EXAM: **II SEM** **PAPER:**
SUBJECT: **COMPUTER ORGANIZATION** **CLASS:** **BCA**
MAXIMUM: **50** **TIME:** **2H**

WEIGHTAGE TO OBJECTIVES TABLE

SL.NO	OBJECTIVES	MARKS	% MARKS
1.	Knowledge (Remembering)	5	10
2.	Understanding	25	50
3.	Application	15	30
4.	Skill	10	20
Total		**50**	**100**

EXAM: **II SEM** **PAPER:**
SUBJECT: **COMPUTER ORGANIZATION** **CLASS:** **BCA**
MAXIMUM: **50** **TIME:** **2H**

BLUE PRINT

Unit	REMEMBERING			UNDERSTAND			APPLICATION			SKILL			Total
	OT	SA	Unit wise Marks	OT	SA	Unit wise Marks	OT	SA	Unit wise Marks	OT	SA	Unit wise Marks	
1	1(1)	1(4)	5	1(1)	1(4)	5							10
2				1(1)		1	1(1)	1(4)	5		1(4)	4	10
3				1(1)	1(4)	5	1(1)	1(4)	5				10
4				1(1)	1(4)	5		1(4)	4	1(1)		1	10
5					1(4)	4	1(1)		1	1(1)	1(4)	5	10
	05			20				15		10			50

BLUE PRINT

EXAM: **II SEM** **PAPER:**
SUBJECT: **COMPUTER ORGANIZATION** **CLASS:** **BCA**
MAXIMUM: **50** **TIME:** **2H**

Unit	REMEMBERING		UNDERSTAND		APPLICATION		SKILL		Total
	OT	SA	OT	SA	OT	SA	OT	SA	
1	1(1)	1(4)	1(1)	1(4)					10
2			1(1)		1(1)	1(4)		1(4)	10
3			1(1)	1(4)	1(1)	1(4)			10
4			1(1)	1(4)		1(4)	1(1)		10
5				1(4)	1(1)		1(1)	1(4)	10
	05		20		15		10		50

UNIT 1

Multiple Choice Questions

(Questions for Remembering)

a. What do you think 1's complement of binary number 11010 is?
   - A. **00101**
   - B. 00010
   - C. 00110
   - D. 11101

b. What do you think the decimal equivalent of $3FA_{16}$ is?
   - **A.** 1024
   - B. **1018**
   - C. 916
   - D. 614

c. What do you think the hexadecimal equivalent of binary number 101101 is?
   - **A.** 37_{16}
   - B. **$2D_{16}$**
   - C. $2E_{16}$
   - D. 27_{16}

d. Do you know the 2's complement of 0011010110011100 number is?
   - A. 1100101011001011
   - B. 1100101001100011
   - C. **1100101001100100**
   - D. 1100101011111111

e. Can you outline how many input signals can a gate have?
   - A. One
   - B. More than one
   - **C.** Two only
   - **D. Both (a) and (b)**

f. Can you outline how many bytes the binary number 1100 0101 has?
   - A. **1**
   - B. 2
   - C. 4
   - D. 8

g. What is the binary equivalent of decimal number 71_{10}?
   - A. 110011_2
   - B. 1110011_2
   - **C.** 0110011_2
   - D. **1000111_2**

h. Can you outline the digits used in a binary number system are?
   - **A.** 9 and 0
   - B. **0 and 1**
   - C. 1 and 2

D. 3 and 4

i. What do you think the number of binary bits required to represent a hexadecimal digit is?
   A. 3
   B. **4**
   C. 6
   D. 8

j. What is the decimal equivalent of 2^{10}?
   A. 4096
   B. **1024**
   C. 1000
   D. 16

k. What do you think the hexadecimal equivalent of an octal number 1438 is?
   A. **63_{16}**
   B. 60_{16}
   C. 50_{16}
   D. 57_{16}

l. Can you outline how many bites are involved in a byte?
   A. Two
   B. Four
   C. **Eight**
   D. Ten

(Questions for Understanding)

m. Can you specify the hexadecimal equivalent of an octal number 112 is?
   a. **$4A_{16}$**
   b. $5A_{16}$
   c. 15_{16}
   d. 20_{16}

n. Can you convert the binary number 101000010111_2 to hexadecimal number?
   A. $D8F9_{16}$
   B. $A8B9_{16}$
   C. **$A17_{16}$**
   D. $D9F8_{16}$

o. Can you specify the hexadecimal equivalent of an octal number 136_8 is?
   A. $7E_{16}$
   B. **$5E_{16}$**
   C. $5A_{16}$
   D. $5D_{16}$

p. Can you convert the octal number 3137_8 to decimal equivalent form?
   A. **1631_{10}**
   B. 1632_{10}
   C. 1531_{10}
   D. 1931_{10}

q. Can you identify the decimal equivalent of an octal number 143_8 is?
   A. 90_{10}
   B. 971_{10}
   C. **99_{10}**
   D. 107_{10}

r. Can you identify the hexadecimal equivalent of binary number 1001001_2 is?
   A. 40_{16}
   B. 39_{16}
   C. **49_{16}**
   D. 42_{16}

s. Can you identify the octal equivalent of decimal number 13_{10}?
   A. **15_8**
   B. 17_8
   C. 13_8
   D. 11_8

t. Can you identify 2's complement of binary number 010111.1100?
   A. 101001.1100
   B. **101000.0100**
   C. 010111.0011
   D. 101000.0011

u. Can you identify the binary equivalent of an octal number 7432_8 is?
   A. 1111000110111_2
   B. **111100011010_2**
   C. 1100110101111_2
   D. 111111111000_2

v. What is the 1's complement of 0000 1111 0010 1101_2 number?
   A. 1111 0000 0010 1101
   B. **1111 0000 1101 0010**
   C. 1111 1100 1010 1100
   D. 1001 0010 1010 1100

w. Can you identify the binary equivalent of hexadecimal number $6B2_{16}$ is?
   A. 1111000110111_2
   B. **011010110010_2**
   C. 01100110011111_2
   D. 11111111_2

x. Can you select the binary equivalent of decimal number 99_{10}?
   A. **1100011_2**
   B. 100011_2
   C. 1110001_2
   D. $111\ 1000_2$

y. Can you identify the hexadecimal equivalent of binary number 1100001_2 is?
   A. 57_{16}
   B. **61_{16}**

C. 51_{16}
D. 43_{16}

Short answer questions
(Questions for Remembering)

1. Describe the steps for performing 1's Complement Subtraction?

We can find the 1's complement of the binary number by simply inverting the given number. The operation is carried out by means of the following steps:

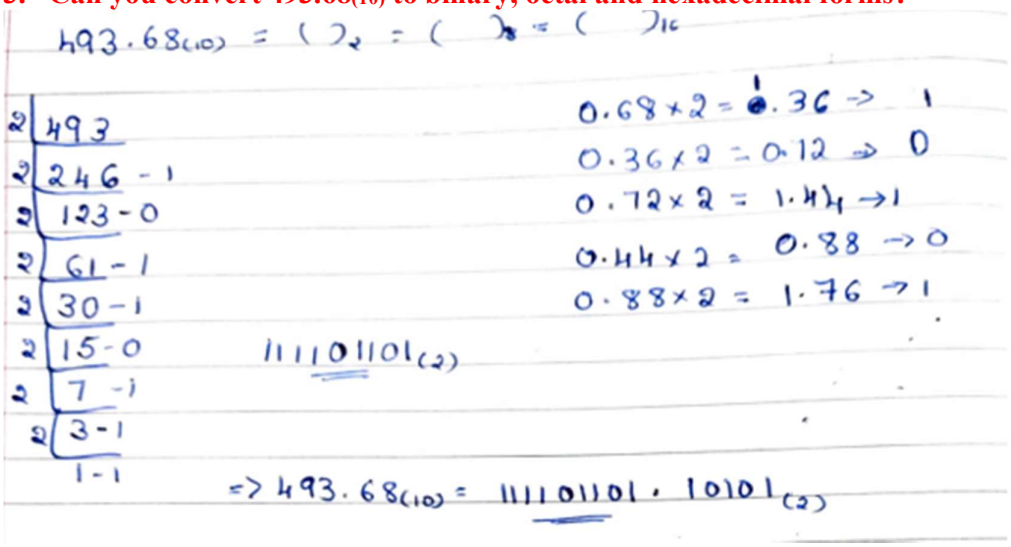
1. To write down 1's complement of the subtrahend.
2. To add this with the minuend.
3. If the result of addition has a carry-over then it is dropped and a 1 is added in the last bit.
4. If there is no carry over, then 1's complement of the result of addition is obtained to get the final result and it is negative.

2. Describe the steps for performing 2's Complement Subtraction?

Just like 1's complement, 2's complement is also used to represent the signed binary numbers. For finding 2's complement of the binary number, we will first find the 1's complement of the binary number and then add 1 to the least significant bit of it. With the help of subtraction by 2's complement method we can easily subtract two binary numbers.
The operation is carried out by means of the following steps:

1. At first, 2's complement of the subtrahend is found.
2. Then it is added to the minuend.
3. If the final carry over of the sum is 1, it is dropped and the result is positive.
4. If there is no carry over, the two's complement of the sum will be the result and it is negative.

3. Can you convert $493.68_{(10)}$ to binary, octal and hexadecimal forms?



8 | 493
8 | 61 - 5
7 - 5

$= 755$

$755.5341_{(8)}$

$0.68 \times 8 = 5.44 \rightarrow 5$
$0.44 \times 8 = 3.52 \rightarrow 3$
$0.52 \times 8 = 4.16 \rightarrow 4$
$0.16 \times 8 = 1.28 \rightarrow 1$

16 | 493
16 | 30 - 13 (D)
1 - 14 (E)

$1ED_{(16)}$

$0.68 \times 16 = 10.88 \rightarrow 10 \rightarrow A$
$0.88 \times 16 = 14.08 \rightarrow 14 \rightarrow E$
$0.08 \times 16 = 1.28 \rightarrow 1$
$0.28 \times 16 = 4.48 \rightarrow 4$

$\Rightarrow 493.68_{(10)} = 1ED.AE14_{(16)}$

4. Can you convert 5137.46 (8) to decimal, hexadecimal and binary forms?

$5137.46_{(8)} = (\quad)_{10} = (\quad)_{16} = (\quad)_2$

$5 \times 8^3 + 1 \times 8^2 + 3 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} + 6 \times 8^{-2}$
$= 2560 + 64 + 24 + 7 + 0.5 + 0.09375$
$= 2655.5937_{(10)}$

$5137.46_{(8)} = (\quad)_{16}$

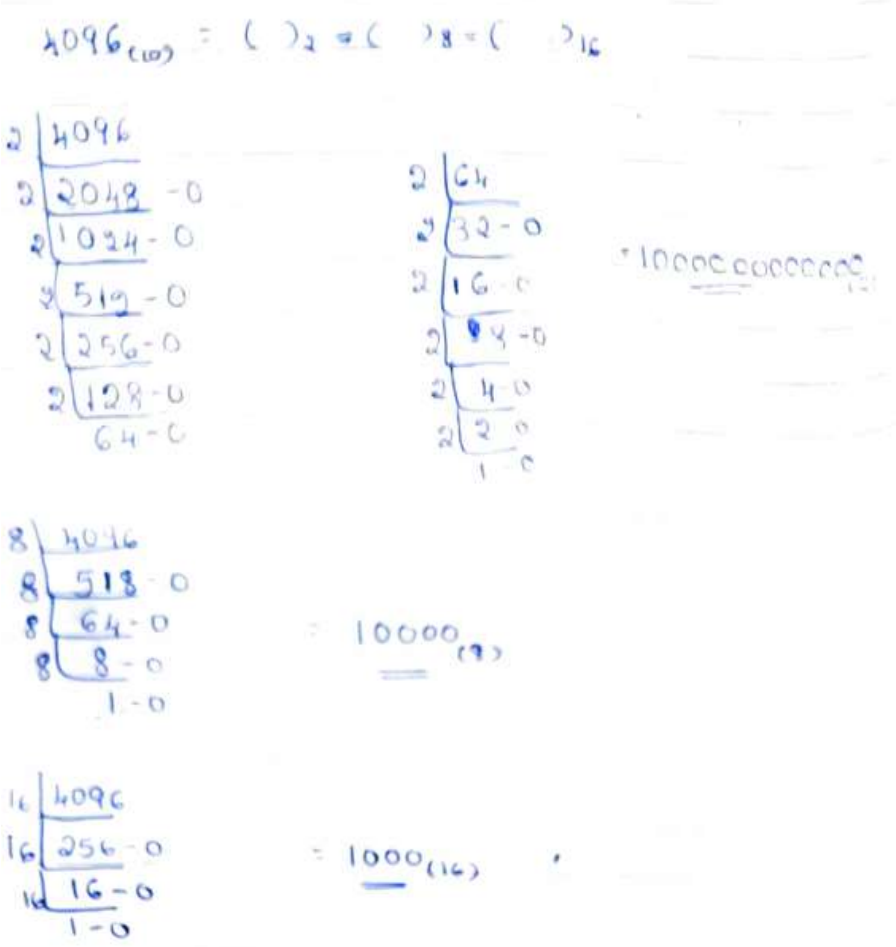
5 1 3 7 . 4 6
101 001 011 111 . 100 110
$101001011111.100110_{(2)}$

101001011111.100110000
A 5 F . 9 8
$A5F.98_{(16)}$

$5137.46_{(8)} = (\quad)_2$

5 1 3 7 . 4 6
$101001011111.100110_{(2)}$

5. Can you convert 4096_{10} to binary, octal and hexadecimal forms?



6. Can you perform the following conversions?

a. **$(1101.11)_2 = (\ )_{10}$**
b. **$(37)_8 = (\ )_{16}$**
c. **$(45)_{10} = (\ )_2$**
d. **$(BCD.A5)_{16} = (\ )_2$**
e. **$(125.48)_{10} = (\ )_8$**

a. $(1101.11)_2 = (\ )_{10}$

$= 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2}$

$= 8 + 4 + 0 + 1 + 0.25 + 0.5$

$= 13.75_{(10)}$

b. $37_{(8)} = (\ )_{16}$

3 7

011 111 Octal to Binary

0001 1111

1 F

$1F_{(16)}$

(c) $45_{(10)} = (\ )_2$

2 | 45
2 | 22 - 1
2 | 11 - 0
2 | 5 - 1
2 | 2 - 1
1 - 0

$101101_{(2)}$

(d) $(BCD.A5)_{16} = (\ )_2$

B C D . A 5
1011 1100 1101 . 1010 0101

$= 101111001101.10100101_{(2)}$

(e) $(125.48)_{10} = (\ )_8$

8 | 125
8 | 15 - 5
1 - 7
$= 175$

$0.48 \times 8 = 3.84 \rightarrow 3$
$0.84 \times 8 = 6.72 \rightarrow 6$
$0.72 \times 8 = 5.76 \rightarrow 5$
$0.76 \times 8 = 6.08 \rightarrow 6$
$= 0.3656$

$= 175.03656_{(8)}$

7. Can you convert each of the following binary numbers to octal and hexadecimal forms?

a. 1011_2
b. 1001.001_2
c. 10110.00101_2

a. 1011_2

Binary to octal

001 011
$13_{(8)}$

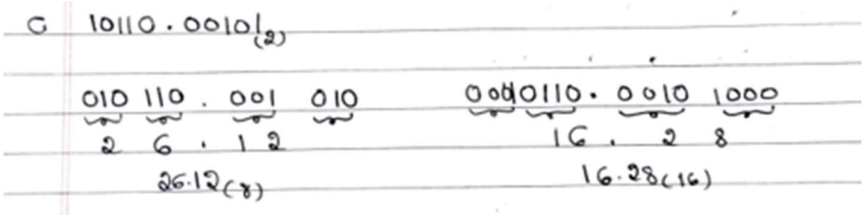
Binary to hexadecimal

1011
11 => B
$B_{(16)}$

b. 1001.001_2

001 001 . 001
1 1 1
$11.1_{(8)}$

1001.0010
9 2
$92_{(16)}$



8. How do you summarize different types of Number System with examples?

Number system is a basis for counting various items.

- **Decimal Number System:** In decimal number system we can express any decimal number in units, tens, hundreds, thousands and so on. When we write a decimal number say, 5678.9, we know it can be represented as

  $5000 + 600 + 70 + 8 + 0.9 = 5678.9 = 5 \times 10^3 + 6 \times 10^2 + 7 \times 10^1 + 8 \times 10^0 + 9 \times 10^{-1} = 5678.9$
- **Binary Number System:** Binary system with its two digits is a base-two system. The two binary digits (bits) are 1 and 0. Like digital system, in binary system each binary digit commonly known as bit, has its own value or weight. However in binary system weight is expressed as a power of 2.

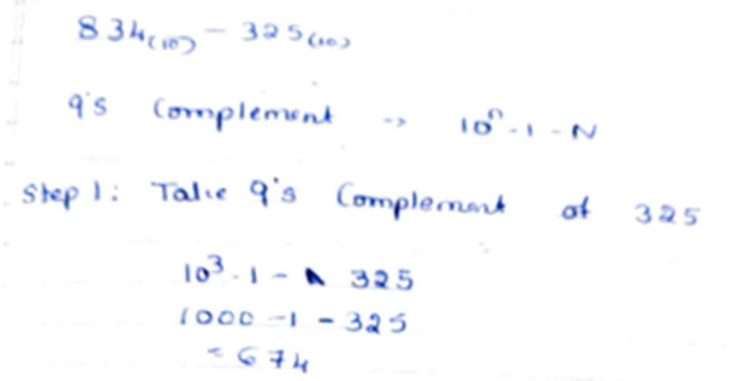
  $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
- **Octal Numbers System:** The octal number system uses first eight digits of decimal number system: 0, 1, 2, 3, 4, 5, 6, and 7. As it uses 8 digits, its base is 8. When we write an octal number say, 567, it can be represented in power of 8 as

  $5 \times 8^2 + 6 \times 8^1 + 7 \times 8^0$
- **Hexadecimal Number System:** The hexadecimal number system has a base of 16 having 16 digits: 0, 1,2,3,4,5,6,7,8,9, A, B, C, D, E and F. It is another number system that is particularly useful for human communications with a computer.

  $3 \times 16^2 + F \times 16^1 + D \times 16^0$

(Questions for Understanding)

9. How would you perform the following subtraction using 9's and 10's complement: 834_{10}-325_{10}?



Step 2: Add 9's Complement of 325 to 834

$$\begin{array}{r} 834 \\ +\ 674 \\ \hline \boxed{1}\ 508 \end{array}$$

$$\begin{array}{r} 508 \\ +1 \\ \hline 509 \end{array}$$

10's Complement → $10^n - N$

Step 1: take 10's Complement of 325

$= 10^3 - 325$

$= 675$

Step 2: Add 10's complement of 325 to 834

$$\begin{array}{r} 834 \\ +\ 675 \\ \hline \boxed{1}\ 509 \end{array}$$

Step 3: Discard the Carry

509

10. How would you perform the following subtraction using 1's and 2's complement $95_{10} - 120_{10}$?

95 - 120

2	95	
2	47	-1
2	23	-1
2	11	-1
2	5	-1
2	2	-1
	1	-0

$= 1011111_{(2)}$

2	120	
2	60	-0
2	30	-0
2	15	-0
2	7	-1
2	3	-1
	1	-1

$= 1111000_{(2)}$

Step 1: 1's complement of 120

1111000

0000111

Step 2: add 1's Complement of 120 to 95

$$\begin{array}{r} 1011111 \\ +\ 0000111 \\ \hline 1100110 \end{array}$$

Step 3: There is No Carry. So take 2's Complement and place -ve sign

0 0 1 1 0 0 0

+1

0 0 1 1 0 0 1

=> −11001

11. How would you perform the following subtraction using 1's and 2's complement: $10011_2 - 01001_2$?

10011 - 01001

1's Complement

Step 1: 1's Complement of 01001

→ 10110

Step 2: add 1's Complement of 01001 to 10011

10011

+ 10110

1 01001

+1

01010

2's Complement

Step 1: 2's Complement of 01001

10110

+1

10111

Step 2: add 2's Complement of 01001 to 10011

10011

10111

1 01010

Step 3: Neglect the Carry

01010

12. How would you perform the following subtraction using 1's and 2's complement: $(1011.11)_2 - (1100.11)_2$?

$1011.11_{(2)} - 1100.11_{(2)}$

1's Complement

Step 1: 1's complement of 1100.11

0011.00

Step 2: add 0011.00 to 1011.11

```
 1011.11
 0011.00
---------
 1110.11
```

Step 3: No carry so take 1's Complement of 1110.11 & put -ve sign

1110.11 => 0001.00

=> -0001.00

2's Complement.

Step 1: 2's Complement of 1100.11

```
 0011.00
    +1
---------
 0011.01
```

Step 2: add 0011.01 to 1011.11

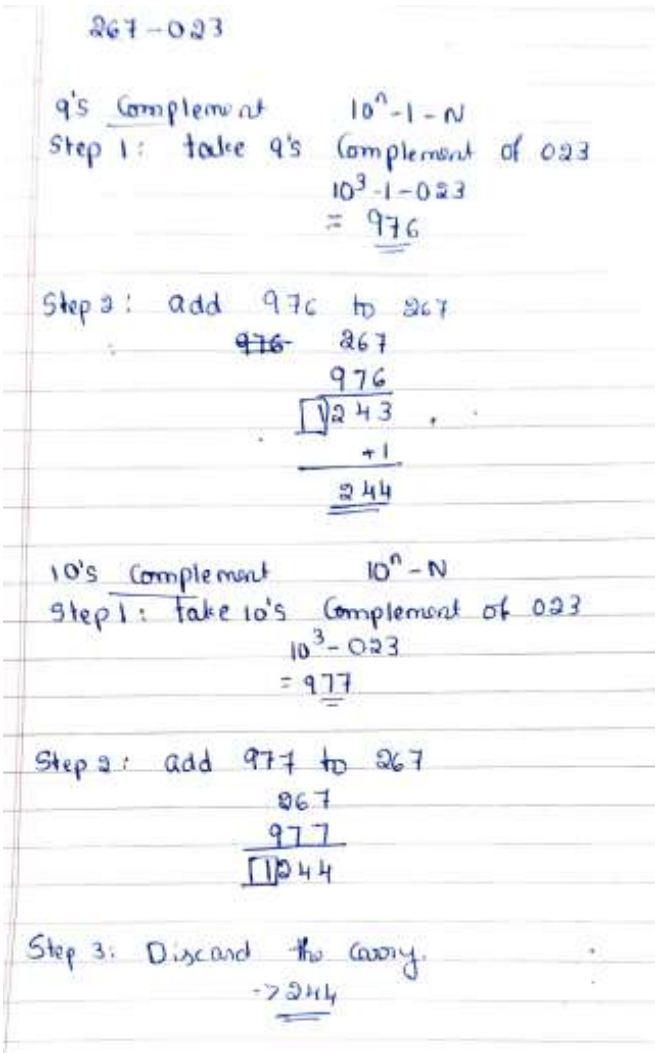
```
 1011.11
+0011.01
---------
 1111.00
```

Step 3: No carry. So take 2's Complement & Place -Ve sign

```
 1111.00
 0000.11
    +1
---------
-0001.00
```

:> -0001.00

13. How would you perform the following subtraction using 9's and 10's complement: 267-023?



267 - 023

9's Complement $10^n - 1 - N$

Step 1: take 9's Complement of 023

$10^3 - 1 - 023$

$= 976$

Step 2: add 976 to 267

~~976~~ 267

976

1243

+1

244

10's Complement $10^n - N$

Step 1: take 10's Complement of 023

$10^3 - 023$

$= 977$

Step 2: add 977 to 267

267

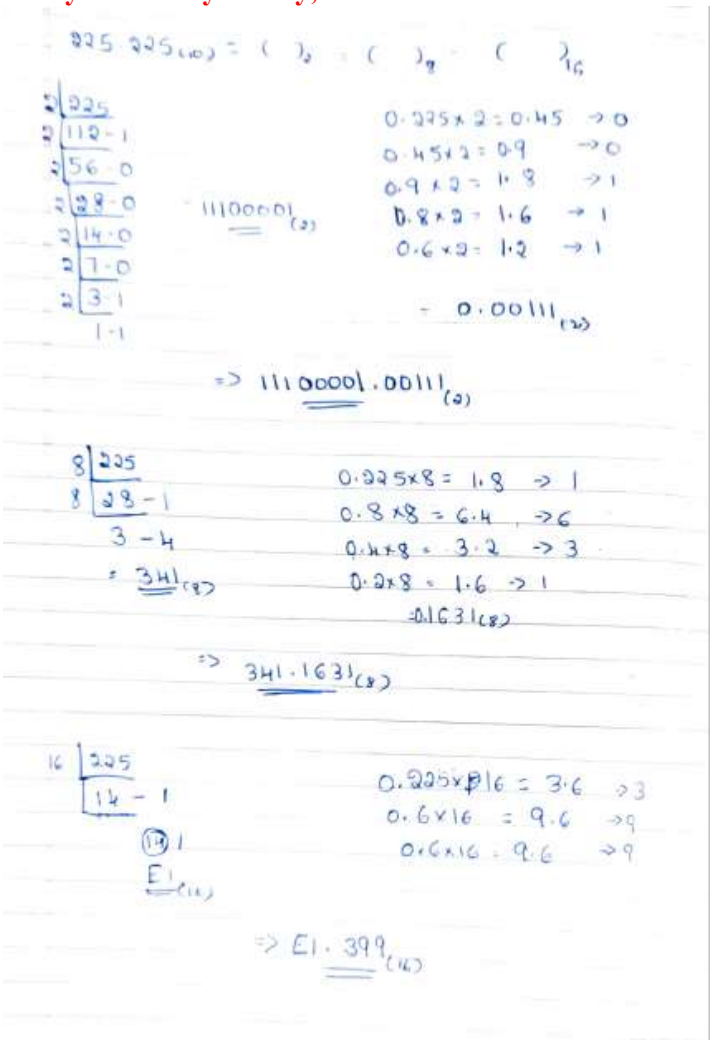
977

1244

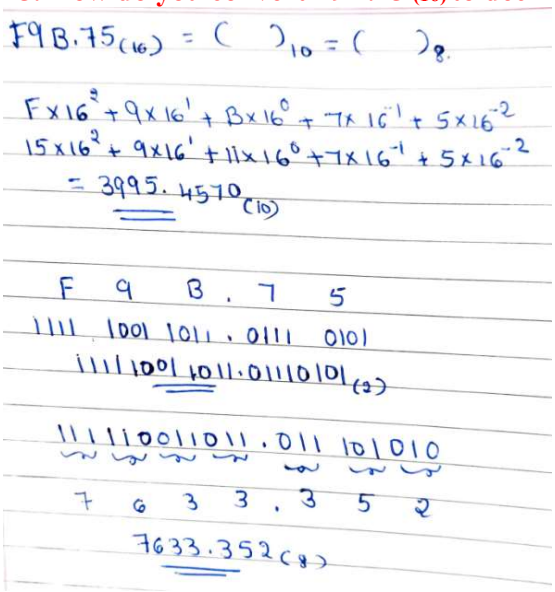
Step 3: Discard the carry.

→ 244

14. How do you identify binary, octal and hexadecimal forms of 225.225_{10}?



15. How do you convert F9B.75 (16) to decimal and octal forms?



UNIT 2

Questions Bank

Multiple Choice Questions

(Questions for Understanding)

a. Do you know the values a binary variable can take?
   A. 0 only
   B. 0 and -1
   C. 0 and 1
   D. 1 and 2

b. What do you think in Boolean algebra 0 is a?
   A. Commutative property
   B. Additive identity
   C. Associative identity
   D. Identity element

c. Which symbol is used to represent AND operation?
   A. (+)
   B. (.)
   C. (-)
   D. (/)

d. What do you think boolean algebra is a collection of objects having
   A. Positive properties
   B. Negative properties
   C. Common properties
   D. Different properties

e. What do you think Truth table is way of expressing
   A. Boolean function
   B. Boolean operators
   C. Boolean addition
   D. Boolean subtraction

f. Do you know NAND gate is a complement of?
   A. AND
   B. OR
   C. NOT
   D. XOR

g. *Which statement below best describes a Karnaugh map?*
   A. *Variable complements can be eliminated by using Karnaugh maps.*
   B. *The Karnaugh map eliminates the need for using NAND and NOR gates.*
   C. *A Karnaugh map can be used to replace Boolean rules.*
   D. **Karnaugh maps provide a cookbook approach to simplifying Boolean expressions.**

h. What do you think (X')' is a?
   A. Complement
   B. Dual complement

C. Duality
D. Reflection

i. What is the alternative term for Minterms?
A. Standard sum
B. Standard product
C. Standard division
D. Standard subtraction

j. What is the alternative term for Maxterms?
A. Standard sum
B. Standard product
C. Standard division
D. Standard subtraction

k. What do you think the first operator precedence for evaluating Boolean expressions is?
A. parenthesis
B. AND
C. OR
D. NOT

l. Do you know the systematic reduction of logic circuits is accomplished by?
A. Using Boolean algebra
B. Symbolic reduction
C. TTL logic
D. Using a truth table

m. Can you tell an 8-square is called as?
A. A pair
B. A quad
C. An octet
D. A cube

(Questions for application)

n. Which is the best answer to perform product of Maxterms Boolean function must be brought into?
A. AND terms
B. OR terms
C. NOT terms
D. NAND terms

o. Which of following is correct according to Boolean algebra absorption law?
A. x+xy=x
B. (x+y)=xy
C. xy+y=x
D. x+y=y

p. What can you say about Boolean function may be transformed into?
A. Logical diagram
B. Logical graph

C. Map
D. Matrix

q. Can you outline x*y = y*x is?
A. Commutative law
B. Inverse property
C. Associative law
D. Identity element

r. Which expression supports sum-of-products (SOP) form?
A. (A + B)(C + D)
B. (A)B(CD)
C. AB(CD)
D. AB + CD

s. Do you know Boolean algebra is defined as a set of _____?
A. Three values
B. Two values
C. Four values
D. Five values

t. What do you think a boolean function can be converted from algebraic expressions to a product of Maxterms by using
A. Graphical representation
B. Truth table
C. Canonical conversion method
D. Both b and c

u. What do you think according to Boolean algebra, postulate 2 with respect to addition is?
A. x+0=x
B. x+0=1
C. x+0=0
D. x+1=0

v. What do you think x+xy =?
A. y
B. 1
C. 0
D. x

w. Do you know X+0=0+X =X is an example of
A. Commutative property
B. Inverse property
C. Associative property
D. Identity element

x. What do you think (x*y)*z=x*(y*z) is?
A. Commutative property
B. Inverse property
C. Identity element

D. Associative property

y. What do you think in this equation A*B=C, where * is the _____

A. Binary operator
B. Logical operator
C. Geometric operators
D. Linear operators

Short answer questions
(Questions for application)

1. Can you state and prove any five postulates of Boolean algebra? (any 4)

Identity Element

x • 1 = x x + 0 = x
if x=1 then,
1.1=1 1+0=1
If x=0 then
0.0=0 0+0=0

Complement

x • x' = 0 x + x' = 1
if x=1 then x'=0
1.0=0 1+0=1
If x=0 then x'=1
0.1=0 0+1=1

Idempotence law

x • x = x x + x = x
when x=0 then,
0.0=0 0+0=0
when x=1 then,
1.1=1 1+1=1

Involution

(x')' = x (x) = x
When x=1 then,
(1')' =1 (1)=1
When x=0 then
(0')'=0 (0)=0

Absorption

x • (x + y) = x x + (x • y) = x
when x=0, y=0 then
0.(0+0)=0 0+(0.0)=0
when x=0, y=1 then
0.(0+1)=0 0+(0.1)=0
when x=1, y=0 then
1.(1+0)=1 1+(1.0)=1
when x=1, y=1 then
1.(1+1)=1 1+(1.1)=1

2. How do you get complement or dual form of a function? Can you find the complement of

a. F1 = X'YZ' + X'Y'Z

b. F2 = X(Y'Z' + YZ)

The duality or complement theorem says that, starting with a Boolean relation, you can derive dual form or complement of another boolean relation by

1. Changing each OR sign to an AND sign
2. Changing each AND sign to an OR sign
3. Complementing any 0 or 1 appearing in the expression

a. F1 = X'YZ' + X'Y'Z
F1= (X+Y'+Z)(X+Y+Z')

b. F2 = X(Y'Z' + YZ)
F2 =XY'Z'+XYZ
= (X'+Y+Z)(X'+Y'+Z')

3. Can you prove the following theorems of Boolean algebra?

a. X + X = X

b. X + XY = X

c. X + X'Y = X + Y

a. **x+x=x**

Solution:
LHS: x+x
= x(1+1)
= x(1)
= x = RHS

b. **x+xy=x**

Solution:
LHS: x+xy
= x(1+y)
= x(1)
= x = RHS

c. **x+x'y=x+y**

Solution:
LHS: x+x'y
= (x+x').(x+y) [By distributive law]
= 1.(x+y)
= x+y = RHS

4. Can you state and prove De Morgan's Theorem for 3 variables?

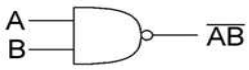
The complement of a product is equal to the sum of the complements.
i.e,
$(X + Y + Z)' = X' \times Y' \times Z'$
$(X + Y + Z)' = (X + (Y + Z))'$ - by associative law
$= X' \times (Y + Z)'$ - by DeMorgan's law
$= X' \times (Y' \times Z')$ by DeMorgan's law
$= X' \times Y' \times Z'$ - by associative law

The complement of a sum is equal to the product of the complements.
i.e,
$(X \times Y \times Z)' = X' + Y' + Z'$
$(X \times Y \times Z)' = (X \times (Y \times Z))'$ - by associative law
$= X' + (Y \times Z)'$ by DeMorgan's law
$= X' + (Y' + Z')$ - by DeMorgan's law
$= X' + Y' + Z'$ - by associative law

5. Illustrate NAND Gate and prove that NAND gate is Universal Gate?

The Boolean expression for NAND function is Y = (AB)'. This expression tells that the two input variables, A& B, are first ANDed and then complemented, as indicated by the bar over the AND expression.

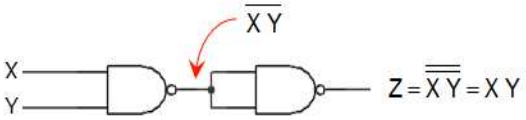
Logic Diagram



Truth Table

	B	(A.B)'
0	0	1
0	1	1
1	0	1
1	1	0

(a) Implementation of AND function using only NAND gates

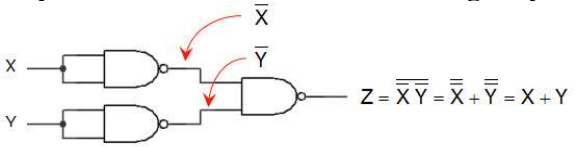


Truth Table

X	Y	X.Y	(X.Y)'	((X.Y)')'
0	0	0	1	0
0	1	0	1	0
1	0	0	1	0
1	1	1	0	1

Equivalent to AND Gate

Implementation of OR function using only NAND gates

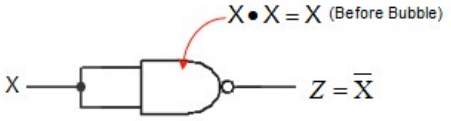


Truth Table

X	Y	X'	Y'	(X')'+(Y')'
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1

Equivalent to OR

Implementation of NOT function using only NAND gates



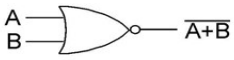
Truth Table

X	X’
0	1
1	0

6. What is NOR Gate and prove that NOR gate is Universal Gate?

The Boolean expression for NOR function is (Y = A +B)’. This equation says that the two input variables are first ORed and then complemented.

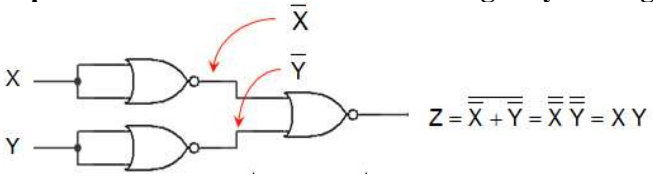
Logic Diagram



Truth Table

A	B	(A+B)'
0	0	1
0	1	0
1	0	0
1	1	0

Implementation of AND function using only NOR gates

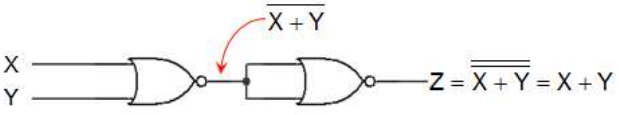


Truth Table

X	Y	X'	Y'	(X'+Y')'
0	0	1	1	0
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

Equivalent to AND

Implementation of OR function using only NOR gates



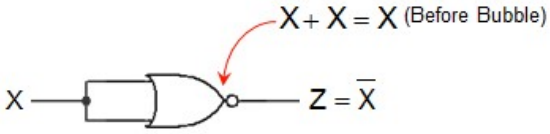
Truth Table

X	Y	X+Y	(X+Y)'	((X+Y)')'
0	0	0	1	0
0	1	1	0	1

Equivalent to OR Gate

1	0	1	0	1
1	1	1	0	1

Implementation of NOT function using only NOR gates



Truth Table

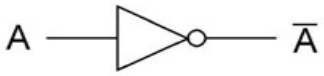
X	X'
0	1
1	0

7. What is a gate? Can you illustrate different types of digital gates with truth table and symbol?

A logic gate is an idealized or physical device implementing a boolean function. That is, it performs a logical operation on one or more logical inputs, and produces a single logical output.

NOT gate: If the input variable is A and the output variable is Y; then Y= A. The output is complement of the input.

Logic Diagram

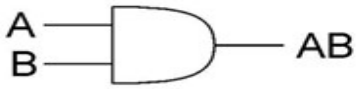


Truth Table

X	X'
0	1
1	0

AND gate: Let the two input variables be A and B and the output variable Y; then the boolean expression is Y = AB or A.B. If there are four input variables A, B, C, D, then the output Y = ABCD.

Logic Diagram

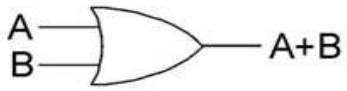


Truth Table

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

OR gate: If one input is A, the other input is B, and the output is Y, then the Boolean expression for OR function is Y = A+B.

1. **Logic Diagram**



Truth Table

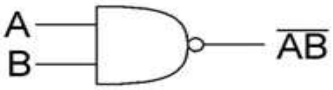
A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

8. What is universal gate? Analyze different universal gates?

A universal gate is the gate using which you can implement all the other gates in the logical system. There are 2 universal gates

1. **NAND gate**: The Boolean expression for NAND function is Y = (AB)'. This expression tells that the two input variables, A& B, are first ANDed and then complemented, as indicated by the bar over the AND expression.

Logic Diagram

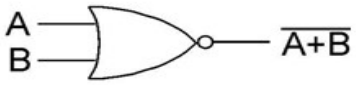


Truth Table

	B	(A.B)'
0	0	1
0	1	1
1	0	1
1	1	0

2. **NOR gate**: The Boolean expression for NOR function is (Y = A +B)'. This equation says that the two input variables are first ORed and then complemented.

Logic Diagram



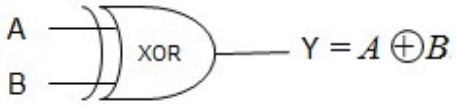
Truth Table

A	B	(A+B)'
0	0	1
0	1	0
1	0	0
1	1	0

9. Can you explain XOR gate with logic diagram and truth table?

The Exclusive-OR gate or XOR gate is achieved by combining standard logic gates together. XOR gate is used extensively in error detection circuits, computational logic comparators and arithmetic logic circuits. If one input is A, the other input is B, and the output is Y. Then the Boolean Expression for X-OR function is A ⊕ B= AB' + A'B

Logic Diagram

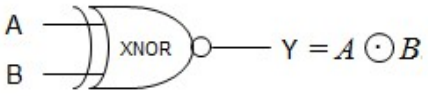


Truth Table

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

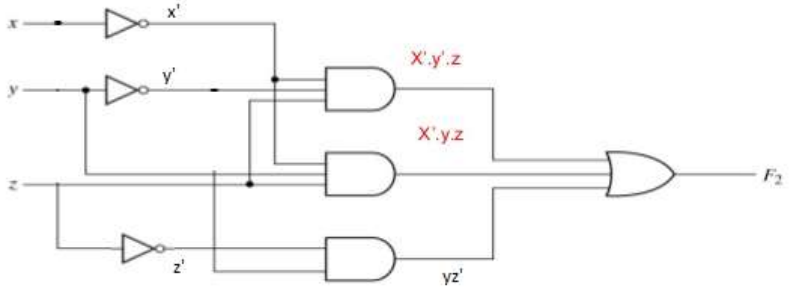
10. Can you explain XNOR gate with logic diagram and truth table?

If one input is A, the other input is B, and the output is Y. Then the Boolean expression for X-OR function is A Θ B = (A ⊕ B)' = (AB' + A'B)'. This equation says that the two input variables are first EX-ORed and then complemented.



A	B	A⊙B
0	0	1
0	1	0
1	0	0
1	1	1

11. Can you implement Boolean function F=X'Y'Z + X'YZ + YZ' with basic gates?



(Questions for skill)

12. **Can you express the Boolean function F = A+BC as sum of Minterm and product of Maxterm.**

F=A+BC

Sum of minterms

F=A(B+B')(C+C')+(A+A')BC

= (AB+AB')(C+C')+ABC+A'BC

= ABC+AB'C+ABC'+AB'C'+ABC+A'BC

= ABC+AB'C+ABC'+AB'C'+ A'BC

Sum of maxterms

F = A+BC

F'= A'+B'C'

= (A'+BB'+CC').(B'+C'+AA')

= (A'+B+C)(A'+B+C')(A'+B'+C)(A'+B'+C')(B'+C'+A)(B'+C'+A')

= (A'+B+C)(A'+B+C')(A'+B'+C)(A'+B'+C')(B'+C'+A)

13. **Can you simplify the Boolean functions using Karnaugh Map method F(X, Y, Z) = Σ (2, 3, 4, 6).**

X\YZ	Y'Z'	Y'Z	YZ	YZ'
X'	0	0	1	1
X	1	0	0	1

X\YZ	Y'Z'	Y'Z	YZ	YZ'
X'	0	0	1	1
X	1	0	0	1

Step 1: Fig. shows the K-map for 3 variables

Step 2: There are no isolated 0s.

Step 3: Pair: X'Y, XZ'

Step 4: No Quad

Step 5: Therefore the final reduced expression is:

F= X'Y+XZ'

14. **Can you simplify the Boolean functions using Karnaugh Map method F(X, Y, Z) = Π (0, 1, 4, 5).**

X\YZ	Y'Z'	Y'Z	YZ	YZ'
X'	0	0	1	1
X	0	0	1	1

X\YZ	Y'Z'	Y'Z	YZ	YZ'
X'	0	0	1	1
X	0	0	1	1

Step 1: Fig. shows the K-map for 3 variables

Step 2: There are no isolated 0s.

Step 3: No Pair

Step 4: Quad: Y'

Step 5: Therefore the final reduced expression is:

F= Y'

F'= Y

15. **Using K-Map can you reduce the following expressions**
F = A'B'C+A'BC+AB'C'+ABC.

A\BC	B'C'	B'C	BC	BC'
A'	0	1	1	0
A	1	0	1	0

A\BC	B'C'	B'C	BC	BC'
A'	0	1	1	0
A	1	0	1	0

Step 1: Fig shows the K-map for 3 variables.

Step 2: Isolated 1's – AB'C'

Step 3: Pairs: A'C, BC

Therefore the final reduced expression is: F= AB'C'+ A'C+BC

16. Using K-Map can you reduce the following expressions F= (A + B'+C) (A' + B'+C') (A + B+C') (A '+ B'+C) (A + B+C).

F=(A + B'+C) (A' + B'+C') (A + B+C') (A '+ B'+C) (A + B+C).

F'=A'BC'+ABC+A'B'C+ABC'+A'B'C'

A\BC	B'C'	B'C	BC	BC'
A'	1	1	0	1
A	0	0	1	1

A\BC	B'C'	B'C	BC	BC'
A'	1	1	0	1
A	0	0	1	1

Step 1: Fig shows the K-map for 3 variables.

Step 2: No isolated 1's

Step 3: Pairs: A'B', AB, BC'

Therefore the final reduced expression is: F'= A'B'+AB+BC'

F= (A+B)(A'+B')(B'+C)

17. Using K-Map can you reduce the following expression F (A, B, C, D) = Σ (0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14, 15).

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	1	0	1
A'B	1	1	0	1
AB	1	1	1	1
AB'	1	1	0	0

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	1	0	1
A'B	1	1	0	1
AB	1	1	1	1
AB'	1	1	0	0

Step 1: Fig shows the K-map for 4 variables.

Step 2: No isolated 1's

Step 3: No pairs

Step 4: Quad: A'D', AB

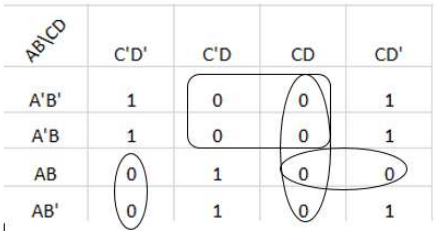
Octant: C'

Therefore the final reduced expression is: A'D'+AB+C'

18. Using K-Map can you reduce the following expression F (A, B, C, D) = Π (1, 3, 5, 7, 8, 11, 12, 14, 15).

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	0	0	1
A'B	1	0	0	1
AB	0	1	0	0
AB'	0	1	0	1

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	0	0	1
A'B	1	0	0	1
AB	0	1	0	0
AB'	0	1	0	1



Step 1: Fig. shows the K-map for 4 variables

Step 2: no isolated 0's

Step 3: Pairs – AC'D', ABC

Step 4: Quads: A'D, CD

No octets

Step 5: Therefore the final reduced expression is:

F= AC'D'+ ABC + A'D+CD

F'= (A'+C+D) + (A'+B'+C')+ (A+D')+(C'+D')

19. Can you simplify the following expressions using K-Map F= Σ (0, 3, 4, 7, 8) + d (10, 11, 12, 13, 14, 15).

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	0	1	0
A'B	1	0	1	0
AB	X	X	X	X
AB'	1	0	X	X

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	0	1	0
A'B	1	0	1	0
AB	1	0	1	0
AB'	1	0	1	0

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	0	1	0
A'B	1	0	1	0
AB	1	0	1	0
AB'	1	0	1	0

Step 1: Fig shows the K-map for 4 variables.

Step 2: No isolated 1's

Step 3: No pairs

Step 4: Quad: C'D', CD

And no octants

Therefore the final reduced expression is: C'D'+CD

20. Can you simplify the following expressions using K-Map F = Π (2, 3, 5, 7, 8, 14) + d (4, 6, 11).

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	1	0	0
A'B	X	0	0	X
AB	1	1	1	0
AB'	0	1	X	1

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	1	0	0
A'B	0	0	0	0
AB	1	1	1	0
AB'	0	1	1	1

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	1	0	0
A'B	0	0	0	0
AB	1	1	1	0
AB'	0	1	1	1

Step 1: Fig. shows the K-map for 4 variables

Step 2: Isolated 0's – AB'C'D'

Step 3: Pairs – BCD'

Step 4: Quads: A'C, A'B

No octets

Step 5: Therefore the final reduced expression is:

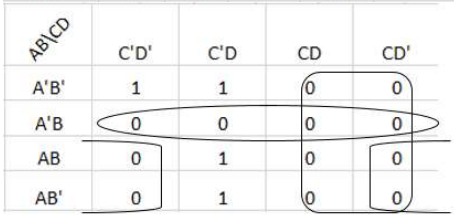
F= AB'C'D'+ BCD'+ A'C+A'B

F'= (A'+B+C+D) + (B'+C'+D)+ (A+C')+(A+B')

21. Can you simplify the following expressions using K-Map F = Π (2, 3, 5, 7, 8, 11, 12, 14) + d (4, 6, 10, 15).

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	1	0	0
A'B	X	0	0	X
AB	0	1	X	0
AB'	0	1	0	X

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	1	0	0
A'B	0	0	0	0
AB	0	1	0	0
AB'	0	1	0	0



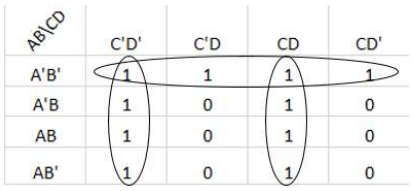
Step 1: Fig. shows the K-map for 4 variables
Step 2: There are no isolated 0s.
Step 3: No pairs
Step 4: Quads: AD', A'B
Octets: C
Step 5: Therefore the final reduced expression is:

F= AD'+A'B+C
F'= (A'+D)(A+B')C'

22. Can you simplify the following expressions using K-Map F= Σ (0, 1, 2, 3, 4, 7, 8) + d (10, 11, 12, 13, 14, 15).

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	1	1	1
A'B	1	0	1	0
AB	X	X	X	X
AB'	1	0	X	X

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	1	1	1
A'B	1	0	1	0
AB	1	0	1	0
AB'	1	0	1	0

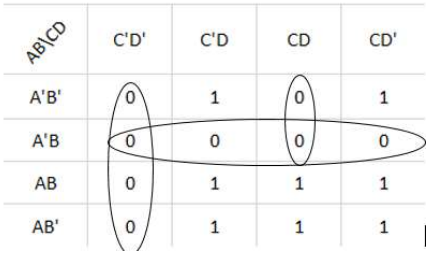


Step 1: Fig shows the K-map for 4 variables.
Step 2: No isolated 1's
Step 3: No pairs
Step 4: Quad: C'D', CD, A'B'
And no octants
Therefore the final reduced expression is: C'D'+CD+A'B'

23. Can you simplify the following expressions using K-Map F= Π (0, 3, 4, 5, 6, 7, 8) + d (10, 11, 12, 13).

AB\CD	C'D'	C'D	CD	CD'
A'B'	0	1	0	1
A'B	0	0	0	0
AB	X	X	1	1
AB'	0	1	X	X

AB\CD	C'D'	C'D	CD	CD'
A'B'	0	1	0	1
A'B	0	0	0	0
AB	0	1	1	1
AB'	0	1	1	1



Step 1: Fig. shows the K-map for 4 variables
Step 2: There are no isolated 0s.
Step 3: Pair: A'CD
Step 4: Quad: C'D', A'B
Step 5: Therefore the final reduced expression is:

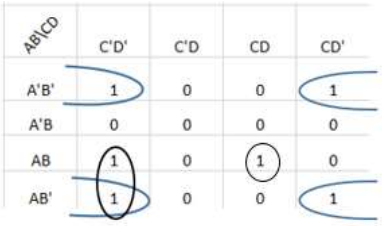
F= A'CD+ C'D'+A'B

F'= (A+C'+D')(C+D)(A+B')

24. Can you simplify the following expressions using K-Map F = Σ (0, 2, 8, 10, 12, 15) + d (3, 5, 9).

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	0	X	1
A'B	0	X	0	0
AB	1	0	1	0
AB'	1	X	0	1

AB\CD	C'D'	C'D	CD	CD'
A'B'	1	0	0	1
A'B	0	0	0	0
AB	1	0	1	0
AB'	1	0	0	1



Step 1: Fig shows the K-map for 4 variables.

Step 2: Isolated 1's -ABCD

Step 3: The pairs – A'B'D', AC'D', AB'D'
Step 4: No quad and octants

Therefore the final reduced expression is: A'B'D'+AC'D', AB'D'+ ABCD

UNIT 3
QUESTION BANK

Multiple Choice Questions

(Questions for Understanding)

a. Which are the fundamental inputs assigned or configured in the full adder circuit?
   A. Addend, Augend & Sum
   B. Augend, Sum & Input Carry
   C. Addend, Augend & Input Carry
   D. **Addend, Sum & Input Carry**

b. What do you think the Boolean functions which can be represented by the sum of Minterms and product of Maxterms can be categorized in _____?
   A. Standard form
   B. **Canonical form**
   C. Both a & b
   D. None of the above

c. Can you tell what half adder does?
   A. Decimal addition operation for 2 decimal inputs
   B. **Binary addition operation for 2 binary inputs**
   C. Decimal addition operation for 2 binary inputs
   D. Binary addition operation for 2 decimal inputs

d. What do you think half Subtractors have an output to specify that 1 has been?
   A. Complemented
   B. **Borrowed**
   C. Carried
   D. Primed

e. Can you tell two bit subtraction is done by
   A. Demultiplexer
   B. Multiplexer
   C. Full Subtractors
   D. **Half Subtractors**

f. What do you think a binary parallel adder produces arithmetic sum in _____?
   A. Serial
   B. **Parallel**
   C. Sequence
   D. Both a and b

g. Can you clarify full adder forms sum of _____?
   A. 2 bits
   B. **3 bits**
   C. 4 bits
   D. 5 bits

h. What do you think a BCD adder is a circuit that adds two BCD digits in parallel and produces a result in _____?
   A. Hexadecimal code

B. Binary code
C. **BCD code**
D. Decimal code

i. Can you clarify addition of two decimal digits in BCD can be done through _____?
A. **BCD adder**
B. Full adder
C. Ripple carry adder
D. Carry look ahead

j. Can you clarify output sum of two decimal digits can be represented in _____?
A. Gray code
B. Excess-3
C. **BCD()**
D. Hexadecimal

k. What do you think the output of a full subtractor is same as _____?
A. Half adder
B. **Full adder**
C. Half subtractor
D. Decoder

l. Do you know in which operation carry is obtained?
A. Subtraction
B. **Addition**
C. Multiplication
D. Both addition and subtraction

(Questions for application)

m. Which of the following statements accurately represents the two BEST methods of logic circuit simplification?
A. **Boolean algebra and Karnaugh mapping**
B. Karnaugh mapping and circuit waveform analysis
C. Actual circuit trial and error evaluation and waveform analysis
D. Boolean algebra and actual circuit trial and error evaluation

n. What is the simplified expression of half subtractor borrow?
A. B=X+Y
B. B=XY
C. **B=X'Y**
D. B=XY'

o. What is the simplified expression of full subtractor borrow?
A. B= XY' Z'+X' Y' Z
B. B= XY' Z'+X' Y' Z+XYZ+X'YZ'
C. **B= X' Z+X' Y+YZ**
D. B=XY'

p. What is the simplified expression of full subtractor difference?
A. D= XY' Z'+X' Y' Z

B. D= X' Z+X' Y+YZ
C. **D= XY' Z'+X' Y' Z+XYZ+X'YZ'**
D. D=XY'

q. What is the simplified expression of full subtractor difference?
A. D= XY' Z'+X' Y' Z
B. D= X' Z+X' Y+YZ
C. **D= XY' Z'+X' Y' Z+XYZ+X'YZ'**
D. D=XY'

r. What do you think BCD adder can be constructed with 3 Integrated circuits (IC) packages each of _____?
A. 2 bits
B. 3 bits
C. **4 bits**
D. 5 bits

s. In the half subtractor combinational circuit, what does 'A' represent in the subtraction operation (A - B)?
A. Minuend bit
B. Maxend bit
C. **Subtrahend bit**
D. Subtrahend bit

t. Can you clarify for subtracting 1 from 0, we use to take a _____ from neighboring bits.
A. Carry
B. **Borrow**
C. Input
D. Output

u. What do you think if A and B is the input of a subtractor then the output will be _____?
A. **A XOR B**
B. A AND B
C. A OR B
D. A EXNOR B

v. What does minuend and subtrahend denotes in a subtractor?
A. Their corresponding bits of input
B. Its outputs
C. **Its inputs**
D. Borrow bits

w. What do you think if A and B is the input of a subtractor then the borrow will be _____?
A. A AND B'
B. **A' AND B**
C. A OR B
D. A AND B

x. What do you think Half-adders have a major limitation in that they cannot _____?

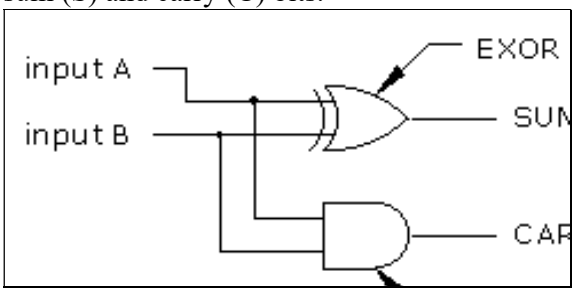
A. Accept a carry bit from a present stage
B. Accept a carry bit from a next stage
C. **Accept a carry bit from a previous stage**
D. Accept a carry bit from the following stages

y. How many AND, OR and EXOR gates are required for the configuration of full adder?
A. 1, 2, 2
B. **2, 1, 2**
C. 3, 1, 2
D. 4, 0, 1

Short answer questions
(Questions for Understanding)

1. With the help of a suitable diagram can you explain the working of a Half-Adder circuit along with the truth table?

An adder is a device that will add together two bits and give the result as the output. ... There are two kinds of adders - half adders and full adders. A half adder just adds two bits together and gives a two-bit output. A full adder adds two inputs and a carried input from another adder, and also gives a two-bit output.
Half adder is a circuit which has only 2 binary inputs (A and B) and two outputs, which are the sum (S) and carry (C) bits.



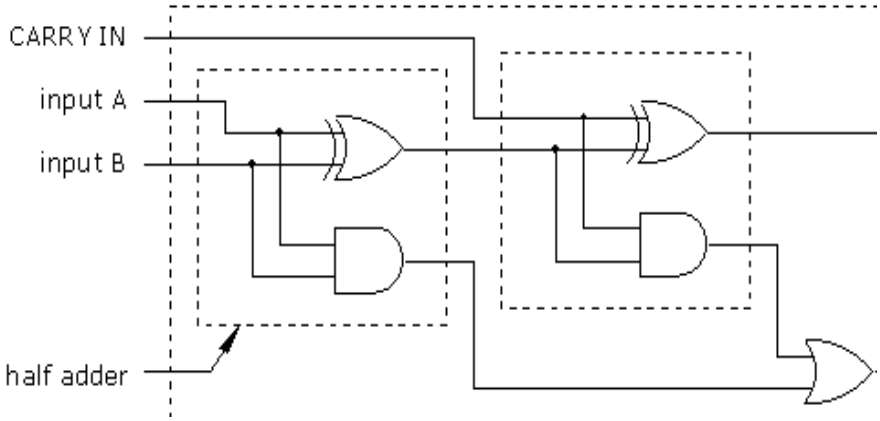
$S = A'B + AB' = A \oplus B$

$C = AB$

Input A	Input B	SUM (S)	CARRY (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

2. Can you illustrate the Full-Adder circuit using truth table and represent the Sum(S) and Carry(C) bits using logic gates?

Full adder is a circuit which has 3 binary inputs - binary input A, binary input B and a Carry-in (C_{in}) from the previous column and produces two outputs, which are the sum (S) and carry-out (C_{out}) bits. You can make a full adder by linking together two half adder circuits:



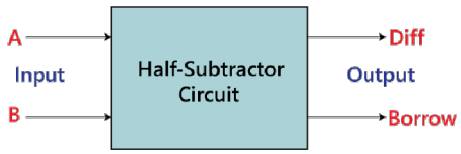
$S = A'BC'_{in} + AB'C'_{in} + A'B'C_{in} + ABC_{in} = A \oplus B \oplus C_{in}$

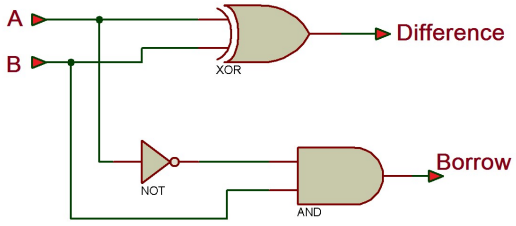
$C = (A \oplus B)C_{in} + AB$

Inputs			Outputs	
A	B	C – IN	Sum	C - Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

3. How would you explain the Half-Subtractor circuit using truth table and a suitable diagram?

The half-subtractor is a combinational circuit that subtracts one bit from the other and produces a difference. It also has an output to specify if a 1 has been borrowed. It is used to subtract the LSB of the subtrahend from the LSB of the minuend when one binary number is subtracted from the other. A half-subtractor is shown below with two inputs A and B and two outputs Diff and Borrow. Diff indicates the difference and Borrow is the output signal which indicates that whether a 1 has been borrowed.





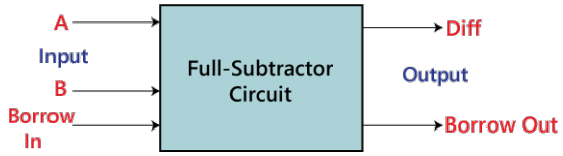
Diff=A'B+AB'
Borrow = A'B

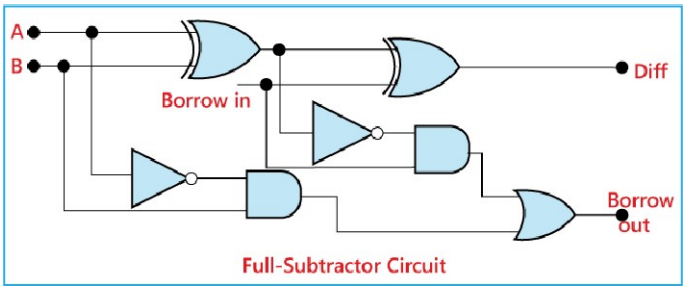
Inputs		Outputs	
A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

4. How would you explain the working of a Full-Subtractor using truth table and a logic diagram?

The half-subtractor can be used only for LSB subtraction. If there is a borrow during the subtraction in the LSBs, it affects the subtraction in the next higher column; the subtrahend bit is subtracted from the minuend bit, considering the borrow from that column used for the subtraction in the preceding column. Such a subtraction is performed by a full-subtractor.

It subtracts one bit (B) from another bit (A), when already there is a Borrow in from this column for the subtraction in the preceding column, and outputs the difference bit and the borrow bit required from the next column. So a full-subtractor is a combinational circuit with three inputs (A, B, Borrow) and two Outputs diff and borrow.





$D = A \oplus B \oplus Bin$

$Bout = A' Bin + A' B + B Bin$

Inputs			Outputs	
A	B	$Borrow_{in}$	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

5. How would you explain the 4-bit binary parallel adder with logic diagram?

A binary parallel adder is a digital circuit that adds two binary numbers in parallel form and Produces the arithmetic sum of those numbers in parallel form. It consists of full adders

connected in a chain, with the output carry from each full-adder connected to the input carry of the next full adder in the chain.
The Figure below shows the interconnection of four full-adder (FA) circuits to providea4-bitparalleladder.The augend bits of A and addend bits of B are designated by subscript numbers from right to left, with subscript 1 denoting the lower-order bit. The carries are connected in a chain through the full-adders. The input carry to the adder is Cin and the output carry is C4. The S outputs generate the required sum bits.

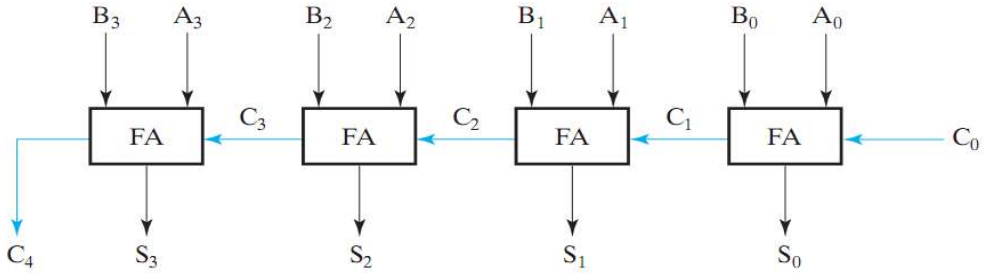


Fig. Logic Diagram for a 4-bitBinary Parallel Adder

6. How would you explain the 4-bit binary parallel subtractor with logic diagram?

The subtraction of 2 binary numbers can be carried out most conveniently by means of complement. The subtraction A - B can be done by taking the 2'scomplement of B and adding it to A. The 2's complement can be obtained by taking the l's complement and then adding 1 to the least significant pair of bits. The l's complement can be implemented with inverters as shown in the figure

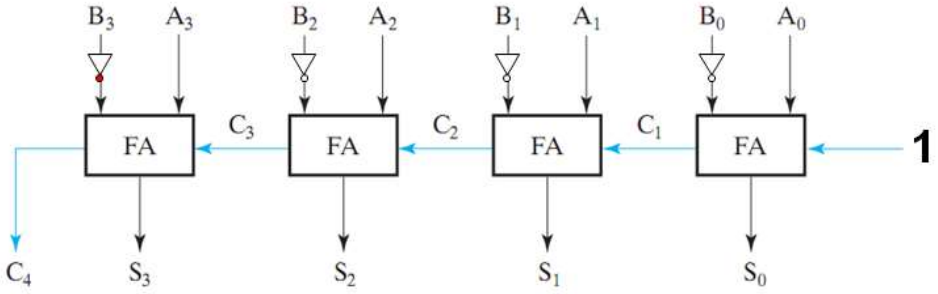


Fig. Logic Diagram for a 4-bitBinary Parallel Subtractor

(Questions for application)

7. With a circuit diagram can you explain the working of a BCD adder?

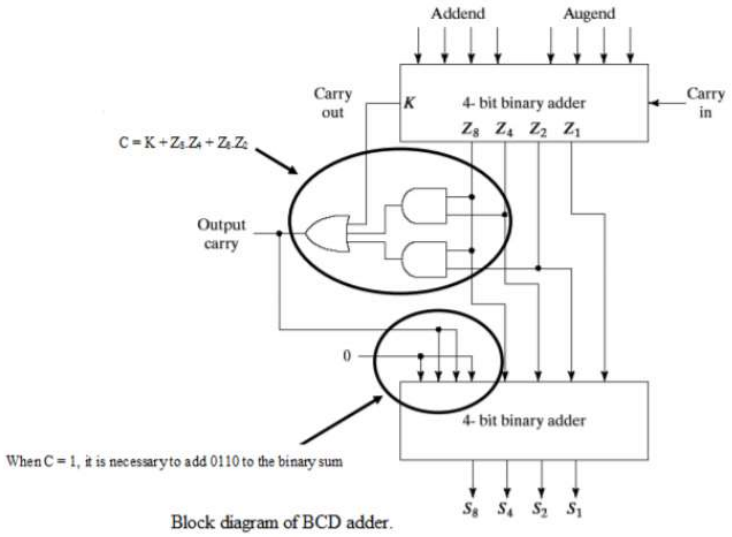
Computers or calculators that perform arithmetic operations directly in the decimal number system represent decimal numbers in binary coded form. An adder for such a computer must employ arithmetic circuits that accept coded decimal numbers and present results in the same code. Suppose we apply two BCD digits to a four-bit binary adder. The adder will form the sum in binary and produce a result that ranges from 0 through 19. These binary numbers are labeled by symbols K, Z_8, Z_4, Z_2, and Z_1. K is the carry, and the subscripts under the letter Z represent the weights 8, 4, 2, and 1 that can be assigned to the four bits in the BCD code.
When the binary sum is equal to or less than 1001_b
BCD Sum = Binary Sum
C = 0 ;
When the binary sum is greater than 1001_b
BCD Sum = Binary Sum + 0110_b

C = 1

The condition for a correction and an output carry can be expressed by the Boolean function

$C = K + Z_8.Z_4 + Z_8.Z_2$

When C = 1, it is necessary to add 0110 to the binary sum and provide an output carry for the next stage.



Block diagram of BCD adder.

8. Can you explain the implementation process of EX-OR, using NAND gates?

This ability of the Exclusive-OR gate to compare two logic levels and produce an output value dependent upon the input condition is very useful in computational logic circuits as it gives us the following Boolean expression of:

$Q = A \oplus B = A'B + AB'$

$= A'B + AB' + AA' + BB'$

$= (A + B)(A' + B')$

Now we need to implement this circuit using NAND gates

$Q = (A + B)(AB)'$

$= A.(AB)' + B.(AB)'$

Taking compliment

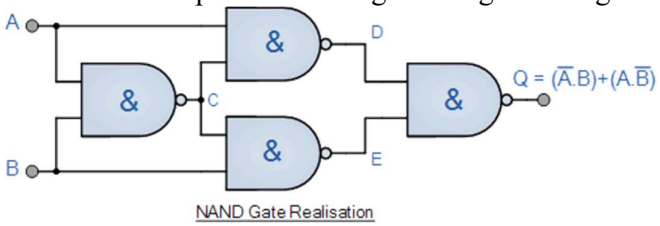
$Q' = (A.(AB)' + B.(AB)')'$

$= (A.(AB)')'.(B.(AB)')'$

Taking compliment again

$Q = ((A.(AB)')'.(B.(AB)')')'$

Now we can implement XOR gate using NAND gates



NAND Gate Realisation

9. Can you explain the implementation process of EX-OR, using NOR gates?

This ability of the Exclusive-OR gate to compare two logic levels and produce an output value dependent upon the input condition is very useful in computational logic circuits as it gives us the following Boolean expression of:

$Q = A \oplus B = A'B + AB'$

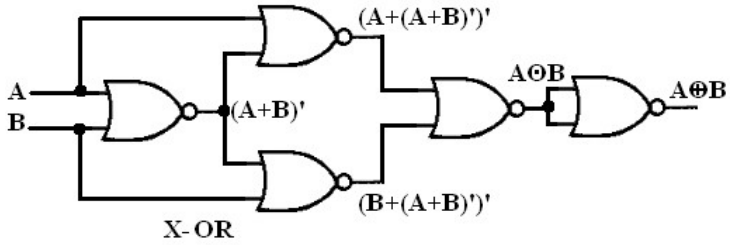
= A’B + AB’ + AA’ + BB’
= (A + B) (A’ + B’)
Q= A’ (A+B) + B’(A+B)

By applying compliment
Q’ = (A’ (A + B) + B’ (A + B))’
= (A’(A + B))’ . (B’ (A + B))’
=(A + (A + B)’). (B + (A + B)’)

By applying compliment again
Q’’=Q= ((A + (A + B)’). (B + (A + B)’))’
= (A + (A + B)’)’ + (B + (A + B)’)’

Applying compliment again
Q’’’=Q’= [(A + (A + B)’)’ + (B + (A + B)’)’]’

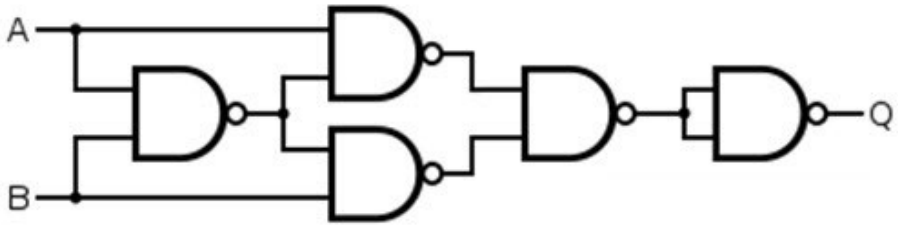
Applying compliment again
Q’’’’=Q’’=Q={ [(A + (A + B)’)’ + (B + (A + B)’)’]’}’
Now we can implement the XOR gate using NOR gates.



10. Can you explain the implementation process of EX-NOR, using NAND gates?
Basically the “Exclusive-NOR” gate is a combination of the Exclusive-OR gate and the NOT gate but has a truth table similar to the standard NOR gate in that it has an output that is normally at logic level “1” and goes “LOW” to logic level “0” when ANY of its inputs are at logic level “1”. However, an output “1” is only obtained if BOTH of its inputs are at the same logic level, either binary “1” or “0”.
Q = (A XOR B)’
= (A’B + AB’)’
Q = (A’B + AB’ + AA’ + BB’)’
= ((A + B) (A’ + B’))’
Q = ((A + B) (AB)’)’
= (A . (AB)’ + B. (AB)’)’
=((A . (AB)’)’ . (B. (AB)’)’
Taking compliment
Q’= (((A . (AB)’)’ . (B. (AB)’)’)’

Taking compliment again
Q’’= (((A . (AB)’)’ . (B. (AB)’)’)’)’

Now we can implement XNOR gate using NAND gates



11. Can you explain the implementation process of EX-NOR, using NOR gates?

Basically the "Exclusive-NOR" gate is a combination of the Exclusive-OR gate and the NOT gate but has a truth table similar to the standard NOR gate in that it has an output that is normally at logic level "1" and goes "LOW" to logic level "0" when ANY of its inputs are at logic level "1". However, an output "1" is only obtained if BOTH of its inputs are at the same logic level, either binary "1" or "0".

If A and B are inputs and Q is the output of XNOR gate then

Q = A XNOR B

= (A XOR B)'

Q = (A'B + AB')'

= (A'B + AB' + AA' + BB')'

Q= [A' (A + B) + B' (A + B)]'

Take compliment

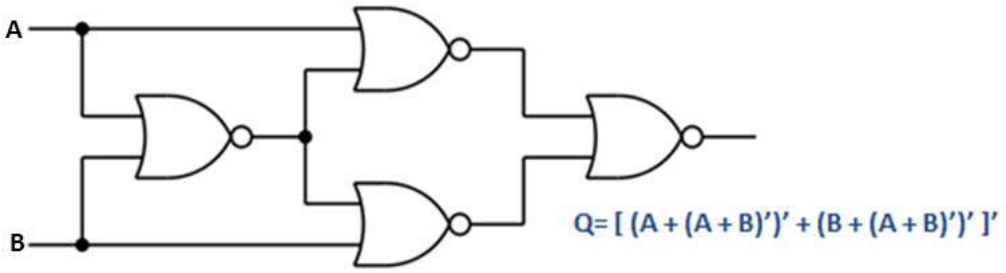
Q' = {[A' (A + B) + B' (A + B)]'}'

= (A + (A + B)')'+ (B + (A + B)')'

Take compliment again

Q''=Q = ((A + (A + B)')'+ (B + (A + B)')')'

Now we can implement the XNOR circuit using NOR gates

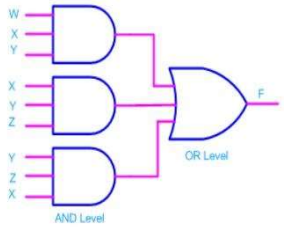


12. Can you explain the implement process of F = W.X.Y + X.Y.Z + Y.Z.W expression using NAND gates?

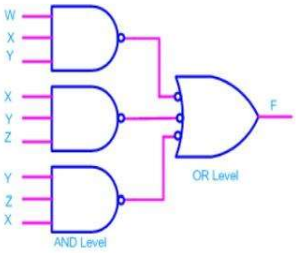
Any logic function can be implemented using NAND gates. To achieve this, first the logic function has to be written in Sum of Product (SOP) form. Once logic function is converted to SOP, then is very easy to implement using NAND gate. In other words any logic circuit with AND gates in first level and OR gates in second level can be converted into a NAND-NAND gate circuit.

F = W.X.Y + X.Y.Z + Y.Z.W

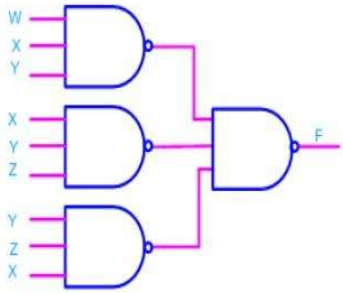
The above expression can be implemented with three AND gates in first stage and one OR gate in second stage as shown in figure.



If bubbles are introduced at AND gates output and OR gates inputs (the same for NOR gates), the above circuit becomes as shown in figure.



Now replace OR gate with input bubble with the NAND gate. Now we have circuit which is fully implemented with just NAND gates.



UNIT 4
QUESTION BANK
Multiple Choice Questions
(Questions for Understanding)

a. On a master-slave flip-flop, when is the master enabled?
   A. When the gate is 0
   B. **When the gate is 1**
   C. Both of the above
   D. Neither of the above

b. Table that is not a part of asynchronous analysis procedure is
   A. Transition table
   B. State table
   C. **Flow table**
   D. Excitation table

c. In excitation table of D flip-flop next state is equal to
   A. Present state
   B. Next state
   C. Input state
   D. **D state**

d. One example of the use of an S-R flip-flop is as a(n):
   A. Racer
   B. Stable oscillator
   C. **Binary storage register**
   D. Transition pulse generator

e. Which of the following is correct for a gated D flip-flop?
   A. The output toggles if one of the inputs is held 1.
   B. Only one of the inputs can be 1 at a time.
   C. The output complement follows the input when enabled.
   D. **Q output follows the input d when the enable is 1.**

f. What is one disadvantage of an S-R flip-flop?
   A. It has no enable input.
   B. **It has an invalid state.**
   C. It has no clock input.
   D. It has only a single output.

g. Why do the D-flip-flop s receives its designation or nomenclature as 'Data Flip-flops'
   A. Due to its capability to receive data from flip-flop
   B. Due to its capability to store data in flip-flop
   C. **Due to its capability to transfer the data into flip-flop**
   D. All of the above

h. The characteristic equation of D-flip-flop implies that _____.
   A. The next state is dependent on previous state
   B. The next state is dependent on present state
   C. The next state is independent of previous state

D. **The next state is independent of present state**

i. Two cross coupled NAND gates make
   A. SR Latch
   B. **RS flip-flop**
   C. D flip-flop
   D. Master slave flip-flop

j. Input clock of RS flip-flop is given to
   A. **Input**
   B. Pulsar
   C. Output
   D. Master slave flip-flop

k. D flip-flop is a circuit having
   A. 2NAND gates
   B. 3NAND gates
   C. **4NAND gates**
   D. 5NAND gates

l. D flip-flop is constructed with
   A. AND gates
   B. OR gates
   C. **NAND gates**
   D. NOR gates

(Questions for Skill)

m. How is a J-K flip-flop made to toggle?
   A. J = 0, K = 0
   B. J = 1, K = 0
   C. J = 0, K = 1
   D. **J = 1, K = 1**

n. A J-K flip-flop is in a "no change" condition when _____.
   A. J = 1, K = 1
   B. J = 1, K = 0
   C. J = 0, K = 1
   D. **J = 0, K = 0**

o. On a J-K flip-flop, when is the flip-flop in a hold condition?
   A. **J = 0, K = 0**
   B. J = 1, K = 0
   C. J = 0, K = 1
   D. J = 1, K = 1

p. The output of a gated S-R flip-flop changes only if the:
   A. Flip-flop is set
   B. **Control input data has changed**
   C. Flip-flop is reset

D. Input data has no change

q. If both inputs of an S-R flip-flop are low, what will happen when the clock goes HIGH?
   A. An invalid state will exist.
   B. **No change will occur in the output.**
   C. The output will toggle.
   D. The output will reset.

r. For an S-R flip-flop to be set or reset, the respective input must be:
   A. Installed with steering diodes
   B. In parallel with a limiting resistor
   C. 0
   D. **1**

s. If both inputs of an S-R flip-flop are 0, what will happen when the clock goes high?
   A. **No change will occur in the output.**
   B. An invalid state will exist.
   C. The output will toggle.
   D. The output will reset.

t. What is the bit storage binary information capacity of any flip-flop?
   A. **1 bit**
   B. 2 bits
   C. 16 bits
   D. Infinite bits

u. What is the difference between combinational logic and sequential logic?
   A. **Combinational circuits are not triggered by timing pulses, sequential circuits are triggered by timing pulses**
   B. Combinational and sequential circuits are both triggered by timing pulses.
   C. Neither circuit is triggered by timing pulses.
   D. No Difference

v. The logic circuits whose outputs at any instant of time depends only on the present input but also on the past outputs are called _____
   A. Combinational circuits
   B. **Sequential circuits**
   C. Latches
   D. Flip-flops

w. Whose operations are faster among the following?
   A. Combinational circuits
   B. Sequential circuits
   C. **Latches**
   D. Flip-flops

x. What is a trigger pulse?
   A. **A pulse that starts a cycle of operation**
   B. A pulse that reverses the cycle of operation
   C. A pulse that prevents a cycle of operation

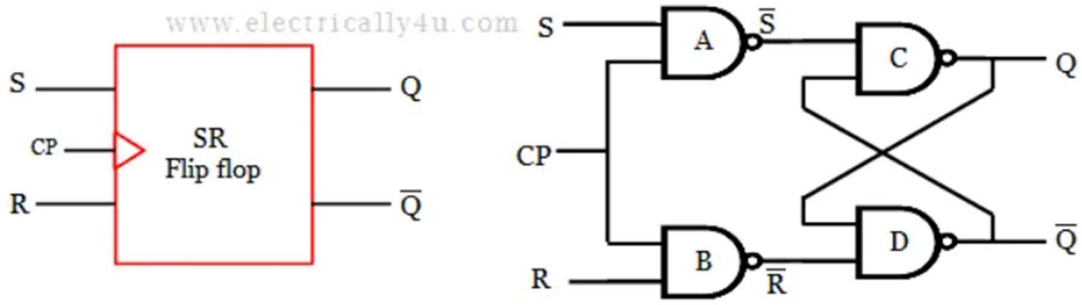
D. A pulse that enhances a cycle of operation

y. The circuits of NOR based S-R latch classified as asynchronous sequential circuits, why?
   A. Because of inverted outputs
   B. Because of triggering functionality
   C. **Because of cross-coupled connection**
   D. Because of inverted outputs & triggering functionality

Short answer questions
(Questions for Understanding)

1. How would you explain the working of clocked- SR flip flop using truth table and logic diagram?

The Clocked SR flip-flop consists of 4 NAND gates, two inputs(S and R) and two outputs(Q and Q'). The clock pulse is given at the inputs of gate A and B. If the clock pulse input is replaced by an enable input, then it is said to be SR latch. Let us assume that this flip flop works under positive edge triggering. The following figure shows the block diagram and the logic circuit of a clocked SR flip flop.



Clocked SR flip flop

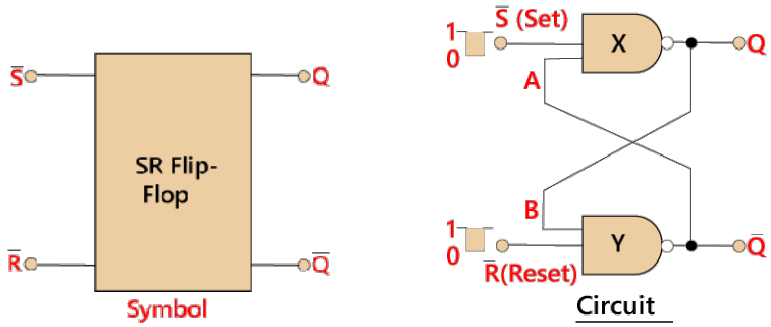
CP	S	R	Q	Q_{+1}	State
1	0	0	0	0	No change
1	0	0	1	1	
1	0	1	0	0	RESET
1	0	1	1	0	
1	1	0	0	1	SET
1	1	0	1	1	
1	1	1	0	x	Indeterminate
1	1	1	1	x	
0	x	x	0	0	No Change
0	x	x	1	1	

Truth table for clocked SR flip flop

2. How would you explain the working of SR latch using NAND gates truth table and logic diagram?

SR flip flop can be designed by cross coupling of two NAND gates. It is an active low

input SR flip – flop. The circuit of SR flip – flop using NAND gates is shown in below figure

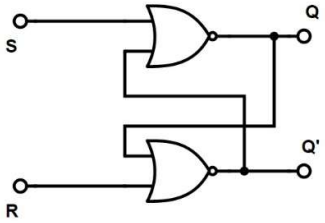


- **Case 1:** When both the SET and RESET inputs are high, then the output remains in previous state i.e. it holds the previous data.
- **Case 2:** When SET input is HIGH and RESET input is LOW, then the flip flop will be in RESET state. Because the low input of NAND gate with R input drives the other NAND gate with 1, as its output is 1. So both the inputs of the NAND gate with S input are 1. This will cause the output of the flip – flop to settle in RESET state.
- **Case 3:** When SET input is LOW and RESET input is HIGH, then the flip flop will be in SET state. Because the low input of NAND gate with S input drives the other NAND gate with 1, as its output is 1. So both the inputs of the NAND gate with R input are 1. This will cause the output of the flip – flop to settle in SET state.
- **Case 4:** When both the SET and RESET inputs are low, then the flip flop will be in undefined state. Because the low inputs of S and R, violates the rule of flip – flop that the outputs should compliment to each other. So the flip flop is in undefined state (or forbidden state).

S	R	Q	Q′
0	0	Memory/hold state	
0	1	0	1
1	0	1	0
1	1	Invalid state (not used)	

3. How would you explain the working of SR latch using NOR gates truth table and logic diagram?

SR flip flop can also be designed by cross coupling of two NOR gates. It is an active high input SR flip – flop. The circuit of SR flip – flop using NOR gates is shown in below figure.



S	R	Q	State
0	0	Previous State	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Forbidden

The operation is same as that of NOR SR Latch.

- **Case 1:** When both the SET and RESET inputs are low, then the output remains in previous state i.e. it holds the previous data
- **Case 2:** When SET input is low and RESET input is high, then the flip flop will be in RESET state. Because the high input of NOR gate with R input drives the other NOR gate with 0, as its output is 0. So both the inputs of the NOR gate with S input are 0. This will cause the output of the flip – flop to settle in RESET state.
- **Case 3:** When SET input is high and RESET input is low, then the flip flop will be in SET state. Because the low input of NOR gate with S input drives the other NOR gate with 1, as its output is 1. So both the inputs of the NOR gate with R input are 1. This will cause the output of the flip flop to settle in SET state.
- **Case 4:** When both the SET and RESET inputs are high, then the flip flop will be undefined state. Because the high inputs of S and R, violates the rule of flip flop that the outputs should complement to each other. So the flip flop is in undefined state (or forbidden state).

4. Can you illustrate State machine notations?

The information available in a state table may be represented graphically in a state diagram. In this diagram, a state is represented by a circle, and the transition between states is indicated by directed lines connecting the circles.

- **Input Variables:** External input variables to sequential machine as inputs.
- **Output Variables:** All variables that exit from the sequential machine are output variables.
- **State:** State of sequential machine is defined by the content of memory, when memory is realized by using FFs.
- **Present State:** The status of all state variable i.e. content of FF for given instant of time t is called as present state.
- **Next State:** The state of memory at t+1 is called as Next state.
- **State Diagram:** State diagram is graphical representation of state variables represented by circle. The connection between two states represented by lives with arrows and also indicates the excitation input and related outputs.

- **Output Variables:** All variables that exit from the sequential machine are output variables.

5. Can you write a brief outline of Flip flops?

Flip flops are actually an application of logic gates. With the help of Boolean logic you can create memory with them. The most commonly used application of flip flops is in the implementation of a feedback circuit. As a memory relies on the feedback concept, flip flops can be used to design it. There are mainly four types of flip flops that are used in electronic circuits. They are

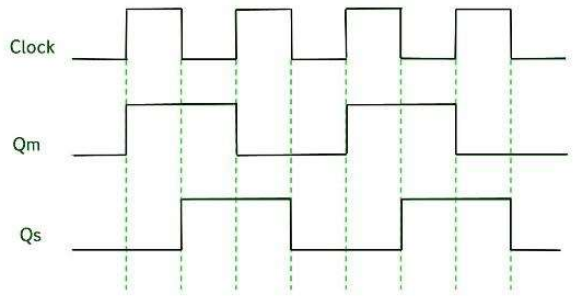
1. The basic Flip Flop or S-R Flip Flop
2. Delay Flip Flop [D Flip Flop]
3. J-K Flip Flop
4. T Flip Flop

The term flip – flop is used as they can switch between the states under the influence of a control signal (clock or enable) i.e. they can 'flip' to one state and 'flop' back to other state.

- Flip – flops are a binary storage device because they can store binary data (0 or 1).
- Flip – flops are edge sensitive or edge triggered devices i.e. they are sensitive to the transition rather than the duration or width of the clock signal.
- They are also known as signal change sensitive devices which mean that the change in the level of clock signal will bring change in output of the flip flop.
- A Flip – flop works depending on clock pulses.
- Flip flops are also used to control the digital circuit's functionality. They can change the operation of a digital circuit depending on the state.

6. Can you explain the process of Timing Diagram of a Master flip flop?

The Master-Slave Flip-Flop is basically a combination of two JK flip-flops connected together in a series configuration. Out of these, one acts as the **"master"** and the other as a **"slave"**. The output from the master flip flop is connected to the two inputs of the slave flip flop whose output is fed back to inputs of the master flip flop.



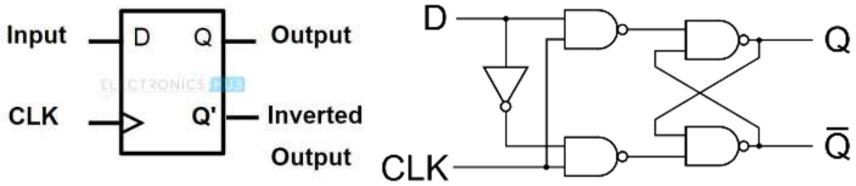
1. When the Clock pulse is high the output of master is high and remains high till the clock is low because the state is stored.
2. Now the output of master becomes low when the clock pulse becomes high again and remains low until the clock becomes high again.
3. Thus toggling takes place for a clock cycle.

4. When the clock pulse is high, the master is operational but not the slave thus the output of the slave remains low till the clock remains high.
5. When the clock is low, the slave becomes operational and remains high until the clock again becomes low.
6. Toggling takes place during the whole process since the output is changing once in a cycle.

(Questions for application)

7. Can you explain the working of D flip flop?

D flip – flops are also called as "Delay flip – flop" or "Data flip – flop". They are used to store 1 – bit binary data. They are one of the widely used flip – flops in digital electronics. Apart from being the basic memory element in digital systems, D flip – flops are also considered as Delay line elements and Zero – Order Hold elements. D flip – flop has two inputs , a clock (CLK) input and a data (D) input and two outputs; one is main output represented by Q and the other is complement of Q represented by Q'.



In a D flip flop, the output can be only changed at the clock edge, and if the input changes at other times, the output will be unaffected.

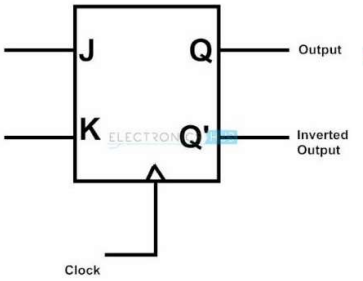
Inputs		Outputs		State
D	CLK	Q	Q'	
0	1	0	1	RESET
1	1	1	0	SET

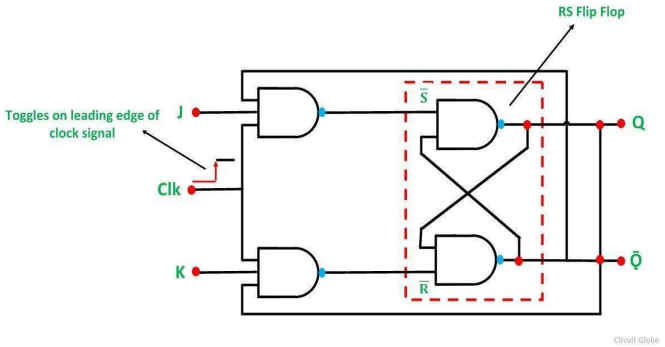
The change of state of the output is dependent on the rising edge of the clock. The output (Q) is same as the input and can only change at the rising edge of the clock.

D flip flop is actually a slight modification of the above explained clocked SR flip-flop. From the figure you can see that the D input is connected to the S input and the complement of the D input is connected to the R input. The D input is passed on to the flip flop when the value of CP is '1'. When CP is HIGH, the flip flop moves to the SET state. If it is '0', the flip flop switches to the clear state.

8. With a neat diagram can you explain the working of a JK flip flop?

A JK flip – flop is the modification of SR flip – flop with no illegal state. In this the J input is similar to the SET input of SR flip – flop and the K input is similar to the RESET input of SR flip – flop. A JK flip – flop is the modification of SR flip – flop with no illegal state. In this the J input is similar to the SET input of SR flip – flop and the K input is similar to the RESET input of SR flip – flop. The design of the JK flip – flop is such that the three inputs to one NAND gate are J, clock signal along with a feedback signal from Q' and the three inputs to the other NAND are K, clock signal along with a feedback signal from Q. This arrangement eliminates the indeterminate state in SR flip – flop.



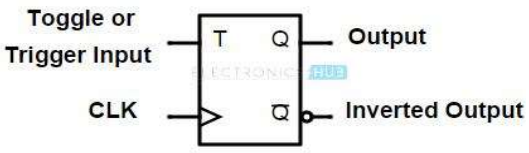


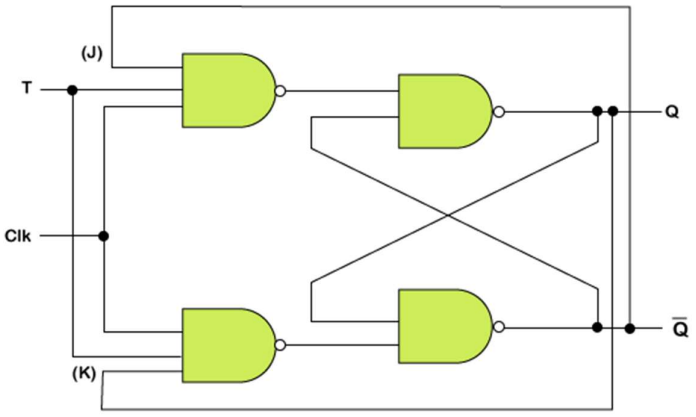
	J	K	Q	Q̄	Description
Same as for the RS Latch	0	0	0	0	Memory No Change
	0	0	0	1	
	0	1	1	0	Reset Q >> 0
	0	1	0	1	
	1	0	0	1	Set Q >> 1
	1	0	1	0	
Toggle	1	1	0	1	Toggle
	1	1	1	0	

9. Can you explain the working of T flip flop? Write the truth table and logical Expression?

T flip – flop is also known as "Toggle Flip – flop". To avoid the occurrence of intermediate state in SR flip – flop, we should provide only one input to the flip – flop called Trigger input or Toggle input (T). Then the flip – flop acts as a Toggle switch. Toggling means 'Changing the next state output to complement of the present state output'.

The logic symbol of T flip – flop is shown below. It has one Toggle input (T) & one clock signal input (CLK).

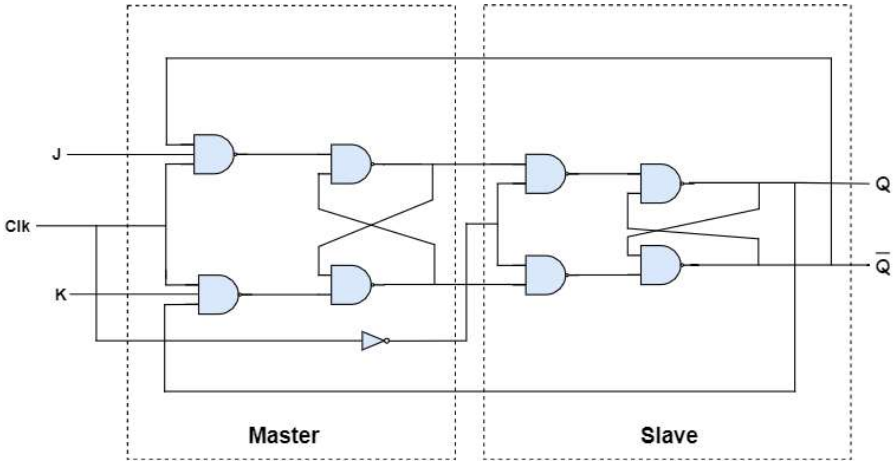




	Previous		Next	
T	Q	Q'	Q	Q'
0	0	1	0	1
0	1	0	1	0
1	0	1	1	0
1	1	0	0	1

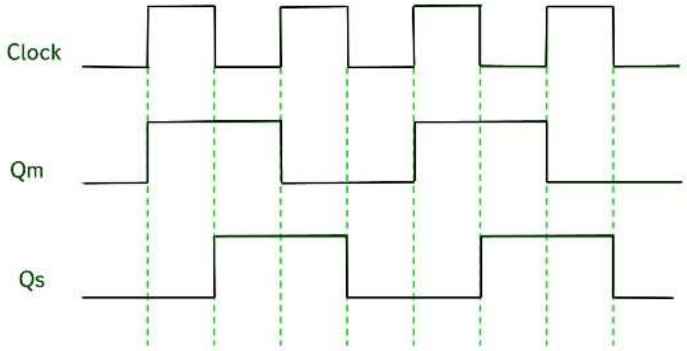
10. Can you explain the working of Master-Slave flip-flop with timing diagram and logic diagram?

The Master-Slave Flip-Flop is basically a combination of two JK flip-flops connected together in a series configuration. Out of these, one acts as the **"master"** and the other as a **"slave"**. The output from the master flip flop is connected to the two inputs of the slave flip flop whose output is fed back to inputs of the master flip flop. In addition to these two flip-flops, the circuit also includes an **inverter**. The inverter is connected to clock pulse in such a way that the inverted clock pulse is given to the slave flip-flop. In other words if CP=0 for a master flip-flop, then CP=1 for a slave flip-flop and if CP=1 for master flip flop then it becomes 0 for slave flip flop.



1. When the clock pulse goes to 1, the slave is isolated; J and K inputs may affect the state of the system. The slave flip-flop is isolated until the CP goes to 0. When the CP goes back to 0, information is passed from the master flip-flop to the slave and output is obtained.

2. Firstly the master flip flop is positive level triggered and the slave flip flop is negative level triggered, so the master responds before the slave.
3. If J=0 and K=1, the high Q' output of the master goes to the K input of the slave and the clock forces the slave to reset, thus the slave copies the master.
4. If J=1 and K=0, the high Q output of the master goes to the J input of the slave and the Negative transition of the clock sets the slave, copying the master.
5. If J=1 and K=1, it toggles on the positive transition of the clock and thus the slave toggles on the negative transition of the clock.
6. If J=0 and K=0, the flip flop is disabled and Q remains unchanged.



11. Write a note on triggering the flip flops.

The output of a flip flop can be changed by bring a small change in the input signal. This small change can be brought with the help of a clock pulse or commonly known as a trigger pulse.When such a trigger pulse is applied to the input, the output changes and thus the flip flop is said to be triggered. Flip flops are applicable in designing counters or registers which stores data in the form of multi-bit numbers. But such registers need a group of flip flops connected to each other as sequential circuits. And these sequential circuits require trigger pulses.

The number of trigger pulses that is applied to the input of the circuit determines the number in a counter. A single pulse makes the bit move one position, when it is applied onto a register that stores multi-bit data. In the case of SR Flip Flops, the change in signal level decides the type of trigger that is to be given to the input. But the original level must be regained before giving a second pulse to the circuit. If a clock pulse is given to the input of the flip flop at the same time when the output of the flip flop is changing, it may cause instability to the circuit. The reason for this instability is the feedback that is given from the output combinational circuit to the memory elements. This problem can be solved to a certain level by making the flip flop more sensitive to the pulse transition rather than the pulse duration.

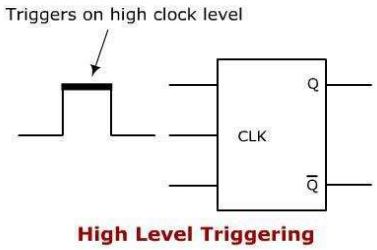
12. Can you explain the types of pulse-triggering methods?

The output of a flip flop can be changed by bring a small change in the input signal. This small change can be brought with the help of a clock pulse or commonly known as a trigger pulse.

There are mainly four types of pulse-triggering methods. They differ in the manner in which the electronic circuits respond to the pulse. They are

1. **High Level Triggering**

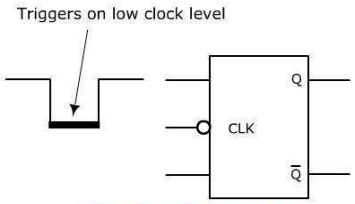
When a flip flop is required to respond at its HIGH state, a HIGH level triggering method is used. It is mainly identified from the straight lead from the clock input. Take a look at the symbolic representation shown below.



High Level Triggering

2. **Low level triggering**

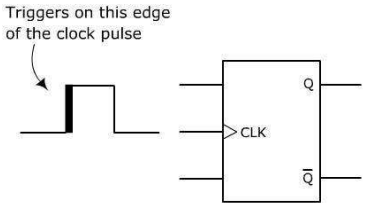
When a flip flop is required to respond at its LOW state, a LOW level triggering method is used.. It is mainly identified from the clock input lead along with a low state indicator bubble. Take a look at the symbolic representation shown below.



Low Level Triggering

3. **Positive edge triggering**

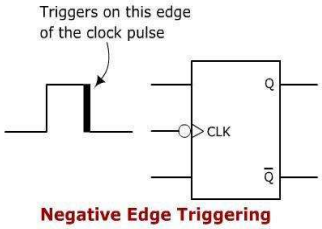
When a flip flop is required to respond at a LOW to HIGH transition state, POSITIVE edge triggering method is used. It is mainly identified from the clock input lead along with a triangle. Take a look at the symbolic representation shown below



Positive Edge Triggering

4. **Negative edge triggering**

When a flip flop is required to respond during the HIGH to LOW transition state, a NEGATIVE edge triggering method is used.. It is mainly identified from the clock input lead along with a low-state indicator and a triangle. Take a look at the symbolic representation shown below.



Negative Edge Triggering

UNIT 5
QUESTION BANK

Multiple Choice Questions

(Questions for application)

a. In digital logic, a counter is a device which _____
   A. Counts the number of outputs
   B. **Stores the number of times a particular event or process has occurred**
   C. Stores the number of times a clock pulse rises and falls
   D. Counts the number of inputs

b. A counter circuit is usually constructed of _____
   A. A number of latches connected in cascade form
   B. A number of NAND gates connected in cascade form
   C. **A number of flip-flops connected in cascade**
   D. A number of NOR gates connected in cascade form

c. How many types of the counter are there?
   A. 2
   B. **3**
   C. 4
   D. 5

d. A decimal counter has _____ states.
   A. 5
   B. **10**
   C. 1
   D. 20

e. Synchronous counter is a type of _____
   A. SSI counters
   B. LSI counters
   C. **MSI counters**
   D. VLSI counters

f. The parallel outputs of a counter circuit represent the _____
   A. Parallel data word
   B. Clock frequency
   C. Counter modulus
   D. **Clock count**

g. Internal propagation delay of asynchronous counter is removed by _____
   A. Ripple counter
   B. Ring counter
   C. Modulus counter
   D. **Synchronous counter**

h. Internal propagation delay of asynchronous counter is removed by _____
   A. Ripple counter
   B. Ring counter
   C. Modulus counter

D. **Synchronous counter**

i. A register is defined as _____
   A. The group of latches for storing one bit of information
   B. The group of latches for storing n-bit of information
   C. The group of flip-flops suitable for storing one bit of information
   D. **The group of flip-flops suitable for storing binary information**

j. What is meant by the parallel load of a shift register?
   A. **All FFs are preset with data**
   B. Each FF is loaded with data, one at a time
   C. Parallel shifting of data
   D. All FFs are set with data

k. The register is a type of _____
   A. **Sequential circuit**
   B. Combinational circuit
   C. CPU
   D. Latches

l. In D register, 'D' stands for _____
   A. Delay
   B. Decrement
   C. **Data**
   D. Decay

(Questions for skill)

m. How many natural states will there be in a 4-bit ripple counter?
   A. 4
   B. 8
   C. **16**
   D. 32

n. One of the major drawbacks to the use of asynchronous counters is that _____
   A. Low-frequency applications are limited because of internal propagation delays
   B. **High-frequency applications are limited because of internal propagation delays**
   C. Asynchronous counters do not have major drawbacks and are suitable for use in high- and low-frequency counting applications
   D. Asynchronous counters do not have propagation delays, which limits their use in high-frequency applications

o. What happens to the parallel output word in an asynchronous binary down counter whenever a clock pulse occurs?
   A. The output increases by 1
   B. **The output decreases by 1**
   C. The output word increases by 2
   D. The output word decreases by 2

p. Why the extent of propagation delay in is synchronous counter much lesser than that of asynchronous counter?

A. **Due to clocking of all flip flops at the same instant**
B. Due to increase in number of states
C. Due to absence of connection between output of preceding flip flop and clock of next on
D. Due to absence of mode control operation

q. A ripple counter's speed is limited by the propagation delay of _____
A. **Each flip-flop**
B. All flip-flops and gates
C. The flip-flops only with gates
D. Only circuit gates

r. An asynchronous 4-bit binary down counter changes from count 2 to count 3. How many transitional states are required
A. 1
B. 2
C. 8
D. **15**

s. A principle regarding most display decoders is that when the correct input is present, the related output will switch _____
A. HIGH
B. To high impedance
C. To an open
D. **LOW**

t. The main difference between a register and a counter is _____
A. **A register has no specific sequence of states**
B. A counter has no specific sequence of states
C. A register has capability to store one bit of information but counter has n-bit
D. A register counts data

u. Registers capable of shifting in one direction is _____
A. Universal shift register
B. **Unidirectional shift register**
C. Unipolar shift register
D. Unique shift register

v. A shift register is defined as _____
A. The register capable of shifting information to another register
B. **The register capable of shifting information either to the right or to the left**
C. The register capable of shifting information to the right only
D. The register capable of shifting information to the left only

w. In serial shifting method, data shifting occurs _____
A. **One bit at a time**
B. Simultaneously
C. Two bit at a time
D. Four bit at a time

x. A shift register that will accept a parallel input or a bidirectional serial load and internal shift features is called as?
   A. Tristate
   B. End around
   C. **Universal**
   D. Conversion

y. How can parallel data be taken out of a shift register simultaneously?
   A. Use the Q output of the first FF
   B. Use the Q output of the last FF
   C. Tie all of the Q outputs together
   D. **Use the Q output of each FF**

Short answer questions
(Questions for application)

1. Can you illustrate the steps of designing Synchronous Counters?

Synchronous counter has one global clock which drives each flip flop so output changes in parallel. The one advantage of synchronous counter over asynchronous counter is, it can operate on higher frequency than asynchronous counter as it does not have cumulative delay because of same clock is given to each flip flop.

Step 1: Number of flip-flops: Based on the description of the problem, determine the required number 'n' of the FFs-the smallest value of n is such that the number of states N < $2^n$and the desired counting sequence.

Step 2: State diagram: Draw the state diagram showing all the possible states. A state diagram, which can also be called the transition diagram, is a graphical means of depicting the sequence of states through which the counter progresses. In case the counter goes to a particular state from the invalid states on the next clock pulse, the same can also be included in the state diagram.

Step 3: Choice of flip-flops and excitation table: Select the type of flip-flops to be used and write the excitation table. An excitation table is a table that lists the present state (PS), the next state (NS) and the required excitations.

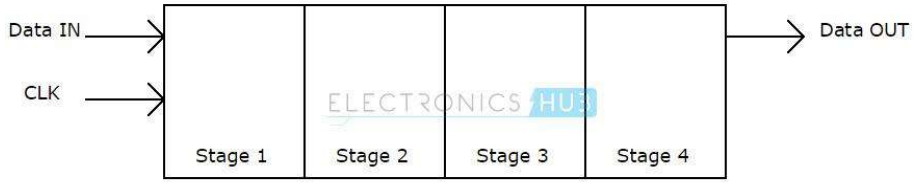
Step 4: Minimal expressions for excitations: Obtain the minimal expressions for the excitations of the FFs using the K-maps drawn for the excitations of the flip-flops in terms of the present states and inputs.

Step 5: Logic diagram: Draw a logic diagram based on the minimal expressions.

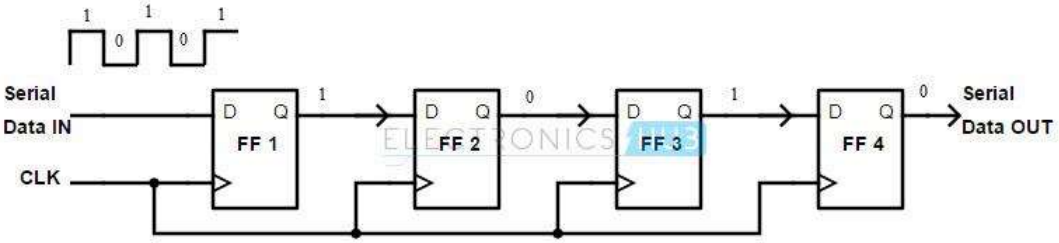
2. Can you explain the process of Serial In/Serial out Shift Registers?

The input to this register is given in serial fashion i.e. one bit after the other through a single data line and the output is also collected serially. The data can be shifted only left or shifted only right. Hence it is called Serial in Serial out shift register or a SISO shift register.

As the data is fed from right as bit by bit, the shift register shifts the data bits to left. A 4-bit SISO shift register consists of 4 flip flops and only three connections. The registers which will shift the bits to left are called "Shift left registers". The registers which will shift the bits to right are called "Shift right registers".



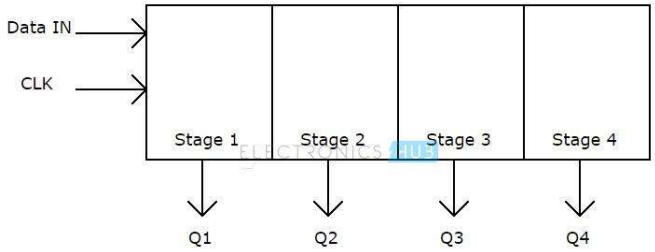
This one is the simplest register among the four types. As the clock signal is connected to all the 4 flip flops, the serial data is connected to the left most or right most flip flop. The output of the first flip flop is connected to the input of the next flip flop and so on. The final output of the shift register is collected at the outmost flip flop.



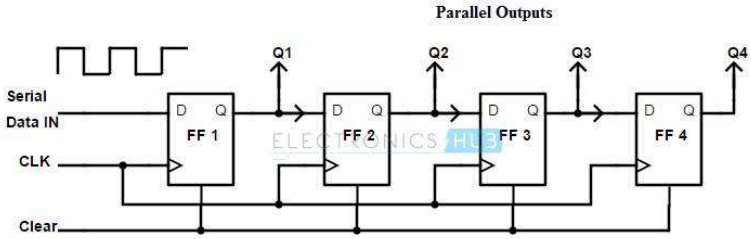
3. Can you explain the process of Serial In/Parallel Out Shift Registers?

Shift registers consist of arrangements of flip-flops and are important in applications involving the storage and transfer of data in a digital system. A register has no specified sequence of states, except in certain very specialized applications. A register, in general, is used solely for storing and shifting data (1s and 0s) entered into it from an external source and typically possesses no characteristic internal sequence of states.

The input to this register is given in serial and the output is collected in parallel



The clear (CLR) signal is connected in addition to clock signal to all the 4 flip flops in order to RESET them and the serial data is connected to the flip flop at either end (depending on shift left register or shift right register). The output of the first flip flop is connected to the input of the next flip flop and so on. All the flip flops are connected with a common clock.

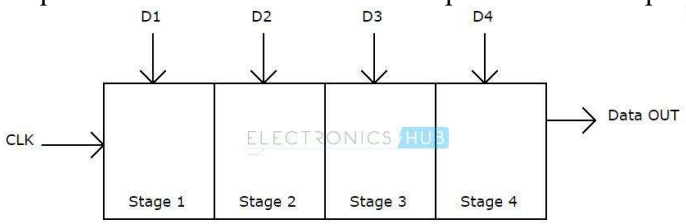


Unlike the serial in serial out shift registers, the output of Serial in Parallel out (SIPO) shift register is collected at each flip flop. Q1, Q2, Q3 and Q4 are the outputs of first, second, third and fourth flip flops, respectively. The main application of Serial in Parallel out shift register is to convert serial data into parallel data.

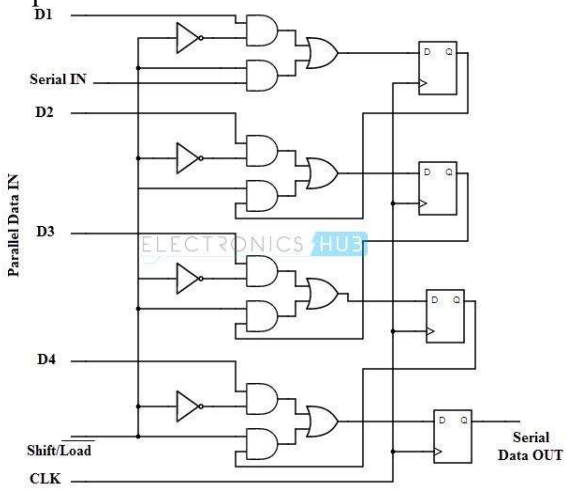
4. Can you explain the process of Parallel In/Serial Out Shift Registers?

Shift registers consist of arrangements of flip-flops and are important in applications involving the storage and transfer of data in a digital system. A register has no specified sequence of states, except in certain very specialized applications. A register, in general, is used solely for storing and shifting data (1s and 0s) entered into it from an external source and typically possesses no characteristic internal sequence of states.

The input to this register is given in parallel i.e. data is given separately to each flip flop and the output is collected in serial at the output of the end flip flop.



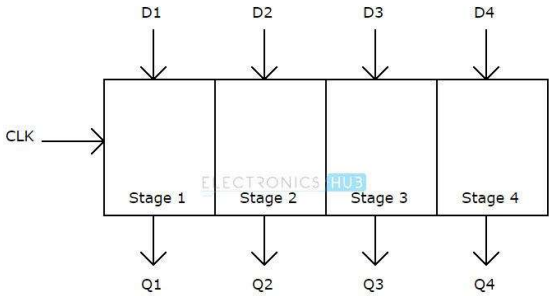
The clock input is directly connected to all the flip flops but the input data is connected individually to each flip flop through a mux (multiplexer) at input of every flip flop. Here D1, D2, D3 and D4 are the individual parallel inputs to the shift register. In this register the output is collected in serial.



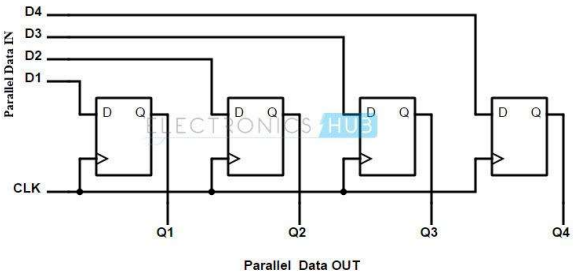
5. Can you explain the process of Parallel In/Parallel Out Shift Registers?

Shift registers consist of arrangements of flip-flops and are important in applications involving the storage and transfer of data in a digital system. A register has no specified sequence of states, except in certain very specialized applications. A register, in general, is used solely for storing and shifting data (1s and 0s) entered into it from an external source and typically possesses no characteristic internal sequence of states.

In this register, the input is given in parallel and the output also collected in parallel. The clear (CLR) signal and clock signals are connected to all the 4 flip flops. Data is given as input separately for each flip flop and in the same way, output also collected individually from each flip flop.



The above diagram shows the 4 stage parallel in parallel out register. Qa, Qb, Qc and Qd are the parallel outputs and P_a, P_b, P_c and P_d are the individual parallel inputs. There are no interconnections between any of the four flip flops.



A Parallel in Parallel out (PIPO) shift register is used as a temporary storage device and also as a delay element similar to a SISO shift register.

6. Can you give a brief outline of Registers?

Shift registers consist of arrangements of flip-flops and are important in applications involving the storage and transfer of data in a digital system. A register has no specified sequence of states, except in certain very specialized applications. A register, in general, is used solely for storing and shifting data (1s and 0s) entered into it from an external source and typically possesses no characteristic internal sequence of states.

SHIFT LEFT:

The following figure shows the block diagram of Shift-left register. Din sets up the right flip-flop, Q0 sets up the second flip-flop, Q1 the third, and so on. When the next positive clock edge strikes, therefore, the stored bits move one position to the left.

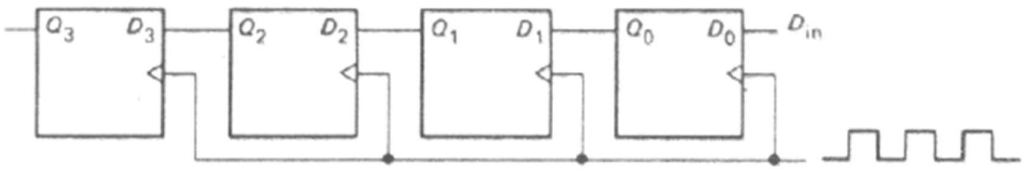


Fig.: Shift Left Register

SHIFT RIGHT:

The following figure shows the block diagram of Shift-right register. Here, each Q output sets up the D input of the preceding flip-flop. When the positive clock pulse arrives, the stored bits move one position to the right.

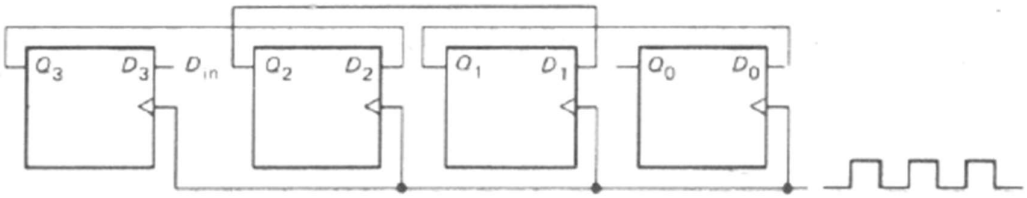


Fig.: Shift Left Register

The main type of shift register is the Serial-In, Serial-Out shift register, which is the basic shift

(Questions for skill)

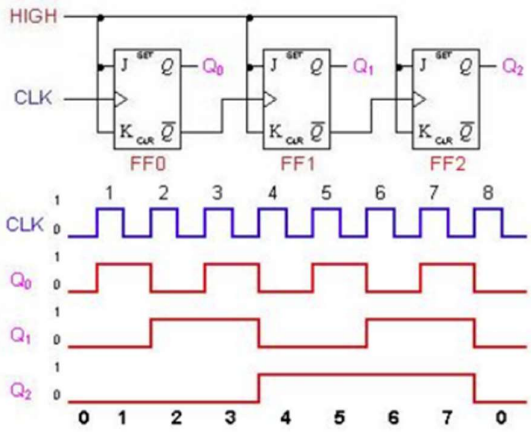
7. Can you explain 3-Bit Asynchronous Binary Counter with circuit diagram?

In asynchronous counter, a clock pulse drives FF0. Output of FF0 drives FF1 which then drives the FF2 flip flop. All J and K inputs are connected to Logic 1. Therefore, each flip flop will toggle with negative transition at its clock input.

The 3 bit MOD-8 asynchronous counter consists of 3 JK flip flops. Overall propagation delay time is the sum of individual delays. Initially all flip flops are reset to produce 0. The output conditions is Q2Q1Q0 = 000.

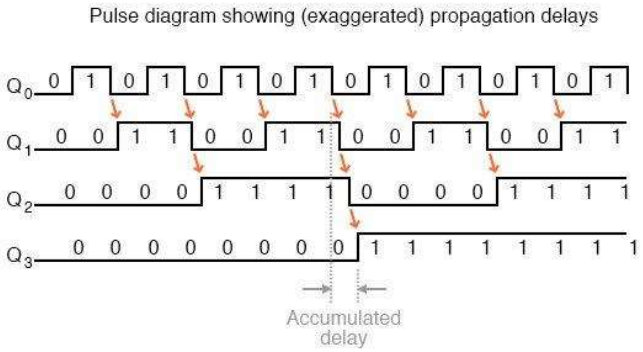
When the first clock pulse is applied, the FF0 changes state on its negative edge. Therefore, Q2Q1Q0 = 001. On the negative edge of second clock pulse flip flop FF0 toggles. Its output changes from 1 to 0. This being negative change, FF1 changes state. Therefore, Q2Q1Q0 = 010. Similarly, the output of flip flop FF2 changes only when there is negative transition at its input when fourth clock pulse is applied.

The output of the flip flops is a binary number equivalent to the number of clock pulses received. On the negative edge of eighth pulse, counter is reset. The counter acts as a frequency divider. FF0 divides clock frequency by 2, FF1 divides clock frequency by 4, FF2 divides clock frequency by 8.



8. Can you explain Propagation Delay in 3-Bit Asynchronous Binary Counter?

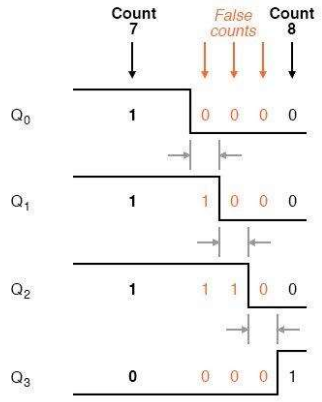
In asynchronous counter, a clock pulse drives FF0. Output of FF0 drives FF1 which then drives the FF2 flip flop. All J and K inputs are connected to Logic 1. Therefore, each flip flop will toggle with negative transition at its clock input. The 3 bit MOD-8 asynchronous counter consists of 3 JK flip flops. Overall propagation delay time is the sum of individual delays.



As you can see, the more bits that toggle with a given clock pulse, the more severe the accumulated delay time from LSB to MSB. When a clock pulse occurs at such a transition point (say, on the transition from 0111 to 1000), the output bits will “ripple” in sequence from LSB to MSB, as each succeeding bit toggles and commands the next bit to toggle as well, with a small amount of propagation delay between each bit toggle.

If we take a close-up look at this effect during the transition from 0111 to 1000, we can see that there will be false output counts generated in the brief time period that the “ripple”

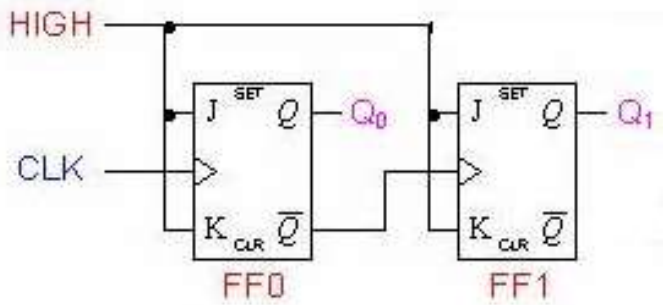
effect takes place:



Instead of cleanly transitioning from a "0111" output to a "1000" output, the counter circuit will very quickly ripple from 0111 to 0110 to 0100 to 0000 to 1000, or from 7 to 6 to 4 to 0 and then to 8. This behavior earns the counter circuit the name of ripple counter or asynchronous counter.

9. Can you explain 2-Bit Asynchronous Binary Counter with circuit diagram?

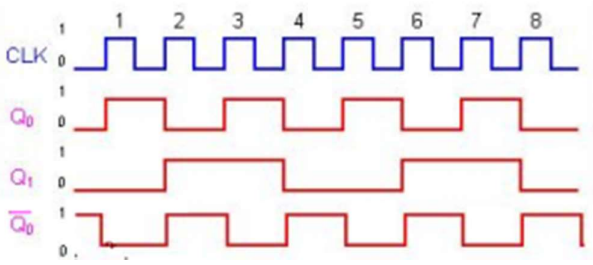
A 2-Bit Asynchronous Binary Counter Fig1-1 shows a 2-bit counter connected for asynchronous operation. Notice that the clock (CLK) is applied to the clock input (C) of only the first flop-flop, FF0, which is always the least significant bit (LSB). The second flip-flop, FF1, is triggered by the Q'0 out-put of FF0. FF0 changes state at the positive-going edge of each clock pulse. But FF1 changes only when triggered by a positive-going transition of the Q'0 output of FF0. Because of the inherent propagation delay tie through a flip-flop, a transition of the input clock pulse (CLK) and a transition of the Q'0 output of FF0 can never occur at exactly the same time. Therefore, the two flip-flops are never simultaneously triggered, so the counter operation is asynchronous.



Applying 4clock pulses to FF0, Both flip-flops are connected for toggle operation (J=1, K=1) and initially RESET (Q LOW). The positive-going edge of CLK1 (clock pulse1) causes the Q 0 output of FF0 to go HIGH. At the same time the Q'0 output goes LOW, but it has no effect on FF1 because a positive-going transition must occur to trigger the flip-flop.

After the leading edge of CLK1, Q0=1 & Q1=0. The positive-going edge of CLK2 causes Q0 to go LOW. Q'0 goes HIGH and triggers FF1, causing Q1 to go HIGH. After the leading edge of CLK2, Q0=0 & Q1=1. The positive-going edge of CLK3 causes Q0 to go HIGH again. Output Q'0goes LOW and has no effect on FF1. Thus, after the leading edge of CLK3, Q'0=1 & Q1=1. The positive-going edge of CLK4 causes Q0 to go LOW, while Q'0 goes HIGH and triggers FF1, causing Q 1 to go LOW.

After the leading edge of CLK4, Q0=0 & Q1=0. The 2-bit counter exhibits four different states, as you would expect with two flip-flops (2^2= 4). The fourth pulse it recycles to its original state (Q0=0, Q1=0). The term recycles; it refers to the transition of the counter from its final state back to its original state.



10. **Can you design a Synchronous 3-bit up-down counter using JK Flip Flops?**

Step 1. Determine the number of flip-flops required: A 3-bit counter requires three FFs. It has 8 states (000, 001, 010, all, 100, 101, 110, 111) and all the states are valid.

Step 2. Draw the state diagram: when M=0 it counts down the order of numbers and when M=1 it counts up the order of numbers.

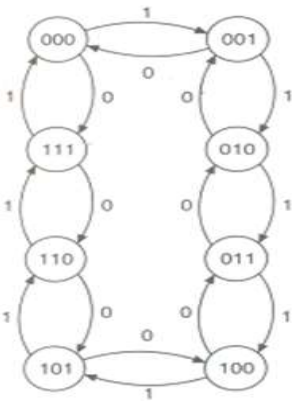


Fig.: State Diagram of a 3 bit up-down counter

Step 3. Select the type of flip-flops and draw the excitation table:

PS			Mode	NS			Required excitations					
Q_3	Q_2	Q_1	M	Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	1	1	1	1	×	1	×	1	×
0	0	0	1	0	0	1	0	×	0	×	1	×
0	0	1	0	0	0	0	0	×	0	×	×	1
0	0	1	1	0	1	0	0	×	1	×	×	1
0	1	0	0	0	0	1	0	×	×	1	1	×
0	1	0	1	0	1	1	0	×	×	0	1	×
0	1	1	0	0	1	0	0	×	×	0	×	1
0	1	1	1	1	0	0	1	×	×	1	×	1
1	0	0	0	0	1	1	×	1	1	×	1	×
1	0	0	1	1	0	1	×	0	0	×	1	×
1	0	1	0	1	0	0	×	0	0	×	×	1
1	0	1	1	1	1	0	×	0	1	×	×	1
1	1	0	0	1	0	1	×	0	×	1	1	×
1	1	0	1	1	1	1	×	0	×	0	1	×
1	1	1	0	1	1	0	×	0	×	0	×	1
1	1	1	1	0	0	0	×	1	×	1	×	1

Fig.: Excitation table of a 3-bit up-down counter

Step 4: Obtain the minimal expressions: From the excitation table we can conclude that J_1 = 1 and K_1=1, because all the entries for J_1 and K_1 are either X or 1. The K-maps for J_3, K_3, J_2 and K_2 based on the excitation table and the minimal expressions obtained from them

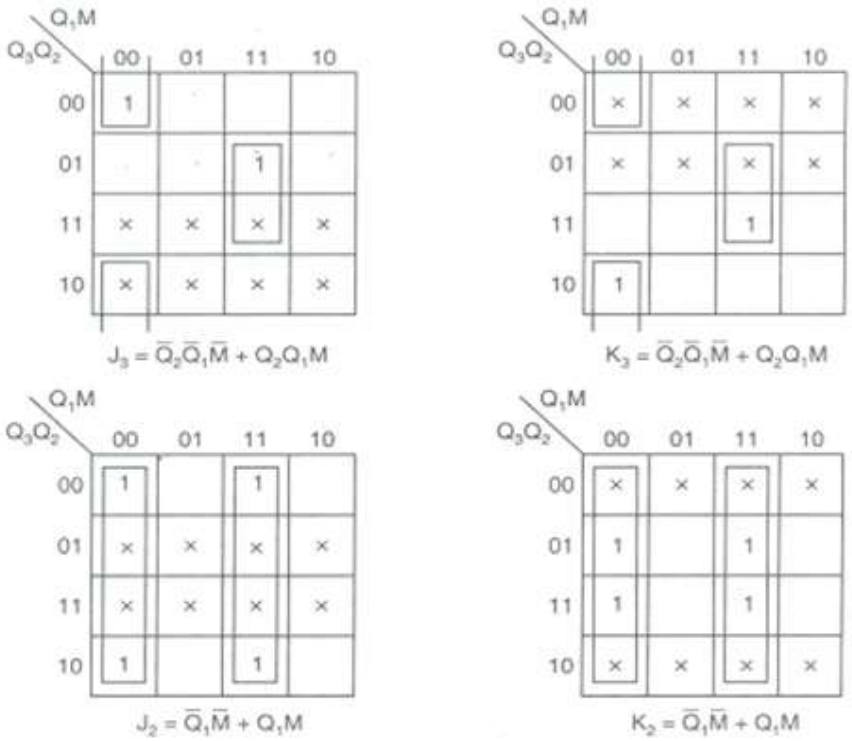
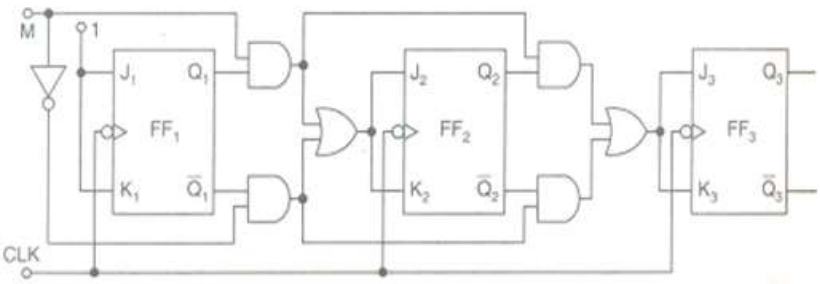


Fig : K-Maps for excitations of synchronous 3-bit up-down counter

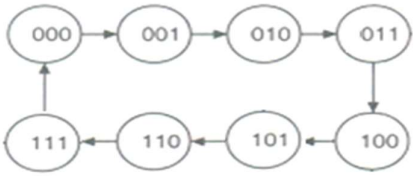
Step 5: Draw the logic diagram. The logic diagram for a synchronous 3-bit up-down counter



11. **Can you design a 3 bit binary counter (mod-8 counter) using T Flip Flops?**

Step 1: No. of flip flops required is 3 as the no. of bits is 3.

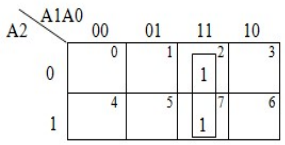
Step 2: State diagram

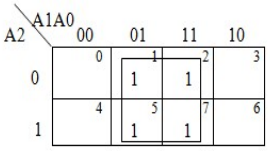


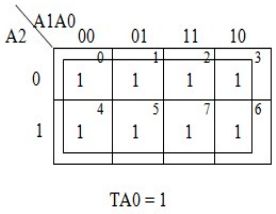
Step 3: Excitation Table

Present State			Next State			Required Excitations		
A2	A1	A0	A2	A1	A0	TA2	TA1	TA0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

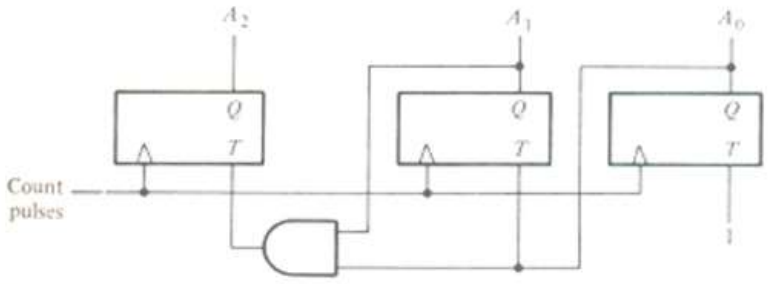
Step 4: Obtaining the minimal expression using K-Map:







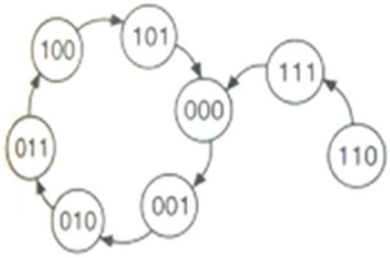
Step 5: Logic Diagram



12. Can you design a Mod·6 Counter Using JK Flip Flops?

Step 1. The number of flip-flops: We know that the counting sequence for a mod-6 counter is 000,001, 010,011, 100, 101, and 000. It has six states. So it requires n = 3 FFs.

Step 2. The state diagram: The state diagram for the mod-6 counter is drawn as shown in Figure.

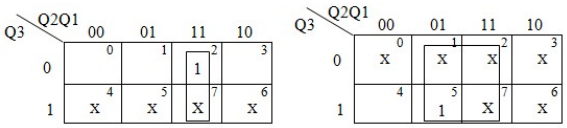


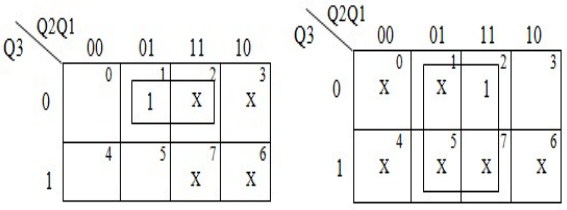
Step 3. The type of flip-flops and the excitation table

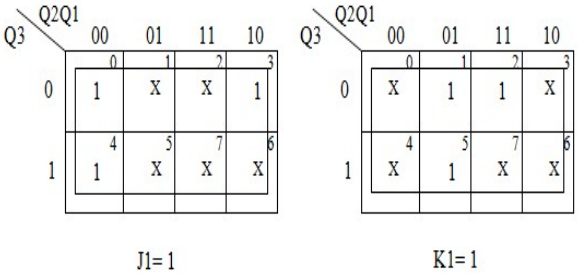
PS			NS			Required Excitations					
Q3	Q2	Q1	Q3	Q2	Q1	J3	K3	J2	K2	J1	K1
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	0	0	0	X	1	0	X	X	1

Fig. 6.18(b) Excitation table of a mod-6 counter using JK FF

Step 4. The minimal expressions: The K-maps for excitations of FFs J3, K3, J2, K2, J1, and K1 in terms of the outputs of FFs Q3, Q2 and Q1, their minimization, and the minimal expressions for excitations obtained from them are shown in fig







Step 5. The logic diagram: The logic diagram based on these minimal expressions is drawn