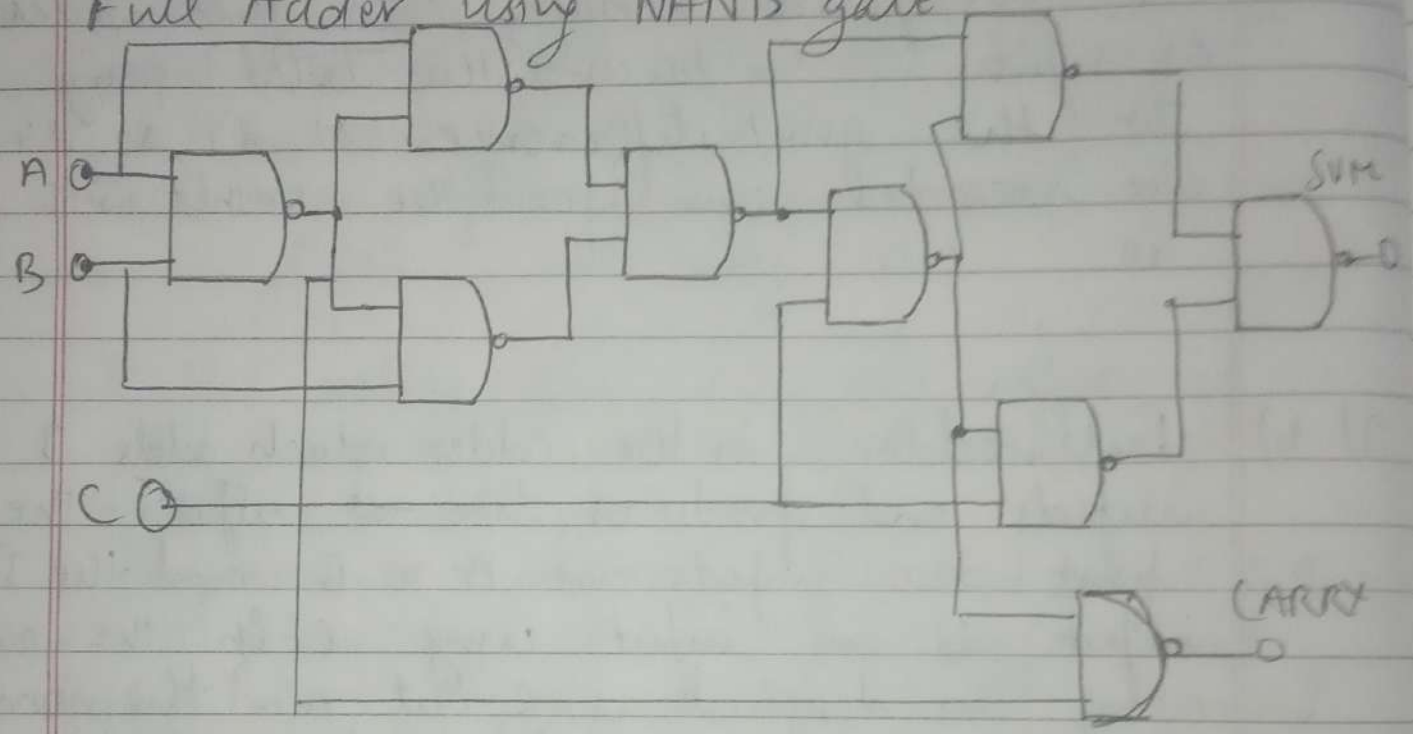
A full adder can be used as a part of many other larger circuits like :-

1- Ripple carry adder, it adds n-bits at a time.
2- Carry out Multiplication, the dedicated multiplication circuit uses it.

20BCS057

3- ALU - Arithmetic Logic Unit (One of the circuits is a full adder).

Full Adder using NAND gate

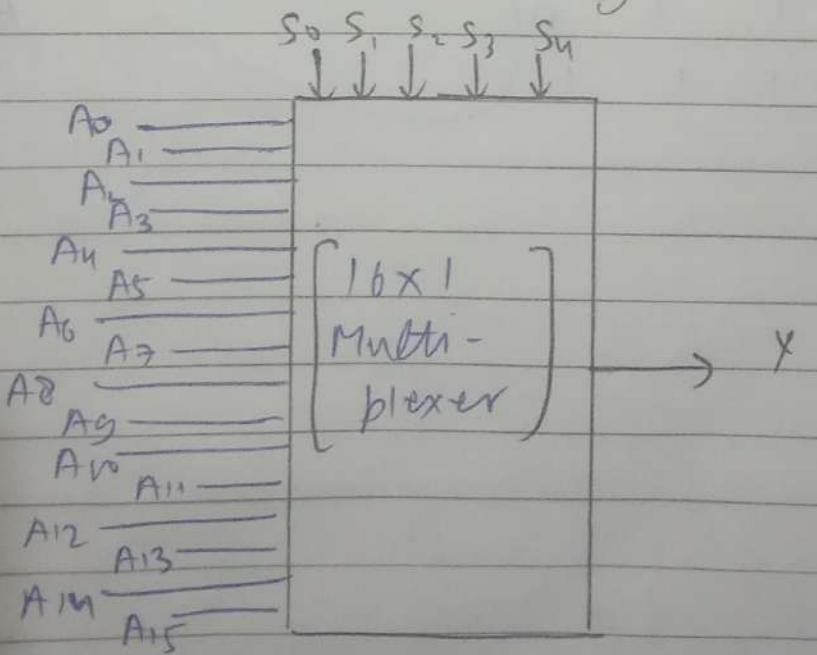


4) a) **Uses of multiplexers :-**

A multiplexer is a combinational circuit that has 2^4 input lines and a single output line. Output will be selected from one of the inputs lines based on the selection lines.
A multiplexer makes it possible for several input signals to share one device or resource. For example, one analog-to-digital conversion or one communications transmission medium., instead of having one device per input signal.

→ Multiplexers can also be used to implement boolean functions of multiple variables.

16 - 1 Multiplexer block diagram



20BCS057

Truth Table

Inputs				Output
S_0	S_1	S_2	S_3	Y
0	0	0	0	A_0
0	0	0	1	A_1
0	0	1	0	A_2
0	0	1	1	A_3
0	1	0	0	A_4
0	1	0	1	A_5
0	1	1	0	A_6
0	1	1	1	A_7
1	0	0	0	A_8
1	0	0	1	A_9
1	0	1	0	A_{10}
1	0	1	1	A_{11}
1	1	0	0	A_{12}
1	1	0	1	A_{13}
1	1	1	0	A_{14}
1	1	1	1	A_{15}

20BCS057

4)

b) **Applications of Comparators**

1) Comparators are used in central processing units (CPUs) and microcontrollers (MCUs)
2) These are used in control applications in which the binary numbers representing physical variables such as temperature, position etc. are compared with the reference value.
3) Comparators are also used as process controllers and for Servo motor control.
4) Used in password verification and biometric applications.

TRUTH TABLE

INPUT				OUTPUT		
A1	A0	B1	B0	A<B	A=B	A>B
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0

contd. on next page.

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

K Maps

A > B

A_1A_0 \ B_1B_0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

A = B

A_1A_0 \ B_1B_0	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	1	0
11	0	0	0	1

A < B

A_1A_0 \ B_1B_0	00	01	10	11
00	0	1	1	1
01	0	0	1	1
10	0	0	0	0
11	0	0	1	0

The logical expressions from K-maps are →

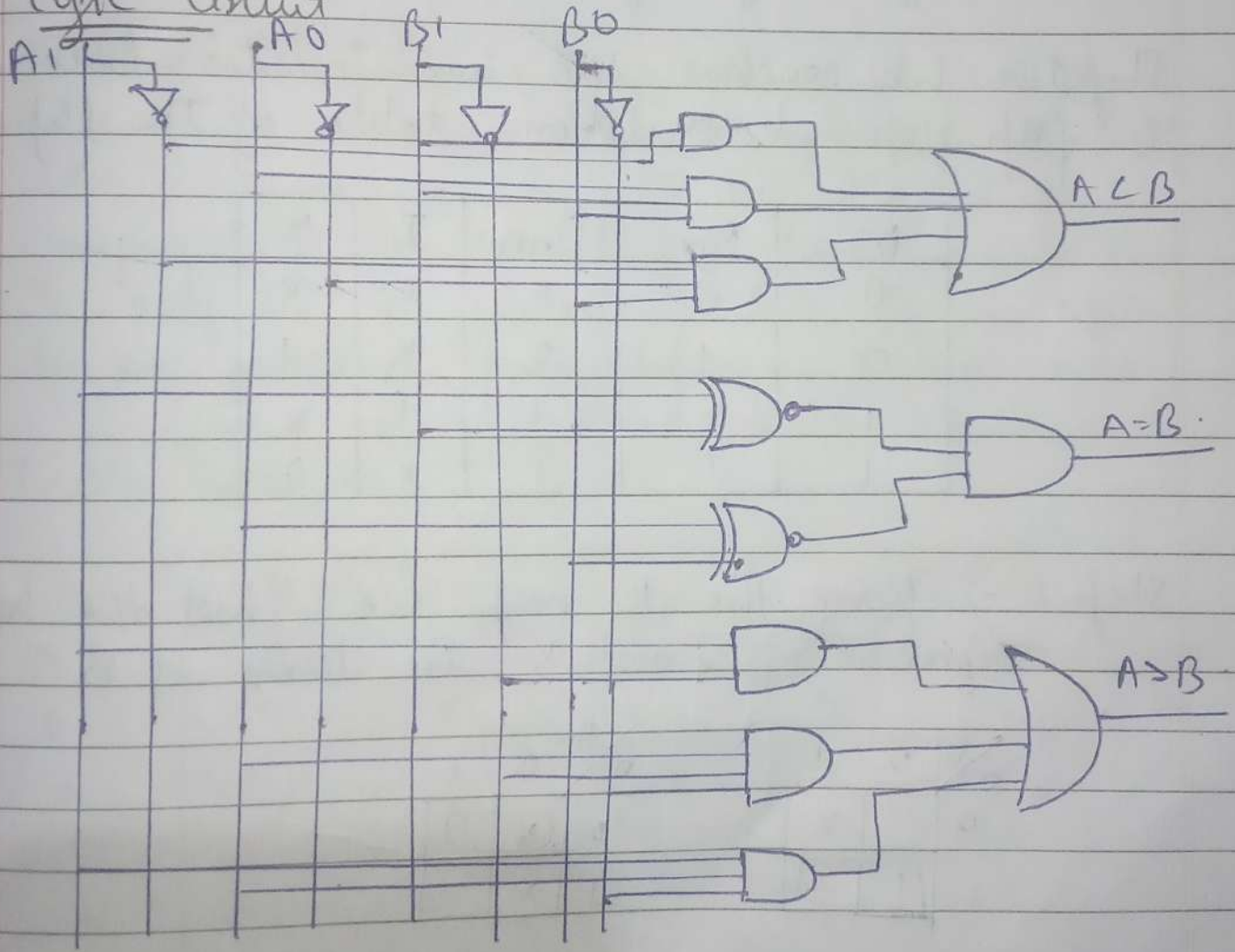
$A > B$: $A_1B_1' + A_0B_1'B_0' + A_1A_0B_0'$

$A = B$: $A_1'A_0'B_1'B_0' + A_1'A_0B_1'B_0 + A_1A_0B_1B_0 + A_1A_0'B_1B_0'$

: $(A_0 \text{ ExNOR } B_0)(A_1 \text{ ExNor } B_1)$

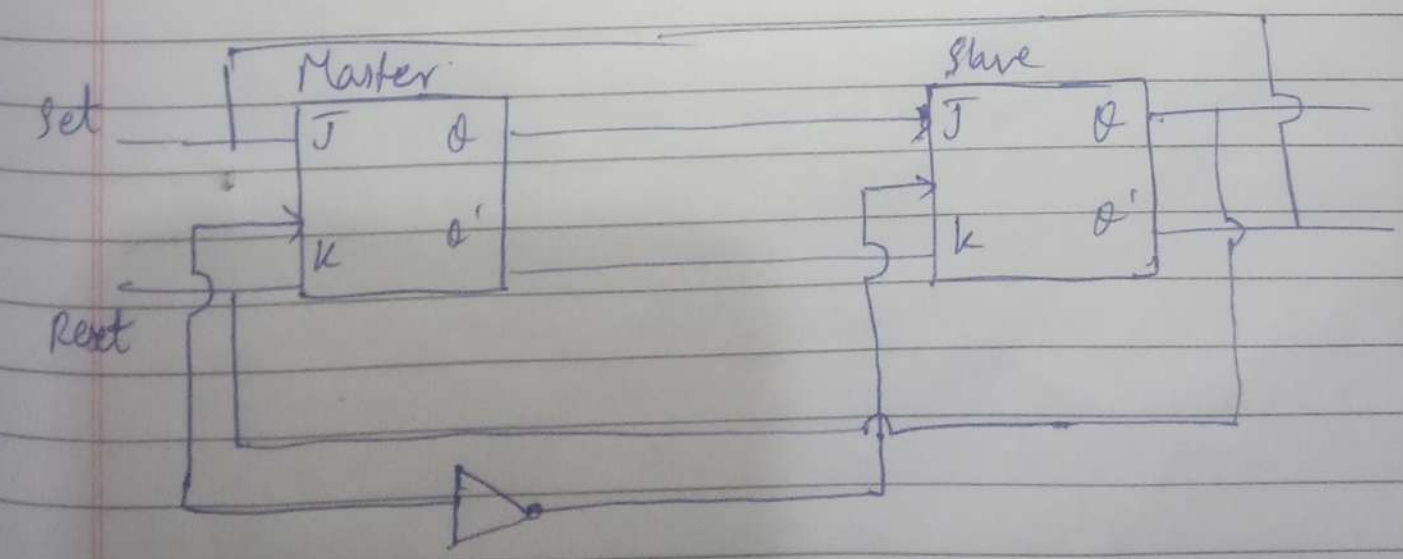
$A < B$: $A_1'B_1 + A_0'B_1B_0 + A_1'A_0'B_0$.

Logic Circuit



5) a) In a JK flip flop, when $j=k=1$ and clock is applied, the output goes on complementing every delay time of flip flop as long as block is present. Therefore, the output at the end of the clock pulse is ambiguous. This condition is known as race around condition.

Master Slave Flip Flop is a combination of two J-K flip flops connected together in a series configuration. In this flip flop, the first flip flop responds to the data input when the clock is high, whereas the second one responds to the output of the first one only when the clock is low. Thus the final output changes only when the clock is low when the data inputs are not effective. Thus, the race-around condition gets eliminated.



Page No. 10
Date

Working :- 20BCS057

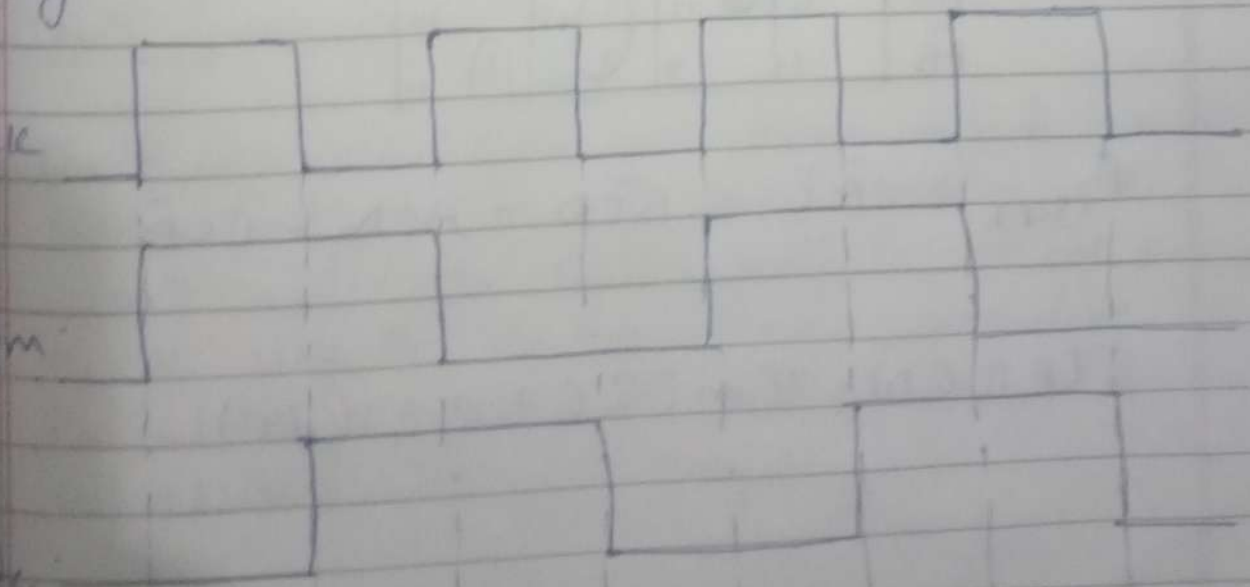
→ Firstly the master flip flop is positive level triggered and the slave flip flop is negative level triggered, as the master responds before the slave.

→ If J=0 & K=1, the high Q' output of the master goes to the K input of the slave and clock forces the slave to reset.

→ If J=1 & K=0, the high Q' output of master goe to the J input of the slave and the -ve transition of the clock sets the value.

→ If J=1 & K=1, it toggles on the positive transaction of the clock and then the slave toggles on the negative transition of the clock.

→ If J=0 & K=0, the flip flop is disabled.

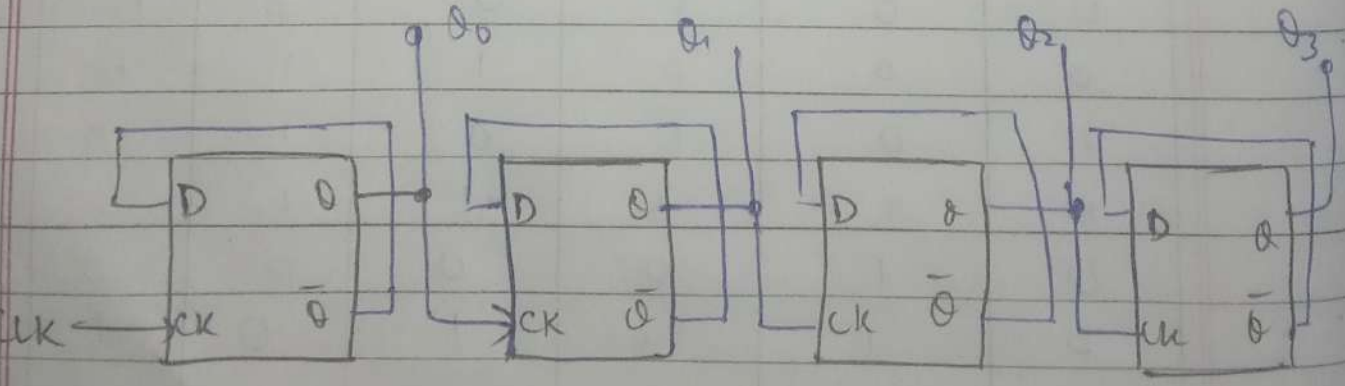


2018B0057

5) b)

Combinational Circuit	Sequential Circuit
→ Output only depends on present input.	Output depends upon the previous output and present input
→ There is no memory unit.	There is a memory unit.
→ There is no clock	There is a clock.
→ It does not have a feedback	It does have a feedback.
→ It can be built using basic gate	It is built using basic gates and combinational circuits.
→ Examples:- Adder, Subtractor, decoder.	Ex:- flip flops, counters, shift registers.

4 bit asynchronous binary down counter.



20BCS057

Previous figure shows a 4 bit Binary Down Counter. 4 bit binary Down Counter will count numbers from 15 to 0, downwards the clock inputs all flip flops are cascaded and the D inputs (Data inputs) of each flip flop is connected to logic 1