

Second Year Manuals

DATA STRUCTURE USING 'C'

(CSE-231-F)

LAB MANUAL
III Semsester

CONTENTS

SUBJECT: DATA STRUCTURE USING 'C' LAB (CSE-231-F)

s.no.	name of experiment	page no.
1	Write a program to search an element in a two-dimensional array using linear search.	
2	Using iteration & recursion concepts write programs for finding the element in the array using Binary Search Method	
3	Write a program to perform following operations on tables using functions only a) Addition b) Subtraction c) Multiplication d) Transpose	
4	Using iteration & recursion concepts write the programs for Quick Sort Technique	
5	Write a program to implement the various operations on string such as length of string concatenation, reverse of a string & copy of a string to another.	
6	Write a program for swapping of two numbers using 'call by value' and 'call by reference' strategies.	
7	Write a program to implement binary search tree. (Insertion and Deletion in Binary search Tree)	
8	Write a program to create a linked list & perform operations such as insert, delete, update, reverse in the link list	
9.	Write the program for implementation of a file and performing operations such as insert, delete, update a record in the file.	
10	Create a linked list and perform the following operations on it a) add a node b) Delete a node	
11	Write a program to simulate the various searching & sorting algorithms and compare their timings for a list of 1000 elements.	
12	Write a program to simulate the various graph traversing algorithms.	
13	Write a program which simulates the various tree traversal algorithms.	

EXPERIMENT NO.1

AIM: - Write a program to search an element in a two-dimensional array using linear search.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,j,item,loc=0,loc1=0;
    int a[2][2];
    clrscr();
    printf("\n\tThis Program is Used To seaech an element in 2Dimensional Array using Linear Search\n");
    printf("\n\tEnter The Value Of Array:");
    for(i=1;i<=2;i++)
    {
        for(j=1;j<=2;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("\n\tEnter The Value To Be Serched:");
    scanf("%d",&item);
    for(i=1;i<=2;i++)
    {
        for(j=1;j<=2;j++)
        {
            if(item==a[i][j])
            {
                loc=i;
                loc1=j;
                break;
            }
        }
    }

    printf("\n\tThe Item is at %d Row And %d Coloumn.",loc,loc1);
    printf("\n\n\t\tSearch Completed.");
    printf("\n\n\n\t\t\tTHANKYOU!");
    getch();
}
```

QUIZ

Q1. How does an array differ from an ordinary variable?

Ans.

Array Vs. Ordinary Variable

Array is made up of similar data structure that exists in any language. Array is set of similar data types. Array is the collection of similar elements. These similar elements could be all int or all float or all char etc. Array of char is known as string. All elements of the given array must be of same type. Array is finite ordered set of homogeneous elements. The number of elements in the array is pre-specified.

An ordinary variable of a simple data type can store a single element only.

For example: Ordinary variable: - int a

Array: - int a[10]

Q2. Two dimensional arrays are also called

- a. tables arrays
- b. matrix arrays
- c. both of above
- d. none of above

Ans. Matrix arrays

Q3. Which data structure can't store the non-homogeneous data elements?

Ans. Array

Q4. What is the main difference between ARRAY and STACK?

Ans: Stack follows LIFO. Thus the item that is first entered would be the last removed. In array the items can be entered or removed in any order. Basically each member access is done using index and no strict order is to be followed here to remove a particular element. Array may be multi dimensional or one dimensional but stack should be one-dimensional.

Size of array is fixed, while stack can be grow or shrink. We can say stack is dynamic data structure.

Q5. What is linear search technique?

Ans: linear search or sequential search is a method for finding a particular value in a list that consists of checking every one of its elements, one at a time and in sequence, until the desired one is found.

Q6. What is the use of clrscr() function?

Ans: To clear the previous result from output screen.

Q7. What is an array?

Ans: An array is defined as a collection of homogeneous elements.

Q8. How to declare a two dimensional array?

Ans: data_type arr_name[][];

Q9. What are the input and output functions used in this program?

Ans: printf() and scanf()

Q10. What is sparse matrix?

Ans: A sparse matrix is a matrix populated primarily with zeros

EXPERIMENT NO.2

AIM: Using iteration & recursion concepts write programs for finding the element in the array using Binary Search Method

PROGRAM USING RECURSION

```
#include <stdio.h>
binarysearch(int a[],int n,int low,int high)
{
    int mid;
    if (low > high)
        return -1;
    mid = (low + high)/2;
    if(n == a[mid])
    {
        printf("The element is at position %d\n",mid+1);
        return 0;
    }
    if(n < a[mid])
    {
        high = mid - 1;
        binarysearch(a,n,low,high);
    }
    if(n > a[mid])
    {
        low = mid + 1;
        binarysearch(a,n,low,high);
    }
}
main()
{
    int a[50];
    int n,no,x,result;
    printf("Enter the number of terms : ");
    scanf("%d",&no);
    printf("Enter the elements :\n");
    for(x=0;x<no;x++)
        scanf("%d",&a[x]);
    printf("Enter the number to be searched : ");
    scanf("%d",&n);
```

```
result = binarysearch(a,n,0,no-1);
if(result == -1)
printf("Element not found");
return 0;
}
```

PROGRAM USING ITERATION

```
#include<stdio.h>
#include<conio.h>
int nr_bin_search(int[],int,int);
void main()
{
int key,i,n,index,l[20];
printf("\n enter the number of elements in the list:");
scanf("%d",n);
printf("\n enter the elements of the list:");
for(i=0;i<n;i++)
scanf("%d",&l[i]);
printf("\n enter the key element to be searched in the
list:");
scanf("%d",&key);
index=nr_bin_search(l,n,key);
if(index== -1)
printf("\n search completed,element%d found in the list at
position %d",key,index);
getch();
}
int nr_bin_search(ints[],int n,int e)
{
int low_val,mid_val,high_val;
low_val=0;
high_val=0;
while(high_val>=low_val)
{
mid_val=(low_val+high_val)/2;
if(s[mid_val]==e)
return(mid_val);
if(s[mid_val]<e)
low_val=mid_val+1;
else
high_val=mid_val-1;
}
return-1;
}
```

QUIZ

Q1. The complexity of searching an element from a set of n elements using Binary search algorithm is

- (A) $O(n)$ (B) $O(\log n)$
(C) $O(n^2)$ (D) $O(n \log n)$

Ans: B

Q2. In binary search, average number of comparison required for searching an element in a list if n numbers is

- (A) $\log_2 n$. (B) $n / 2$.
(C) n . (D) $n - 1$.

Ans. (A)

Q3 Which of the following is not the required condition for binary search algorithm?

- a. The list must be sorted
- b. there should be the direct access to the middle element in any sublist
- c. There must be mechanism to delete and/or insert elements in list
- d. none of above

Ans: There must be mechanism to delete and/or insert elements in list

Q4 Which of the following is not a limitation of binary search algorithm?

- a. must use a sorted array
- b. requirement of sorted array is expensive when a lot of insertion and deletions are needed
- c. there must be a mechanism to access middle element directly
- d. binary search algorithm is not efficient when the data elements are more than 1000.

Ans: requirement of sorted array is expensive when a lot of insertion and deletions are needed

Q5. What is binary search technique?

Ans: In computer science, a binary search or half-interval search algorithm finds the position of a specified value (the input "key") within a sorted array.

Q6. What is the maximum number of leaves in a binary tree of height H ?

Ans: 2^H

Q7. What is the difference between recursion and iteration?

Ans: During recursion, a global variable could be used to control the depth of recursion, or data pushed onto some stack could provide a termination condition; and in the latter case this termination condition is usually related to the form of the data structure being traversed.

Q8. What is the main difference between linear and binary search?

Ans: A linear search looks down a list, one item at a time, without jumping. In complexity terms this is an $O(n)$ search - the time taken to search the list gets bigger at the same rate as the list does. A binary search is when you start with the middle of a sorted list, and see whether that's greater than or less than the value you're looking for, which determines whether the value is in the first or second half of the list.

Q9. Give a real world example of binary search technique.

Ans: Number guessing game, word lists.

Q10. What is noisy binary search?

Ans: binary search solves the same class of projects as regular binary search, with the added complexity that any given test can return a false value at random.

EXPERIMENT NO.3

AIM: Write a program to perform following operations on tables using functions only

a) Addition b) Subtraction c) Multiplication d) Transpose

PROGRAM

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
    int a[4][4],b[4][4],c[4][4],i,j;
    printf("enter the elements of matrix a");
    for(i=0;i<=3;i++)
    for(j=0;j<=3;j++)
    scanf("%d",&a[i][j]);
    printf("the first matrix is");
    for(i=0;i<=3;i++)
    {
        printf("\n");
        for(j=0;j<=3;j++)
        printf("%d",a[i][j]);
    }
    printf("Enter the elements of second matrix");
    for(i=0;i<=3;i++)
    for(j=0;j<=3;j++)
    scanf("%d",&b[i][j]);
    printf("the second matrix is");
    for(i=0;i<=3;i++)
    {
        printf("\n");
        for(j=0;j<=3;j++)
        printf("%d",b[i][j]);
    }
    for(i=0;i<=3;i++)
    for(j=0;j<=3;j++)
    c[i][j]=a[i][j] + b[i][j];
    printf("the addition of matrix is");
    for(i=0;i<=3;i++)
    {
        for(j=0;j<=3;j++)
        printf("%d\t",c[i][j]);
    }
```

```
printf("\n");
}
printf("\nSubtraction of matrix is\n");
for(i=0;i<r1;i++)
{
for(j=0;j<c2;j++)
{
printf("%d\t",c[i][j]);
}
printf("\n");
}
}
else
printf("\nSubraction is not possible pls enter same order");
}
Printf("Multiplication is");
for(i=0;i<=3;i++)
{
for(j=0;j<=3;j++)
{
for(k=0;k<=3;k++)
{
c[i][j]=c[i][j]+a[i][k]*b[k][j];
}
}
}
printf("multiplication matrix is");
for(i=0;i<=5;i++)
for(j=0;j<=5;j++)
printf("%d",c[i][j]);
}
printf("Transpose of the Matrix :\n\n");
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
{
t=m[i][j];
m[i][j]=m[j][i];
m[j][i]=t;
}
}
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
printf("\t%d",m[i][j]);
}
```

```
}  
printf("\n");  
}  
getch();  
}
```

QUIZ

Q1. What is a matrix?

Ans: A matrix is a representation of certain rows and columns, to persist homogeneous data.

Q2. What are the uses of matrix?

Ans: Uses:

- To represent class hierarchy using Boolean square matrix
- For data encryption and decryption
- To represent traffic flow and plumbing in a network
- To implement graph theory of node representation

Q3. How we declare two-dimensional array?

Ans: `int a[2][2];`

Q4. What is sparse matrix?

Ans: A sparse matrix is a matrix populated primarily with zeros.

Q5. What is transpose of a matrix?

Ans: The transpose of a matrix A is another matrix A^T (also written A' , A_{tr} or A_t) created by any one of the following equivalent actions:

- reflect A over its main diagonal (which runs top-left to bottom-right) to obtain A^T
- write the rows of A as the columns of A^T
- write the columns of A as the rows of A^T

Q6 Define rank of matrix.

Ans: The rank of a matrix A is the maximum number of linearly independent row vectors of the matrix, which is the same as the maximum number of linearly independent column vectors.

Q7. What is invertible matrix?

Ans: A square matrix A is called invertible or non-singular if there exists a matrix B such that
$$AB = I_n$$

Q8. What is diagonal matrix?

Ans: If all entries outside the main diagonal are zero, A is called a diagonal matrix.

Q9. What is symmetric matrix?

Ans: A square matrix A that is equal to its transpose, i.e., $A = A^T$, is a symmetric matrix.

Q10. Write a real world application of matrix.

Ans: Game theory.

EXPERIMENT NO.4

AIM: Using iteration & recursion concepts write the programs for Quick Sort Technique

PROGRAM

```
#include<stdio.h>
#include<conio.h>
#define max 100
int a[max],n,i,h,l;
void main()
{
void input(void);
input();
getch();
}
void input(void)
{
void quicksort(int a[],int l,int h);
void output(int a[],int n);
printf("how many:");
scanf("%d",&n);
printf("\n");
printf("Enter the elements:\n");
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
l=0;
h=n-1;
quicksort(a,l,h);
printf("sorted array:\n");
output(a,n);
}
void quicksort(int a[],int l,int h)
{
int temp,key,low,high;

low=l;
high=h;
key=a[(low+high)/2];
do{
while(key>a[low])
```

```
low++;
}
while(key<a[high])
{
high--;
}
if(low<=high)
{
temp=a[low];
a[low++]=a[high];
a[high--]=temp;
}
}
while(low<=high);
if(l<high)
quicksort(a,l,high);
if(low<h)
quicksort(a,low,h);
}
void output(int a[],int n)
{
for(i=0;i<n;i++)
{
printf("%d\n",a[i]);
}
}
```

QUIZ

Q1. What would be the order of Quick Sort in worst case?

Ans: $O(n^2/2)$

Q2. The quick sort algorithm exploits _____ design technique

- (A) Greedy (B) Dynamic programming
(C) Divide and Conquer (D) Backtracking

Ans: C

Q3. Quick sort is also known as

- (A) merge sort (B) heap sort
(C) bubble sort (D) none of these

Ans: D

Q4. What is quick sort?

Ans: Quick sort is one of the fastest sorting algorithm used for sorting a list. A pivot point is chosen. Remaining elements are portioned or divided such that elements less than the pivot point are on the left and those greater than the pivot are on the right. Now, the elements on the left and right can be recursively sorted by repeating the algorithm.

Q5. What is divide and conquer algorithm?

Ans: Algorithms that solve (conquer) problems by dividing them into smaller sub-problems until the problem is so small that it is trivially solved.

Q6. What is in place algorithm?

Ans: In place sorting algorithms don't require additional temporary space to store elements as they sort; they use the space originally occupied by the elements.

Q7. Define sorting algorithm.

Ans: A sorting algorithm is an algorithm that puts elements of a list in a certain order.

Q8. What are the other techniques for sorting?

Ans: Bubble sort, insertion sort, selection sort, merge sort, radix sort, heap sort.

Q9. What is the possible way to classify sorting algorithms? _____

Ans: Complexity

Q10. What is swap based sorting?

Ans: Swap-based sorts begin conceptually with the entire list, and exchange particular pairs of elements (adjacent elements or elements with certain step like in Shell sorts) moving toward a more sorted list.

EXPERIMENT NO.5

AIM: Write a program to implement the various operations on string such as length of string concatenation, reverse of a string & copy of a string to another.

PROGRAM

```
#include<conio.h>
#include<stdio.h>
void main()
{
    char a[20], b[20];
    int i;
    clrscr();
    gets(a);
    i=0;
    while(a[i]!='\0')
        i++; //counts no of chars till encountering null char
    printf("string length=%d,i);
    printf("Before concatenation:"
    "\n string1 = %s \n string2 = %s", string1, string2);
    finalstr = strcat(string1, string2);
    printf("\nAfter concatenation:");
    printf("\n finalstr = %s", finalstr);
    printf("\n string1 = %s", string1);
    printf("\n string2 = %s", string2);

    printf("Enter the string to be reversed : ");
    scanf("%s",str);
    for(i=strlen(str)-1;i>=0;i--)
    {
        revstr[j]=str[i];
        j++;
    }
    revstr[j]='\0';
    printf("Input String : %s",str);
    printf("\nOutput String : %s",revstr);
    CHAR szTest[32] = {"The Quick Brown Fox"};
    CHAR szCopy[32] = {0};
    size_t thelen = strlen(szTest);
    int i = 0;
    for(;i<(int)thelen;i++)
    {
        szCopy[i] = szTest[i];
        getch();
    }
}
```

QUIZ

Q1. Define string.

Ans: A string is traditionally a sequence of characters, either as a literal constant or as some kind of variable.

Q2. What is literal constant?

Ans: A literal is a notation for representing a fixed value in source code.

Q3. When a string is said to be prefix?

Ans: A string s is said to be a prefix of t if there exists a string u such that $t = su$. If u is nonempty, s is said to be a proper prefix of t .

Q4. When a string is said to be suffix?

Ans: A string s is said to be a suffix of t if there exists a string u such that $t = us$. If u is nonempty, s is said to be a proper suffix of t .

Q5. Which function we use to calculate length of string?

Ans: `strlen(s)`

Q6. Which function we use for concatenation of strings?

Ans: `strcat(s1,s2)`

Q7. What is string datatype?

Ans: A string datatype is a datatype modeled on the idea of a formal string. Strings are such an important and useful datatype that they are implemented in nearly every programming language. In some languages they are available as primitive types and in others as composite types.

Q8. Which function we use for comparison of two strings?

Ans: `strcmp(s1,s2)`.

Q9. How a string is declared?

Ans: `char string[] = "Hello, world!";`

Q10. How we mark the end of a string?

Ans: The end of the string is marked with a special character, the null character, which is simply the character with the value 0.

EXPERIMENT NO.6

AIM: Write a program for swapping of two numbers using 'call by value' and 'call by reference' strategies.

Call By Value

```
#include<stdio.h>
#include<conio.h>

main()
{
    int x, y, temp;
    printf("Enter the value of x and y ");
    scanf("%d%d",&x, &y);
    printf("Before Swapping\nx = %d\ny = %d\n",x,y);
    temp = x;
    x = y;
    y = temp;
    printf("After Swapping\nx = %d\ny = %d\n",x,y);
    getch();
    return 0;
}
```

Call By Reference

```
#include<stdio.h>
#include<conio.h>
main()
{
    int i, j;
    clrscr();
    printf("Please Enter the First Number in A : ");
    scanf("%d",&i);
    printf("\nPlease Enter the Second Number in B : ");
    scanf("%d",&j);
    swapr(&i,&j);
    printf("A is now in B : %d",i);
    printf("B is now in A : %d",j);
}

swapr(int *x, int *y)
{
    int t;
    t=*x;
    *x=*y;
    *y=t; }

```

QUIZ

Q1. What is function?

Ans: A function definition specifies the name of the function, the types and number of parameters it expects to receive, and its return type. A function definition also includes a function body with the declarations of its local variables, and the statements that determine what the function does.

Q2. Difference between Call by value and call by reference.

Ans: The arguments passed to function can be of two types

1. Values passed

2. Address passed

The first type refers to call by value and the second type refers to call by reference.

Q3. What is function prototype?

Ans: A function prototype tells the compiler what kind of arguments a function is looking to receive and what kind of return value a function is going to give back. This approach helps the compiler ensure that calls to a function are made correctly and that no erroneous type conversions are taking place.

Q4. When should we declare a function?

Ans: Function declaration should be declared in the current source file along with the definition of the function.

Q5. What is the scope of function variable?

Ans: Variables declared within the calling function can't be accessed unless they are passed to the called function as arguments. The only other contact a function might have with the outside world is through global variables.

Q6. Define recursive function.

Ans: A recursive function is one which calls itself.

Q7. What is the use of making a function inline?

Ans: The point of making a function `inline` is to hint to the compiler that it is worth making some form of extra effort to call the function faster than it would otherwise - generally by substituting the code of the function into its caller.

Q8. What is the return type of the function with prototype: "int func(char x, float v, double t);"

Ans: int

Q9. Which of the following is a valid function call (assuming the function exists)?

A. func;

B. func x, y;

C. func();

D. int func();

Ans: C

Q10. Which is not a proper prototype?

A. int func(char x, char y);

B. double func(char x)

C. void func();

D. char x();

Ans: B

EXPERIMENT NO.7

AIM: Write a program to implement binary search tree.
(Insertion and Deletion in Binary search Tree)

```
# include<stdio.h>
# include<conio.h>
struct node
{
    int info;
    struct node *lchild;
    struct node *rchild;
}*root;
main()
{
    int choice,num;
    root=NULL;
    while(1)
    {
        printf("\n");
        printf("1.Insert\n");
        printf("2.Delete\n");
        printf("3.Display\n");
        printf("4.Quit\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
                printf("Enter the number to be inserted : ");
                scanf("%d",&num);
                insert(num);
                break;
            case 2:
                printf("Enter the number to be deleted : ");
                scanf("%d",&num);
                del(num);
                break;
            case 3:
                display(root,1);
                break;
            case 4:
                exit();
            default:
```

```
printf("Wrong choice\n");
}
}
}
find(int item,struct node **par,struct node **loc)
{
    struct node *ptr,*ptrsave;
    if(root==NULL) /*tree empty*/
    {
        *loc=NULL;
        *par=NULL;
        return;
    }
    if(item==root->info) /*item is at root*/
    {
        *loc=root;
        *par=NULL;
        return;
    }
    /*Initialize ptr and ptrsave*/
    if(item<root->info)
        ptr=root->lchild;
    else
        ptr=root->rchild;
    ptrsave=root;
    while(ptr!=NULL)
    {
        if(item==ptr->info)
        { *loc=ptr;
          *par=ptrsave;
          return;
        }
        ptrsave=ptr;
        if(item<ptr->info)
            ptr=ptr->lchild;
        else
            ptr=ptr->rchild;
    } /*End of while */
    *loc=NULL; /*item not found*/
    *par=ptrsave;
} /*End of find()*/
```

```
insert(int item)
{
```

```
struct node *tmp,*parent,*location;  
find(item,&parent,&location);  
if(location!=NULL)  
{  
    printf("Item already present");  
    return;  
}  
tmp=(struct node *)malloc(sizeof(struct node));  
tmp->info=item;  
tmp->lchild=NULL;  
tmp->rchild=NULL;
```

```
if(parent==NULL)  
    root=tmp;  
else  
    if(iteminfo)  
        parent->lchild=tmp;  
    else  
        parent->rchild=tmp;  
}/*End of insert()*/
```

```
del(int item)  
{  
    struct node *parent,*location;  
    if(root==NULL)  
    {  
        printf("Tree empty");  
        return;  
    }  
}
```

```
find(item,&parent,&location);  
if(location==NULL)  
{  
    printf("Item not present in tree");  
    return;  
}
```

```
if(location->lchild==NULL && location->rchild==NULL)  
    case_a(parent,location);  
if(location->lchild!=NULL && location->rchild==NULL)  
    case_b(parent,location);  
if(location->lchild==NULL && location->rchild!=NULL)  
    case_b(parent,location);  
if(location->lchild!=NULL && location->rchild!=NULL)  
    case_c(parent,location);
```

```
free(location);
}/*End of del()*/
```

```
case_a(struct node *par,struct node *loc )
{
if(par==NULL) /*item to be deleted is root node*/
root=NULL;
else
if(loc==par->lchild)
par->lchild=NULL;
else
par->rchild=NULL;
}/*End of case_a()*/
```

```
case_b(struct node *par,struct node *loc)
{
struct node *child;

/*Initialize child*/
if(loc->lchild!=NULL) /*item to be deleted has lchild */
child=loc->lchild;
else /*item to be deleted has rchild */
child=loc->rchild;
```

```
if(par==NULL ) /*Item to be deleted is root node*/
root=child;
else
if( loc==par->lchild) /*item is lchild of its parent*/
par->lchild=child;
else /*item is rchild of its parent*/
par->rchild=child;
}/*End of case_b()*/
```

```
case_c(struct node *par,struct node *loc)
{
struct node *ptr,*ptrsave,*suc,*parsuc;
```

```
/*Find inorder successor and its parent*/
ptrsave=loc;
ptr=loc->rchild;
while(ptr->lchild!=NULL)
{
ptrsave=ptr;
ptr=ptr->lchild;
}
```

```
suc=ptr;
parsuc=ptrsave;

if(suc->lchild==NULL && suc->rchild==NULL)
case_a(parsuc,suc);
else
case_b(parsuc,suc);

if(par==NULL) /*if item to be deleted is root node */
root=suc;
else
if(loc==par->lchild)
par->lchild=suc;
else
par->rchild=suc;

suc->lchild=loc->lchild;
suc->rchild=loc->rchild;
}/*End of case_c()*/

display(struct node *ptr,int level)
{
int i;
if ( ptr!=NULL )
{
display(ptr->rchild, level+1);
printf("\n");
for (i = 0; i < level; i++)
printf(" ");
printf("%d", ptr->info);
display(ptr->lchild, level+1);
}/*End of if*/
}/*End of display()*/
```

QUIZ

Q1. What is the number of possible ordered trees with three nodes?

Ans: 12

Q2. What is strictly binary tree?

Ans: A binary tree in which every non leaf node has non empty left and right subtrees is called a strictly binary tree.

Q3. What is the depth of a complete binary tree with n nodes?

Ans: $\log(n+1)-1$

Q4. which kind of traversal is similar to preorder?

Ans: Depth-first order

Q5. Which traversal technique lists the nodes of a binary search tree in ascending order?

Ans: In-order

Q6. What is the order of binary search algorithm?

Ans: $\log(n)$

Q7. A binary tree has n leaf nodes. What would be the number of nodes of degree 2 in this tree?

Ans: n-1

Q8. What is the number of binary trees with 3 nodes which when traversed in post order?

Ans: 5

Q9. What is the infix priorities of +, *, ^, /

Ans: 5,2,2,4

Q10. What is 3-ary tree?

Ans: It is a tree in which every internal node has exactly 3 children.

EXPERIMENT NO.8

AIM: Write a program to create a linked list & perform operations such as insert, delete, update, reverse in the link list

```
#include "stdio.h"
#define NULL 0
struct node
{
    int data;
    struct node *next;
}*p;

delnode(int num)
{
    struct node *temp, *m;
    temp=p;
    while(temp!=NULL)
    {
        if(temp->data==num)
        {
            if(temp==p)
            {
                p=temp->next;
                free(temp);
                return;
            }
            else
            {
                m->next=temp->next;
                free(temp);
                return;
            }
        }
        else
        {
            m=temp;
            temp= temp->next;
        }
    }

    printf("
ELEMENT %d NOT FOUND
", num);
```

```
append( int num )
{
    struct node *temp,*r;
    temp= (struct node *)malloc(sizeof(struct node));
    temp->data=num;
    r=(struct node *)p;

    if (p == NULL)
    {
        p=temp;
        p->next =NULL;
    }
    else
    { /* GO TO LAST AND ADD*/

        while( r->next != NULL)
            r=r->next;
        r->next =temp;
        r=temp;
        r->next=NULL;
    }
    /* ADD A NEW NODE AT BEGINNING */

    addbeg( int num )
    {
        /* CREATING A NODE AND INSERTING VALUE TO IT */

        struct node *temp;
        temp=(struct node *)malloc(sizeof(struct node));
        temp->data=num;

        /* IF LIST IS NULL ADD AT BEGINNING */
        if ( p== NULL)
        {
            p=temp;
            p->next=NULL;
        }

        else
        {
            temp->next=p;
            p=temp;
        }
    }
}
```

```
/* ADD A NEW NODE AFTER A SPECIFIED NO OF NODES */
```

```
addafter(int num, int loc)
{
    int i;
    struct node *temp, *t, *r;
    r=p; /* here r stores the first location */
    if(loc > count()+1 || loc <= 0)
    {
        printf("
insertion is not possible :
");
        return;
    }
    if (loc == 1)/* if list is null then add at beginning */
    {
        addbeg(num);
        return;
    }
    else
    {
        for(i=1;i<loc;i++)
        {
            t=r; /* t will be holding previous value */
            r=r->next;
        }
        temp=(struct node *)malloc(sizeof(struct node));
        temp->data=num;
        t->next=temp;
        t=temp;
        t->next=r;
        return;
    }
}/* THIS FUNCTION DISPLAYS THE CONTENTS OF THE LINKED LIST */
```

```
display(struct node *r)
{
    r=p;
    if(r==NULL)
    {
        printf("NO ELEMENT IN THE LIST :");
        return;
    }
    /* traverse the entire linked list */
    while(r!=NULL)
```

```
{  
printf(" -> %d ",r->data);  
r=r->next;  
}  
printf("<BR>");  
}
```

//THIS FUNCTION REVERSES A LINKED LIST

```
reverse(struct node *q)  
{  
struct node *m, *n,*l,*s;  
m=q;  
n=NULL;  
while(m!=NULL)  
{  
s=n;  
n=m;  
m=m->next;  
n->next=s;  
}  
p=n;  
}
```

/* THIS IS THE MAIN PROGRAM */

```
main()  
{  
int i;  
p=NULL;  
while(1) /* this is an indefinite loop */  
{  
printf("  
1.INSERT A NUMBER AT BEGINNING:<BR>");  
printf("  
2.INSERT A NUMBER AT LAST:<BR>");  
printf("  
3.INSERT A NUMBER AT A PARTICULAR LOCATION INIIST:<BR>");  
printf("  
4.PRINT THE ELEMENTS IN THE LIST :<BR>");  
printf("  
5.PRINT THE NUMBER OF ELEMENTS IN THE LIST <BR>");  
printf("  
6.DELETE A NODE IN THE LINKED LIST:<BR>");  
printf("  
7.REVERSE A LINKED LIST :<BR>");  
}
```

```

printf("
8.GET OUT OF LINKED LIST (BYEE BYEE);
printf("
PLEASE, ENTER THE NUMBER:");

scanf("%d",&i); /* ENTER A VALUE FOR SWITCH */

switch(i)
{
case 1:
{
int num;
printf("PLEASE ENTER THE NUMBER :-");
scanf("%d",&num);
addbeg(num);
break;
}
case 2:
{
int num;
printf("
PLEASE ENTER THE NUMBER :-");
scanf("%d",&num);
append(num);
break;
}

case 3:
{
int num, loc,k;
printf("
PLEASE ENTER THE NUMBER :-");
scanf("%d",&num);
printf("
PLEASE ENTER THE LOCATION NUMBER :-");
scanf("%d",&loc);
addafter(num,loc);
break;
} case 4:
{
struct node *n;
printf("

THE ELEMENTS IN THE LIST ARE : <BR>);
display(n);

```

```
break;
}
case 5:
{
struct node *n;
display(n);
printf(" TOTAL NO OF ELEMENTS IN THE LSIT ARE %d",count());
break;
} case 6:
{
int num;
printf("PLEASE ENTER A NUMBER FROM THE LIST :");
scanf("%d",&num);
delnode(num);
break;
}
case 7:
{
reverse(p);
display(p);
break;
}
case 8:
{
exit();
}
}/* end if switch */
}/* end of while */
}/* end of main *
```

QUIZ

Q1. In a linked list with n nodes, the time taken to insert an element after an element pointed by some pointer is

- (A) $O(1)$ (B) $O(\log n)$
(C) $O(n)$ (D) $O(n \log n)$

Ans:A

Q2. Consider a linked list of n elements. What is the time taken to insert an element after an element pointed by some pointer?

- (A) $O(1)$ (B) $O(\log^2 n)$
(C) $O(n)$ (D) $O(n \log^2 n)$

Ans:A

Q3. Clarify whether Linked List is linear or Non-linear data structure ?

Answer:Link list is always linear data structure because every element (NODE) having unique position and also every element has its unique successor and predecessor. Also,linear collection of data items called nodes and the linear order is given by means of pointers. Each node is divided into two parts. First part contains information of the element and another part contains the address of the next node in the list.

Q4. Explain the types of linked list.

Ans: The types of linked lists are:

Singly linked list: It has only head part and corresponding references to the next nodes.

Doubly linked list: A linked list which both head and tail parts, thus allowing the traversal in bi-directional fashion. Except the first node, the head node refers to the previous node.

Circular linked list: A linked list whose last node has reference to the first node.

Q5. How would you sort a linked list?

Ans: Step 1: Compare the current node in the unsorted list with every element in the rest of the list.

If the current element is more than any other element go to step 2 otherwise go to step 3.

Step 2: Position the element with higher value after the position of the current element. Compare the next element. Go to step1 if an element exists, else stop the process.

Step 3: If the list is already in sorted order, insert the current node at the end of the list. Compare the next element, if any and go to step 1 or quit.

Q6. In general, linked lists allow:

- Insertions and removals anywhere.
- Insertions and removals only at one end.
- Insertions at the back and removals from the front.
- None of the above.

ANS a. Insertions and removals anywhere.

Q7. What kind of linked list begins with a pointer to the first node, and each node contains a pointer to the next node, and the pointer in the last node points back to the first node?

- Circular, singly-linked list.
 - Circular, doubly-linked list.
 - Singly-linked list.
 - Doubly-linked list.
-

ANS a. Circular, singly-linked list.

Q8. How many pointers are contained as data members in the nodes of a circular, doubly linked list of integers with five nodes?

Ans: 10

Q9. Which part in a linked list index represents the position of a node in a linked list?

Ans: An integer

Q10. What is the value of first linked list index?

Ans: Zero

EXPERIMENT NO.9

AIM: Write the program for implementation of a file and performing operations such as insert, delete, update a record in the file.

```
#include<stdio.h>
#include<conio.h>
void append();
void list();
void search();
void modify();
void del();
struct employee
{
int no, sal;
char gen, name[20];
};
void main()
{ int a;
char ch;
do{
printf("\nEMPLOYEE DATABASE\n\n");
printf("1.Append Employee Record\n2.List Employee Record\n3.Modify Employee
Record\n4.Delete Employee Record\n5.Search Employee Record\n Enter Choice : ");
scanf("%d",&a);
switch(a)
{
case 1:
append();
break;
case 2:
list();
break;
case 3:
modify();
break;
case 4:
del();
```

```
break;
case 5:
search();
break;
default :
printf("Invalid Choice!");
}
printf("\n More Actions ? (Y/N) :");
fflush(stdin);
scanf("%c", &ch);
}while(ch=='y' || ch=='Y');
}
void append()
{ int i,n;
struct employee e;
FILE *fp;
fp=fopen("Employee.dat", "a");
if(fp==NULL)
{
printf("File Creation Failed!");
exit(0);
}
printf("Enter the nos. of employees : ");
scanf("%d", &n);
for(i=0;i<n;i++)
{
printf("Enter the Employee Number : ");
scanf("%d", &e.no);
printf("Enter the Employee Salary : ");
scanf("%d", &e.sal);
printf("Enter the Employee gender: ");
fflush(stdin);
scanf("%c", &e.gen);
printf("Enter the Employee Name : ");
fflush(stdin);
gets(e.name);
printf("\n\n");</n;i++}
```

```
fwrite((char *)&e, sizeof(e), 1, fp);
}
fclose(fp);
}
void list()
{ int nofrec=0;
  struct employee e;
  FILE *fp;
  fp=fopen("Employee.dat", "rb");
  if(fp==NULL)
  {
    printf("\n\tFile doesn't exist!!!\TRY AGAIN");
    exit(0);
  }
  while((fread((char *)&e, sizeof(e), 1, fp))==1)
  { nofrec++;
    printf("\nEmployee Number : %d", e.no);
    printf("\nEmployee Salary : %d", e.sal);
    printf("\nEmployee gender : %c", e.gen);
    printf("\nEmployee Name : %s", e.name);
    printf("\n\n");
  }
  printf("Total number of records present are : %d", nofrec);
  fclose(fp);}
```

```
void modify()
{ int recno, nofrec=0;
  char ch;
```

```
  struct employee e;
  FILE *fp;
  fp=fopen("Employee.dat", "rb+");
  printf("Enter the Employee Number to modify : ");
  scanf("%d", &recno);
  while((fread((char *)&e, sizeof(e), 1, fp))==1)
  { nofrec++;
    if(e.no==recno)
```

```
{
printf("\nEmployee Number : %d", e.no);
printf("\nEmployee Salary : %d", e.sal);
printf("\nEmployee gender : %c",e.gen);
printf("\nEmployee Name : %s",e.name);
printf("\n");
printf("Do you want to modify this record : ? (Y/N)");
fflush(stdin);
scanf("%c", &ch);
fseek(fp, ((nofrec-1)*sizeof(e)), 0);
if(ch=='Y' || ch=='y')
{
printf("Enter the Employee Salary : ");
scanf("%d", &e.sal);
printf("Enter the Employee gender: ");
fflush(stdin);
scanf("%c", &e.gen);
printf("Enter the Employee Name : ");
fflush(stdin);
gets(e.name);
fwrite((char *)&e, sizeof(e), 1, fp);
printf("Record Modified");
}
else
printf("No modifications were made");
fclose(fp);

}
}
}
void del()
{
int recno;
char ch;
struct employee e;
FILE *fp, *ft;
```

```
fp=fopen("Employee.dat", "rb");
ft=fopen("Temp.dat", "wb");
printf("Enter the Employee Number to delete : ");
scanf("%d", &recno);
while((fread((char *)&e, sizeof(e), 1, fp))==1)
{
if(e.no==recno)
{
printf("\nEmployee Number : %d", e.no);
printf("\nEmployee Salary : %d", e.sal);
printf("\nEmployee gender : %c",e.gen);
printf("\nEmployee Name : %s",e.name);
printf("\n");
printf("Do you want to delete this record : ? (Y/N)");
fflush(stdin);
scanf("%c", &ch);
}
}
if(ch=='y' || ch=='Y')
{
rewind(fp);
while((fread((char *)&e, sizeof(e), 1, fp))==1)
{
if(recno!=e.no)
{
fwrite((char *)&e, sizeof(e), 1, ft);
}
}
}
else
printf("No Record was deleted");
fclose(fp);
fclose(ft);
remove("Employee.dat");
rename("Temp.dat", "Employee.dat");
}
void search()
```

```
{ int s,recno;
char sname[20];
struct employee e;
FILE *fp;
fp=fopen("Employee.dat", "rb");
printf("\n1.Search by Name\n2.Search by Employee No.\n Enter choice : ");
scanf("%d", &s);
switch(s)
{
case 1:
printf("Enter the Employee Name to Search : ");
fflush(stdin);
gets(sname);
while((fread((char *)&e, sizeof(e), 1, fp))==1)
{
if(strcmp(sname,e.name)==0)
{
printf("\nEmployee Number : %d", e.no);
printf("\nEmployee Salary : %d", e.sal);
printf("\nEmployee gender : %c",e.gen);
printf("\nEmployee Name : %s",e.name);
printf("\n");
}
}
break;
case 2:printf("Enter the Employee Number to Search : ");
scanf("%d", &recno);
while((fread((char *)&e, sizeof(e), 1, fp))==1)
{
if(e.no==recno)
{
printf("\nEmployee Number : %d", e.no);
printf("\nEmployee Salary : %d", e.sal);
printf("\nEmployee gender : %c",e.gen);
printf("\nEmployee Name : %s",e.name);
printf("\n");
}
```

```
}  
}  
break;  
}  
}
```

QUIZ

Q1. What is File Handling?

We frequently use files for storing information which can be processed by our programs. In order to store information permanently and retrieve it we need to use files. A file is collection of related data. Placed on the disk. Files are not only used for data. Our programs are also stored in files.

Q2. Why we use file Handling:

The input and out operation that we have performed so far were done through screen and keyboard only. After the termination of program all the entered data is lost because primary memory is volatile. If the data has to be used later, then it becomes necessary to keep it in permanent storage device. So the C language provides the concept of file through which data can be stored on the disk or secondary storage device. The stored data can be read whenever required.

Q3. Explain Types of File Handling in C:

The file handling in C can be categorized in two types-

1. High level (Standard files or stream oriented files)- High level file handling is managed by library function. High level file handling is commonly used because it is easier and hides most of the details from the programmer.

2. Low level (system oriented files)- low level file handling is managed by system calls.

Q4. A random access file is organized most like a(n):

- a. Array.
- b. Object.
- c. Class.
- d. Pointer.

ANS: a. Array.

Q5. What are the typical operations on file?

Ans: Open, Read, Write and close

Q6. How is a file stored?

Ans: Stored as sequence of bytes, logically contiguous (may not be physically contiguous on disk).

Q7. In C, what we use to set a pointer to a file?

Ans: File *

Q8. Which command is used to open a file?

Ans: fopen() and it returns null if unable to open a file.

Q9. What are the modes used in fopen()?

Ans: read, write and append

Q10. What is fprintf() function?

Ans: It works like printf() and sprintf() except that its first argument is a file pointer.

EXPERIMENT NO.10

AIM: Create a linked list and perform the following operations on it

a) add a node b) Delete a node

```
#include<stdio.h>
#include<stdlib.h>

struct node
{
    int data;
    struct node *next;
}*head;

void append(int num)
{
    struct node *temp,*right;
    temp= (struct node *)malloc(sizeof(struct node));
    temp->data=num;
    right=(struct node *)head;
    while(right->next != NULL)
        right=right->next;
    right->next =temp;
    right=temp;
    right->next=NULL;
}

void add( int num )
{
    struct node *temp;
    temp=(struct node *)malloc(sizeof(struct node));
    temp->data=num;
    if (head== NULL)
    {
        head=temp;
        head->next=NULL;
    }
    else
    {
        temp->next=head;
        head=temp;
    }
}
```

```
void addafter(int num, int loc)
{
    int i;
    struct node *temp,*left,*right;
    right=head;
    for(i=1;i<loc;i++)
    {
        left=right;
        right=right->next;
    }
    temp=(struct node *)malloc(sizeof(struct node));
    temp->data=num;
    left->next=temp;
    left=temp;
    left->next=right;
    return;
}
```

```
void insert(int num)
{
    int c=0;
    struct node *temp;
    temp=head;
    if(temp==NULL)
    {
        add(num);
    }
    else
    {
        while(temp!=NULL)
        {
            if(temp->data<num)
                c++;
            temp=temp->next;
        }
        if(c==0)
            add(num);
        else if(c<count())
            addafter(num,++c);
        else
            append(num);
    }
}
```

```
int delete(int num)
{
    struct node *temp, *prev;
    temp=head;
    while(temp!=NULL)
    {
        if(temp->data==num)
        {
            if(temp==head)
            {
                head=temp->next;
                free(temp);
                return 1;
            }
            else
            {
                prev->next=temp->next;
                free(temp);
                return 1;
            }
        }
        else
        {
            prev=temp;
            temp= temp->next;
        }
    }
    return 0;
}
```

```
void display(struct node *r)
{
    r=head;
    if(r==NULL)
    {
        return;
    }
    while(r!=NULL)
    {
        printf("%d ",r->data);
        r=r->next;
    }
    printf("\n");
}
```

[illegible]

```
        }
        display(n);
        break;
    case 3: printf("Size of the list is %d\n",count());
        break;
    case 4: if(head==NULL)
        printf("List is Empty\n");
        else{
            printf("Enter the number to delete : ");
            scanf("%d",&num);
            if(delete(num))
                printf("%d deleted successfully\n",num);
            else
                printf("%d not found in the list\n",num);
        }
        break;
    case 5: return 0;
    default: printf("Invalid option\n");
}

}
}
return 0;
}
```

QUIZ

Q1. Consider a linked list of n elements. What is the time taken to insert an element after an element pointed by some pointer?

- (A) $O(1)$ (B) $O(\log_2 n)$
(C) $O(n)$ (D) $O(n \log_2 n)$

Ans: A

Q2. In a linked list with n nodes, the time taken to insert an element after an element pointed by some pointer is

- (A) $O(1)$ (B) $O(\log n)$
(C) $O(n)$ (D) $O(n \log n)$

Ans: A

Q3. Why linked list implementation of sparse matrices is superior to the generalized sparse vector?

Ans: Because linked list implementation of sparse matrices is conceptually easier and completely dynamic.

Q4. Which list implementation could be used for concatenation of two lists in $O(1)$ time?

Ans: Circular doubly linked list

Q5. Linked lists are not suitable for implementing:

- (A) insertion sort (B) Binary search
(C) radix sort (D) polynomial manipulation

Ans: B

Q6. How many pointers are contained as data members in the nodes of a circular, doubly linked list of integers with five nodes?

Ans: 10

Q7. Which part in a linked list index represents the position of a node in a linked list?

Ans: An integer

Q8. What is the value of first linked list index?

Ans: Zero

Q9. Write an application of linked list in data structure.

Ans: Memory management

Q10. What are the uses of linked list?

Ans: They can be used to implement several other common abstract data types, including stacks, queues, associative arrays, and symbolic expressions

EXPERIMENT NO.11

AIM: Write a program to simulate the various searching & sorting algorithms and compare their timings for a list of 1000 elements.

SELECTION SORT

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,j,small,pos,temp;
    int size,i;
    int l[10];
    printf("\n enter d size of list");
    scanf("%d",&size);
    printf("\n enter the elements");
    for(i=0;i<size;i++)
    {
        scanf("%d",&l[i]);
    }
    for(i=0;i<size-1;i++)
    {
        small=l[i];
        pos=i;
        for(j=i+1;j<size;j++)
        {
            if(small>l[j])
            {
                small=l[j];
                pos=j;
            }
        }
        temp=l[i];
        l[i]=l[pos];
        l[pos]=temp;
    }
    printf("\n sorted list...");
    for(i=0;i<size;i++)
    {
        printf("%d",l[i]);
    }
    getch();
}
```

BUBBLE SORT

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,j,small,pos,temp;
    int size,i;
    int l[10];
    printf("\n enter d size of list");
    scanf("%d",&size);
    printf("\n enter the elements");
    for(i=0;i<size;i++)
    {
        scanf("%d",&l[i]);
    }
    int f=0;
    int c=0;
    while(f==0)
    {
        f=1;
        c++;
        for(j=0;j<size-1;j++)
        {
            if(l[j]>l[j+1])
            {
                temp=l[j];
                l[j]=l[j+1];
                l[j+1]=temp;
                f=0;
            }
        }
    }
    printf("sorted list is.....");
    for(i=0;i<size;i++)
    {
        printf("%d",l[i]);
        printf("no. of passes made are %d",c);
        getch();
    }
    for(i=0;i<n-1;i++)
    {
        small=l[i];
        pos=i;
        for(j=i+1;j<n;j++)
```

```
{
if(small>l[j])
{
small=l[j];
pos=j;
}
temp=l[i];
l[i]=l[pos];
l[pos]=temp;
}
}
printf("\n sorted list...");
for(i=0;i<n;i++)
{
printf("%d",l[i]);
}
getch();
}
```

INSERTION SORT

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,j,small,pos,temp;
int size,i;
int l[10];
printf("\n enter d size of list");
scanf("%d",&size);
printf("\n enter the elements");
for(i=0;i<size;i++)
{
scanf("%d",&l[i]);
}
for(i=1;i<size;i++)
{
temp=l[i];
j=i-1;
while((temp<l[j]) && (j>=0))
{
l[j+1]=l[j];
j=j-1;
}
l[j+1]=temp;
}
```

```
}
printf("sorted list is.....");
for(i=0;i<size;i++)
{
printf("%d",l[i]);
}
//printf("no. of passes made are %d",c);
getch();
}
```

SHELL SORT

```
#include<stdio.h>
#include<conio.h>
void main()
{
int list[20];
int size,i,j,k,p;
int temp,s,step;
int dimstep[]={5,3,1};
clrscr();
printf("\n enter size of list");
scanf("%d",&size);
printf("\n enter elements");
for (i=0;i<size;i++)
{
scanf("%d",&list[i]);
}
for (step=0;step<3;step++)
{
k=dimstep[step];
s=0;
for(i=s+k;i<size;i=i+k)
{
temp=list[i];
j=i-k;
while((temp<list[j])&&(j>=0))
{
list[j+k]=list[j];
j=j-k;
}
list[j+k]=temp;
s++;
}
}
```

```
}  
printf("\n sorted list is..");  
for(i=0;i<size;i++)  
{  
printf("%d",list[i]);  
}  
getch();  
}
```

QUIZ

Q1. You have to sort a list L consisting of a sorted list followed by a few “random” elements. Which sorting method would be especially suitable for such a task?

Ans: Insertion Sort

Q2. A sort which relatively passes through a list to exchange the first element with any element less than it and then repeats with a new first element is called

(A) insertion sort. (B) selection sort.
(C) heap sort. (D) quick sort.

Ans: D

Q3. Which of the following sorting algorithms does not have a worst case running time of $O(n^2)$ On

(A) Insertion sort (B) Merge sort
(C) Quick sort (D) Bubble sort

Ans: B

Q4. Which sorting algorithm is best if the list is already sorted? Why?

Ans. Insertion sort as there is no movement of data if the list is already sorted and complexity is of the order $O(N)$.

Q5. What are the various kinds of sorting techniques? Which is the best case?

Ans: heap sort is the best sorting technique bcoz its complexity in best case ,worst and avg case is of $O(n \log n)$. In worst case quick sort the complexity in best case and avg case is of $O(n \log n)$ and worst case $O(n^2)$.

Q6. Sorting is useful for:

(A) report generation (B) minimizing the storage needed
(C) making searching easier and efficient (D) responding to queries easily

Ans: D

Q7. Which sorting method is best if the number of swapping done is the only measure of efficiency?

Ans: Selection sort

Q8. Which sorting method is preferred for sorting 15 randomly generated numbers?

Ans: Bubble sort

Q9. What are the maximum numbers of comparisons needed to sort 7 items using radix sort?

Ans: 280

Q10. Which sorting algorithm has the worst time complexity of $n \log(n)$?

Ans: Heap sort

EXPERIMENT NO.12

AIM: Write a program to simulate the various graph traversing algorithms.

```
#include<stdio.h>
#define MAX 20

typedef enum boolean{false,true} bool;
int adj[MAX][MAX];
bool visited[MAX];
int n; /* Denotes number of nodes in the graph */
main()
{
    int i,v,choice;

    create_graph();
    while(1)
    {
        printf("\n");
        printf("1. Adjacency matrix\n");
        printf("2. Depth First Search using stack\n");
        printf("3. Depth First Search through recursion\n");
        printf("4. Breadth First Search\n");
        printf("5. Adjacent vertices\n");
        printf("6. Components\n");
        printf("7. Exit\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);

        switch(choice)
        {
            case 1:
                printf("Adjacency Matrix\n");
                display();
                break;
            case 2:
                printf("Enter starting node for Depth First Search : ");
                scanf("%d",&v);
                for(i=1;i<=n;i++)
                    visited[i]=false;
                dfs(v);
                break;
            case 3:
                printf("Enter starting node for Depth First Search : ");
```

```
scanf("%d",&v);
for(i=1;i<=n;i++)
visited[i]=false;
dfs_rec(v);
break;
case 4:
printf("Enter starting node for Breadth First Search : ");
scanf("%d", &v);
for(i=1;i<=n;i++)
visited[i]=false;
bfs(v);
break;
case 5:
printf("Enter node to find adjacent vertices : ");
scanf("%d", &v);
printf("Adjacent Vertices are : ");
adj_nodes(v);
break;
case 6:
components();
break;
case 7:
exit(1);
default:
printf("Wrong choice\n");
break;
}/*End of switch*/
}/*End of while*/
}/*End of main()*/
```

```
create_graph()
{
int i,max_edges,origin,destin;
```

```
printf("Enter number of nodes : ");
scanf("%d",&n);
max_edges=n*(n-1);
```

```
for(i=1;i<=max_edges;i++)
{
printf("Enter edge %d( 0 0 to quit ) : ",i);
scanf("%d %d",&origin,&destin);
```

```
if((origin==0) && (destin==0))
break;
```

```
if( origin > n || destin > n || origin<=0 || destin<=0)
{
printf("Invalid edge!\n");
i--;
}
else
{
adj[origin][destin]=1;
}
}/*End of for*/
}/*End of create_graph()*/
```

```
display()
{
int i,j;
for(i=1;i<=n;i++)
{
for(j=1;j<=n;j++)
printf("%4d",adj[i][j]);
printf("\n");
}
}/*End of display()*/
```

```
dfs_rec(int v)
{
int i;
visited[v]=true;
printf("%d ",v);
for(i=1;i<=n;i++)
if(adj[v][i]==1 && visited[i]==false)
dfs_rec(i);
}/*End of dfs_rec()*/
```

```
dfs(int v)
{
int i,stack[MAX],top=-1,pop_v,j,t;
int ch;
```

```
top++;
stack[top]=v;
while (top>=0)
{
pop_v=stack[top];
top--; /*pop from stack*/
```

```
if( visited[pop_v]==false)
{
printf("%d ",pop_v);
visited[pop_v]=true;
}
else
continue;
for(i=n;i>=1;i--)
{
if( adj[pop_v][i]==1 && visited[i]==false)
{
top++; /* push all unvisited neighbours of pop_v */
stack[top]=i;
}/*End of if*/
}/*End of for*/
}/*End of while*/
}/*End of dfs()*/
```

```
bfs(int v)
{
int i,front,rear;
int que[20];
front=rear= -1;
```

```
printf("%d ",v);
visited[v]=true;
rear++;
front++;
que[rear]=v;
```

```
while(front<=rear)
{
v=que[front]; /* delete from queue */
front++;
for(i=1;i<=n;i++)
{
/* Check for adjacent unvisited nodes */
if( adj[v][i]==1 && visited[i]==false)
{
printf("%d ",i);
visited[i]=true;
rear++;
que[rear]=i;
}
}
}
```

```
/*End of while*/
/*End of bfs()*/

adj_nodes(int v)
{
int i;
for(i=1;i<=n;i++)
if(adj[v][i]==1)
printf("%d ",i);
printf("\n");
}/*End of adj_nodes()*/
components()
{
int i;
for(i=1;i<=n;i++)
visited[i]=false;
for(i=1;i<=n;i++)
{
if(visited[i]==false)
dfs_rec(i);
}
printf("\n");
}
```

QUIZ

Q. Let A be an adjacency matrix of a graph G. The k^{th} ij entry in the matrix $K A$, gives

- (A) The number of paths of length K from vertex V_i to vertex V_j .
- (B) Shortest path of K edges from vertex V_i to vertex V_j .
- (C) Length of a Eulerian path from vertex V_i to vertex V_j .
- (D) Length of a Hamiltonian cycle from vertex V_i to vertex V_j .

Ans:B

Q2. For an undirected graph with n vertices and e edges, what would be the sum of the degree of each vertex?

Ans:2e

Q3. An undirected graph G with n vertices and e edges is represented by adjacency list. What is the time required to generate all the connected components?

- (A) $O(n)$ (B) $O(e)$
- (C) $O(e+n)$ (D) $O(2e)$

Ans:C

Q4. A graph with n vertices will definitely have a parallel edge or self loop if the total number of edges are

- (A) more than n (B) more than $n+1$
- (C) more than $(n+1)/2$ (D) more than $n(n-1)/2$

Ans: D

Q5. Which data structure is required for Breadth First Traversal on a graph is

Ans: Queue

Q6. An adjacency matrix representation of a graph cannot contain information of:

- (A) nodes (B) edges
- (C) direction of edges (D) parallel edges

Ans:D

Q7. Does the minimum spanning tree of a graph give the shortest distance between any 2 specified nodes ?

Answer: Minimal spanning tree assures that the total weight of the tree is kept at its minimum but it doesn't mean that the distance between any two nodes involved in the minimum-spanning tree is minimum.

Q8. What is the maximum degree of any vertex in a simple graph with n vertices?

Ans: $n-1$

Q9. Which technique is useful in traversing a graph by breadth first search?

Ans: Queue

Q10. What is the minimum number of edges in a connected cyclic graph on n vertices?

Ans: n

EXPERIMENT NO.13

AIM: Write a program which simulates the various tree traversal algorithms.

```
#include<stdio.h>
#include<conio.h>
struct node
{
int data;
struct node *right, *left;
}*root,*p,*q;

struct node *make(int y)
{
struct node *newnode;
newnode=(struct node *)malloc(sizeof(struct node));
newnode->data=y;
newnode->right=newnode->left=NULL;
return(newnode);
}

void left(struct node *r,int x)
{
if(r->left!=NULL)
printf("\n Invalid !");
else
r->left=make(x);
}

void right(struct node *r,int x)
{
if(r->right!=NULL)
printf("\n Invalid !");
else
r->right=make(x);
}

void inorder(struct node *r)
{
if(r!=NULL)
```

```
{
inorder(r->left);
printf("\t %d",r->data);
inorder(r->right);
}
}
```

```
void preorder(struct node *r)
{
if(r!=NULL)
{
printf("\t %d",r->data);
preorder(r->left);
preorder(r->right);
}
}
```

```
void postorder(struct node *r)
{
if(r!=NULL)
{
postorder(r->left);
postorder(r->right);
printf("\t %d",r->data);
}
}
```

```
void main()
{
int no;
int choice;
clrscr();
printf("\n Enter the root:");
scanf("%d",& no);
root=make(no);
p=root;
while(1)
{
printf("\n Enter another number:");
```

```
scanf("%d",& no);
if(no==-1)
break;
p=root;
q=root;
while(no!=p->data && q!=NULL)
{
p=q;
if(nodata)
q=p->left;
else
q=p->right;
}
if(nodata)
{
printf("\n Left branch of %d is %d",p->data,no);
left(p,no);
}
else
{
right(p,no);
printf("\n Right Branch of %d is %d",p->data,no);
}
}
while(1)
{
printf("\n 1.Inorder Traversal \n 2.Preorder Traversal \n 3.Postorder Traversal \n 4.Exit");
printf("\n Enter choice:");
scanf("%d",&choice);
switch(choice)
{
case 1 :inorder(root);
break;
case 2 :preorder(root);
break;
case 3 :postorder(root);
break;
```

```
case 4 :exit(0);  
default:printf("Error ! Invalid Choice ");  
break;  
}  
getch();  
}
```

QUIZ

Q1. If a node having two children is deleted from a binary tree, it is replaced by which type of predecessor/successor?

Ans: Inorder Successor

Q2. How many leaf nodes are there in a full binary tree with $2n+1$ nodes?

Ans: n non-leaf nodes

Q3. If a node in a BST has two children, then its inorder predecessor has

- (A) no left child (B) no right child
- (C) two children (D) no child

Ans: B

Q4. What we call to a binary tree in which if all its levels except possibly the last, have the maximum number of nodes and all the nodes at the last level appear as far left as possible.

Ans: Full Binary Tree

Q5. How many nodes are there in a full binary tree with n leaves?

Ans: $2n - 1$ nodes

Q6. A BST is traversed in the following order recursively: Right, root, left

The output sequence will be in

- (A) Ascending order (B) Descending order
- (C) Bitomic sequence (D) No specific order

Ans: B

Q7. The pre-order and post order traversal of a Binary Tree generates the same output. The tree can have maximum

- (A) Three nodes (B) Two nodes
- (C) One node (D) Any number of nodes

Ans: C

Q8. What is the maximum possible number of nodes in a binary tree at level 6?

Ans: $2^6 = 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 64$

Q9. How to find the number of possible tree in the given tree.

Ans: The number of possible tree = $(2^{\text{power of } n}) - n$.

For example: A tree contain three node. So if $n=3$, the possible number of trees = $8 - 3 = 5$.

Q10. What is almost complete binary tree?

Ans: An almost complete binary tree is a tree in which each node that has a right child also has a left child. Having a left child does not require a node to have a right child. Stated alternately, an almost complete binary tree is a tree where for a right child, there is always a left child, but for a left child there may not be a right child. The number of nodes in a binary tree can be found using this formula: $n = 2^h$ Where n is the amount of nodes in the tree, and h is the height of the tree.

NETWORK THEORY

(EE – 223 – FF)

LAB MANUAL

IIIrd SEMESTER

LIST OF EXPERIMENTS

EXP NO.	NAME OF THE EXPERIMENT	PAGE NO.
1	To determine and verify Thevenin's and Norton's theorem.	3
2	To calculate and verify 'Z' parameters of two-port network.	6
3	To calculate and verify 'Y' parameters of two-port network.	8
4	To calculate and verify 'ABCD' parameters of two-port network.	10
5	To calculate and verify 'H' parameters of two-port network.	13
6	To determine equivalent parameters of parallel connection of two-port network.	15
7	To determine equivalent parameters of parallel connection of two-port network.	17
8	To determine the equivalent parameters of series connection of two port network.	19
9	To determine the A'B'C'D' parameters of the cascade connection of two-port network.	21
10	To Study P-Spice Software.	23
11	Introduction to circuit creation and simulation software TINAPRO	25
12	Introduction to Layout Tool, and creating Layout board using TINAPRO	27
13	Design a RLC resonance circuit & verify the transient response for different values of R, L & C	28

EXPERIMENT NO 10:

AIM: To determine and verify Thevenin's and Norton's Theorem.

APPARATUS REQUIRED: Power Supply, Bread Board, Connecting Lead

Voltmeter, Ammeter

THEORY:

THEVENIN'S THEOREM:

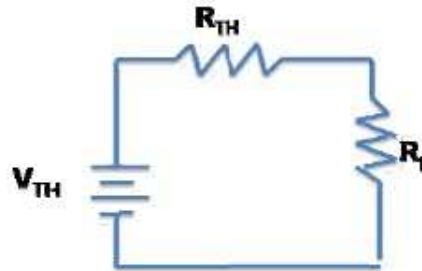
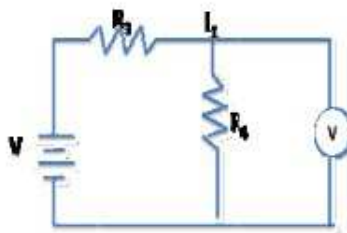
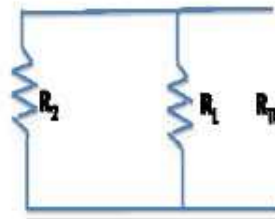
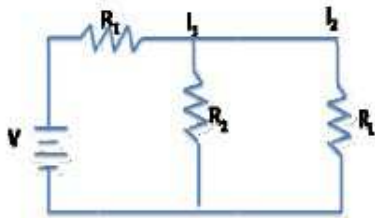
As applied to the network circuit, it can be stated that the current flowing through a load resistance R_L connected across any two terminals A and B of a linear bilateral network is given by $V_{TH} / (R_{TH} + R_L)$ where V_{TH} is the open circuit voltage and R_{TH} is the internal resistance of the network looking into the terminals A and B with all voltage sources replaced with their internal resistances and all current sources with infinite resistance.

NORTON'S THEOREM:

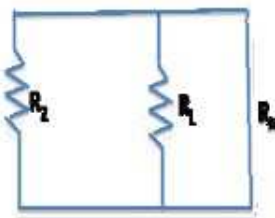
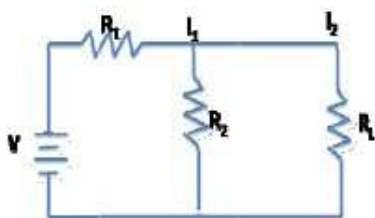
Replaces the electrical network by an equivalent constant current source and a parallel resistance. Norton's equivalent resistance $R_N = R_1 \times R_2 / (R_1 + R_2)$. Actual load current in the circuit $I_L = I_{SC} \times R_N / (R_N + R_L)$, I_{SC} is the short circuit current.

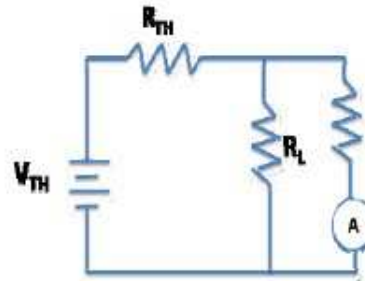
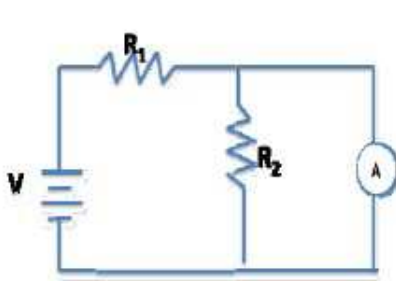
THEVENIN CIRCUIT DIAGRAM:

THEVENIN CIRCUIT DIAGRAM: 1) T



NORTON'S CIRCUIT DIAGRAM: 2) N





DURE:PROCED

NINCEDEURE:THEVEN PROC

Thevenin Theorem is a way to reduce a nn'smonetwork to an equivalent circuit coomposed of a single vooltage source series resise,stance, and sseries load.

Steps to ffollow for Thevenin's ThTheorem:

- (1) To find the curre flowing tenthrough the load resistance R_L as sshown in fig Remove R_L from the ckt temporaarily and leav the termin A and B open circuivenalsited.
- (2) Calcuulate the ope ckt voltag V_{TH} which appears acrengehross termina A and B.al
 $V_{TH} = I R_{TH}$. This is ccalled Theveenin's voltagege.
- (3) Now calculate $R_{TH} = R_1 R_2 / (R_1 + R_2)$. This i called TheRiseevenin's Resistance.τ
- (4) Analyze voltage and current for the load resistor folldowing the ru for serie circuits.uleses

$$I_L = V_{TH} / (R_L + R_{TH})$$

$$V_{TH} = E \times R_2 / (R_1 + R_2)$$

NORTO THEOREONEM:

Norton's Theorem is a way to reduce a network to an eqquivalent circ composed of a singlecuit ource, parall resistance and parall load.lele,lcurrent so

Steps to ffollow for Norton's TheoNorem:

- (1) Find the Norton source curre by removsentvng the load resistor fro the origin circuit andomnalnd calculatin current thnghrough a short (wire) juumping acro the open connection points wherosre the load rresistor used to be.d
- (2) Find the Norton resistance by removing all power soyources in the original cieircuit (voltage sources sshorted and current sources open) a calculati total resiandingistance between the oopen connectio points.on
- (3) Draw the Norton equivalent circuit, wit the Norto current sownththonource in parallel with thhe Norton reesistance. Th load resis re-attach between the two ope points of t equivalenhestorhesenthent circuit.
- (4) Analy voltage a current f the load resistor folloyzeandforowing the ru for paral circuits.ulesllel

OBSERVVATION TABLE:

1) TTHEVENIN TABLEN'S

S. No.	Appliedd Voltagee (volts)	V_{THH} (volts) Theo.	V_{TTH} (volts) Praact.	R_{th} (ohhms)	I_L (amp) Ththeo.	I_L (amp) Prract.	Result

NETWORK THEORY LAB (EE-223-F)

2) NORTON'S TABLE

S. No.	Applied Voltage (volts)	IN (amp.)	R _N (ohms)	I _{L1} (amp)	I _{L2} (amp)	Error I _{L1} -I _{L2}

RESULT: THEVENIN'S and NORTON'S THEOREM has been verified.

DISCUSSION: Thevenin's and Norton's theorems are dual theorems and can be used in the reduction of circuit analysis.

PRECAUTIONS:

- a) Make the connections according to the circuit diagram. Power supply should be switched off.
- b) Connections should be tight.
- c) Note the readings carefully.

QUIZ /ANSWERS:

Q.1 To what type of circuit thevenin's theorem is applicable

A. Linear and bilateral.

Q.2 What is the use of thevenin's theorem?

A. To convert the complex ckt into a voltage source and a series resistance.

Q.3 How R_{th} is connected with the ckt?

A. In series.

Q.4 How is R_{th} connected with the load resistance?

A. In series

Q.5 What modification is done in galvanometer to convert it into a ammeter?

A. A large resistance in parallel

Q.6 What modification is done in the galvanometer to convert it into a voltmeter?

A. A series resistance

Q.7 Resistance is an active element or the passive?

A. Passive

Q.8 How will you calculate the R_{th}?

A. The resistance between the two terminals

Q.9 In place of current source, what is placed while calculating R_{th}?

A. Replace current source by open ckt

Q.10 In place of voltage source which electrical parameters is placed?

A. A short ckt.

Q.11 To what type of network Norton's theorem applicable?

A. Two terminal linear network containing independent voltage and current sources.

Q.12 How is R_n connected to I_n ?

A. In the parallel

Q.13 What is placed in place of voltage sources while calculating the R_n ?

A. Their internal resistance replaces these.

Q.14 Give an example of unilateral ckt?

A. Diode rectifier

Q.15 What is unilateral ckt?

A. Whose characteristics changes with the change in direction of operation

Q.16 Give one example of the bilateral n/w?

A. Transmission lines

Q.17 What is the limitation of ohm's law?

A. Provided physical conditions do not change

Q.18 What is the reason that ground pin are made of greater diameter in the plugs?

A. $R = \rho L/A$

Q.19 Where is the voltage divider rule applicable?

A. Two resistance in series

Q.20 Where is the current divider rule applicable?

A. When there are two resistances in parallel.

EXPERIMENT NO: 2

AIM: To calculate and verify 'Z' parameters of two-port network.

APPARATUS REQUIRED: Power Supply, Bread Board, Five resistances, Connecting Leads. Voltmeter , Ammeter

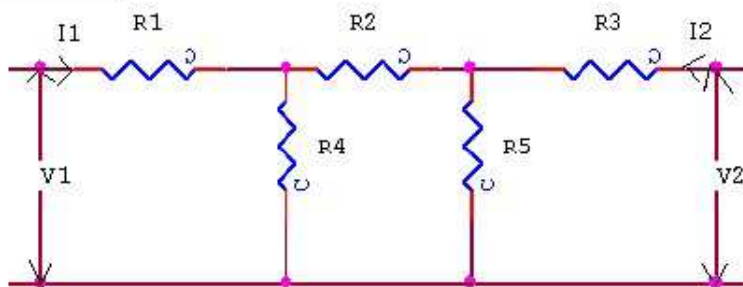
BRIEF THEORY: In Z parameters of a two-port, the input & output voltages V_1 & V_2 can be expressed in terms of input & output currents I_1 & I_2 . Out of four variables (i.e V_1 , V_2 , I_1 , I_2) V_1 & V_2 are dependent variables whereas I_1 & I_2 are independent variables. Thus,

$$V_1 = Z_{11}I_1 + Z_{12}I_2 \quad \text{----(1)}$$

$$V_2 = Z_{21}I_1 + Z_{22}I_2 \quad \text{----(2)}$$

Here Z_{11} & Z_{22} are the input & output driving point impedances while Z_{12} & Z_{21} are the reverse & forward transfer impedances.

CIRCUIT DIAGRAM:



PROCEDURE:

- Connect the circuit as shown in fig. & switch 'ON' the experimental board.
- First open the O/P terminal & supply 5V to I/P terminal. Measure O/P Voltage & I/P Current.
- Secondly, open I/P terminal & supply 5V to O/P terminal. Measure I/P Voltage & O/P current using multi-meter.
- Calculate the values of Z parameter using Equation (1) & (2).
- Switch 'OFF' the supply after taking the readings.

OBSERVATION TABLE:

S.No	When I/P is open ckt			When O/P is open ckt		
	V_2	V_1	I_2	V_2	V_1	I_1

SAMPLE CALCULATION:

- (1) When O/P is open circuited i.e. $I_2 = 0$
 $Z_{11} = V_1/I_1$ $Z_{21} = V_2/I_1$

(2) When I/P is open circuited i.e. $I_1 = 0$

$$Z_{12} = V_1/I_2 Z_{22} = V_2/I_2$$

RESULT/CONCLUSION: The Z-parameters of the two port network has been calculated and verified.

DISCUSSION: The Z-parameters are open circuit parameters.

PRECAUTIONS:

- a) Make the connections according to the circuit diagram. Power supply should be switched off.
- b) Connections should be tight.
- c) Note the readings carefully.

QUIZ /ANSWERS:

Q1. Define Z parameters?	A1. In Z parameters, the input & output voltages V_1 & V_2 can be expressed in terms of input & output currents I_1 & I_2 .
Q2. List the four variables used in Z-parameter representation	A2. The four variables are V_1 , V_2 , I_1 & I_2
Q3. List the two dependent variables used in Z- parameter representation	A3. The two dependent variables are V_1 & V_2
Q4. List the two independent variables used in Z- parameter representation.	A4. The two independent variables are I_1 & I_2
Q5. Define input driving point impedance	A5. The input driving point impedance is defined as the ratio of input voltage to the input current
Q6. Define output driving point impedance	A6. The output driving point impedance is defined as the ratio of output voltage to the output current.
Q7. Define reverse transfer impedance.	A7. The reverse transfer impedance is defined as ratio of input voltage to the output current
Q8. Define forward transfer impedance	A8. The forward transfer impedance is defined as ratio of output voltage to the input current
Q9. Write condition for reciprocity.	A9. Condition for reciprocity is $Z_{12} = Z_{21}$.
Q10. Write condition for symmetry.	A10. Condition for symmetry is $Z_{11} = Z_{22}$.

EXPERIMENT NO: 3

AIM: To calculate and verify 'Y' parameters of two-port network.

APPARATUS REQUIRED: Power supply, Bread Board, Five resistances, Connecting Leads, Voltmeter, and Ammeter.

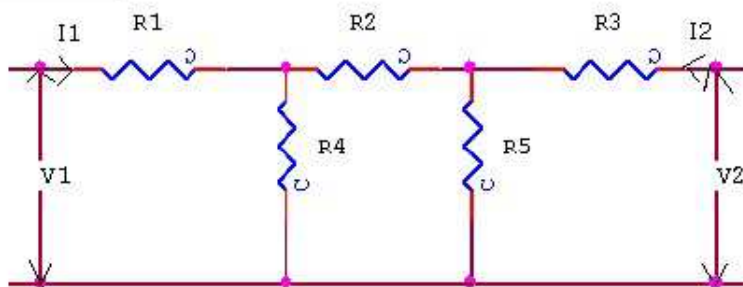
BRIEF THEORY : In Y parameters of a two-port, the input & output currents I_1 & I_2 can be expressed in terms of input & output voltages V_1 & V_2 . Out of four variables (i.e I_1 , I_2 , V_1 , V_2) I_1 & I_2 are dependent variables whereas V_1 & V_2 are independent variables.

$$I_1 = Y_{11}V_1 + Y_{12}V_2 \quad \text{-----(1)}$$

$$I_2 = Y_{21}V_1 + Y_{22}V_2 \quad \text{-----(2)}$$

Here Y_{11} & Y_{22} are the input & output driving point admittances while Y_{12} & Y_{21} are the reverse & forward transfer admittances.

CIRCUIT DIAGRAM:



PROCEDURE :

- Connect the circuit as shown in fig. & switch 'ON' the experimental board.
- First short the O/P terminal & supply 5V to I/P terminal. Measure O/P & I/P current
- Secondly, short I/P terminal & supply 5V to O/P terminal. Measure I/P & O/P current using multi-meter.
- Calculate the values of Y parameter using Eq. (1) & (2).
- Switch 'off' the supply after taking the readings.

OBSERVATION TABLE:

S.No	When I/P is short ckt			When O/P is short ckt		
	V_2	I_1	I_2	V_1	I_1	I_2

SAMPLE CALCULATION:

- (1) When O/P is short circuited i.e. $V_2 = 0$
 $Y_{11} = I_1/V_1$ $Y_{21} = I_2/V_1$

(2) When I/P is short circuited i.e. $V_1 = 0$
 $Y_{12} = I_1/V_2 = I_2/V_2$

RESULT/CONCLUSION: The Y-parameters of the two port network has been calculated and verified.

DISCUSSION: The Y-parameters are short circuit parameters

PRECAUTIONS:

- a) Make the connections according to the circuit diagram. Power supply should be switched off.
- b) Connections should be tight.
- c) Note the readings carefully.

QUIZ/ANSWERS:

Q1. Define Y parameters?	A1. In Y-parameters the input & output currents I_1 & I_2 can be expressed in terms of input & output voltages V_1 & V_2 .
Q2. List the four variables used in Y-parameter representation	A2. The four variables are I_1 , I_2 , V_1 and V_2 .
Q3. List the two dependent variables used in Y-parameter representation	A3. The two dependent variables are I_1 & I_2
Q4. List the two independent variables used in Y-parameter representation	A4. The two independent variables are V_1 & V_2 .
Q5. Define input driving point admittance	A5. The input driving point admittance is defined as the ratio of input current to the input voltage.
Q6. Define output driving point admittance	A6. The output driving point admittance is defined as the ratio of output current to the output voltage.
Q7. Define reverse transfer admittance	A7. The reverse transfer ratio is defined as ratio of input current to the output voltage
Q8. Define forward transfer admittance	A8. The forward transfer ratio is defined as ratio of output current to the input voltage
Q9. Write condition for reciprocity.	A9. Condition for reciprocity is $Y_{12} = Y_{21}$
Q10. Write condition for symmetry.	A10. Condition for symmetry is $Y_{11} = Y_{22}$

EXPERIMENT NO: 4

AIM: To calculate and verify 'ABCD' parameters of two-port network

APPARATUS REQUIRED: Power Supply, Bread Board, Five resistances, Connecting Leads, Voltmeter, and Ammeter.

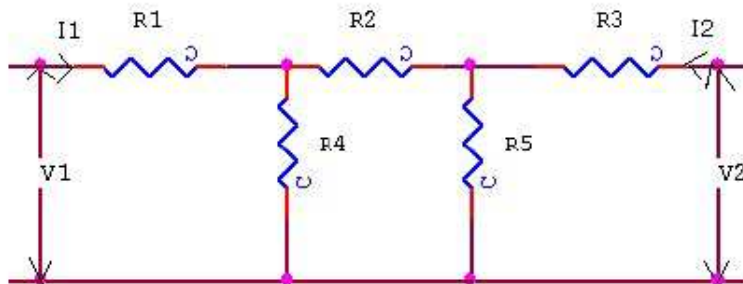
BRIEF THEORY: ABCD parameters are widely used in analysis of power transmission engineering where they are termed as "Circuit Parameters". ABCD parameters are also known as "Transmission Parameters". In these parameters, the voltage & current at the sending end terminals can be expressed in terms of voltage & current at the receiving end. Thus,

$$V_1 = AV_2 + B(-I_2)$$

$$I_1 = CV_2 + D(-I_2)$$

Here "A" is called reverse voltage ratio, "B" is called transfer impedance "C" is called transfer admittance & "D" is called reverse current ratio.

CIRCUIT DIAGRAM:



PROCEDURE :

- Connect the circuit as shown in fig. & switch 'ON' the experimental board.
- First open the O/P terminal & supply 5V to I/P terminal. Measure O/P voltage & I/P current
- Secondly, short the O/P terminal & supply 5V to I/P terminal. Measure I/P & O/P current using multi-meter.
- Calculate the A, B, C, & D parameters using the Eq. (1) & (2).
- Switch 'off' the supply after taking the readings.

OBSERVATION TABLE:

S.No	When O/P is open ckt			When O/P is short ckt		
	V ₁	V ₂	I ₁	V ₁	I ₂	I ₁

SAMPLE CALCULATION:

(1) When O/P is open circuited i.e. $I_2 = 0$

$$A = V_1/V_2C = I_1/V_2$$

(2) When O/P is short circuited i.e. $V_2 = 0$

NETWORK THEORY LAB (EE-223-F)

$$B = -V_1/I_2$$

$$D = -I_1/I_2$$

RESULT/CONCLUSION: The ABCD-parameters of the two-port network has been calculated and verified.

DISCUSSION: ABCD parameters are transmission parameters.

PRECAUTIONS:

- a) Make the connections according to the circuit diagram. Power supply should be switched off.
- b) Connections should be tight.
- c) Note the readings carefully.

QUIZ/ANSWERS:

Q1. Define transmission parameters	A1. In these parameters, the voltage & current at the sending end terminals can be expressed in terms of voltage & current at the receiving end.
Q2. Why ABCD parameters are also called as transmission parameters?	A2. ABCD parameters are also called as transmission parameters because these are used in the analysis power transmission lines
Q3. Where they are used?	A3. Transmission line theory & cascade network
Q4. Define reverse voltage ratio (A) .	A4. It is defined as the ratio of sending end voltage to the receiving end voltage
Q5. Define transfer impedance (B).	A5. It is defined as the ratio of sending end voltage to the receiving end current with the receiving end current assumed to be in reverse direction
Q6. Define transfer admittance (C).	A6. It is defined as the ratio of sending end current to the receiving end voltage
Q7. Define reverse current ratio (D).	A7. It is defined as the ratio of sending end current to the receiving end current with the receiving end current assumed to be in reverse direction
Q8. Write the units of parameters B & C.	A8. Unit of parameter B is ohm & of C is mho.
Q9. Write the units of parameters A & D.	A9. Both parameters A & D are unit less.
Q10. Write the condition for symmetry & Reciprocity.	A10. The condition for symmetry is $A = D$ & the condition for reciprocity is $AD - BC = 1$.

EXPERIMENT NO: 5

AIM: To calculate and verify 'H' parameters of two-port network

APPARATUS REQUIRED: Power supply, Bread Board, Five resistances, Connecting Leads, Multimeter.

BRIEF THEORY: In 'h' parameters of a two port network, voltage of the input port and the current of the output port are expressed in terms of the current of the input port and the voltage of the output port. Due to this reason, these parameters are called as 'hybrid' parameters, i.e. out of four variables (i.e. V_1, V_2, I_1, I_2) V_1, I_2 are dependent variables. Thus,

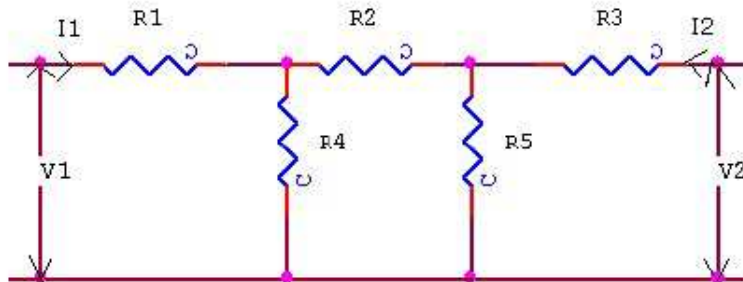
$$V_1 = h_{11}I_1 + h_{12}V_2 \text{ ----- (1)}$$

$$I_2 = h_{21}I_1 + h_{22}V_2 \text{ ----- (2)}$$

h_{11} and h_{22} are input impedance and output admittance.

h_{21} and h_{12} are forward current gain and reverse voltage gain.

CIRCUIT DIAGRAM:



PROCEDURE :

- Connect the circuit as shown in fig. & switch 'ON' the experimental board.
- Short the output port and excite input port with a known voltage source V_s . So that $V_1 = V_s$ and $V_2 = 0$. We determine I_1 and I_2 to obtain h_{11} and h_{21} .
- Input port is open circuited and output port is excited with the same voltage source V_s . So that $V_2 = V_s$ and $I_1 = 0$, we determine I_2 and V_1 to obtain h_{12} and h_{22} .
- Switch 'off' the supply after taking the readings.

OBSERVATION TABLE:

S.No	When O/P is short ckt			When I/P is short ckt		
	V_1	I_1	I_2	V_2	V_1	I_2

SAMPLE CALCULATION:

(1) When O/P is short circuited i.e. $V_2 = 0$

$$h_{11} = V_1/I_1 \quad h_{21} = I_2/I_1$$

(2) When I/P is open circuited i.e. $I_1 = 0$

$$h_{12} = V_1/V_2 \quad h_{22} = I_2/V_2$$

NETWORK THEORY LAB (EE-223-F)

RESULT/CONCLUSION: The h-parameters of the two port network has been calculated and verified.

DISCUSSION: The h-parameters are short circuit parameters

PRECAUTIONS:

- Make the connections according to the circuit diagram. Power supply should be switched off.
- Connections should be tight.
- Note the readings carefully.

QUIZ/ANSWERS:

Q1. Define H parameters?	A1. In 'h' parameters of a two port network, voltage of the input port and current of the output port are expressed in terms of the current of the input port and voltage of the output port.
Q2. List the four variables used in h-parameter representation	A2. The four variables are V_1 , V_2 , I_1 and I_2 .
Q3. List the two dependent variables used in h-parameter representation	A3. The two dependent variables are V_1 & I_2 .
Q4. List the two independent variables used in h-parameter representation	A4. The two independent variables are I_1 & V_2 .
Q5. Define input impedance	A5. $h_{11} = V_1/I_1$.
Q6. Define output admittance	A6. $h_{22} = I_2/V_2$.
Q7. Define forward current gain	A7. $h_{21} = I_2/I_1$.
Q8. Define reverse current gain	A8. $h_{12} = V_1/V_2$.
Q9. Write condition for reciprocity.	A9. Condition for reciprocity is $h_{12} = h_{21}$.
Q10. Write condition for symmetry.	A10. Condition for symmetry is $h_{11} = h_{22}$.

EXPERIMENT NO: 6

AIM: To calculate and verify 'G' parameters of two-port network.

APPARATUS REQUIRED: Power supply, Bread Board, Five resistances, Connecting Leads, Multimeter.

BRIEF THEORY: In 'g' parameters of a two port network, the current at the input port I_1 & The voltage at the output port V_2 can be expressed in terms of I_2 & V_1 .

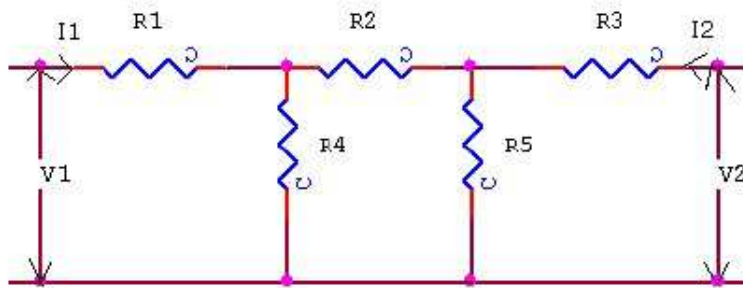
$$I_1 = g_{11}V_1 + g_{12}I_2 \quad (1)$$

$$V_2 = g_{21}V_1 + g_{22}I_2 \quad (2)$$

G_{11} and G_{22} are input driving point admittance and output driving point impedance.

G_{21} and G_{12} are forward current gain and reverse voltage gain.

CIRCUIT DIAGRAM:



PROCEDURE:

- 1) Connect the circuit as shown in fig. & switch 'ON' the experimental board.
- 2) Open the output port & excite input port with a known voltage source V_s , So that $V_1 = V_s$ & $I_2 = 0$. We determine I_1 & V_2 to obtain g_{11} & g_{21} .
- 3) Input port is short circuited and out port is excited with the same voltage source V_s , so that $V_2 = V_s$ & $V_1 = 0$. We determine I_1 & I_2 to obtain g_{12} & g_{22} .
- 4) Switch 'off' the supply after taking the readings.

OBSERVATION TABLE:

S.No	When O/P is open ckt			When I/P is short ckt		
	V_1	I_1	V_2	V_2	I_2	

SAMPLE CALCULATION:

- (1) When O/P is open circuited i.e. $I_2 = 0$

$$g_{11} = I_1/V_1 \quad g_{21} = V_2/V_1$$

- (2) When I/P is short circuited i.e. $V_1 = 0$

$$g_{12} = I_1/I_2 \quad g_{22} = V_2/I_2$$

RESULT/CONCLUSION: The G-parameters of the two port network has been calculated and verified.

PRECAUTIONS:

- a) Make the connections according to the circuit diagram. Power supply should be switched off.
- b) Connections should be tight.
- c) Note the readings carefully.

QUIZ/ANSWERS:

Q1. Define G parameters?	A1. In 'G' parameters of a two port, the current at the input port I_1 & the voltage at the output port V_2 can be expressed in terms of I_2 & V_1 network.
Q2. List the four variables used in g-parameter representation	A2. The four variables are V_1 , V_2 , I_1 and I_2 .
Q3. List the two dependent variables used in g-parameter representation	A3. The two dependent variables are I_1 & V_2 .
Q4. List the two independent variables used in h-parameter representation	A4. The two independent variables are V_1 & I_2 .
Q5. Define input driving point admittance	A5. $g_{11} = I_1/V_1$.
Q6. Define output driving point impedance	A6. $g_{22} = V_2/I_2$.
Q7. Define forward current gain	A7. $g_{21} = V_2/V_1$.
Q8. Define reverse voltage gain	A8. $g_{12} = I_1/I_2$.
Q9. Write condition for reciprocity.	A9. Condition for reciprocity is $g_{12} = g_{21}$.
Q10. Write condition for symmetry.	A10. Condition for symmetry is $g_{11} = g_{22}$.

EXPERIMENT NO: 7

AIM: To determine equivalent parameters of parallel connection of two-port network

APPARATUS REQUIRED: Power Supply, Bread Board, Five Resistances, Connecting Leads, Voltmeter, and Ammeter.

BRIEF THEORY: Consider two port N/Ws connected in parallel so that they have common reference node, then the equation of the N/Ws A&B in terms of Y parameters are given by -

$$Y_{11} = Y_{11A} + Y_{11B}$$

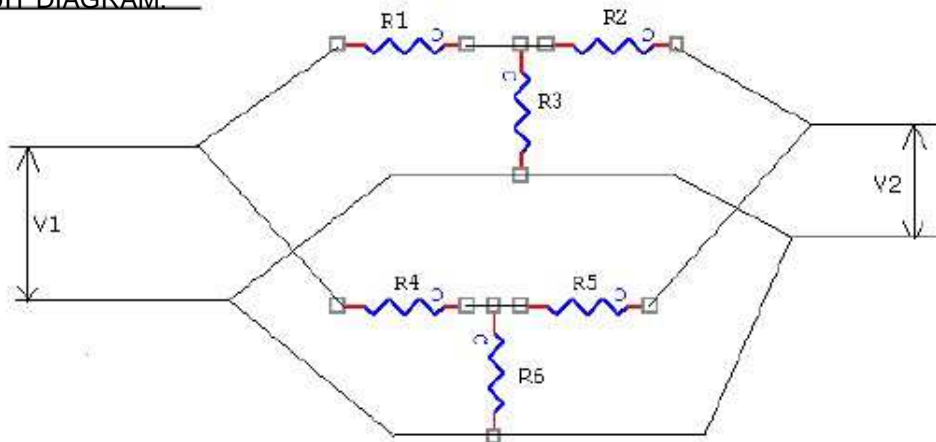
$$Y_{12} = Y_{12A} + Y_{12B}$$

$$Y_{21} = Y_{21A} + Y_{21B}$$

$$Y_{22} = Y_{22A} + Y_{22B}$$

Thus we see that each Y parameter of the parallel N/W is given as the sum of the corresponding parameters of the individual N/Ws.

CIRCUIT DIAGRAM:



PROCEDURE:

- Connect the N/Ws A&B separately on the Bread board according to the fig.
- Take the Reading according to the observation table & calculate Y parameters for both N/Ws & add them.
- Connect two N/Ws A&B in parallel & take the readings.
- Calculate the Y parameters of parallel connected N/Ws.
- Verify that the sum of parameters of A&B N/Ws is equal to the parameters of parallel connected N/Ws.

OBSERVATION TABLE:

S. No.	When I/P is short ckt			When O/P is short ckt		
	V ₂ I ₁ I ₂			V ₁ I ₁ I ₂		

SAMPLE CALCULATION:

NETWORK THEORY LAB (EE-223-F)

(1) When O/P is short circuited i.e. $V_2 = 0$
 $Y_{11} = I_1/V_1$ $Y_{21} = I_2/V_1$

(2) When I/P is short circuited i.e. $V_1 = 0$
 $Y_{12} = I_1/V_2$ $Y_{22} = I_2/V_2$

RESULT/CONCLUSION: The Y-parameters of parallel connection of two-port network has been determined.

DISCUSSION: The overall Y-parameters of a parallel connection is equal to sum of individual network parameters.

PRECAUTIONS:

- a) Make the connections according to the circuit diagram. Power supply should be switched off.
- b) Connections should be tight.
- c) Note the readings carefully.

QUIZ/ANSWERS:

Q1. What will be the total admittance if the two networks are connected in series?	A1. The total admittance (Z) = $Z_1 + Z_2$
Q2. What will be the total admittance if the two networks are connected in parallel?	A2. The total admittance (Y) = $Y_1 + Y_2$
Q3. Which parameter is used for the representation of parallel connection of two port network?	A3. Y-parameters
Q4. Which parameter is used for the representation of series connection of two port network?	A4. Z-parameters
Q5. Difference between Z & Y parameters	A5. Z-parameters are called open ckt while Y-parameters are called short ckt parameters. Z-parameters are used for series connection while Y-parameters are used for parallel connection.
Q6. What do you mean by cascade connection?	A6. The network is said to be in cascade when the o/p of one port becomes the input for second n/w.
Q7. Is Z inversely proportional to Y in one port network?	A7. Yes.
Q8. Is Z inversely proportional to Y in two port network?	A8. No.
Q9. A two port network is simply a network inside a black box & the network has only a) two terminals b) two pairs of accessible terminals two pairs of ports	A9. (b)
Q10. The number of possible combinations generated by four variables taken two at a time in a two-port network is (a) Four (b) two (c) six	A10. (c)

EXPERIMENT NO: 8

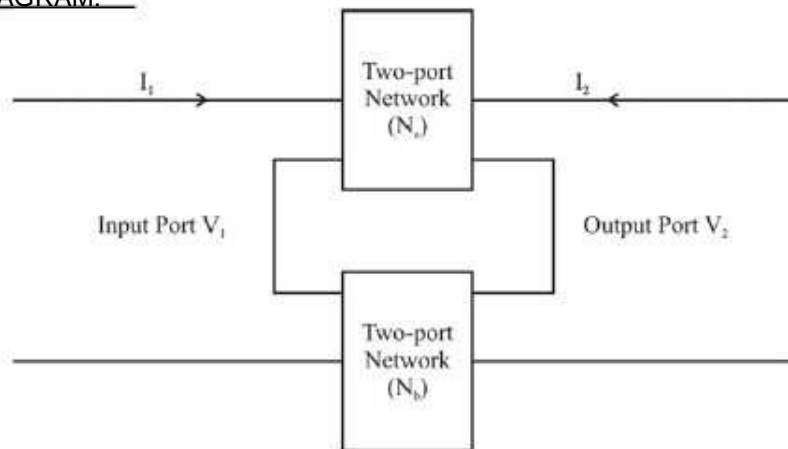
AIM: To determine the equivalent parameters of series connection of two port network

APPARATUS REQUIRED: Power Supply, Bread Board, Resistances, Connecting Leads, Multimeter

BRIEF THEORY: The Series connection is also called as Series-Series connection, Since both input ports & output ports are series connected. Thus,
 $V_1 = Z'_{11}I_1 + Z'_{12}I_2$
 $V_2 = Z'_{21}I_1 + Z'_{22}I_2$

V_1 & V_2 are dependent variables, I_1 & I_2 are independent variables, Z'_{11} & Z'_{22} is Input & Output driving point impedance. Z'_{12} & Z'_{21} is Reverse & Forward transfer impedance.

CIRCUIT DIAGRAM:



PROCEDURE:

- 1) Connect the circuit as shown in fig and switch 'ON' the experiment board.
- 2) Open the output port & excite input port with a known voltage source V_s so that $V_1 = V_s$ & $I_2 = 0$. We determine I_1 & I_2 to obtain Z'_{11} & Z'_{21} .
- 3) Input port is open circuited & Output port is excited with the same voltage source V_s so that $V_2 = V_s$ & $I_1 = 0$. We determine I_2 & V_1 to obtain Z'_{22} & Z'_{12} .
- 4) Switch OFF the supply after taking the readings.

OBSERVATION TABLE:

S.N.O	When O/P is open cktd			When I/P is open cktd		
	$V_1 I_1 V_2$			$V_2 V_1 I_2$		

SAMPLE CALCULATION:

- (1) When O/P is open circuited i.e. $I_2 = 0$
 $Z'_{11} = V_1 / I_1$ $Z'_{21} = V_2 / I_1$
- (2) When I/P is open circuited i.e. $I_1 = 0$
 $Z'_{12} = V_1 / I_2$ $Z'_{22} = V_2 / I_2$

NETWORK THEORY LAB (EE-223-F)

RESULT/CONCLUSION: The Z'_{11} , Z'_{12} , Z'_{21} , Z'_{22} parameters of two-port network has been determined.

PRECAUTIONS:

- a) Make the connections according to the circuit diagram. Power supply should be switched off.
- b) Connections should be tight.
- c) Note the readings carefully.

QUIZ/ANSWERS:

Q1 What do you mean by cascade connection?	A1. The network is said to be in cascade when the output of one port becomes the input for second network
Q2 A two port network is simply a network inside a black box & the network has only c) two terminals d) two pairs of accessible terminals two pairs of ports	A9.(b)
Q3.What is Input driving point impedance	A3. $Z'_{11} = V_1/I_1$
Q4.What is Output driving point impedance	A4. $Z'_{22} = V_2/I_2$
Q5. What is Reverse Transfer Impedance	A5. $Z'_{12} = V_1/I_2$
Q6. What is Reverse Transfer Admittance	A6. $Z'_{21} = V_2/I_1$
Q7. Is Z inversely proportional to Y in one port network?	A7.Yes.
Q8. Is Z inversely proportional to Y in two port network?	A8.No.
Q9. The number of possible combinations generated by four variables taken two at a time in a two-port network is (a) Four (b) two (c) six	A9.(c)

EXPERIMENT NO: 9

AIM: To determine the A'B'C'D' parameters of the cascade connection of two-port network

APPARATUS REQUIRED: Power Supply, Bread Board, Resistances, Connecting Leads, Multi-meter.

BRIEF THEORY: Two port networks are said to be connected in cascade if the output port of the first becomes the input port of the second as shown in fig.

$$V_1 = A'V_2 + B'(-I_2)(1)$$

$$I_1 = C'V_2 + D'(-I_2)(2)$$

V₁ and I₁ are dependent variables; V₂ and I₂ are independent variables

A', D' is Reverse Voltage Ratio & Reverse Current Ratio.

B', C' is Reverse Transfer Impedance & Reverse Transfer Admittance.

CIRCUIT DIAGRAM:



PROCEDURE:

- 1) Connect the circuit as shown in fig and switch 'ON' the experiment board.
- 2) In A, B, C, D parameters, open the output port and excite input port with a known voltage source Vs so that V₁ = Vs and V₂ = 0. We determine I₁ and V₂ to obtain A' & C'.
- 3) The output port is short circuited and input port is excited with the same voltage source Vs so that V₁ = Vs & V₂ = 0. We determine I₁ & I₂ to obtain B' & D'.
- 4) Switch OFF the supply after taking the readings.

OBSERVATION TABLE:

S.N.O	When O/P is open ckt			When O/P is short ckt		
	V ₂ I ₁ I ₂			V ₁ I ₁ I ₂		

SAMPLE CALCULATION:

- (1) When O/P is open circuited i.e. I₂ = 0
A' = V₁/V₂C' = I₁/V₂
- (2) When O/P is short circuited i.e. V₂ = 0
B' = V₁/-I₂D' = I₁/-I₂

RESULT/CONCLUSION: The A, B, C, D parameters of two-port network have been determined.

PRECAUTIONS:

- a) Make the connections according to the circuit diagram. Power supply should be switched off.
- b) Connections should be tight.
- c) Note the readings carefully.

QUIZ/ANSWERS:

Q1 What do you mean by cascade connection?	A1. The network is said to be in cascade when the output of one port becomes the input for second network
Q2 A two port network is simply a network inside a black box & the network has only e) two terminals f) two pairs of accessible terminals two pairs of ports	A9.(b)
Q3.What is Reverse Voltage Ratio	A3. $A' = V_1/V_2$
Q4.What is Reverse Current Ratio	A4. $D' = I_1/-I_2$
Q5. What is Reverse Transfer Impedance	A5. $B' = V_1/-I_2$
Q6. What is Reverse Transfer Admittance	A6. $C' = I_1/V_2$
Q7. Is Z inversely proportional to Y in one port network?	A7.Yes.
Q8.Is Z inversely proportional to Y in two port network?	A8.No.
Q9.The number of possible combinations generated by four variables taken two at a time in a two-port network is (a) Four (b) two (c) six	A9.(c)

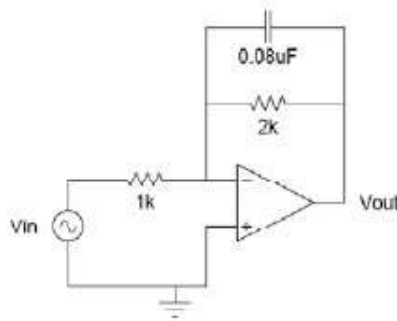
EXPERIMENT-10

AIM: To Study P-Spice Software.

THEORY: SPICE (Simulation Program for Integrated Circuits Emphasis) is an analog circuit simulator developed at Berkeley. Many different versions of SPICE are available from many different vendors. Common SPICEs include HSPICE, PSPICE, and B 2 SPICE. SPICE takes a circuit net list and performs mathematical simulation of the circuit's behavior. A net list describes the components in the circuit and how they are connected. SPICE can simulate DC operating point, AC response, transient response, and other useful simulations.

Inputting your Circuit

1. From the Start Menu, select ProgramsPSPICE StudentSchematics. You are now running the schematic capture program, which you will use to enter the simple integrator
2. Add an ideal op-amp:
 - Open the Get New Part window by clicking on the binocular icon or using the Draw menu.
 - Search for available op-amp models by typing "op-amp" in the description search box.
 - You should see a list of available op-amp models. Notice the 741 and 411 are both available in lab. It is possible to add more models if you ever need to do so.
 - Select OPAMP, the ideal op-amp model. Click on "Place and Close"
 - Place the op-amp on the sheet by left-clicking the mouse.
 - Double click on the op-amp. You should see a window listing the parameters for the part. Notice you can set the op-amp gain. You don't need to change any parameters for this part. If you need to know what the parameters mean for a part you are using, the help contents has a list of parameters and their meaning.
3. Add the resistors and capacitors:
 - Use Get New Part to place the resistors and capacitors on the sheet. Use the R and C components (ideal resistor and ideal capacitor.)
 - Set the component values by double clicking on the value already displayed next to the part. You can also change the component value by double clicking on the component itself and changing the correct parameter.
4. Add an AC voltage source.
 - Use Get New Part to place an AC voltage source (VAC).
 - Set the AC magnitude (ACMAG) to 1V.
5. Add a ground (GND_EARTH)
6. Connect the components using the Draw Wire button, or use the Draw menu.
7. Name the nodes to useful names
 - Double click on the wire between the output of the op-amp and the resistor. Name this node something like "Vout".
 - Rename the other nodes as well. The ground node is already labelled "0" and cannot be changed to anything else.



Simulating Your Circuit

1. Set the simulation parameters.
 - Go to Analysis Setup.
 - Enable AC Sweep
 - Click on the AC Sweep button.
 - Set the AC parameters to:
 - o Sweep Type = Decade
 - o Pts/Decade = 101
 - o Start Freq = 10
 - o End Freq = 1meg
2. Save your schematic. Save it in D:\SAVE STUFF HERE\ your_folder_here
3. Simulate the circuit by clicking on the simulate button, or use the Analysis menu.

Plotting the Results

1. When the plotting window appears, plot the results:
 - Click on the Add Trace button or use the Trace menu.
 - Uncheck Alias Names
 - You can plot any of the parameters seen on the left. Select and plot V (Vout). Make sure the plot matches what you expect.
 - Set the y-axis to a log scale by double clicking on the axis.
 - Now, plot the same data using dB for the y-axis.
 - o Trace Delete All Traces
 - o Set the y-axis back to the linear scale.
 - o Add Trace
 - o In the Trace Expression box, type: $20 \cdot \log_{10}(V(Vout))$ Click OK.
 - Find the -3dB point on the plot using the markers:
 - o Turn on the cursors using the Toggle Cursors button or the Trace Cursor menu.
 - o A small window has appeared showing the position of the cursors and their values. The left mouse button moves one cursor and the right mouse button moves the second.
 - o Using the cursors and the -3dB point by setting one cursor to 10Hz and move the second cursor until their difference is 3dB. Is this the frequency you expect?

CONCLUSION: Hence we have studied the P-Spice Software.

QUIZ QUESTIONS & ANSWERS

Q. 1 What is P-SPICE Design Suite?

Ans. P-SPICE Design Suite is a powerful yet affordable circuit simulation and PCB design software package.

Q.2 Why P-SPICE Design Suite is used?

Ans. It is used for analyzing, designing, and real time testing of analog, digital, VHDL, MCU, and mixed electronic circuits and their PCB layouts.

Q.3 What is unique feature of P-SPICE?

Ans. A unique feature of PSPICE is that you can bring your circuit to life with the optional USB controlled.

Q.4 How it is used to compute computational power?

Ans. To meet this requirement PSPICE v9 has the ability to utilize the increasingly popular scalable multi-thread CPUs.

Q.5 What is the use of schematic editor ?

Ans. It enhances your schematics by adding text and graphics elements such lines, arcs arrows, frames around the schematics and title blocks.

Q.6 How many manufacturer models it contain?

Ans. More than 20,000.

Q.7 Name all the Virtual Instruments used in it.

Ans. Virtual Instruments: Oscilloscope, Function Generator, Multimeter, Signal Analyzer/Bode Plotter, Network Analyzer, Spectrum Analyzer, Logic Analyzer, Digital Signal Generator, XY Recorder.

Q.8 What is Real-time Storage Oscilloscope?

Ans. It observes the actual time response of your circuit.

Q.9 What is Real-time Signal Analyzer?

Ans. It measure the frequency response of your circuit in real-time.

Q.10 What's role P-SPICE play in Integrated PCB design?

Ans. The new fully integrated layout module of P-SPICE has all the features you need for advanced PCB design, including multilayer PCB's with split power plane layers, powerful autoplacement & autorouting, rip-up and reroute, manual and "follow-me" trace placement, DRC, forward and back annotation, pin and gate swapping, keep-in and keep-out areas, copper pour, thermal relief, fanout, 3D view of your PCB design from any angle, Gerber file output and much more.

EXPERIMENT NO: 11

AIM: Introduction to circuit creation and simulation software TINAPRO.

APPARATUS: PC installed with TINAPRO

THEORY: The study of TINAPRO mainly consists of:
Introduction to TINAPRO Capture: Generation of Schematic, Generation of net list.

SCHEMATIC:

A schematic is merely a collection of electronic symbols connected together with virtual "wires." The main reason you need a schematic when fabricating a printed circuit board is to provide input (a net list) to your layout and routing tool.

NETLIST:

A net list is a file, usually ASCII text, which defines the connections between the components in your design.

CREATING PROJECT:

To create a new project, use Capture's Project Wizard. The Project Wizard provides you with the framework for creating any kind of project.

- 1 Launch Capture
- 2 From the File menu, choose New > Project.
- 3 In the New Project dialog box, specify the project name
- 4 To specify the project type, select Analog or Mixed A/D.
- 5 Specify the location where you want the project files to be created and click OK.
- 6 In the Create P Spice Project dialog box, select the Create a blank project option button.

Adding parts:

To add parts to your design:

- 1 From the Place menu in Capture, select Part.
 - 2 In the Place Part dialog box, first select the library from which the part is to be added and then instantiate the part on the schematic page
- While working with the Capture, add parts from "Name of Library". OLB.
To add libraries to the project, select the Add Library button.
- 3 Browse to <install_dir>/tools/capture/library/pspice/eval.olb.

Connecting parts:

After placing the required parts on the schematic page, you need to connect the parts.
From the Place menu, choose Wire.

Net List:

Save the schematic and close the schematic page.
Open tools and create net list.
Select Layout tab
Be sure to put the *.MNL file in a unique folder.

Latter many more design files will be generated and it will be much easier to sort them out if the design is by itself.

Click on OK

The net list that Layout needs has been created. The file has the name of the project.MNL

CONCLUSION: Thus we have studied the TINAPRO circuit capturing, Generation of Schematic and net list.

QUIZ QUESTIONS & ANSWERS

Q. 1 What is TINA Design Suite?

Ans. TINA Design Suite is a powerful yet affordable circuit simulation and PCB design software package.

Q.2 Why TINA Design Suite is used?

Ans. It is used for analyzing, designing, and real time testing of analog, digital, VHDL, MCU, and mixed electronic circuits and their PCB layouts.

Q.3 What is unique feature of TINA?

Ans. A unique feature of TINA is that you can bring your circuit to life with the optional USB controlled.

Q.4 How it is used to compute computational power?

Ans. To meet this requirement TINA v9 has the ability to utilize the increasingly popular scalable multi-thread CPUs.

Q.5 What is the use of schematic editor in Tina Pro?

Ans. It enhances your schematics by adding text and graphics elements such lines, arcs arrows, frames around the schematics and title blocks.

Q.6 How many manufacturer models it contain?

Ans. More than 20,000.

Q.7 Name all the Virtual Instruments used in it.

Ans. Virtual Instruments: Oscilloscope, Function Generator, Multimeter, Signal Analyzer/Bode Plotter, Network Analyzer, Spectrum Analyzer, Logic Analyzer, Digital Signal Generator, XY Recorder.

Q.8 What is Real-time Storage Oscilloscope?

Ans. It observes the actual time response of your circuit.

Q.9 What is Real-time Signal Analyzer?

Ans. It measure the frequency response of your circuit in real-time.

Q.10 What's role Tina play in Integrated PCB design?

Ans. The new fully integrated layout module of TINA has all the features you need for advanced PCB design, including multilayer PCB's with split power plane layers, powerful autoplacement & autorouting, rip-up and reroute, manual and "follow-me" trace placement, DRC, forward and back annotation, pin and gate swapping, keep-in and keep-out areas, copper pour, thermal relief, fanout, 3D view of your PCB design from any angle, Gerber file output and much more.

EXPERIMENT NO:12

AIM: Introduction to Layout Tool, and creating Layout board using TINAPRO.

APPARATUS: PC installed with TINAPRO

THEORY:

TINAPRO Layout

TINAPRO tool used for PCB routing and floor-planning

LAYOUT:

Layout is a circuit board layout tool that accepts a layout-compatible circuit net list (ex. from Capture CIS) and generates an output layout files that suitable for PCB fabrication

CREATING LAYOUT BOARD:

Having created the layout net list, the next step is to create a new board in Layout.

Launch Layout

Create the Layout board file

When you create a new board file in OrCAD Layout, you merge the electrical information from the layout net list (.MNL) and physical information from a template file (.TPL) or a technology file (.TCH) to create a new board design (.MAX). Therefore, to be able to create a board file for a new design in Layout, you need to provide a template file and a net list. A template (.TPL) file describes the characteristics of a physical board. A template can include information, such as the board outline, the design origin, the layer definitions, grid settings, spacing rules, and default track widths.

1. From the File menu in OrCAD Layout, choose New. The Auto ECO dialog box appears.
2. In the Input Layout TCH or TPL or MAX file text box, specify the name and the location of the technology file to be used for your board
3. In the Input MNL net list file text box, specify the location of the FULLADD.MNL created in the Creating Layout net list section.
4. From the drop-down list in the Options section, select Auto ECO.

CONCLUSION: Thus we have studied the TINAPRO circuit TINAPRO Layout and tool used for PCB routing and floor-planning.

QUIZ QUESTIONS & ANSWERS

Q.1 What is layout?

Ans. Layout is a circuit board layout tool that accepts a layout-compatible circuit netlist (ex. from Capture CIS) and generates an output layout files that suitable for PCB fabrication.

Q.2 What do you mean by SCHEMATIC?

Ans. A schematic is merely a collection of electronic symbols connected together with virtual "wires." The main reason you need a schematic when fabricating a printed circuit board is to provide input (a netlist) to your layout and routing tool.

Q.3 What is netlist?

Ans. A netlist is a file, usually ASCII text, which defines the connections between the components in your design.

Q.4 What's new in version 9.2 of Tina?

Ans. Schematic Symbol Editor (useable with the Macro Wizard) is included, so you can create your own symbols for imported SPICE macro models.

Q.5 What's new in version 9.3 of Tina?

Ans. It does not require active or non-linear components for analysis (so you can now run a circuit using just passives).

Q.6 What is user-friendly interface?

Ans. The user-friendly interface makes it easy to find the necessary tools for designing a circuit. There are several switches, meters, sources, semiconductors and spice macros to choose from. When evaluating a circuit in function mode, a multimeter, oscilloscope, a signal analyzer and others can also be used.

Q.7 What is the use of schematic editor in Tina Pro?

Ans. It enhances your schematics by adding text and graphics elements such as lines, arcs, arrows, frames around the schematics and title blocks.

Q.8 How many manufacturer models does it contain?

Ans. More than 20,000.

Q.9 What is a unique feature of TINA?

Ans. A unique feature of TINA is that you can bring your circuit to life with the optional USB controller.

Q.10 How many number of devices and nodes can be included in it?

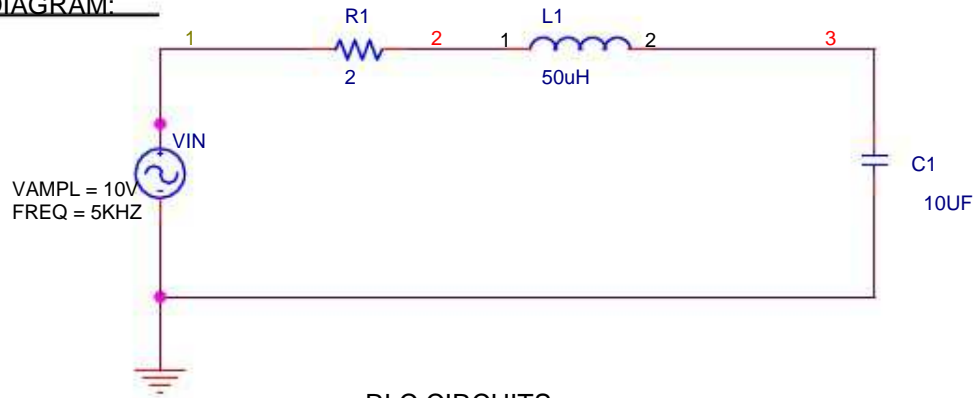
Ans. The number of devices and nodes that can be included in a circuit is not limited.

EXPERIMENT NO: 13

AIM: Design a RLC resonance circuit & verify the transient response for different values of R, L & C.

APPARATUS: PC installed with TINAPRO

CIRCUIT DIAGRAM:



RLC CIRCUITS

PROGRAM:

****Exp Transient Response of an RLC-circuit with a sinusoidal input voltage

* SIN (VO VA FREQ); Simple sinusoidal source

VIN 1 0 SIN (0 10V 5KHZ); Sinusoidal input voltage

R1 1 2 2

L1 2 3 50UH

C1 3 0 10UF

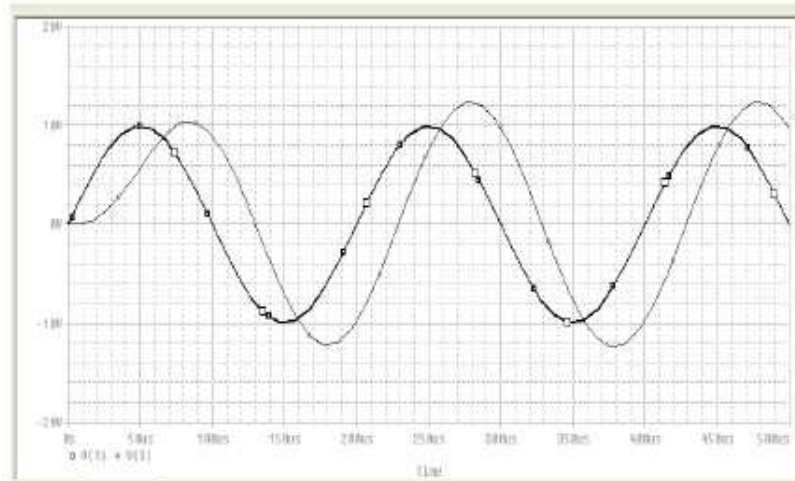
.TRAN 1US 500US; Transient analysis

.PLOT TRAN V (3) V (1) ; Plots on the output file

.PROBE; Graphical waveform analyzer

.END; End of circuit file

RESULT:



CONCLUSION: Thus we have studied transient response of RLC circuit for different values of R, L & C.

QUIZ QUESTIONS & ANSWERS

Q.1 What is layout?

Ans. Layout is a circuit board layout tool that accepts a layout-compatible circuit netlist (ex. from Capture CIS) and generates an output layout files that suitable for PCB fabrication.

Q.2 What do you mean by SCHEMATIC?

Ans. A schematic is merely a collection of electronic symbols connected together with virtual "wires." The main reason you need a schematic when fabricating a printed circuit board is to provide input (a netlist) to your layout and routing tool.

Q.3 What is netlist?

Ans. A netlist is a file, usually ASCII text, which defines the connections between the components in your design.

Q.4 What's new in version 9.2 of Tina?

Ans. Schematic Symbol Editor (useable with the Macro Wizard) is included, so you can create your own symbols for imported SPICE macro models.

Q.5 What's new in version 9.3 of Tina?

Ans. It does not require active or non-linear components for analysis (so you can now run a circuit using just passives).

Q.6 What is user-friendly interface?

Ans. The user-friendly interface makes it easy to find the necessary tools for designing a circuit. There are several switches, meters, sources, semiconductors and spice macros to choose from. When evaluating a circuit in function mode, a multimeter, oscilloscope, a signal analyzer and others can also be used.

Q.7 What is the use of schematic editor in Tina Pro?

Ans. It enhances your schematics by adding text and graphics elements such as lines, arcs, arrows, frames around the schematics and title blocks.

Q.8 How many manufacturer models does it contain?

Ans. More than 20,000.

Q.9 What is a unique feature of TINA?

Ans. A unique feature of TINA is that you can bring your circuit to life with the optional USB controller.

Q.10 How many number of devices and nodes can be included in it?

Ans. The number of devices and nodes that can be included in a circuit are not limited.

ELECTRONIC WORKSHOP, PCB DESIGN & CIRCUIT LAB

(EE-221-F)

LAB MANUAL
IIIrd SEMESTER

LIST OF EXPERIMENTS

EXP NO.	NAME OF THE EXPERIMENT	PAGE NO.
1	Introduction to circuit creation and simulation software TINAPRO	3
2	Introduction to Layout Tool, and creating Layout board using TINAPRO	7
3	Design a half wave rectifier using TINAPRO and observe its output on a virtual oscilloscope.	9
4	Design a full wave centre tapped rectifier using TINAPRO & its output on a virtual oscilloscope.	13
5	Design a clipper circuits using TINAPRO.	16
6	Design a clamper circuits using TINAPRO.	26
7	Design a RLC resonance circuit & verify the transient response for different values of R, L & C	33
8	Convert the power supply circuit into PCB & simulates its 2D & 3D view	37
9	Introduction of the materials required for the fabrication of PCB's	41
10	Development of PCB in hardware lab.	45

EXPERIMENT NO:1

AIM: Introduction to circuit creation and simulation software TINAPRO

APPARATUS: PC installed with TINAPRO.

THEORY: The study of TINAPRO mainly consists of:

Introduction to TINAPRO Capture:
Generation of Schematic
Generation of net list

SCHEMATIC:

A schematic is merely a collection of electronic symbols connected together with virtual “wires.” The main reason you need a schematic when fabricating a printed circuit board is to provide input (a net list) to your layout and routing tool.

NETLIST:

A net list is a file, usually ASCII text, which defines the connections between the components in your design.

CREATING PROJECT:

To create a new project, use Capture's Project Wizard. The Project Wizard provides you with the framework for creating any kind of project.

- 1 Launch Capture.
- 2 From the File menu, choose New > Project.
- 3 In the New Project dialog box, specify the project name
- 4 To specify the project type, select Analog or Mixed A/D.
- 5 Specify the location where you want the project files to be created and click OK.
- 6 In the Create PSpice Project dialog box, select the Create a blank project option button.

Adding parts:

To add parts to your design:

- 1 From the Place menu in Capture, select Part.
- 2 In the Place Part dialog box, first select the library from which the part is to be added and then instantiate the part on the schematic page. While working with the Capture, add parts from "Name of Library".OLB. To add libraries to the project, select the Add Library button.
- 3 Browse to <install_dir>/tools/capture/library/pspice/eval.olb.

Connecting parts:

After placing the required parts on the schematic page, you need to connect the parts.
From the Place menu, choose Wire.

Net List:

Save the schematic and close the schematic page.

Open tools and create net list.

Select Layout tab.

Be sure to put the *.MNL file in a unique folder.

Latter many more design files will be generated and it will be much easier to sort them out if the design is by itself.

Click on OK

The net list that Layout needs has been created. The file has the name of the project.MNL

CONCLUSION: Thus we have studied the TINAPRO circuit capturing, Generation of Schematic and net list..

QUIZ QUESTIONS & ANSWERS

Q. 1 What is TINA Design Suite?

Ans. TINA Design Suite is a powerful yet affordable circuit simulation and PCB design software package.

Q.2 Why TINA Design Suite is used?

Ans. It is used for analyzing, designing, and real time testing of analog, digital, VHDL, MCU, and mixed electronic circuits and their PCB layouts.

Q.3 What is unique feature of TINA?

Ans. A unique feature of TINA is that you can bring your circuit to life with the optional USB controlled.

Q.4 How it is used to compute computational power?

Ans. To meet this requirement TINA v9 has the ability to utilize the increasingly popular scalable multi-thread CPUs.

Q.5 What is the use of schematic editor in Tina Pro?

Ans. It enhances your schematics by adding text and graphics elements such lines, arcs arrows, frames around the schematics and title blocks.

Q.6 How many manufacturer models it contain?

Ans. More than 20,000.

Q.7 Name all the Virtual Instruments used in it.

Ans. Virtual Instruments: Oscilloscope, Function Generator, Multimeter, Signal Analyzer/Bode Plotter, Network Analyzer, Spectrum Analyzer, Logic Analyzer, Digital Signal Generator, XY Recorder.

Q.8 What is Real-time Storage Oscilloscope?

Ans. It observes the actual time response of your circuit.

Q.9 What is Real-time Signal Analyzer?

Ans. It measure the frequency response of your circuit in real-time.

Q.10 What's role Tina play in Integrated PCB design?

Ans. The new fully integrated layout module of TINA has all the features you need for advanced PCB design, including multilayer PCB's with split power plane layers, powerful autoplacement & autorouting, rip-up and reroute, manual and "follow-me" trace placement, DRC, forward and back annotation, pin and gate swapping, keep-in and keep-out areas, copper pour, thermal relief, fanout, 3D view of your PCB design from any angle, Gerber file output and much more.

EXPERIMENT NO:2

AIM: Introduction to Layout Tool, and creating Layout board using TINAPRO

APPARATUS: PC installed with TINAPRO

THEORY:

TINAPRO Layout

TINAPRO tool used for PCB routing and floor-planning

LAYOUT:

Layout is a circuit board layout tool that accepts a layout-compatible circuit netlist (ex. from Capture CIS) and generates an output layout files that suitable for PCB fabrication

CREATING LAYOUT BOARD:

Having created the layout netlist, the next step is to create a new board in Layout.

Launch Layout

Create the Layout board file

When you create a new board file in OrCAD Layout, you merge the electrical information from the layout netlist (.MNL) and physical information from a template file (.TPL) or a technology file (.TCH) to create a new board design (.MAX). Therefore, to be able to create a board file for a new design in Layout, you need to provide a template file and a netlist. A template (.TPL) file describes the characteristics of a physical board. A template can include information, such as the board outline, the design origin, the layer definitions, grid settings, spacing rules, and default track widths.

1. From the File menu in OrCAD Layout, choose New. The AutoECO dialog box appears.
2. In the Input Layout TCH or TPL or MAX file text box, specify the name and the location of the technology file to be used for your board
3. In the Input MNL netlist file text box, specify the location of the FULLADD.MNL created in the Creating Layout netlist section.
4. From the drop-down list in the Options section, select AutoECO.

CONCLUSION: Thus we have studied the TINAPRO circuit TINAPRO Layout and tool used for PCB routing and floor-planning

QUIZ QUESTIONS & ANSWERS

Q.1 What is layout?

Ans. Layout is a circuit board layout tool that accepts a layout-compatible circuit netlist (ex. from Capture CIS) and generates an output layout files that suitable for PCB fabrication.

Q.2 What do you mean by SCHEMATIC?

Ans. A schematic is merely a collection of electronic symbols connected together with virtual "wires." The main reason you need a schematic when fabricating a printed circuit board is to provide input (a netlist) to your layout and routing tool.

Q.3 What is netlist?

Ans. A netlist is a file, usually ASCII text, which defines the connections between the components in your design.

Q.4 What's new in version 9.2 of Tina?

Ans. Schematic Symbol Editor (useable with the Macro Wizard) is included, so you can create your own symbols for imported SPICE macro models.

Q.5 What's new in version 9.3 of Tina?

Ans. It does not require active or non-linear components for analysis (so you can now run a circuit using just passives).

Q.6 What is user-friendly interface?

Ans. The user-friendly interface makes it easy to find the necessary tools for designing a circuit. There are several switches, meters, sources, semiconductors and spice macros to choose from. When evaluating a circuit in function mode, a multimeter, oscilloscope, a signal analyzer and others can also be used.

Q.7 What is the use of schematic editor in Tina Pro?

Ans. It enhances your schematics by adding text and graphics elements such as lines, arcs, arrows, frames around the schematics and title blocks.

Q.8 How many manufacturer models does it contain?

Ans. More than 20,000.

Q.9 What is unique feature of TINA?

Ans. A unique feature of TINA is that you can bring your circuit to life with the optional USB controlled.

Q.10 How many number of devices and nodes can be included in it?

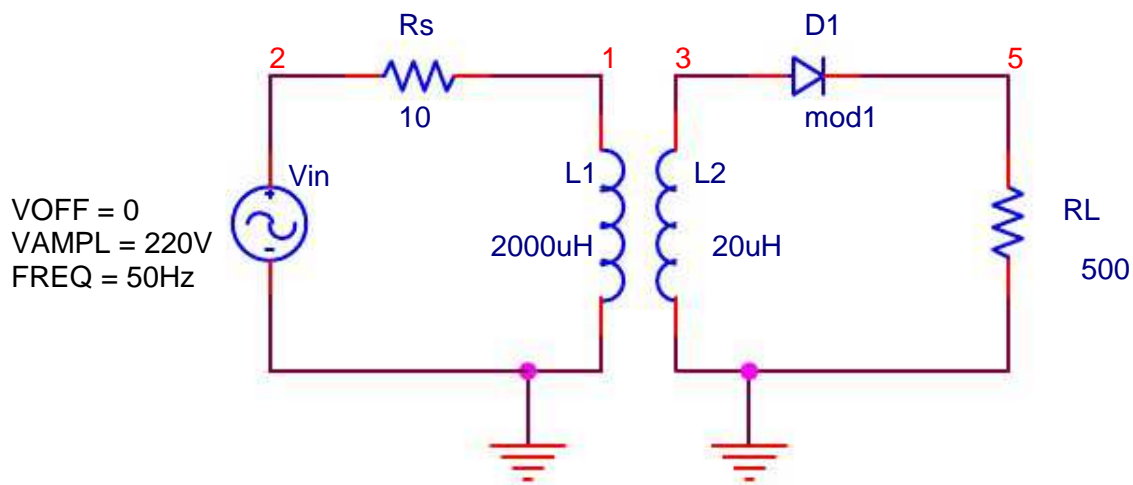
Ans. The number of devices and nodes that can be included in a circuit is not limited.

EXPERIMENT NO:3

AIM: Design a half wave rectifier using TINAPRO and observe its output on a virtual oscilloscope.

APPARATUS: PC installed with TINAPRO

CIRCUIT DIAGRAM:

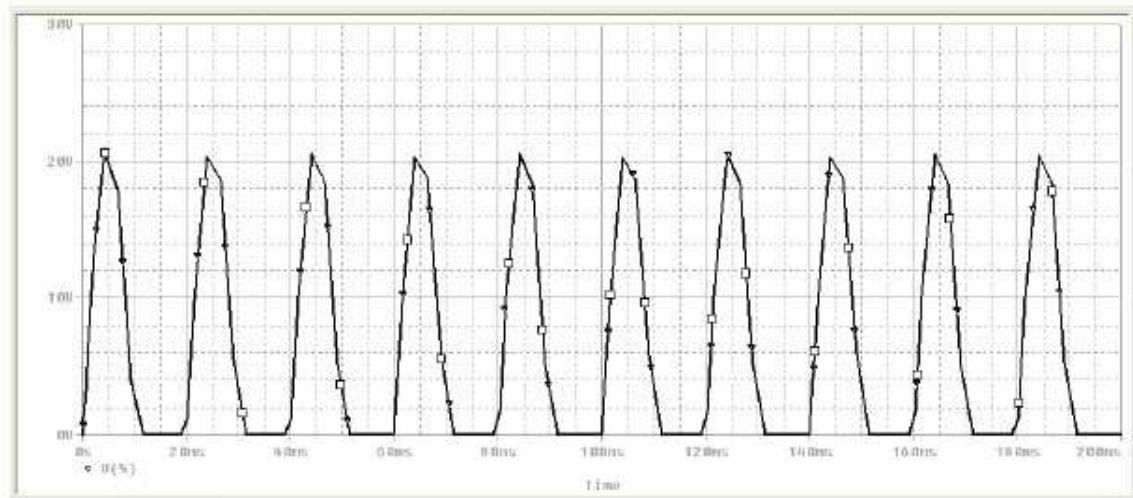


HALF WAVE RECTIFIER

PROGRAM:

```
*HALF WAVE rectifier
Vin 2 0 sin(0 220 50 )
RL 5 0 500
RS 2 1 10
L1 1 0 2000
L2 3 0 20
K1 L1 L2 0.99999
D1 3 5 mod1
.model mod1 D (IS=1e-14, n=1)
.tran 0.2m 200m
.plot tran v(3), v(5)
.probe
.end
```

RESULT:



HALF WAVE RECTIFIER

CONCLUSION: Thus we have studied the half wave rectifier using TINAPRO window and observed its output on the virtual CRO.

QUIZ QUESTIONS & ANSWERS

Q.1What is rectifier?

Ans. A rectifier is an electrical device that converts alternating current (AC), which periodically reverses direction, to direct current (DC), which flows in only one direction. The process is known as rectification.

Q.2 What is half wave rectifier?

Ans. In half wave rectification of a single-phase supply, either the positive or negative half of the AC wave is passed, while the other half is blocked.

Q.3 Relate input frequency and the output frequencies of a half wave rectifier and a full wave rectifier?

Ans. In half-wave rectification, we get pulsating output for half cycle only. Therefore output ripple frequency is same as the input frequency. But in full wave rectification we get output for both half cycle. Hence the output at ripple frequency is full-wave rectification is double the input frequency.

Q.4 What is the application of rectifier?

Ans. The primary application of rectifiers is to derive DC power from an AC supply. Virtually all electronic devices require DC, so rectifiers are used inside the power supplies of virtually all electronic equipment.

Q.5 What do you mean by forward biased?

Ans. When +ve terminal of battery is connected to P side & -ve terminal to N side of diode.

Q.6 What do you mean by reverse biased?

Ans. When +ve terminal of battery is connected to N side & -ve terminal to P side of diode.

Q.7 Define Knee voltage.

Ans. The forward voltage at which current through the junction starts increasing rapidly.

Q.8 Define breakdown voltage.

Ans. Reverse voltage at which PN junction breaks down with sudden rise in reverse current.

Q.9 Define max. Forward current.

Ans. It is highest instantaneous forward current that a PN junction can conduct without damage to Junction.

Q.10 What is the difference between active and passive components?

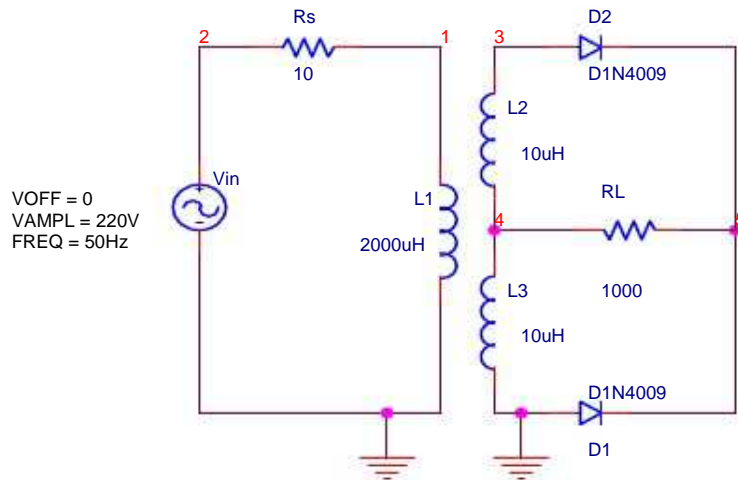
Ans. Passive elements don't require power from the supply to produce its effect on a signal. They derive the power of the input signal to perform its action. For example, a resistor doesn't require a separate supply to provide its action of resistance in a circuit. Whereas in active elements there should be a power source for its working. They require a supply for their working.

EXPERIMENT NO:4

AIM: Design a full wave centre tapped rectifier using TINAPRO & its output on a virtual oscilloscope.

APPARATUS: PC installed with TINAPRO

CIRCUIT DIAGRAM:

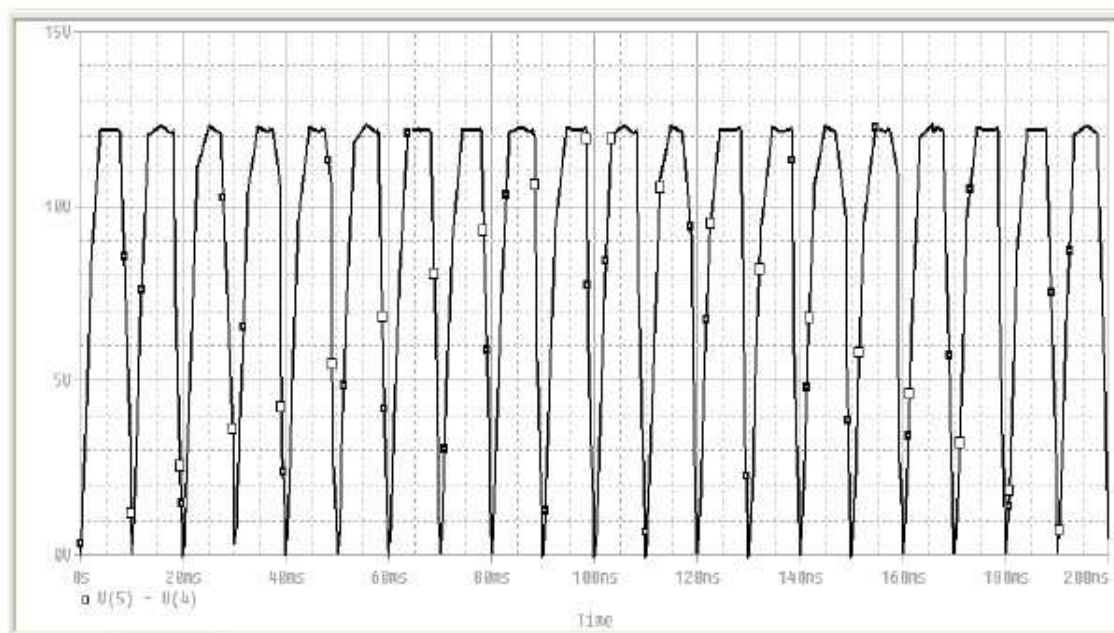


FULL WAVE RECTIFIER

PROGRAM:

```
*FULL WAVE rectifier
Vin 2 0 sin(0 230V 50HZ)
RL 5 4 1000
RS 2 1 10
L1 1 0 2000
L2 3 4 10
L3 4 0 10
K1 L1 L2 L3 0.99
D1 0 5 D1N4009
D2 3 5 D1N4009
.model D1N4009 D(Is=544.7E-21 N=1 Rs=.1 Ikf=0 Xti=3 Eg=1.11 Cjo=4p M=.3333
+Vj=.75 Fc=.5 Isr=30.77n Nr=2 Bv=25 Ibv=100u Tt=2.885n)
.tran 0.2ms 200ms
.probe
.end
```


RESULT:



FULL WAVE RECTIFIER

CONCLUSION: Thus we have studied the full wave rectifier using TINAPRO window and observed its output on the virtual CRO.

QUIZ QUESTIONS & ANSWERS

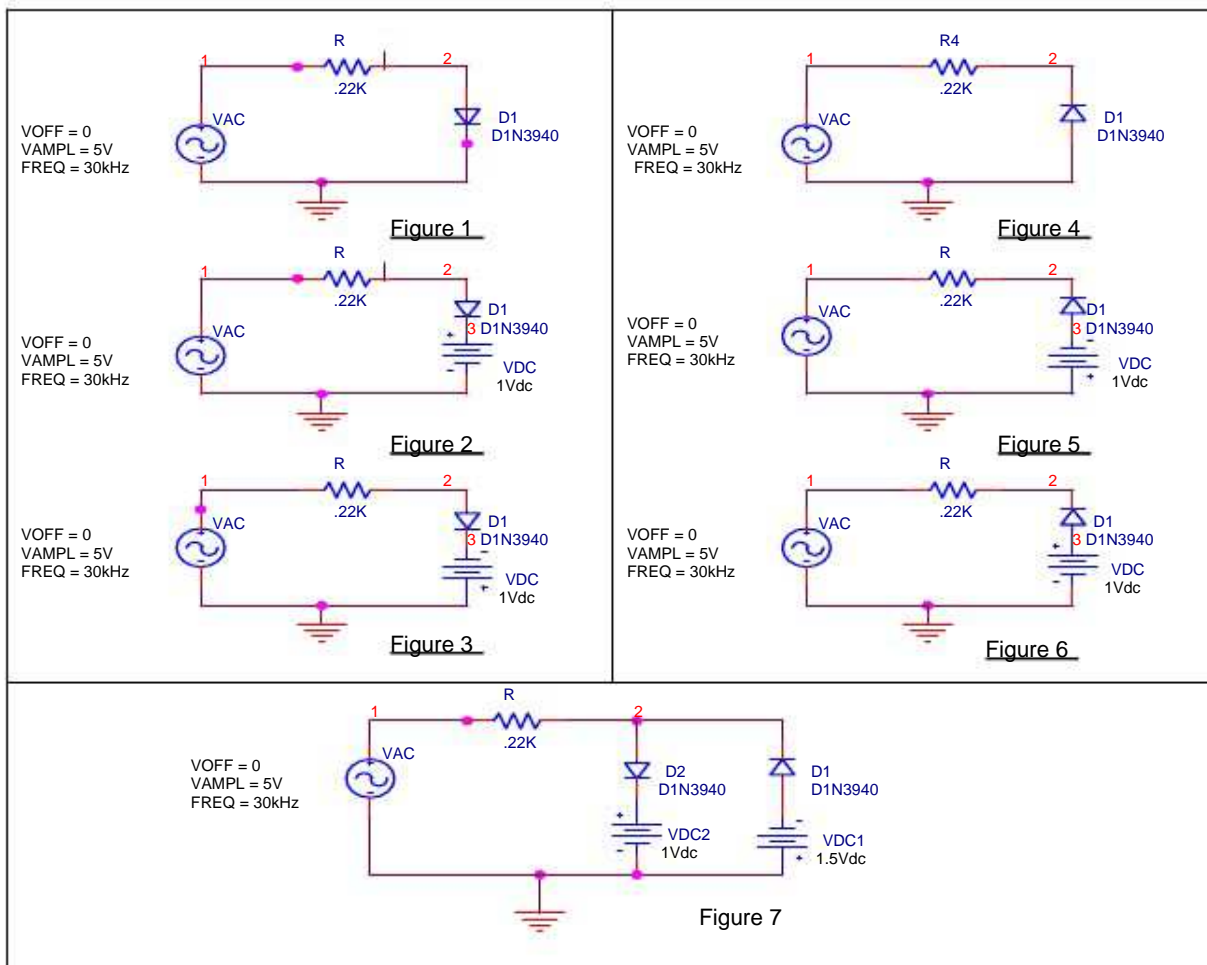
Q. No.	QUESTION	ANSWER
1	Define Full wave rectifier?	In both the half cycles current flows through the load in the same direction.
2	Which are different types of Full Wave rectifier?	Different types of full wave rectifier are Centre-Tap full wave rectifier & Bridge rectifier
3	How many no. of diodes are used in full wave rectifier?	4 No. of diodes are used for Bridge rectifier.
4	Give disadvantage of centre-Tap full wave rectifier?	Necessity of transformer with secondary winding.
5	Write ripple factor for FW rectifier?	The ripple factor for Full wave rectifier is 0.48.
6	What is the efficiency of FW rectifier?	Efficiency of full wave rectifier is 81.2%.
7	Write advantages of bridge rectifier?	Suitable for high-voltage applications.
8	Write one feature of Full wave rectifier?	The current drawn in both the primary & secondary of the supply transformer is Sinusoidal.
9	Define Transformer Utilization Factor?	Transformer Utilization Factor (TUF) is the ratio of d.c power to be delivered to the load to the a.c rating of the Transformer secondary.
10	What is DC current?	DC current is $I_{dc} = I_m / \pi$.

EXPERIMENT NO:5

AIM: Design a clipper circuits using TINAPRO.

APPARATUS: PC installed with TINAPRO

CIRCUIT DIAGRAM:



CLIPPER CIRCUITS

PROGRAM:

Prog. 1

```
*****CLIPPER 1
R 1 2 .22K
D1 2 0 D1N3940
VAC 1 0 SIN(0 5V 30KHZ)
.MODEL D1N3940 D(
+IS = 4E-10
+RS = .105
+N = 1.48
+TT = 8E-7
+CJO = 1.95E-11
+VJ = .4
+M = .38
+EG = 1.36
+XTI = -8
+KF = 0
+AF = 1
+FC = .9
+BV = 600
+IBV = 1E-4
+)
.PROBE
.TRAN 0US 100US
.END
```

Prog. 2

```
*****CLIPPER 1
R 1 2 .22K
D1 2 3 D1N3940
VAC 1 0 SIN(0 5V 30KHZ)
VDC 3 0 DC 1V
.MODEL D1N3940 D(
+IS = 4E-10
+RS = .105
+N = 1.48
+TT = 8E-7
+CJO = 1.95E-11
+VJ = .4
+M = .38
+EG = 1.36
+XTI = -8
```

```
+KF = 0
+AF = 1
+FC = .9
+BV = 600
+IBV = 1E-4
+)
.PROBE
.TRAN 0US 100US
.END
```

Prog. 3

```
*****CLIPPER 3
R 1 2 .22K
D1 2 3 D1N3940
VAC 1 0 SIN(0 5V 30KHZ)
VDC 0 3 DC 1V
.MODEL D1N3940 D(
+IS = 4E-10
+RS = .105
+N = 1.48
+TT = 8E-7
+CJO = 1.95E-11
+VJ = .4
+M = .38
+EG = 1.36
+XTI = -8
+KF = 0
+AF = 1
+FC = .9
+BV = 600
+IBV = 1E-4
+)
.PROBE
.TRAN 0US 100US
.END
```

Prog. 4

```
*****CLIPPER 4
R 1 2 .22K
D1 0 2 D1N3940
VAC 1 0 SIN(0 5V 30KHZ)
.MODEL D1N3940 D(
+IS = 4E-10
```

```
+RS = .105
+N = 1.48
+TT = 8E-7
+CJO = 1.95E-11
+VJ = .4
+M = .38
+EG = 1.36
+XTI = -8
+KF = 0
+AF = 1
+FC = .9
+BV = 600
+IBV = 1E-4
+)
.PROBE
.TRAN 0US 100US
.END
```

Prog. 5

```
*****CLIPPER 5
R 1 2 .22K
D1 3 2 D1N3940
VAC 1 0 SIN(0 5V 30KHZ)
VDC 0 3 DC 1V
.MODEL D1N3940 D(
+IS = 4E-10
+RS = .105
+N = 1.48
+TT = 8E-7
+CJO = 1.95E-11
+VJ = .4
+M = .38
+EG = 1.36
+XTI = -8
+KF = 0
+AF = 1
+FC = .9
+BV = 600
+IBV = 1E-4
+)
.PROBE
.TRAN 0US 100US
.END
```

Prog. 6

*****CLIPPER 6

```
R 1 2 .22K
D1 3 2 D1N3940
VAC 1 0 SIN(0 5V 30KHZ)
VDC 3 0 DC 1V
.MODEL D1N3940 D(
+IS = 4E-10
+RS = .105
+N = 1.48
+TT = 8E-7
+CJO = 1.95E-11
+VJ = .4
+M = .38
+EG = 1.36
+XTI = -8
+KF = 0
+AF = 1
+FC = .9
+BV = 600
+IBV = 1E-4
+)
.PROBE
.TRAN 0US 100US
.END
```

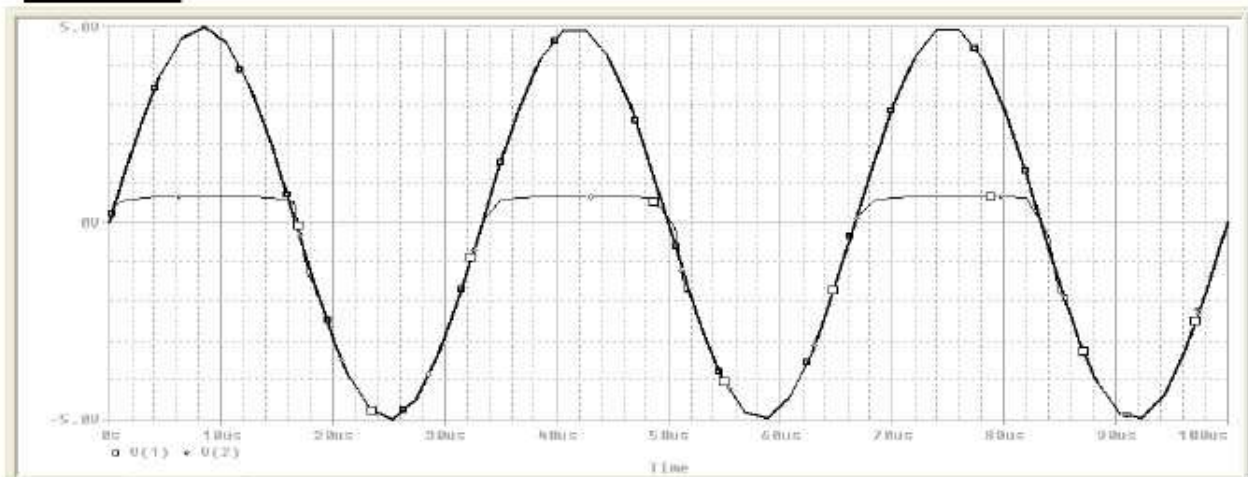
Prog. 7

*****CLIPPER 7

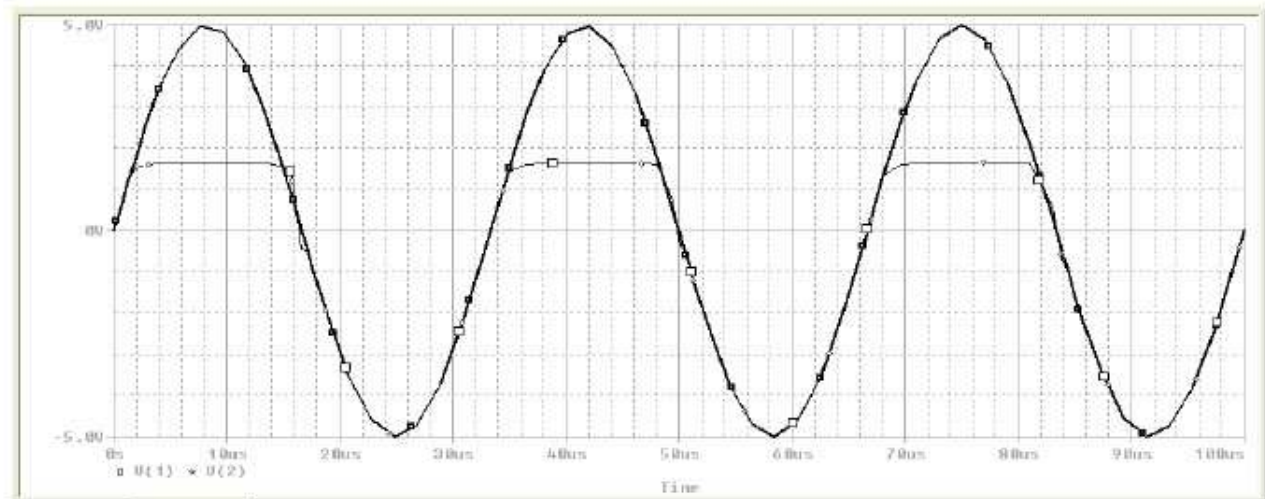
```
R 1 2 .22K
D1 3 2 D1N3940
D2 2 4 D1N3940
VAC 1 0 SIN(0 5V 30KHZ)
V1_VDC1 4 0 DC 1V
V2_VDC2 0 3 DC 1.5V
.MODEL D1N3940 D(
+IS = 4E-10
+RS = .105
+N = 1.48
+TT = 8E-7
+CJO = 1.95E-11
+VJ = .4
+M = .38
```

```
+EG = 1.36  
+XTI = -8  
+KF = 0  
+AF = 1  
+FC = .9  
+BV = 600  
+IBV = 1E-4  
+)  
.PROBE  
.TRAN 0US 100US  
.END
```

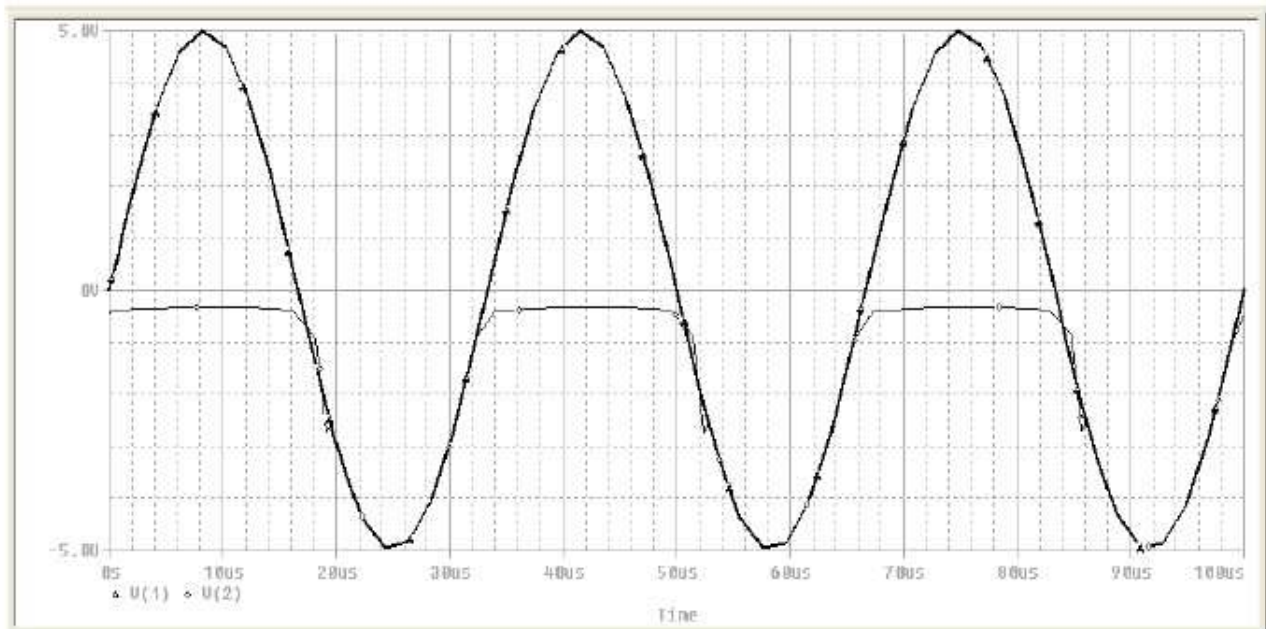
RESULT:



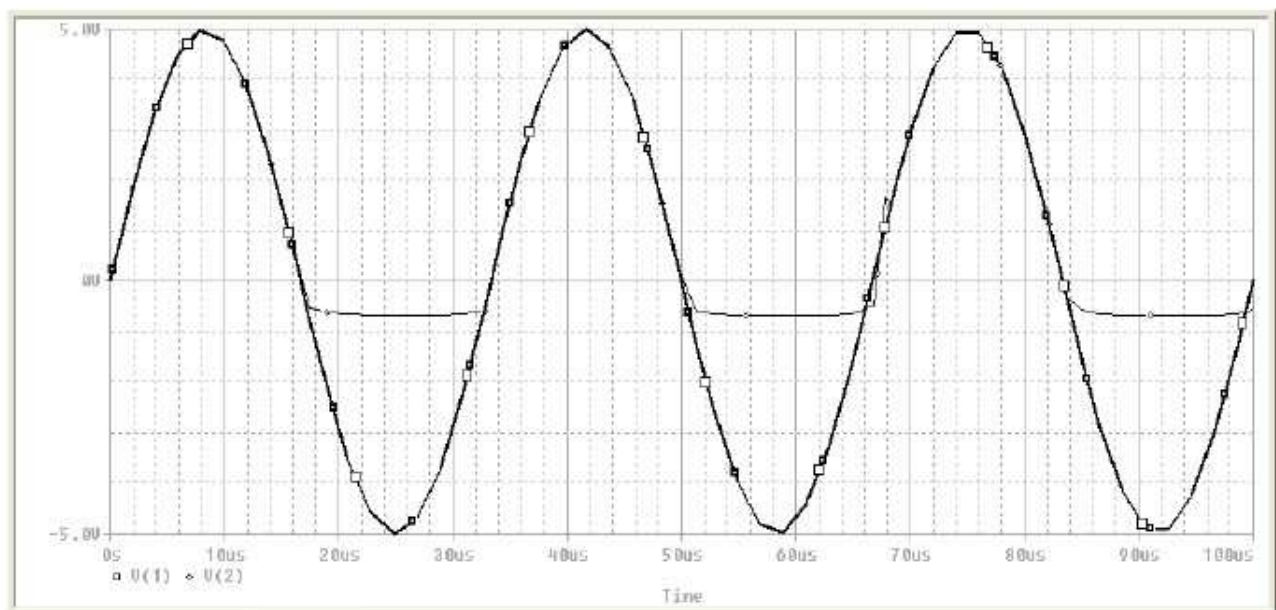
Result of CLIPPER 1



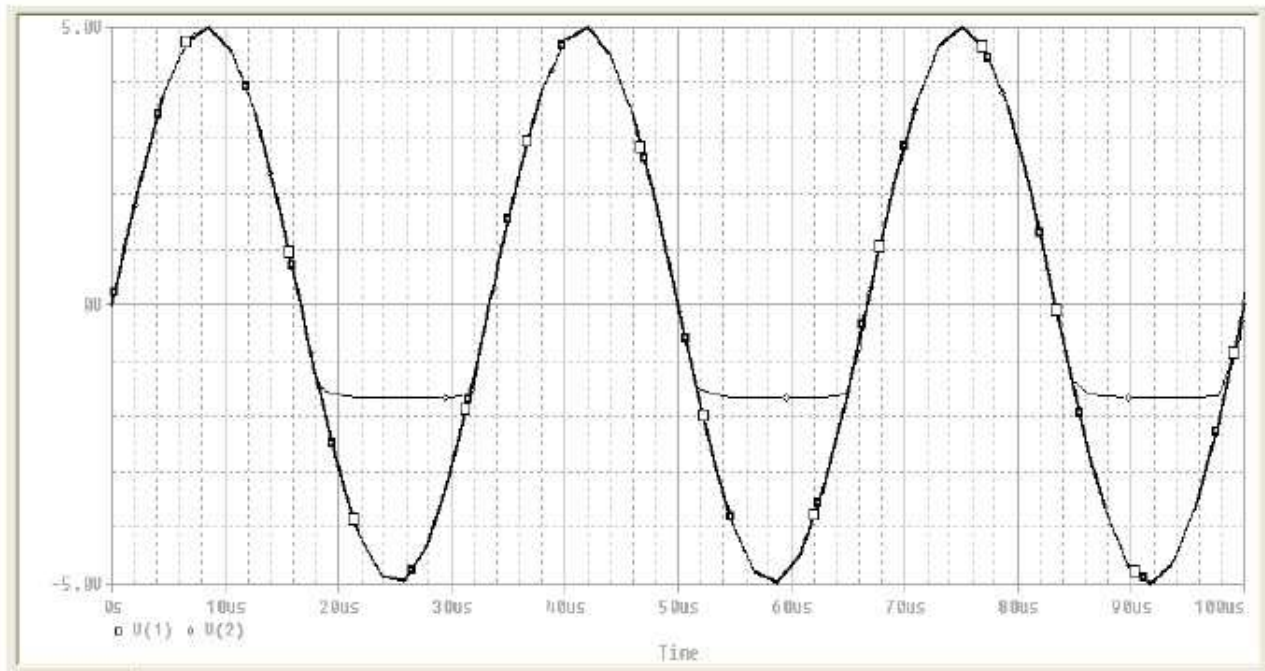
Result of CLIPPER 2



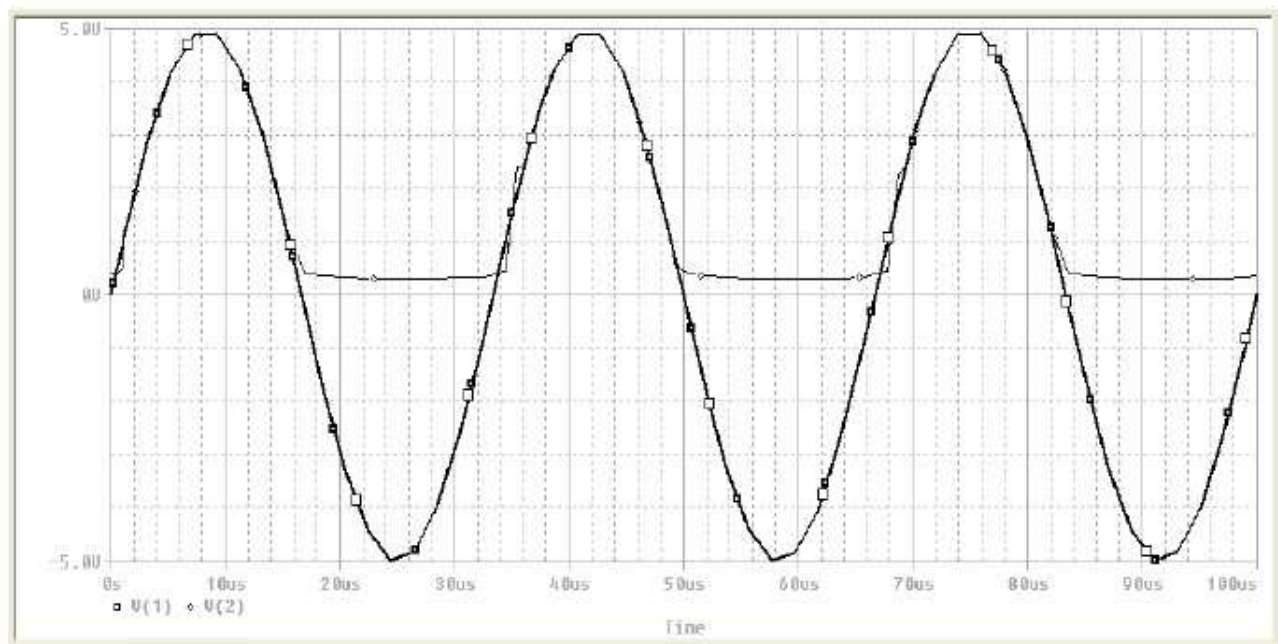
Result of CLIPPER 3



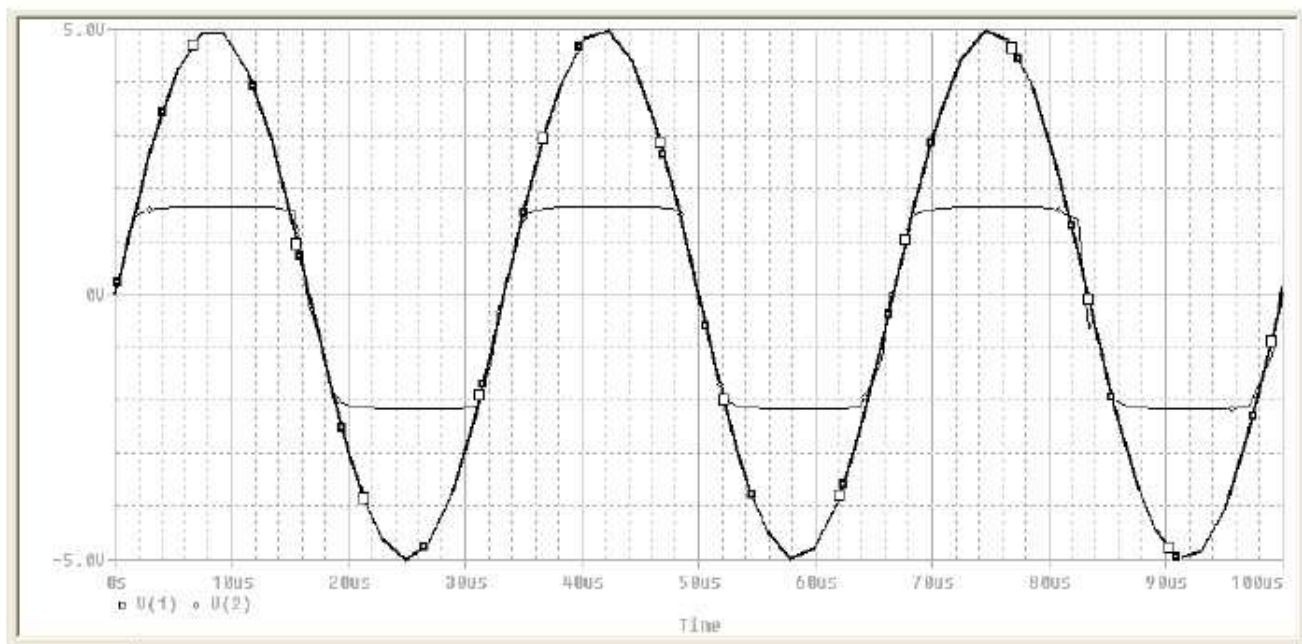
Result of CLIPPER 4



Result of CLIPPER 5



Result of CLIPPER 6



Result of CLIPPER 7

CONCLUSION: Thus we have studied the full wave rectifier using TINAPRO window and observed its output on the virtual CRO.

QUIZ QUESTIONS & ANSWERS

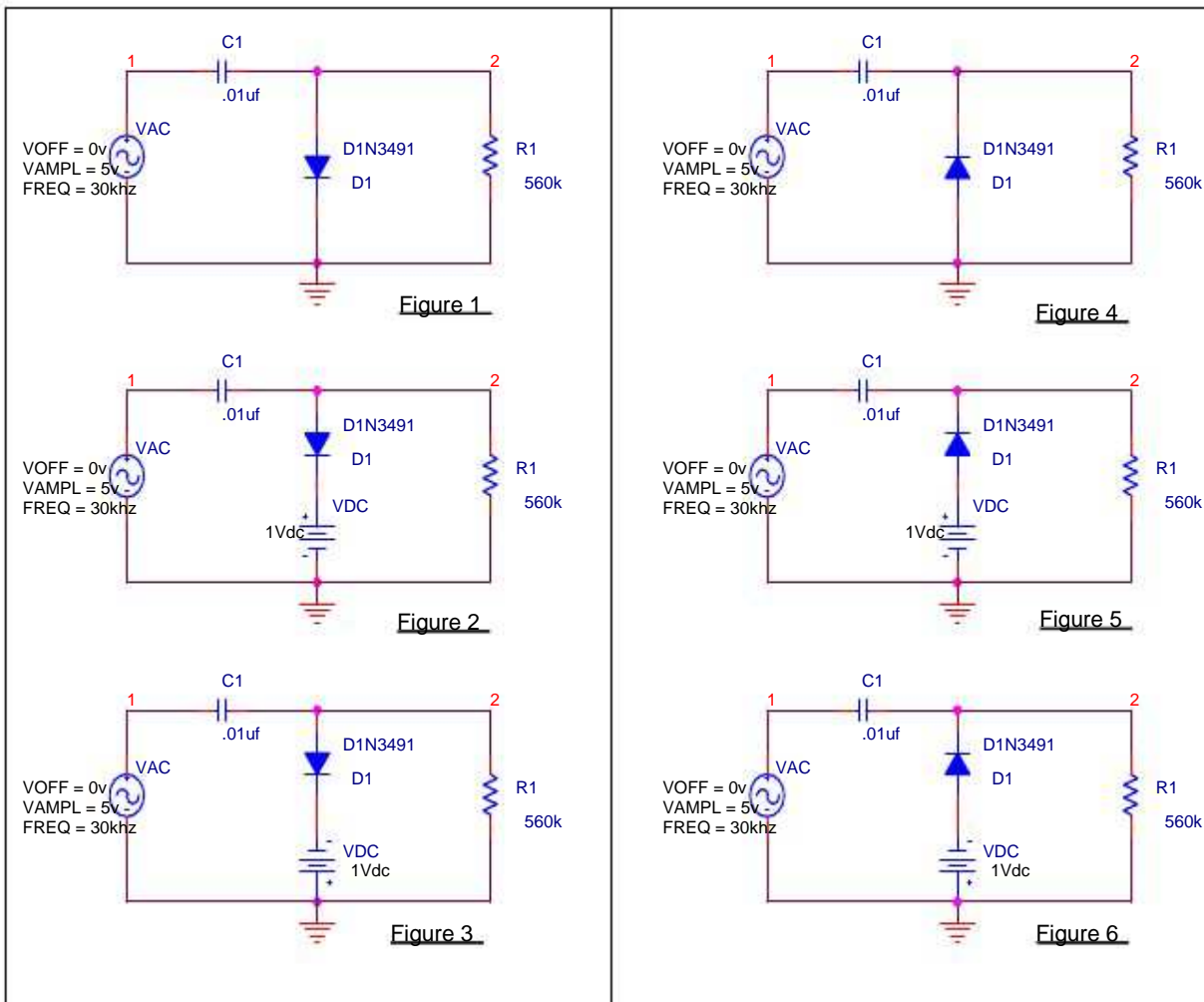
Q. No	QUESTION	ANSWER
1.	What is non-linear wave shaping?	Non linear wave shaping is the process, on applying any wave at input of a non-linear device the shape of the output wave varies non-linearly with the input wave.
2.	Which are the circuits for non- linear wave shaping?	Clipping circuit & Clamping circuits are the circuits for non-linear wave shaping.
3.	What is clipping circuit ?	A wave-shaping circuit which controls the output waveform by removing or clipping a portion of the applied wave is known as clipping circuit.
4.	According to non-linear devices how clippers can be classified.	According to non-linear devices clippers can be classified as diode clippers & transistor clippers.
5.	According to configuration used classify clippers.	According to configuration used classify clippers can be classified as a) Series diode clipper. b) Parallel or shunt diode clipper. c) Combination clippers.
6.	Classify clippers according to level of clippers.	According to level of clipping the clippers may be a) Positive clippers. b) Negative clippers c) Biased clippers d) Combinational clippers.
7.	What is positive clipper circuit?	Positive clipper is one which removes the positive half cycles of the input voltage
8.	What is negative clipper circuit?	Negative clipper is one which removes the negative half cycles of the input voltage
9.	What is clamping?	A circuit that places either the positive or negative peak of a signal at a desired level is known as Clamping circuit.
10.	How many types of clammers are there?	There are 2 types of clammers a) Positive clamper. b) Negative clamper

EXPERIMENT NO:6

AIM: Design a clamper circuits using TINAPRO.

APPARATUS: PC installed with TINAPRO

CIRCUIT DIAGRAM:



CLAMPER CIRCUITS

PROGRAM:

Prog. 1

```
****CLAMPER1
C_C1 1 2 .01uf
R_R1 2 0 560k
D_D1 2 0 D1N3491
V_VAC 1 0 sin(0 5v 30khz)
.model D1N3491D(Is=68.65f Rs=3.786m Ikf=1.774 N=1 Xti=2 Eg=1.11 Cjo=1.457n
+M=.9735 Vj=.75 Fc=.5 Isr=11.02u Nr=2 Tt=6.059u)
*Motorolapid=1N3491 case=DO21
*88-08-24 rmn
.probe
.tran 0us 100us
.end
```

Prog. 2

```
****CLAMPER2
C_C1 1 2 .01uf
R_R1 2 0 560k
D_D1 2 3 D1N3491
V_VDC 3 0 1V
V_VAC 1 0 sin(0 5v 30khz)
.model D1N3491D(Is=68.65f Rs=3.786m Ikf=1.774 N=1 Xti=2 Eg=1.11 Cjo=1.457n
+M=.9735 Vj=.75 Fc=.5 Isr=11.02u Nr=2 Tt=6.059u)
*Motorolapid=1N3491 case=DO21
*88-08-24 rmn
.probe
.tran 0us 100us
.end
```

Prog. 3

```
****CLAMPER3
C_C1 1 2 .01uf
R_R1 2 0 560k
D_D1 2 3 D1N3491
V_VDC 0 3 1V
V_VAC 1 0 sin(0 5v 30khz)
.model D1N3491D(Is=68.65f Rs=3.786m Ikf=1.774 N=1 Xti=2 Eg=1.11 Cjo=1.457n
+M=.9735 Vj=.75 Fc=.5 Isr=11.02u Nr=2 Tt=6.059u)
*Motorolapid=1N3491 case=DO21
*88-08-24 rmn
```

```
.probe  
.tran 0us 100us  
.end
```

Prog. 4

```
****CLAMPER4  
C_C1 1 2 .01uf  
R_R1 2 0 560k  
D_D1 0 2 D1N3491  
V_VAC 1 0 sin(0 5v 30khz)  
.model D1N3491D(Is=68.65f Rs=3.786m Ikf=1.774 N=1 Xti=2 Eg=1.11 Cjo=1.457n  
+M=.9735 Vj=.75 Fc=.5 Isr=11.02u Nr=2 Tt=6.059u)  
*Motorolapid=1N3491 case=DO21  
*88-08-24 rmn  
.probe  
.tran 0us 100us  
.end
```

Prog. 5

```
****CLAMPER5  
C_C1 1 2 .01uf  
R_R1 2 0 560k  
D_D1 3 2 D1N3491  
V_VDC 3 0 1V  
V_VAC 1 0 sin(0 5v 30khz)  
.model D1N3491D(Is=68.65f Rs=3.786m Ikf=1.774 N=1 Xti=2 Eg=1.11 Cjo=1.457n  
+M=.9735 Vj=.75 Fc=.5 Isr=11.02u Nr=2 Tt=6.059u)  
*Motorolapid=1N3491 case=DO21  
*88-08-24 rmn  
.probe  
.tran 0us 100us  
.end
```

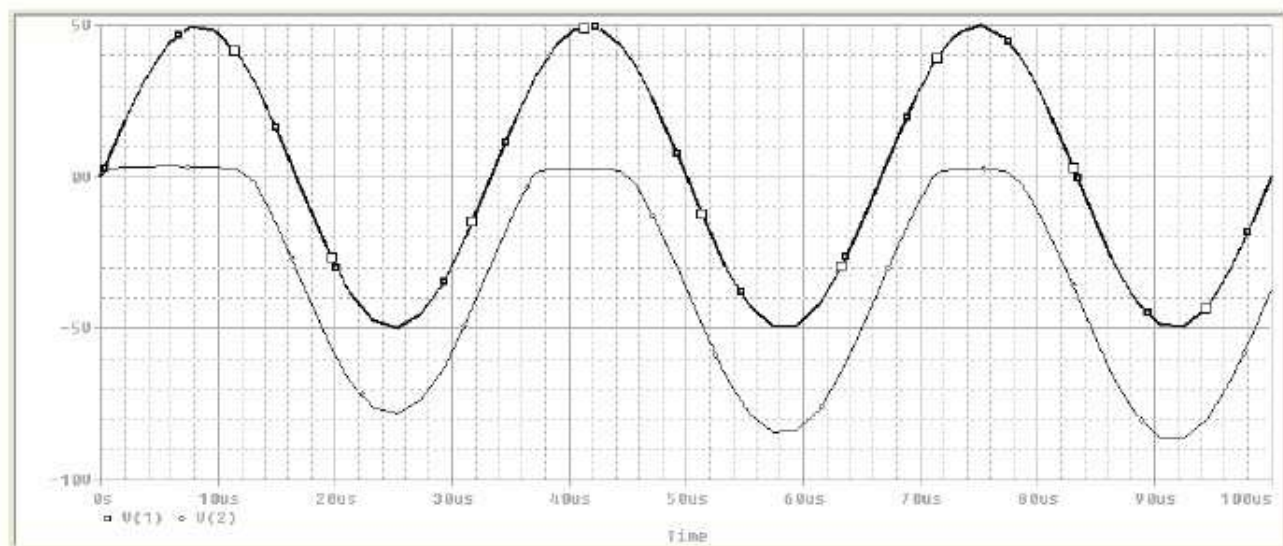
Prog. 6

```
****CLAMPER6  
C_C1 1 2 .01uf  
R_R1 2 0 560k  
D_D1 3 2 D1N3491  
V_VDC 0 3 1V  
V_VAC 1 0 sin(0 5v 30khz)  
.model D1N3491D(Is=68.65f Rs=3.786m Ikf=1.774 N=1 Xti=2 Eg=1.11 Cjo=1.457n  
+M=.9735 Vj=.75 Fc=.5 Isr=11.02u Nr=2 Tt=6.059u)  
*Motorolapid=1N3491 case=DO21
```

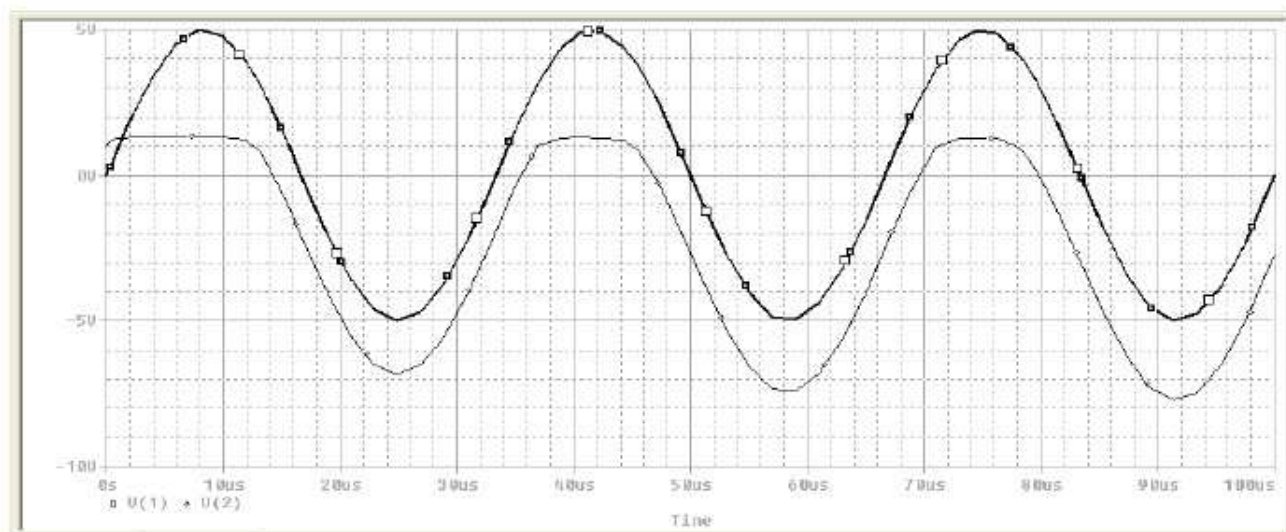


```
*88-08-24 rmn  
.probe  
.tran 0us 100us  
.end
```

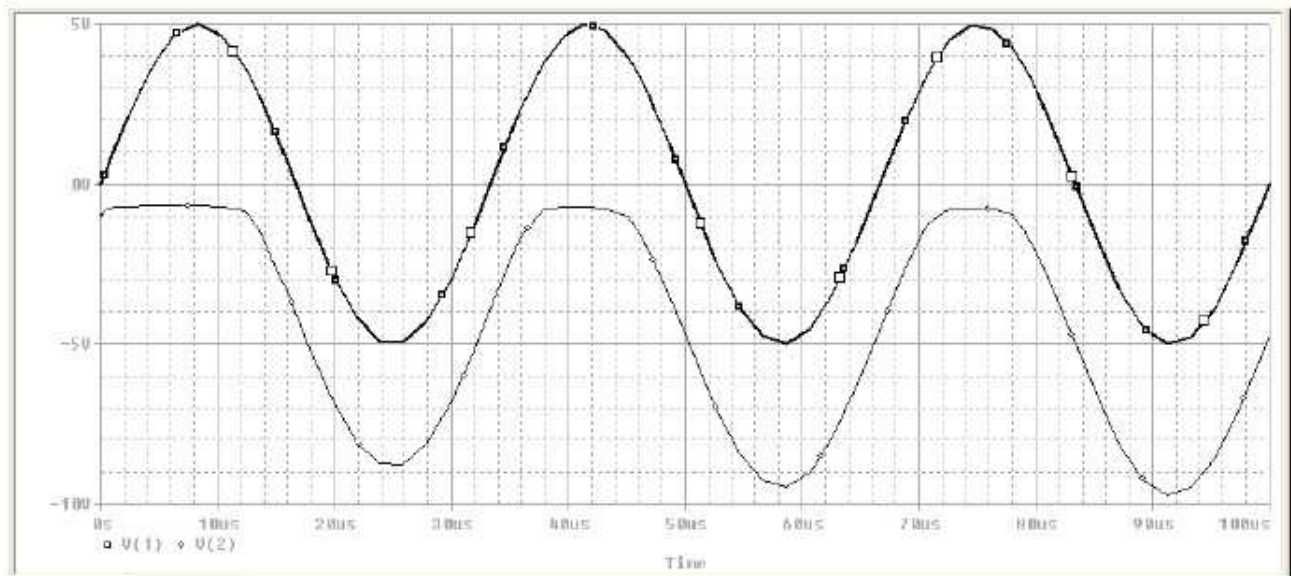
RESULT:



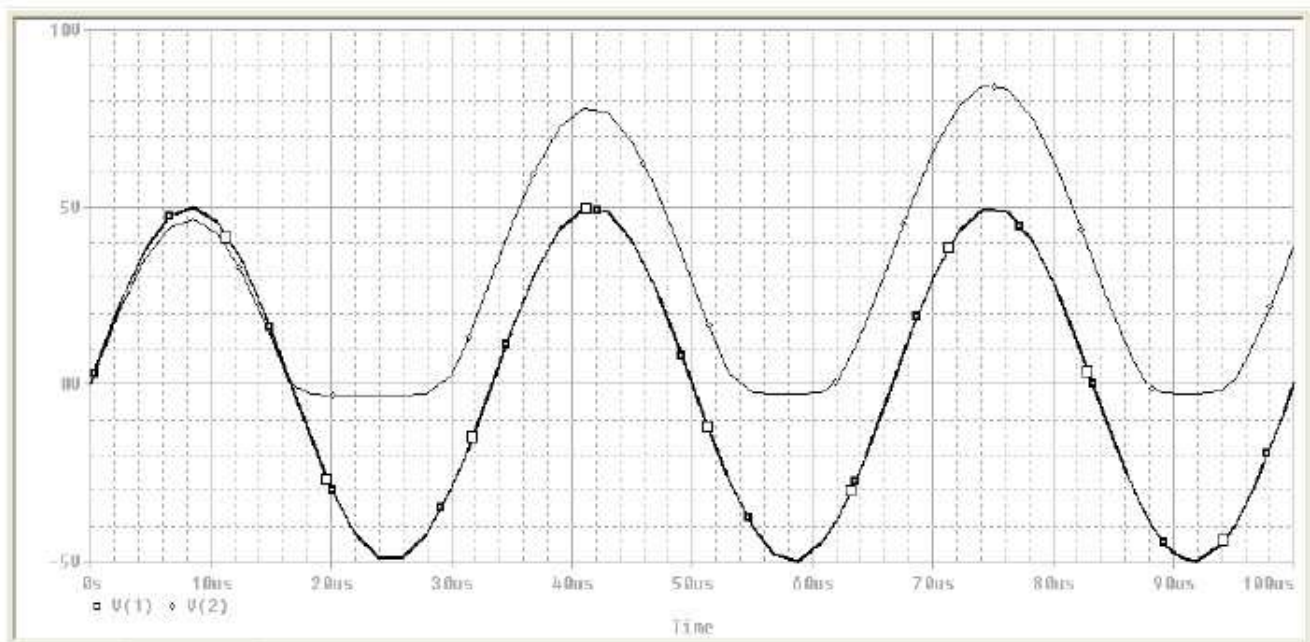
Result of CLAMPER1



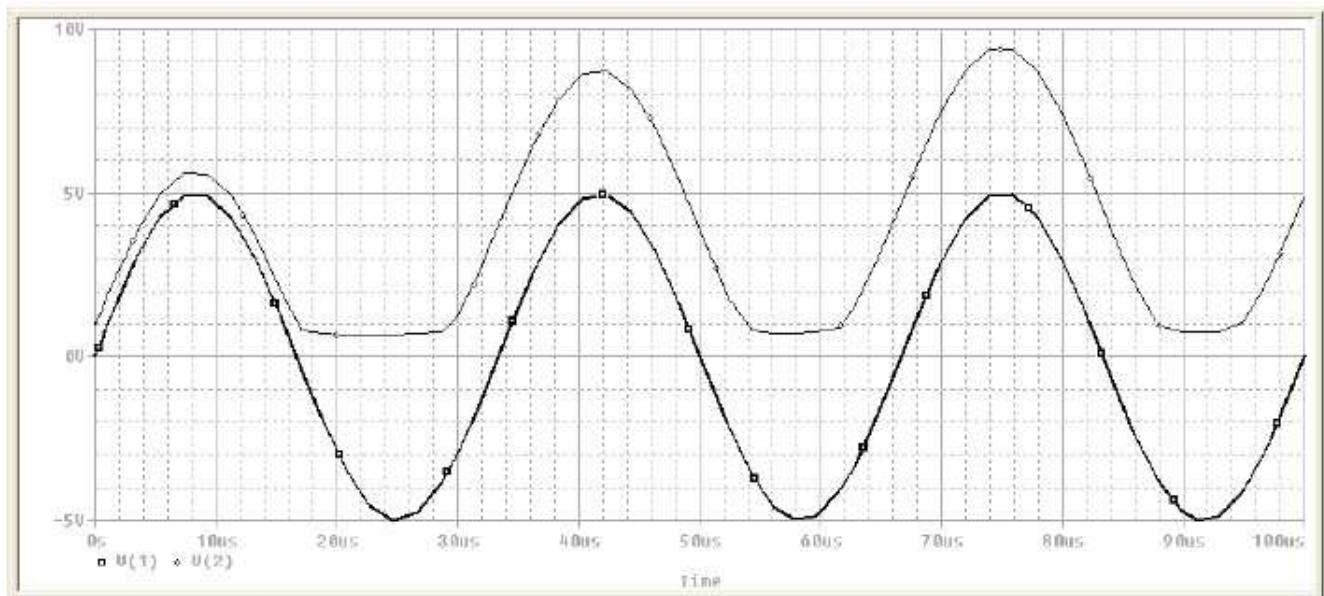
Result of CLAMPER2



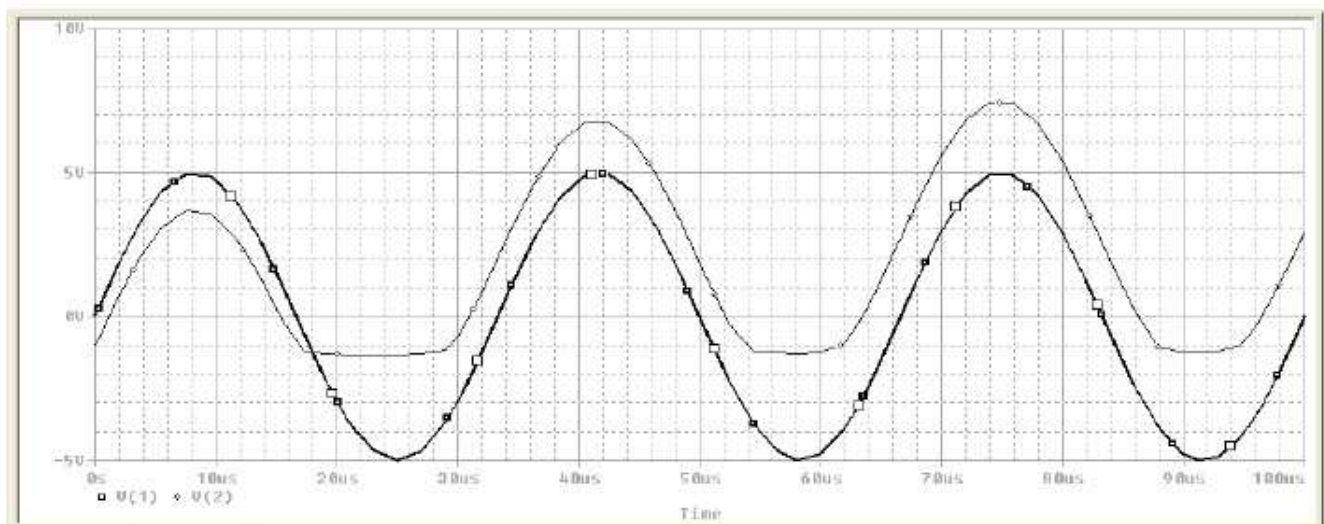
Result of CLAMPER3



Result of CLAMPER4



Result of CLAMPER5



Result of CLAMPER6

CONCLUSION: Thus we have studied the clamper circuits using TINAPRO window and observed its output on the virtual CRO.

QUIZ QUESTIONS & ANSWERS

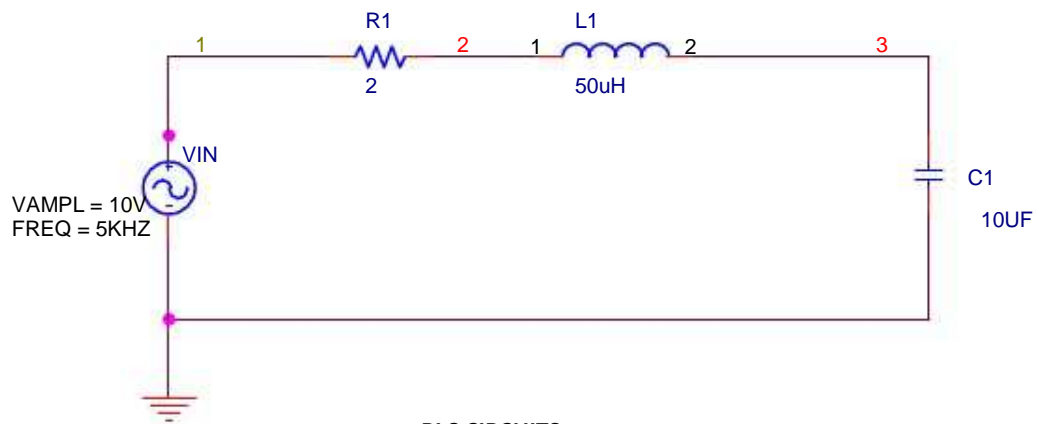
Q. No	QUESTION	ANSWER
1.	What is non-linear wave shaping?	Non linear wave shaping is the process, on applying any wave at input of a non-linear device the shape of the output wave varies non-linearly with the input wave.
2.	Which are the circuits for non- linear wave shaping?	Clipping circuit & Clamping circuits are the circuits for non-linear wave shaping.
3.	What is clipping circuit ?	A wave-shaping circuit which controls the output waveform by removing or clipping a portion of the applied wave is known as clipping circuit.
4.	According to non-linear devices how clippers can be classified.	According to non-linear devices clippers can be classified as diode clippers & transistor clippers.
5.	According to configuration used classify clippers.	According to configuration used classify clippers can be classified as a) Series diode clipper. b) Parallel or shunt diode clipper. c) Combination clippers.
6.	Classify clippers according to level of clippers.	According to level of clipping the clippers may be a) Positive clippers. b) Negative clippers c) Biased clippers d) Combinational clippers.
7.	What is positive clipper circuit?	Positive clipper is one which removes the positive half cycles of the input voltage
8.	What is negative clipper circuit?	Negative clipper is one which removes the negative half cycles of the input voltage
9.	What is clamping?	A circuit that places either the positive or negative peak of a signal at a desired level is known as Clamping circuit.
10.	How many types of clammers are there?	There are 2 types of clammers a) Positive clamper. b) Negative clamper

EXPERIMENT NO:7

AIM: Design a RLC resonance circuit & verify the transient response for different values of R, L & C

APPARATUS: PC installed with TINAPRO

CIRCUIT DIAGRAM:



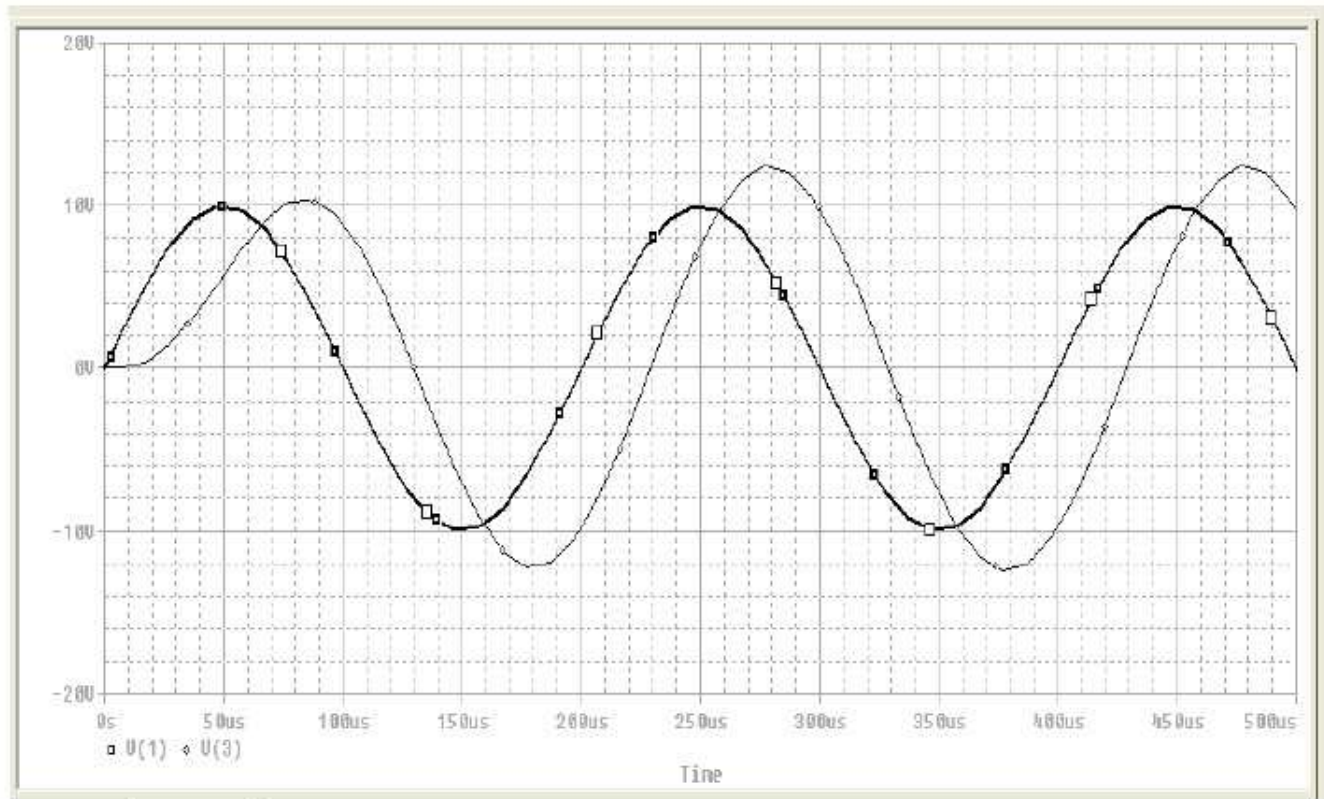
RLC CIRCUITS

PROGRAM:

****Exp Transient Response of an RLC-circuit with a sinusoidal input voltage

```
* SIN (VO VA FREQ); Simple sinusoidal source
VIN 1 0 SIN (0 10V 5KHZ) ; sinusoidal input voltage
R1 1 2 2
L1 2 3 50UH
C1 3 0 10UF
.TRAN 1US 500US; Transient analysis
.PLOT TRAN V(3) V(1); Plots on the output file
.PROBE; Graphical waveform analyzer
.END; End of circuit file
```

RESULT:



CONCLUSION: Thus we have studied transient response of RLC circuit for different values of R, L & C

QUIZ QUESTIONS & ANSWERS

Q1 What is RLC circuit?

Ans. An RLC circuit (or LCR circuit) is an electrical circuit consisting of a resistor, an inductor, and a capacitor, connected in series or in parallel.

Q2 Why it is called RLC?

Ans. The RLC part of the name is due to those letters being the usual electrical symbols for resistance, inductance and capacitance respectively.

Q.3 Name application of RLC.

Ans. There are many applications for this circuit. They are used in many different types of oscillator circuit. Another important application is for tuning, such as in radio receivers or television sets.

Q4. What is relation between angular frequency and resonance frequency?

Ans. An important property of this circuit is its ability to resonate at a specific frequency, the resonance frequency, f_0 . Frequencies are measured in units of hertz. In this article, however, angular frequency, ω , is used which is more mathematically convenient. This is measured in radians per second. They are related to each other by a simple proportion,

$$\omega_0 = 2\pi f_0$$

Q5 What is resonance frequency?

Ans. The resonance frequency is defined as the frequency at which the impedance of the circuit is at a minimum.

Q6 What is Damping?

Ans. Damping is caused by the resistance in the circuit. It determines whether or not the circuit will resonate naturally (that is, without a driving source).

Q7 What is Q factor?

Ans. The Q factor is a widespread measure used to characterise resonators. It is defined as the peak energy stored in the circuit divided by the average energy dissipated in it per cycle at resonance.

Q8 What do you mean by transient response?

Ans. Transient response or natural response is the response of a system to a change from equilibrium. The transient response is not necessarily tied to "on/off" events but to any event that affects the equilibrium of the system.

Q.9 What is Rise time?

Ans. Rise time refers to the time required for a signal to change from a specified low value to a specified high value. Typically, these values are 10% and 90% of the step height.

Q. 10 What is Overshoot?

Ans. Overshoot is when a signal or function exceeds its target. It is often associated with ringing.

EXPERIMENT NO:8

AIM: Convert the power supply circuit into PCB & simulates its 2D & 3D view.

APPARATUS: PC installed with TINAPRO.

THEORY:

Creating a PCB for the circuit simulated in the software involves certain basic steps.

Setting and checking footprint names:

To achieve an accurate PCB design, one that is easy to build, every part in your schematic must have a physical representation with exact physical size. This is realized through so-called footprints: drawings showing the outline and the pins of the parts. Default footprint names to all parts are already defined that represent real components. Some parts used for theoretical investigations, controlled sources, for example, do not represent real physical parts and cannot be placed on a PCB. If your design contains such components, you should replace them with real physical parts.

Of course there is no guarantee that the default physical packages parts are the same as needed by your design.

There are two ways to check this.

- 1) You can use "Footprint name editor", which you can invoke from Tools menu. In this dialog you can see all of components and their corresponding footprint names.
- 2) The second way to examine the assigned footprints is to double-click on each part and check the Footprint Name in the component property dialog that appears.

2D/3D view checking:

After completing the checking of the footprint names check the 2D/3D shape by clicking the 2D/3D view button or the F6 key. Unless a component is only meaningful for analysis, it will have a 3D view. If the physical part association is OK, we can begin the PCB layout design.

PCB Designing :

To begin PCB design, select the "PCB Design" command on the Tools menu. Set the parameters as required. Select "Start New Project," "Auto placement" and "Use board template." With the Browse button find and select the 2layer_A.tpt template file from Template folder. When starting with a template, you are choosing the level of manufacturing complexity of your project. The following three levels of manufacturing technology are defined by the IPC-2221 generic standard:

Level A : General Design Complexity
Level B : Moderate Design Complexity
Level C : High Design Complexity

The template file specifies the number of layers, including their properties, system grid size, auto router settings, spacing and track width.

In choosing a PCB template, you should take into consideration technology, density, and package pitch. To complete the set up, set the PCB size in inches or mm depending on the measurement unit settings in the View/Options dialog. Now that everything is set properly press the OK button and the PCB layout design will appear with all the components automatically placed on the PCB board. While all the parts and nets are placed, we need to adjust their positions for good placement and easier routing.

Press F4 to reach the Net Editor and set net routing width. First, click on "Modify all" and enter 12.5 into "Track width" field. Then select power nets (Ground, VCC, -VCC) and set their width to 25mil. To automatically route the PCB, press the F5 button or select "Autoroute board" command from the Tools menu.

Finally, check your design in full 3D. Press F3 or select 3D View from the View menu. After some calculation, 3D view appears on the screen.

Lastly take the print out of the PCB generated on the trace paper to develop the PCB in hardware lab.

CONCLUSION: Thus we have obtained the print of the power supply PCB to develop the PCB in hardware lab.

QUIZ QUESTIONS & ANSWERS

Q.1 What do you mean by 2D/3D view checking?

Ans. 2D/3D view checking: After completing the checking of the footprint names check the 2D/3D shape by clicking the 2D/3D view button or the F6 key. Unless a component is only meaningful for analysis, it will have a 3D view. If the physical part association is OK, we can begin the PCB layout design.

Q.2 What's role Tina play in Integrated PCB design?

Ans. The new fully integrated layout module of TINA has all the features you need for advanced PCB design, including multilayer PCB's with split power plane layers, powerful auto placement & auto routing, rip-up and reroute, manual and "follow-me" trace placement, DRC, forward and back annotation, pin and gate swapping, keep-in and keep-out areas, copper pour, thermal relief, fan out, 3D view of your PCB design from any angle, Gerber file output and much more.

Q.3 What is PCB printing using screen printing?

Ans. Screen printing techniques actually the process that patterns the metal conductor to form the circuit.

Q.4 What do you mean by PCB fabrication process?

Ans. This PCB fabrication process involves a multistep integration of imaging materials, imaging equipment, and processing conditions with the metallization process to reduce the master pattern on a substrate.

Q5 What is Patterning (etching)?

Ans. The vast majority of printed circuit boards are made by bonding a layer of copper over the entire substrate, sometimes on both sides, (creating a "blank PCB") then removing unwanted copper after applying a temporary mask (e.g., by etching), leaving only the desired copper traces.

Q.6 What is Silk screen?

Ans. Silk screen printing uses etch-resistant inks to protect the copper foil. Subsequent etching removes the unwanted copper. Alternatively, the ink may be conductive, printed on a blank (non-conductive) board. The latter technique is also used in the manufacture of hybrid circuits.

Q.7 What is Photoengraving?

Ans. Photoengraving uses a photo mask and developer to selectively remove a photoresist coating. The remaining photoresist protects the copper foil. Subsequent etching removes the unwanted copper. The photomask is usually prepared with a photoplotter from data produced by a technician using CAM, or computer-aided manufacturing software. Laser-printed transparencies are typically employed for phototools; however, direct laser imaging techniques are being employed to replace phototools for high-resolution requirements.

Q.8 What is PCB milling?

Ans. PCB milling uses a two or three-axis mechanical milling system to mill away the copper foil from the substrate. A PCB milling machine (referred to as a 'PCB Prototyper') operates in a similar way to a plotter, receiving commands from the host software that control the position of the milling head in the x, y, and (if relevant) z axis.

Q.9 What material is used for Chemical etching?

Ans. Chemical etching is done with ferric chloride, ammonium persulfate, or sometimes hydrochloric acid.

Q.10 What is lamination?

Ans. Lamination: Some PCBs have trace layers inside the PCB and are called multi-layer PCBs. These are formed by bonding together separately etched thin boards.

EXPERIMENT NO:9

AIM: Introduction of the materials required for the fabrication of PCB's

THEORY:

The designer as well as manufactures prefers to use both the imperial as well as no imperial system of units. The most important to remember is:

1mil = inch by 1000

1mil = 25.4 micron

1micron = 1mm by 1000
Tracks on a PCB add inductance, resistance and capacitance to the circuit.

INDUCTANCE:

The amount of inductance is relatively constant across substrate types and depends on the length of track. The inductance per unit length of copper track is similar to that for a component lead.

RESISTANCE:

Resistance of the track depends on the cross-sectional area of the track as well as the length, hence values are usually quoted in resistance per square for each weight of copper the most popular copper weight, 1 oz., gives a typical value of 0.49mΩ/square.

CAPACITANCE:

$$C = \frac{A}{h} \epsilon$$

$$\text{Where as } \epsilon = 8.854 \text{ pF/m} \\ = 4.7 \text{ for FR4}$$

A = coverage area

H = distance between tracks

Therefore a 1 oz. copper track, .5mm (0.020 ") wide, 20 mm (.8") long over a ground plane on a .25 mm (.010") thick FR4 laminate would exhibit a resistance of 9.8mΩ, an inductance of 20nH, and a capacitive coupling to ground of 1.66 pF. These values may seem like low and negligible but when we talk of so many track then these values add up. These parasitic effects are under designers control very much like components values. There are other design constraints like production, marketing cost etc. Some important tables are given here for the ready reference and handy, fast calculations.

LAMINATE MATERIAL PROPERTIES:

Base material	Ref. Name	Relative Dielectric Constant(Cr)	Max. Temp. Tmax(degree Celsius)	Thermal Conductivity K(W/m per K)	Remarks
Dysfunctional epoxy	FR4	4.2 - 4.9	120 130	0.18	

CONCLUSION:

The availability of the copper plate of the proper size is the first and foremost requirement for the making of the PCB. In addition to it, you also require the precision high speed cutter blade for the fine and précised cutting of these copper plates to the required size. There can be the manual drilling machine for the drilling of the holes for components to be placed or the automatic plant as to the availability of the resources at your hand. There is the list of other equipment to add this like a personal computer printer, saw, scale, photo plotter, screen printing facility, chemicals, chemical treatment plant for the exposing and lamination of the PCB.

QUIZ QUESTIONS & ANSWERS

Q.1 What's role Tina play in Integrated PCB design?

Ans. The new fully integrated layout module of TINA has all the features you need for advanced PCB design, including multilayer PCB's with split power plane layers, powerful autoplacement & autorouting, rip-up and reroute, manual and "follow-me" trace placement, DRC, forward and back annotation, pin and gate swapping, keep-in and keep-out areas, copper pour, thermal relief, fanout, 3D view of your PCB design from any angle, Gerber file output and much more.

Q.2 What is PCB printing using screen printing?

Ans. Screen printing techniques actually the process that patterns the metal conductor to form the circuit.

Q.3What do you mean by PCB fabrication process?

Ans. This PCB fabrication process involves a multistep integration of imaging materials, imaging equipment, and processing conditions with the metallization process to reduce the master pattern on a substrate.

Q.4 What is Solder resist?

Ans. Areas that should not be soldered may be covered with a polymer solder resist.

Q.5 What is Copper thickness?

Ans.Copper thickness of PCBs can be specified in units of length, but is often specified as weight of copper per square foot, in ounces, which is easier to measure. Each ounce of copper is approximately 1.4 mils (0.0014 inch) or 35 μm of thickness.

Q.6 What material used for PCB fabrication process?

Ans.These new materials include eco-friendly FR polymers, ultra-fine PCB substrate yarns, thermally enhanced substrates, halogen-free LCPs, as well as flex circuit materials.

Q.7What is Silver Conductive Inks?

Ans.These silver conductive inks are formulated for use in printed electronics, to meet the need for low-cost processing in touchscreens and OLEDs.

Q8 What is Patterning (etching)?

Ans. The vast majority of printed circuit boards are made by bonding a layer of copper over the entire substrate, sometimes on both sides, (creating a "blank PCB") then removing unwanted copper after applying a temporary mask (e.g., by etching), leaving only the desired copper traces.

Q.9 What is Silk screen?

Ans. Silk screen printing uses etch-resistant inks to protect the copper foil. Subsequent etching removes the unwanted copper. Alternatively, the ink may be conductive, printed on a blank (non-conductive) board. The latter technique is also used in the manufacture of hybrid circuits.

Q.10 What material is used for Chemical etching?

Ans. Chemical etching is done with ferric chloride, ammonium persulfate, or sometimes hydrochloric acid.

EXPERIMENT NO:10

AIM: Development of PCB in hardware lab.

APPARATUS:

- 1) PCB art work film maker NV180
- 2) Artwork table NV181
- 3) PCB shearing machine NV182
- 4) Photo resist dip coating machine NV183
- 5) UV exposure unit NV184
- 6) Dye tank NV185
- 7) Development tank NV186
- 8) PCB etching machine NV187
- 9) Drill machine NV188

ACCESORIES:

- 1) Tray
- 2) Brush
- 3) PCB Laminate
- 4) Spray
- 5) Hand gloves

THEORY:

The development of the PCB involves following steps.

- 1) PCB printing using screen printing
- 2) Etching of the PCB
- 3) Drilling of PCB.
- 4) Coating of etched PCB to protect it from oxidation.

PCB printing using screen printing:

Screen printing techniques actually the process that patterns the metal conductor to form the circuit. This PCB fabrication process involves a multistep integration of imaging materials, imaging equipment, and processing conditions with the metallization process to reduce the master pattern on a substrate. Screen printing is considered as the most versatile of all printing processes as it can be done on wide variety of substrates of any shape, thickness and size. The

screen printing process is simple, and a wide variety of inks and dyes are available for use in screen printing than for use in any other printing process.

Etching of the PCB:

The final copper pattern is formed by selective removal of the unwanted copper which is not protected by an electric resist. FeCl_3 solution is popularly used etching solution. FeCl_3 powder will remove the copper from the unprotected part of the PCB. After removing the PCB it is dried for some time.

Drilling of PCB:

After etching of the PCB the next step is to drill the PCB for the interconnection of the various components on the PCB. The drill hole is having a diameter of generally one mm but the resistance sometimes require 1.5mm diameter. The drilling of the PCB is very important in terms of the working of the PCB hence the drilling is done by drilling machine of large precision and accuracy.

Coating of etched PCB to protect it from oxidation:

Since the upper layer of the PCB is a copper clad material which gets oxidised when comes in contact with the environment that affects the performance of the PCB. Hence the copper layer is coated with the laminates that are basically an insulator, to protect the Etched PCB to get oxidized.

PROCEDURE:

1. Take 50 ml water in a beaker and add 3 gm of sensitizer powder to it.
2. Add 50 ml water to sensitizer solution to make 100 ml solution.
3. Take 10 ml screen coating solution and add 10 drops of sensitizer solution to it.
3. Cut the Light Sensitive film as per the size of PCB layout. Arrange the film and on PCB screen Printing Unit.
4. Coat the Light Sensitive film on the screen with the Squeeze and dry the screen in the curing machine for 5 minutes. Remove the plastic paper from film and dry it again for 5 minutes.
5. Take the print of PCB layout on the plotting paper and place it on UV exposure such that solder side is in contact with glass.

6. Place screen then Rubber sheet and then weight.
7. Develop the screen by spraying water from 1 feet and dry the screen for 15 minutes in the open air.
8. Mount the Screen with the help of clamp on PCB Screen Printing Unit and cut it with the help of shearing machine
9. Place PCB Laminate to print and pour the ink inside the screen.
10. Pour 7 ltr water in the tank and add 2kg Ferric Chloride and stir it.
11. Mount the PCB on the clamp of Dipping Arrangement and dip the PCB on clamp from the opening of cover plate.
12. Drill the PCB with appropriate size of drill bit.

CONCLUSION: Thus we have developed the PCB in the hardware Lab.

QUIZ QUESTIONS & ANSWERS

Q.1 What's role Tina play in Integrated PCB design?

Ans. The new fully integrated layout module of TINA has all the features you need for advanced PCB design, including multilayer PCB's with split power plane layers, powerful autoplacement & autorouting, rip-up and reroute, manual and "follow-me" trace placement, DRC, forward and back annotation, pin and gate swapping, keep-in and keep-out areas, copper pour, thermal relief, fanout, 3D view of your PCB design from any angle, Gerber file output and much more.

Q.2 What is PCB printing using screen printing?

Ans. Screen printing techniques actually the process that patterns the metal conductor to form the circuit.

Q.3 What do you mean by PCB fabrication process?

Ans. This PCB fabrication process involves a multistep integration of imaging materials, imaging equipment, and processing conditions with the metallization process to reduce the master pattern on a substrate.

Q.4 Which Screen printing is most versatile?

Ans. Screen printing is considered as the most versatile of all printing processes as it can be done on wide variety of substrates of any shape, thickness and size. The screen printing process is simple, and a wide variety of inks and dyes are available for use in screen printing than for use in any other printing process.

Q.5 What do you mean by Etching of the PCB?

Ans. The final copper pattern is formed by selective removal of the unwanted copper which is not protected by an electric resist. FeCl_3 solution is popularly used etching solution. FeCl_3 powder will remove the copper from the unprotected part of the PCB. After removing the PCB it is dried for some time.

Q.6 What do you mean by Drilling of PCB?

Ans. After etching of the PCB the next step is to drill the PCB for the interconnection of the various components on the PCB. The drill hole is having a diameter of generally one mm but the resistance sometimes require 1.5mm diameter. The drilling of the PCB is very important in terms of the working of the PCB hence the drilling is done by drilling machine of large precision and accuracy.

Q. 7 What is TINA Design Suite?

Ans. TINA Design Suite is a powerful yet affordable circuit simulation and PCB design software package.

Q.8 Why TINA Design Suite is used?

Ans. It is used for analyzing, designing, and real time testing of analog, digital, VHDL, MCU, and mixed electronic circuits and their PCB layouts.

Q.9 What is unique feature of TINA?

Ans. A unique feature of TINA is that you can bring your circuit to life with the optional USB controlled.

Q.10 How it is used to compute computational power?

Ans. To meet this requirement TINA v9 has the ability to utilize the increasingly popular scalable multi-thread CPUs.

COMMUNICATION SYSTEMS

(EE-226-F)

LAB MANUAL

IV SSEMESSTER

LIST OF EXPERIMENTS

S.No.	NAME OF THE EXPERIMENT	Page No.
1.	GENERATION OF DSB-SC AM SIGNAL USING BALANCED MODULATOR.	3-8
2.	GENERATION OF SSB AM SIGNAL.	9-15
3.	TO STUDY ENVELOPE DETECTOR FOR DEMODULATION OF AM SIGNAL AND OBSERVE DIAGONAL PEAK CLIPPING EFFECT.	16-24
4.	FREQUENCY MODULATION USING VOLTAGE CONTROLLED OSCILLATOR.	25-29
5.	TO GENERATE A FM SIGNAL USING VARACTOR & REACTANCE MODULATION.	30-38
6.	DETECTION OF FM SIGNAL USING PLL & FOSTER SEELEY METHOD.	39-45
7.	TO STUDY SUPER HETERODYNE AM RECEIVER AND MEASUREMENT OF RECEIVER PARAMETERS	46-52
8.	TO STUDY PAM/PWM & PPM MODULATION & DEMODULATION.	53-62
9.	STUDY OF FREQUENCY DIVISION MULTIPLEXING/DEMULTIPLEXING WITH SINUSOIDAL & AUDIO INPUTS. STUDY OF 4 CHANNEL TIME DIVISION MULTIPLEXING SYSTEM.	63-67
10.	STUDY OF PULSE CODE MODULATION AND DEMODULATION	68-72
11.	WITH PARITY & HAMMING CODE. STUDY PULSE DATA CODING & DECODING TECHNIQUES FOR VARIOUS FORMATS.	73-75
12.	STUDY OF ASK, FSK MODULATOR AND DEMODULATOR. STUDY OF PSK & QPSK MODULATOR AND DEMODULATOR.	76-80
13.		81-85
14.		86-87

EXPERIMENT No.1

AIM:- To generate DSB-SC AM signal using balanced modulator.

APPARATUS REQUIRED:- (i) C.R.O. (ii) CRO Probe (ii) DSB/SSB Transmitter (ST 2201) and Receiver (ST2202) Trainer (iv) Connecting leads.

THEORY:-

A double sideband suppressed carrier signal, or DSBSC, is defined as the modulating signal and the carrier wave.

$$\text{DSBSC} = E \cdot \cos \mu t \cdot \cos \omega t \quad (1)$$

Generally, and in the context of this experiment, it is understood that: $\omega \gg \mu$ (2)

Equation (3) can be expanded to give:

$$\cos \mu t \cdot \cos \omega t = (E/2) \cos(\omega - \mu)t + (E/2) \cos(\omega + \mu)t \quad (3)$$

Equation (3) shows that the product is represented by two new signals, one on the sum frequency $(\omega + \mu)$, and one on the difference frequency $(\omega - \mu)$ - see Figure 1.

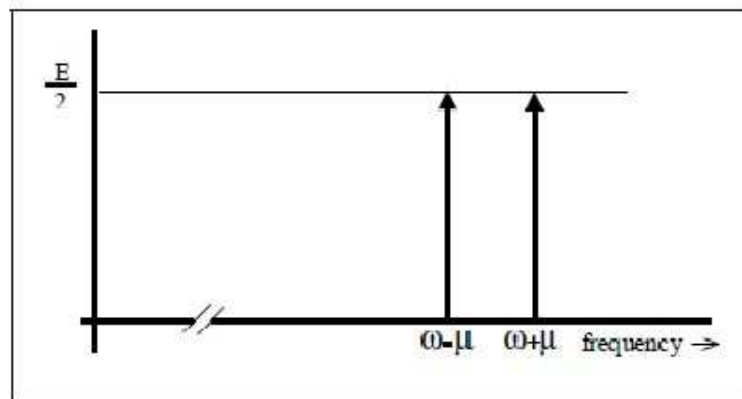


Figure 1: Spectral components

Remembering the inequality of eqn. (2) the two new components are located close to the frequency ω rad/s, one just below, and the other just above it. These are referred to as the lower and upper sidebands respectively.

These two components were derived from a 'carrier' term on ω rad/s, and a message on μ rad/s. Because there is no term at carrier frequency in the product signal it is described as a double sideband suppressed carrier (DSBSC) signal.

The term 'carrier' comes from the context of 'double sideband amplitude modulation' (commonly abbreviated to just AM).

The time domain appearance of a DSBSC (eqn. 1) in a text book is generally as shown in Figure 2.

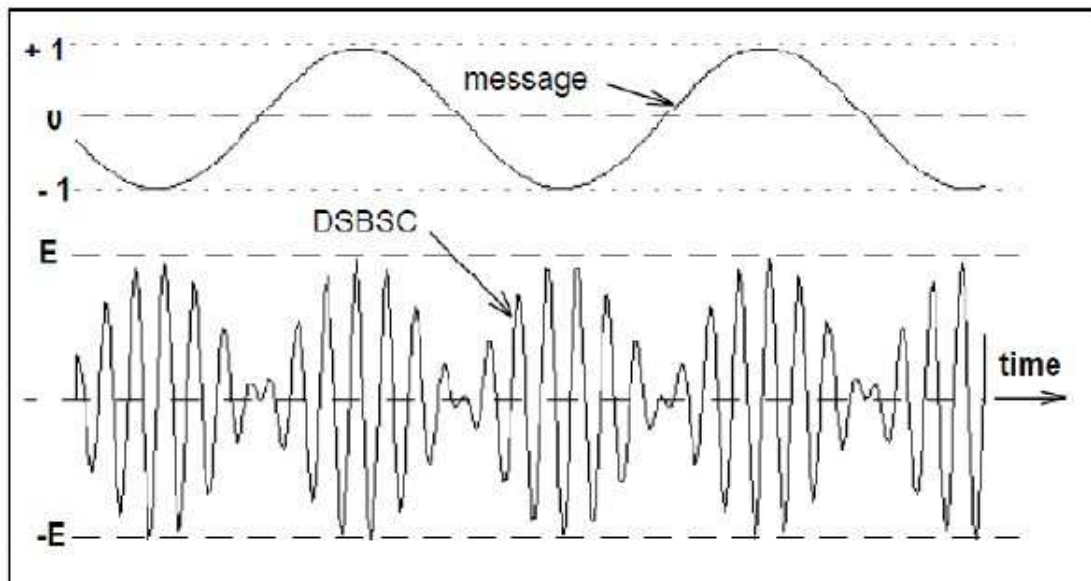


Figure 2: DSBSC - seen in the time domain

Notice the waveform of the DSBSC in Figure 2, especially near the times when the message amplitude is zero. The fine detail differs from period to period of the message. This is because the ratio of the two frequencies μ and ω has been made non-integral. Although the message and the carrier are periodic waveforms (sinusoids), the DSBSC itself need not necessarily be periodic.

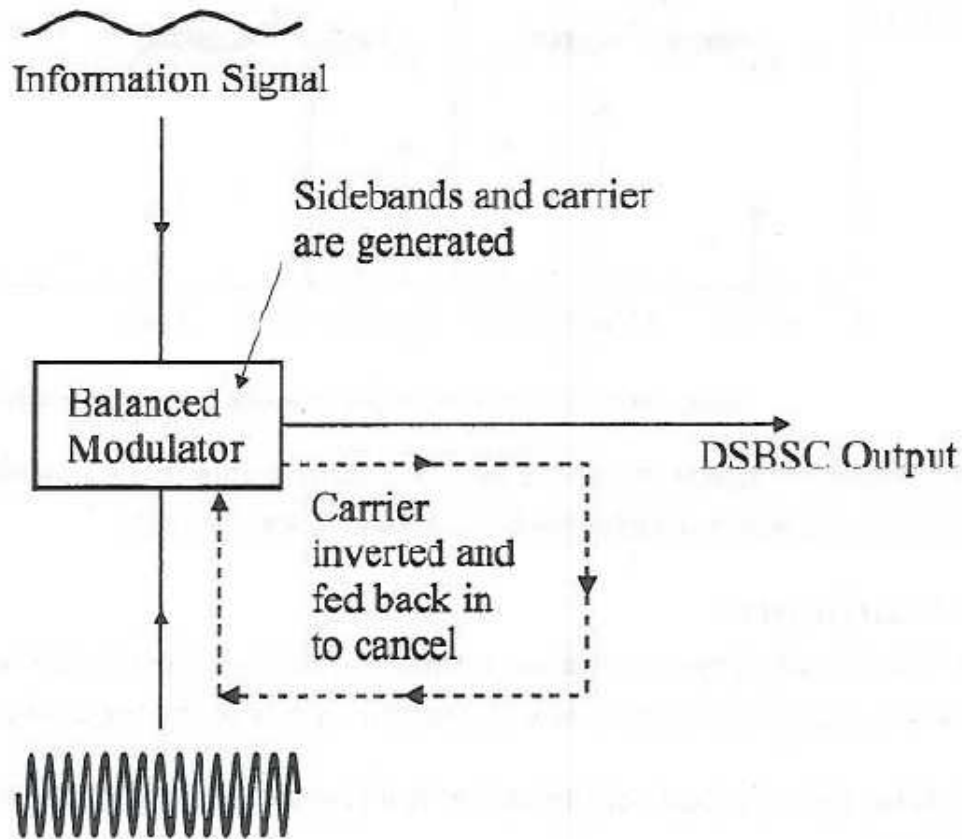


Figure 3: DSBSC Generation using balanced modulator
By removing the carrier from an AM waveforms, the generation of double sideband suppressed carrier (DSBSC) AM is generated.

Properties of DSB-SC Modulation:

- There is a 180 phase reversal at the point where $m(t)$ goes negative. This is typical of DSB-SC modulation.
- The bandwidth of the DSB-SC signal is double that of the message signal, that is, $BW=2B$ (Hz).
- The modulated signal is centered at the carrier frequency ω_c with two identical sidebands (double-sideband) – the lower sideband (LSB) and the upper sideband (USB). Being identical, they both convey the same message component.
- The spectrum contains no isolated carrier. Thus the name suppressed carrier.
- The 180 phase reversal causes the positive (or negative) side of the envelope to have a shape different from that of the message signal, see Figure 2.

A balanced modulator has two inputs: a single-frequency carrier and the modulating signal. For the modulator to operate properly, the amplitude of the carrier must be sufficiently greater than the amplitude of the modulating signal (approximately six to seven times greater).

PROCEDURE:-

1. Ensure that the following initial conditions exist on the board.
 - a. Audio input select switch in INT position.
 - b. Mode switches in DSB position.
 - c. Output amplifier's gain pot in full clockwise position.
 - d. Speakers switch in OFF position.
2. Turn on power to the ST2201 board.
3. Turn the audio oscillator block's amplitude pot to its full clockwise (MAX) position, and examine the block's output (t.p.14) on an oscilloscope. This is the audio frequency sine wave which will be our modulating signal. Note that the sine wave's frequency can be adjusted from about 300 Hz to approximately 3.4 KHz, by adjusting the audio oscillator's frequency potmeter. Note also that the amplitude of this audio modulating signal can be reduced to zero, by turning the Audio oscillator's amplitude potmeter to its fully counterclockwise (MIN) position. Return the amplitude present to its max position.
4. Turn the balance pot, in the balanced modulator & band pass filter circuit 1 block, to its fully clockwise position. It is this block that we will use to perform double-side band amplitude modulation.
5. Monitor, in turn, the two inputs to the balanced modulator & band pass filter circuits block, at t.p.1 and t.p.9. Note that:
 - a. The signal at t.p.1 is the audio-frequency sine wave from the audio oscillator block. This is the modulating input to our double-sideband modulator.
 - b. Test point 9 carries a sine wave of 1MHz frequency and amplitude 120mVpp approx. This is the carrier input to our double-sideband modulator.
6. Next, examine the output of the balanced modulator & band pass filter circuit 1 block (at t.p.3), together with the modulating signal at t.p.1. Trigger the oscilloscope on the t.p. 1 signal. The output from the balanced modulator & band pass filter circuit 1 block (at t.p. 3) is a DSBFC AM waveform, which has been formed by amplitude-modulating the 1MHz carrier sine wave with the audio-frequency sine wave from the audio oscillator.

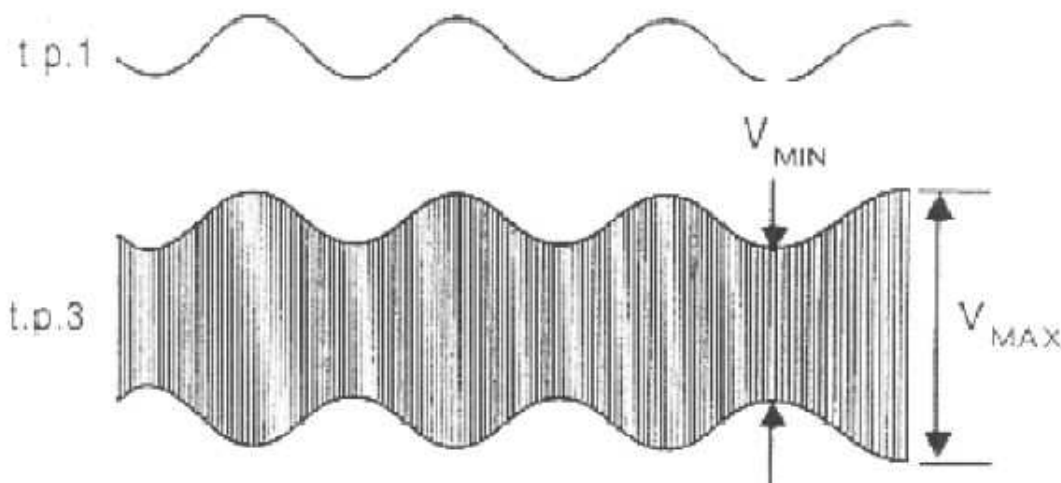


Figure 4: DSB FC (AM) waveforms

7. Now vary the amplitude and frequency of the audio-frequency sine wave, by adjusting the amplitude and frequency present in the audio oscillator block. Note the effect that varying each pot has on the amplitude modulated waveform. The amplitude

and frequency amplitudes of the two sidebands can be reduced to zero by reducing the amplitude of the modulating audio signal to zero. Do this by turning the amplitude pot to its MIN position, and note that the signal at t.p.3 becomes an un-modulated sine wave of frequency 1 MHz, indicating that only the carrier component now remains. Return the amplitude pot to its maximum position.

Now turn the balance pot in the balanced modulator & band pass filter circuit 1 block, until the signal at t.p. 3 is as shown in Fig. 5

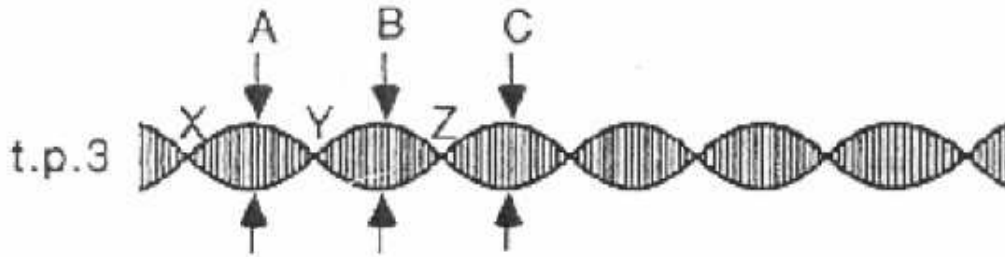


Figure 5: Output of BPF

The balance pot varies the amount of the 1 MHz carrier component, which is passed from the modulator's output. By adjusting the pot until the peaks of the waveform (A, B, C and so on) have the same amplitude, we are removing the carrier component altogether. We say that the carrier has been 'balanced out' (or 'suppressed') to leave only the two sidebands.

Note that once the carrier has been balanced out, the amplitude of t.p.3's waveform could be zero at minimum points X, Y & Z etc. If this is not the case, it is because one of the two sidebands is being amplified more than the other. To remove this problem, the band pass filter in the balanced modulator & band pass filter circuit 1 block must be adjusted so that it passes both sidebands equally. This is achieved by carefully trimming transformer T1, until the waveform's amplitude is as close to zero as possible at the minimum points. The waveform at t.p.3 is known as a double-side suppressed carrier (DSBSC) waveform, and its frequency spectrum is as shown in Fig.1. Note that now only the two sidebands remain, the carrier component has been removed.

8. Change the amplitude and frequency of the modulating audio signal (by adjusting the audio oscillator block's amplitude and frequency pots), and note the effect that these changes on the DSBSC waveform. The amplitudes of the two sidebands can be reduced to zero by reducing the amplitude of the modulating audio signal to zero. Do these by turning the amplitude present to its MIN position, and note that the monitored signal becomes a D C level, indicating that there are now no frequency components present. Return the amplitude pot to its MAX position.

9. Examine the output from the output amplifier block (t.p.13), together with the audio modulating signal (at t.p.1), triggering the scope with the audio modulating signal. Note that the DSBSC waveform appears, amplified slightly at t.p.13, as we will see later, it is the output amplifier's output signal which will be transmitted to the receiver.

10. By using the microphone, the human voice can be used as the modulating signal, instead of using ST2201's audio oscillator block. Connect the module's output to the external audio input on the ST2201 board, and put the audio input select switch in the ext position. The input signal to the audio input module may be taken from an external

COMMUNICATION SYSTEMS LAB (EE-226-F)

microphone or from a cassette recorder, by choosing the appropriate switch setting on the module.

RESULT:-

The DSBSC signal has been generated using balanced modulator.

WAVE FORMS OBSERVED:-

Draw wave forms as observed on CRO and label the different waveforms appropriately.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.
2. Before connecting the power supply plug into socket, ensure power supply should be switched off
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.
5. Power supply should be switched off after completion of experiment.

QUIZ / ANSWERS:-

Q. 1. What is DSBSC?

Ans. Double Sideband Suppressed Carrier.

Q. 2. Which are the discrete frequencies in DSBSC?

Ans. (1) Lower sideband frequency (2) Upper sideband frequency

Q. 3 In DSBSC, how many sidebands are there?

Ans. There are two sidebands in DSCBSC i.e. LSB and USB

Q. 4. Mention advantages of DSBSC over DSBFC.

Ans. Transmission efficiency is more.

Q. 5. Which type of carrier is used in Ring modulator?

Ans. Square wave carrier.

Q. 6. Write the methods of DSBSC generation.

Ans. (1) Balanced Modulator (2) Ring Modulator (3) Switching Modulator

Q. 7. What is the BW of DSBSC for a single tone modulating signal with frequency w ?

Ans. $2w$.

Q. 8. Where the modulation index lies?

Ans. modulation index always lies between 0 and 1. More than 1 is over modulation.

Q. 9. What happens in case of over modulation?

Ans. The wave will get distorted.

Q. 10. What is the range of audio frequencies?

Ans. 20 Hz to 20 KHz.

EXPERIMENT No.2

AIM:- To generate SSB-AM signal.

COMMUNICATION SYSTEMS LAB (EEE-226-F)

APPARATUS REQUIRED (i) C.R.O (ii) CRO PRD:-O.Probe (ii) DSSB/SSB Transmitter (STT 2201 and Receiver Trainer (ST 2202) (iii) Connecting leads

THEORY:-

Single Sideband Suppressed Carrier (SSB-SC) modulation was the basis for all long distance telephone communications up until the last decade. It was called "L carrier." It consisted of groups of telephone conversations modulated on upper and/or lower sidebands of contiguous suppressed carriers. The groupings and sideband orientations (USB LSB) supported hundreds and thousands of individual telephone conversations.

SSB Transmitter:

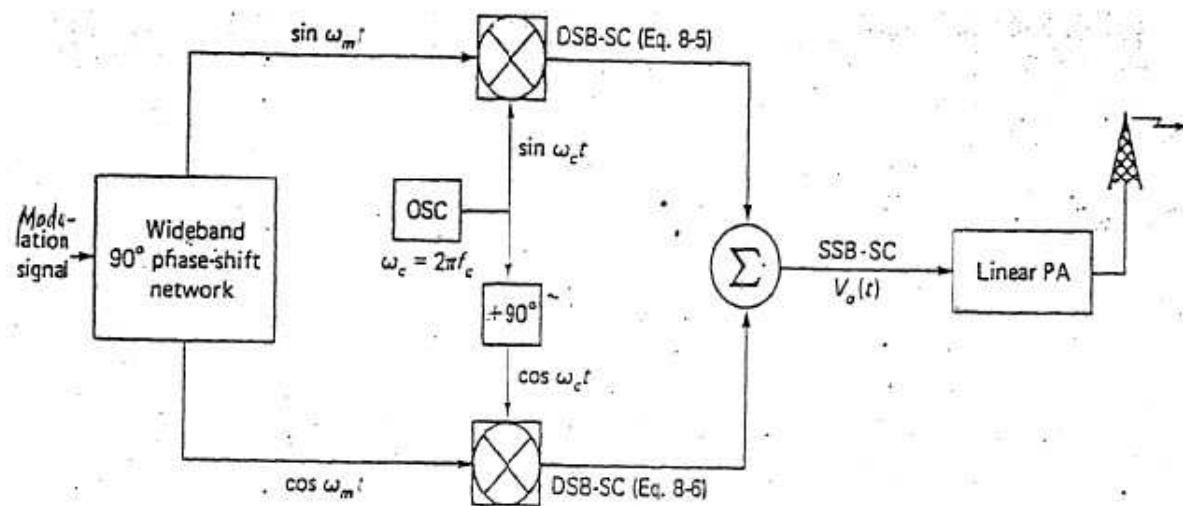


Figure 1: SSB Transmitter

Double side band transmission was the first method of modulation developed and for broadcast station is still the most popular. Indeed, for medium and long range broadcast stations is still the most popular. In medium and long range broadcast stations is still the most popular. The reason for such wide spread use is that the receiver design is simple and reliable. Radio is also used for communication in which the signal is simple and reliable. Radio is also used for communication in which the signal is simple and reliable.

addressed to a receiving station or a group of stations. For this type of communication other systems are used, one of which is investigated.

PROCEDURE:-

Ensure that the following initial conditions exist on the board:

- Audio input select switch in IN position.
- Mode switch in SS position.
- Output amplifier's gain pot in fully clockwise position.
- Speak switch in OFF position.

2. Turn on power to the ST2201 board.

3. Turn the audio oscillator block's amplitude pot to its fully clockwise (MAX) position, and examine the block's output (t.p.14) on an oscilloscope. This is the audio frequency sine wave which will be used as out modulating signal. Note that the sine wave's frequency can be adjusted from about 300Hz to approximately 3.4 KHz, by adjusting the audio oscillator's frequency pot.

Note: That the amplitude of this audio modulating signal can be reduced to zero, by turning the audio oscillator's pot to its fully counter-clockwise (MIN) position. Leave the amplitude pot on its full clockwise position, and adjust the frequency pot for an audio frequency of 2 KHz, approx. (mid-way).

4. To achieve signal- sideband amplitude modulation, we will utilize the following three blocks on the ST2201 module.

- Balanced modulator.
- Ceramic band pass filter
- Balanced modulator & band pass filter circuit 2.

We will now examine the operation of each of these blocks in detail.

5. Monitor the two inputs to the balanced modulator block, at t.p.15 and t.p.6 noting that:

- The signal t.p. 15 is the audio frequency sine wave from the audio oscillator block. This is the modulating input to the balanced modulator block.
- The signal at t.p. 6 is a sinewave whose frequency is slightly less than 455 KHz. It is generated by the 455 KHz oscillator block, and is the carrier input to the balanced modulator block.

6. Next, examine the output of the balanced modulator block (at t.p.17), together with the modulating signal at t.p.15 trigger the oscilloscope on the modulating signal. Check that the waveforms are as shown Fig. 2.

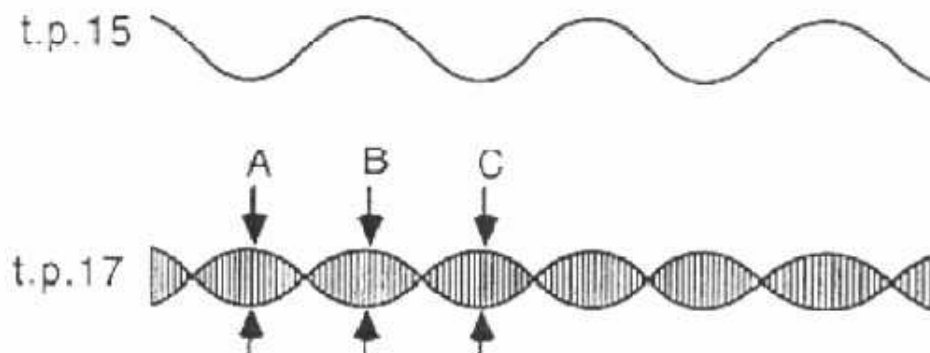


Figure 2: Modulating and Modulated Signal waveforms

Note that it may be necessary to adjust the balanced modulator block's balance pot, in order to ensure that the peaks of t.p.17's waveform envelope (labeled A, B, C etc. in the above diagram) all have equal amplitude. You will recall that the waveform at t.p.17 was encountered in the previous experiment this is a double-sideband suppressed carrier (DSBSC) AM waveform, and it has been obtained by amplitude-modulating the carrier sine wave at t.p. 6 of frequency f_c with the audio-frequency modulating signal at t.p. 15 of frequency f_m , and then removing the carrier component from the resulting AM signal, by adjusting the balance pot. The frequency spectrum of this DSBSC waveform is shown in Fig.3.

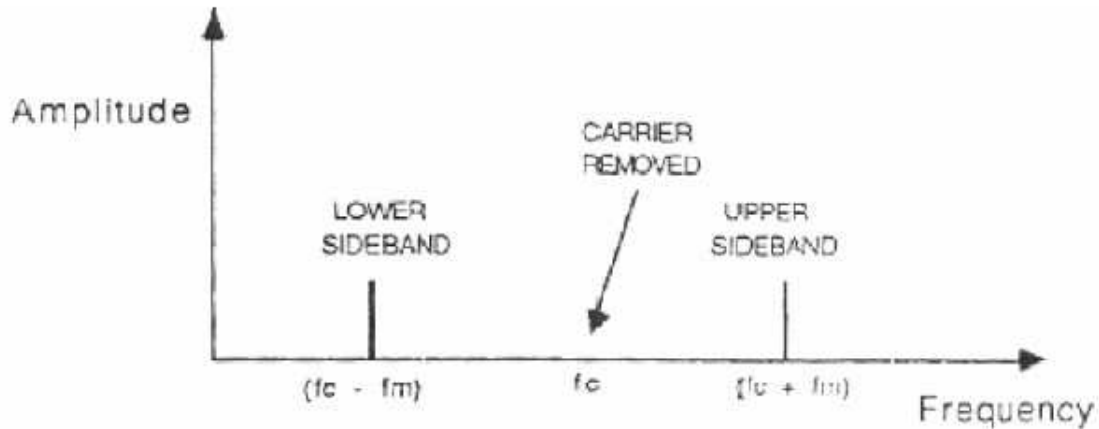


Figure 3:DSBSC Sidebands

7. The DSBSC output from the balanced modulator block is next passed on to the ceramic filter block, whose purpose is to pass the upper sideband, but block the lower sideband. We will now investigate how this is achieved. First note that the ceramic band pass filter has a narrow pass band centered around 455 KHz. It was mentioned earlier that the frequency of the carrier input to the balanced modulator block has been arranged to be slightly less than 455 KHz. In fact, the carrier is chosen so that, whatever the modulating frequency f_m , the upper sideband (at $f_c + f_m$) will fall inside the filter's pass band, while the lower sideband (at $f_c - f_m$) always falls outside. Consequently, the upper sideband will suffer little attenuation, while the lower sideband will be heavily attenuated to such an extent that it can be ignored. This is shown in the frequency spectrum in fig 4.

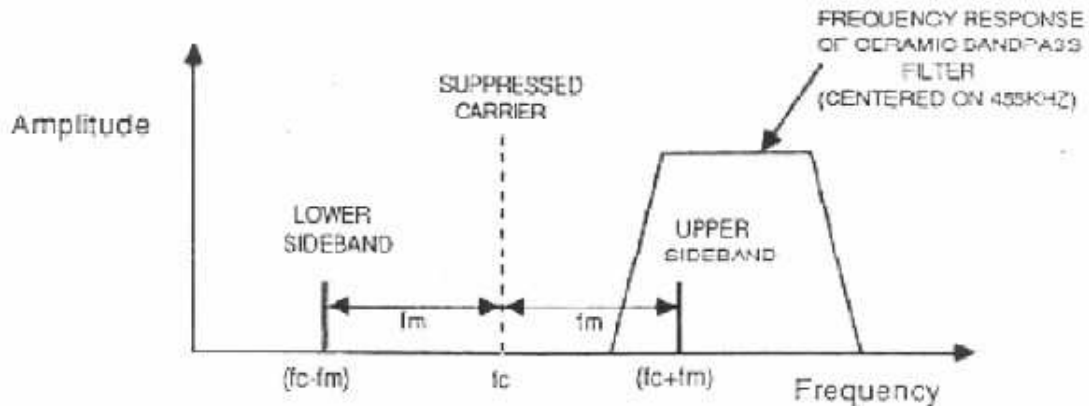


Figure 4: Frequency Response of Ceramic BPF

8. Monitor the output of the ceramic band pass filter block (at t.p. 20) together with the audio modulating signal (at t.p.15) using the later signal to trigger the oscilloscope. Note that the envelope of the signal at t.p. 20 now has fairly constant amplitude, as shown in Fig.5.

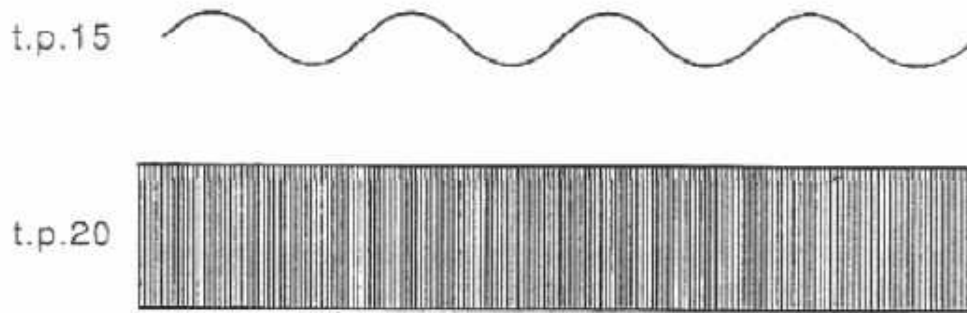


Figure 5: Input Audio Signal and SSB output Signal

If the amplitude of the signal at t.p. 20 is not reasonably constant, adjust the balance pot in the balance modulator block to minimize variations in the signal's amplitude. If the constant-amplitude waveform still cannot be obtained, then the frequency of the 455 KHz oscillator needs to be trimmed.

9. Now, trigger the oscilloscope with the ceramic band pass filter's output signal (t.p.20) and note that the signal is a good, clean sine wave, indicating that the filter has passed the upper sideband only. Next, turn the audio oscillator block's frequency pot throughout its range. Note that for most audio frequencies, the waveform is a good, clean sine wave, indicating that the lower sideband has been totally rejected by the filter. For low audio frequencies, you may notice that the monitored signal is not such a pure sinusoid. This is because the upper and lower sidebands are now very close to each other, and the filter can no longer completely remove the lower sideband. Nevertheless, the lower sideband's amplitude is sufficiently small compared with the upper sideband, that its presence can be ignored. Since the upper sideband dominates for all audio modulating frequencies, we say that single sideband (SSB) amplitude modulation has taken place.

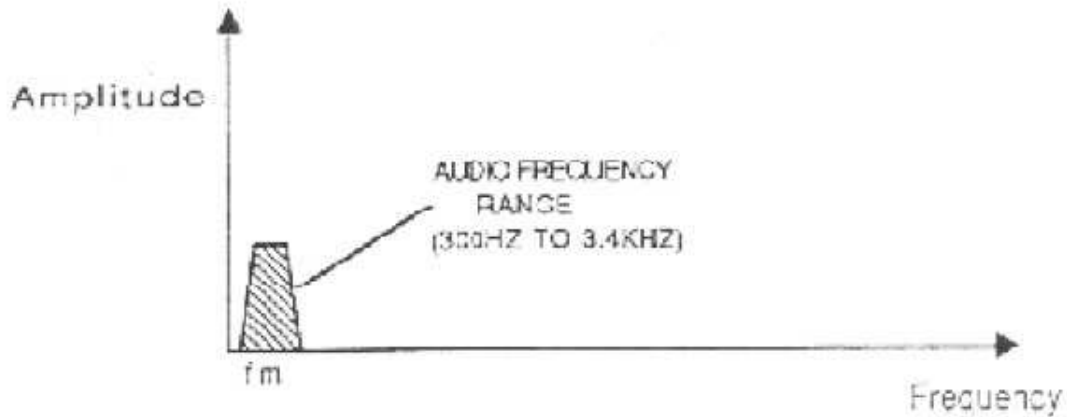
Note: If the monitored waveform is not a good sine wave at higher modulating frequencies (i.e. when the frequency pot is near the MAX position), then it is likely that the frequency of the 455 KHz oscillator needs to be trimmed

10. Note that there is some variation in the amplitude of the signal at the filter's output (t.p. 20) as the modulating frequency changes. This variation is due to the frequency response of the ceramic band pass filter, and is best explained by considering the spectrum of the filter's input signal at the MIN and MAX positions of the frequency pot, as shown in Fig. 4.

- a. Modulating frequency $f_m = 300\text{Hz}$ (pot in MIN position)
- b. Modulating frequency $f_m = 3.4\text{ KHz}$ (pot in MAX position)

Notice that, since the upper sideband cuts rising edge of the filter's frequency response when $f_m = 300\text{ Hz}$, there will be a certain amount of signal attenuation then the frequency pot is in its 'MIN' position.

11. Note that, by passing only the upper side band of frequency (f_c+f_m) , all we have actually done is to shift out audio modulating signal of frequency f_m up in frequency by an amount equal to the carrier frequency f_c . This is shown in Fig.7.



(a). Range of frequencies available from audio oscillator



(b). Corresponding range of output frequencies from ceramic band pass filter block
Figure 7: Range of frequency output from audio oscillator and ceramic BPF

12. With the audio oscillator block's frequency pot roughly in its midway position (arrowhead pointing towards the top), turn the block's amplitude pot to its MIN position, and note that the amplitude of the signal at the ceramic band pass filter's output (t.p. 20) drops to zero. This highlights one of the main advantages of SSB amplitude modulation if there is no modulating signal, then the amplitude of the SSB waveform drops to zero, so that no power is wasted. Return the amplitude pot to its MAX position.
13. The particular filter we are using has a pass band centered on 455 KHz, and this is why we have arranged for the wanted upper sideband to also be at about 455 KHz. However, there is a disadvantage of this type of filter is the range of frequencies that the filter will pass is fixed during the filter's manufacture, and cannot subsequently be altered. Note that since there is a large gap between the upper and lower sidebands (a gap of about 910 KHz), a band pass filter with a very sharp response is not needed to reject the lower sideband, a simple tuned circuit band pass filter is quite sufficient.
14. Now examine the output of the balanced modulator & band pass filter circuit 2 blocks (t.p.22), and check that the waveform is a good sine wave of frequency approximately 1.45MHz. This indicates that only the upper sideband is being passed by the block. Check that the waveform is reasonably good sinusoid for all audio

modulating frequencies (i.e. all positions of the audio oscillator's frequency pot). If this is not the case, it may be that the balance pot (in the balanced modulator & band pass filter circuit 2 blocks) needs adjusting, to remove any residual carrier component at 1 MHz. If a reasonably clean sine wave still cannot be obtained for all audio frequencies, then the response of the tuned circuit band pass filter needs adjusting. This is achieved by adjusting transformer T4 in the balanced modulator & band-pass filter circuit 2 block. When the modulating audio signal is swept over its entire range (a range of 3.4 KHz – 300 Hz = 3.1 KHz), the SSB waveform at t.p. 22 sweeps over the same frequency range. So single-sideband modulation has simply served to shift our range of audio frequencies up so they are centered on 1.455 MHz.

15. Monitor the 1.455 MHz SSB signal (at t.p. 22) together with the audio modulating signal (t.p. 15), triggering the scope with the later. Reduce the amplitude of the audio modulating signal to zero (by means of the audio oscillator block's amplitude pot), and note that the amplitude of the SSB signal also drops to zero, as expected. Return the amplitude pot to its MAX position.

16. Examine the final SSB output (at t.p. 22) together with the output from the output amplifier block (t.p. 13). Note that the final SSB waveform appears, amplified slightly, at t.p. 13. As we still see later, it is the output signal which will be transmitted to the receiver.

17. By using the microphone the human voice can be used as the audio modulating signal, instead of using ST2201's audio oscillator block. Connect the microphone to the external audio input on the ST2201 board, and put the audio input select switch in the EXT position.

The input signal to the audio input select may be taken from an external microphone (supplied with the module) or from a cassette recorder, by choosing the appropriate switch setting on the module.

RESULT:-

The SSB signal has been generated using balanced modulator.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.
2. Before connecting the power supply plug into socket, ensure power supply should be switched off
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.
5. Power supply should be switched off after completion of experiment

QUIZ / ANSWERS:-

Q.1. What is the most commonly used demodulator?

Ans. Diode detector.

Q.2. What is AGC?

Ans. AGC stands for automatic gain control.

Q.3. What is the use of AGC?

Ans. AGC circuit is used to prevent overloading receiver and also reduce the effect of fluctuations in the received signal strength.

Q.4. What is the required oscillator frequency in AM receiver?

Ans. The required oscillator frequency in AM receiver is always higher than the signal frequency.

Q.5. What is the use of pilot carrier in SSB?

Ans. For frequency stabilization.

Q.6. What are the methods of SSB generation?

Ans. Frequency discrimination and (b) Phase discrimination.

Q.7. What are the advantages of SSB over DSB?

Ans. 1. Transmitter circuit is more stable. 2. Increased transmission efficiency 3. Reduced BW.

Q.8. Which type of modulation is used in India for video transmission?

Ans. Amplitude Modulation.

Q.9. Which filter is used in SSB generation?

Ans. Mechanical filters.

Q.10. How AM signals with large carrier are detected?

Ans. By using envelope detector.

EXPERIMENT No.3

AIM:- To study envelope detector for AM signal and observe peak diagonal clipping effect.

APPARATUS REQUIRED:- (i) C.R.O. (ii) CRO Probe (iii) DSB/SSB Transmitter (ST 2201) and Receiver Trainer (ST 2202) (iv) Connecting leads

THEORY:-

The AM Transmitter:

The transmitter circuits produce the amplitude modulated signals which are used to carry information over the transmission to the receiver. The main parts of the transmitter are shown in Fig.11. In Fig.11 & 12, we can see that the peak-to-peak voltage in the AM waveform increase and decrease in sympathy with the audio signal.

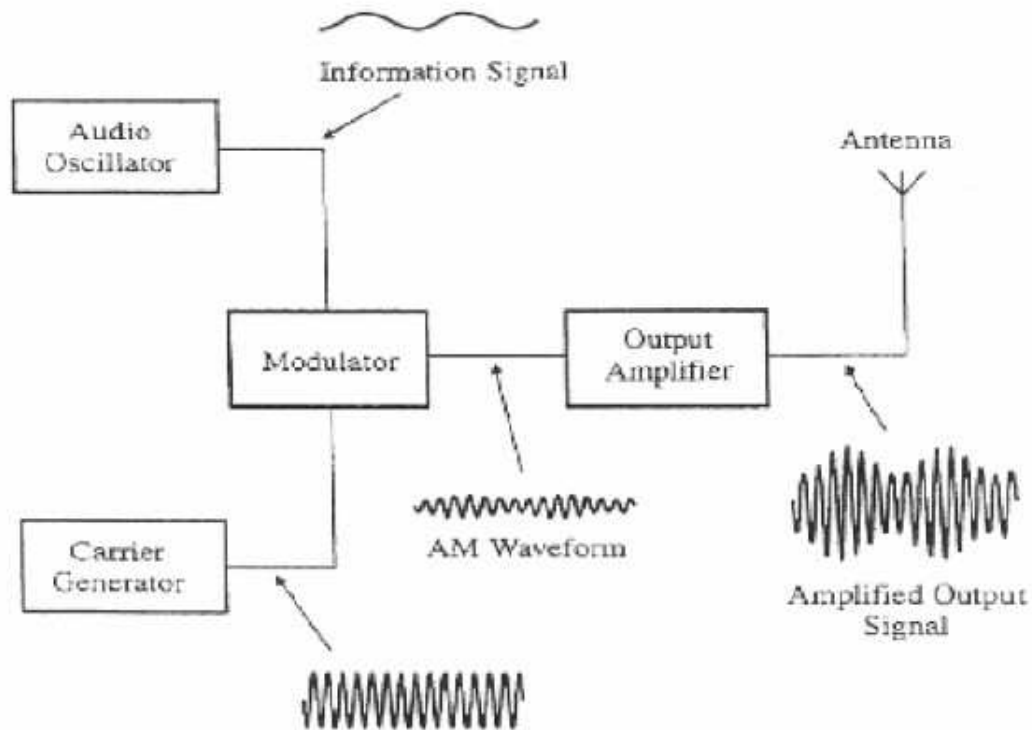


Fig. 1: AM Transmitter System

To emphasize the connection between the information and the final waveform, a line is sometimes drawn to follow the peaks of the carrier wave as shown in Fig.12. This shape, enclosed by a dashed line in our diagram, is referred to as an 'envelope', or a 'modulation envelope'. It is important to appreciate that it is only a guide to emphasize of the AM waveform.

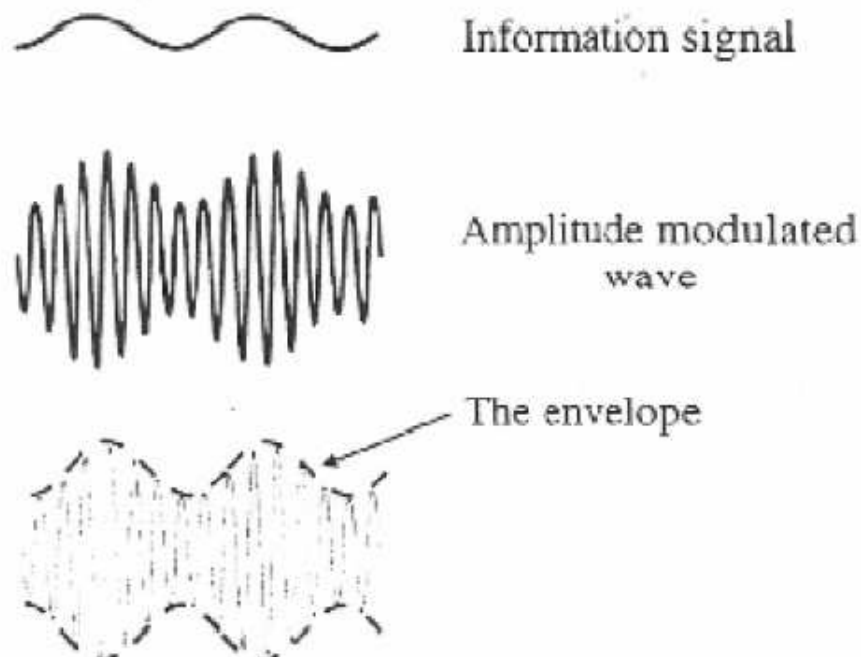


Figure 2: Waveforms in AM transmitter

AM Reception: The 'em' wave from the transmitting antenna will travel to the receiving antenna carrying the information with it. The stages of AM reception are shown in Fig. 3. :

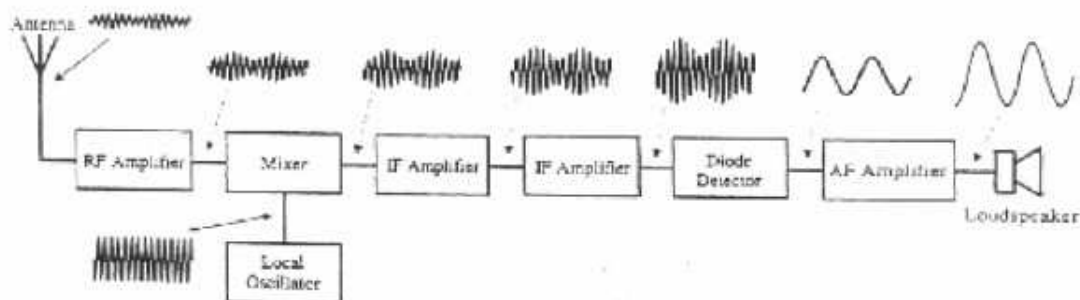


Figure 3: AM Reception

Envelope Detector:

The simplest form of envelope detector is diode detector. The function of the diode detector is to extract the audio signal from the signal at the output of the IF amplifiers. It performs this task in a very similar way to a half wave rectifier converting an AC input to a DC output. Fig.4 shows a simple circuit diagram of the diode detector.

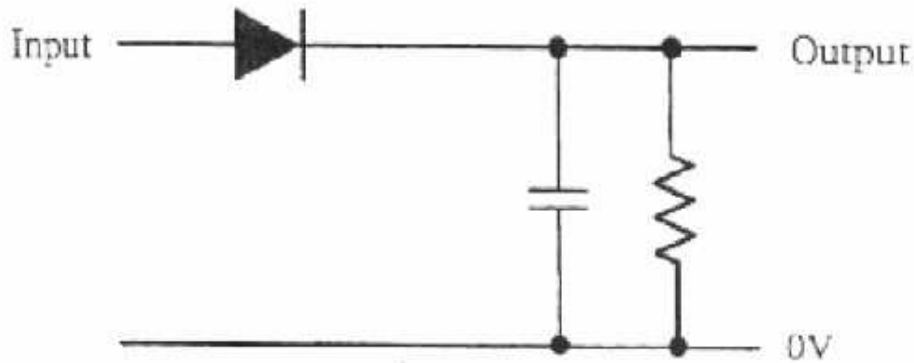


Figure 4: Diode Detector

In Fig.4, the diode conducts every time the input signal applied to its anode is more positive than the voltage on the top plate of the capacitor. When the voltage falls below the capacitor voltage, the diode ceases to conduct and the voltage across the capacitor leaks away until the next time the input signal is able to switch it on again. See fig. 5

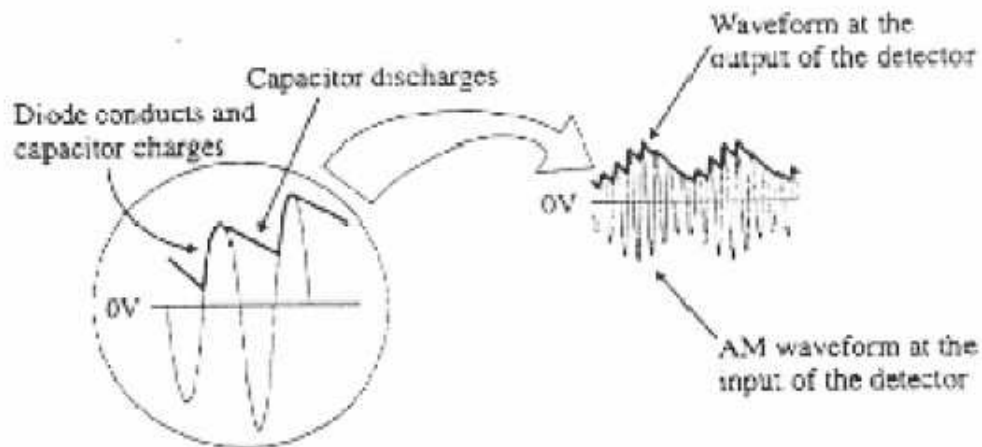


Fig. 5 Clipping in Diode Detector

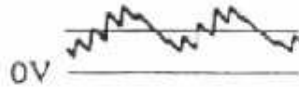
The result is an output which contains three components :

1. The wanted audio information signal.
2. Some ripple at the IF frequency.
3. A positive DC voltage level.

At the input to the audio amplifier, a low pass filter is used to remove the IF ripple and a capacitor blocks the DC voltage level. Fig.6 shows the result of the information signal passing through the diode detector and audio amplifier. The remaining audio signals are then amplified to provide the final output to the loudspeaker.



The input to the diode detector from the last IF amplifier



Output of diode detector includes : a DC level, the audio signal, ripple at IF frequency



Output after filtering

Figure 6: Output of Diode Detector and output Filter

PROCEDURE:-

1. Position the ST2201 & ST2202 modules, with the ST2201 board on the left, and a gap of about three inches between them.
2. Ensure that the following initial conditions exist on the ST2201 board.
 - a. Audio oscillator's amplitude pot in fully clockwise position.
 - b. Audio input select switch in INT position.
 - c. Balance pot in balanced modulator & band pass filter circuit 1 block, in full clockwise position;
 - d. Mode switch in DSB position.
 - e. Output amplifier's gain pot in full counter-clockwise position.
 - f. TX output select switch in ANT position:
 - g. Audio amplifier's volume pot in fully counter-clockwise position.
 - h. Speaker switch in ON position.
 - i. On-board antenna in vertical position, and fully extended.
3. Ensure that the following initial conditions exist on the ST2102 board:
 - a. RX input select switch in ANT position.
 - b. R.F. amplifier's tuned circuit select switch in INT position.
 - c. R.E amplifier's gain pot in fully clock-wise position;
 - d. AGC switch in INT position.
 - e. Detector switch in diode position.
 - f. Audio amplifier's volume pot in fully counter-clockwise position.
 - g. Speaker switch in ON position.
 - h. Beat frequency oscillator switch in OFF position.
 - i. On-board antenna in vertical position, and fully extended.
4. Turn on power to the modules.
5. On the ST2202 module, slowly turn the audio amplifier's volume pot clockwise, until sounds can be heard from the on-board loudspeaker. Next, turn the vernier tuning dial until a broad cast station can be heard clearly, and adjust the volume control to a comfortable level.

Note: If desired, headphones (supplied with the module) may be used instead of the on-board loudspeaker. To use the headphones, simply plug the headphone jack into the audio amplifier block's headphones socket, and adjust controlled block's volume pot.

6. The first stage or 'front end' of the ST2202 AM receiver is the R.F. amplifier stage. This is a wide -bandwidth tuned amplifier stage, which is tuned into the wanted station by means of the tuning dial. Once it has been tuned into the wanted station, the R.F. amplifier, having little selectivity, will not only amplify, but also those frequencies that are close to the wanted frequency. As we will see later, these nearby frequencies will be removed by subsequent stages of the receiver, to leave only the wanted signal. Examine the envelope of the signal at the R.F. amplifier's output (at t.p. 12), with an a.c. - coupled oscilloscope channel. Note that:

a. The amplifier's output signal is very small in amplitude (a few tens of millivolts at the most). This is because one stage of amplification is not sufficient to bring the signal's amplitude up to a reasonable level.

b. Only a very small amount of amplitude modulation can be detected, if any.

This is because there are many unwanted frequencies getting through to the amplifier output, which tend to 'drown out' the wanted AM Signal. You may notice that the waveform itself drifts up and down on the scope display, indicating that the waveform's average level is changing. This is due to the operation of the AGC circuit, which will be explained later.

7. The next stage of the receiver is the mixer stage, which mixes the R.F. amplifier's output with the output of a local oscillator. The Frequency of the local oscillator is also tuned by means of the tuning dial, and is arranged so that its frequency is always 455 KHz above the signal frequency that the R.F. amplifier is tuned to. This fixed frequency difference is always present, irrespective of the position of the tuning dial, and is arranged so that its frequency is always 455 KHz above the signal frequency that the R.F. amplifier is tuned to. This fixed frequency difference is always present, irrespective of the position of the tuning dial, and is known as the intermediate frequency (IF for short). This frequency relationship is shown below, for some arbitrary position of the tuning dial.

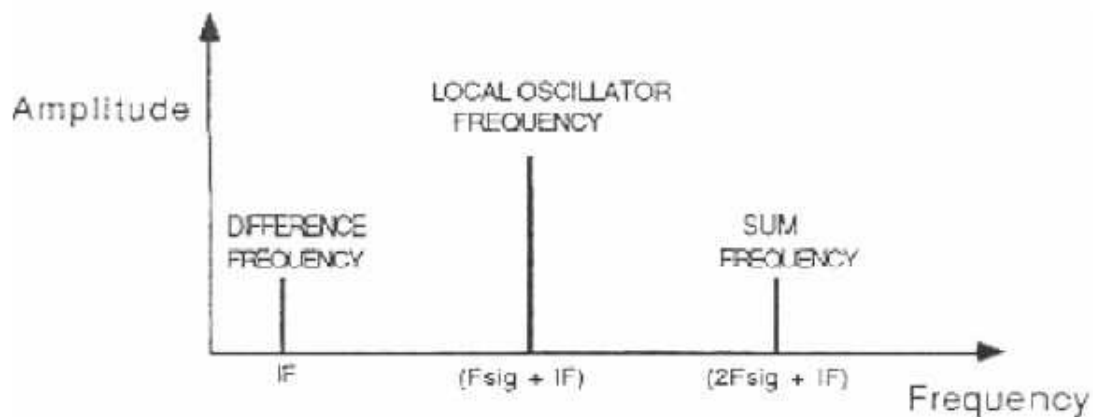


Figure 7: Frequency Contents in DSB AM

Examine the output of the local oscillator block, and check that its frequency varies as the tuning dial is turned. Re-time the receiver to a radio station.

8. The operation of the mixer stage is basically to shift the wanted signal down to the IF frequency, irrespective of the position of the tuning dial. This is achieved in two stages.

a. By mixing the local oscillator's output sine wave with the output from the R.F. amplifier block. This produces three frequency components:

The local oscillator frequency = $(f_{sig} + IF)$

The sum of the original two frequencies, $f_{sum} = (2 f_{sig} + IF)$

The difference between the original two frequencies,

b. By strongly attenuating all components. Except the difference frequency, IF this is done by putting a narrow-bandwidth band pass filter on the mixer's output.

The end result of this process is that the carrier frequency of the selected AM station is shifted down to 455 KHz (the IF Frequency), and the sidebands of the AM signal are now either side of 455 KHz.

9. Note that, since the mixer's band pass filter is not highly selective, it will not completely remove the local oscillators and sum frequency components from the mixer's output. This is the case particularly with the local oscillator component, which is much larger in amplitude than the sum and difference components. Examine the output of the mixer block (t.p. 20) with an a.c. coupled oscilloscope channel, and note that the main frequency component present changes as the tuning dial is turned. This is the local oscillator component, which still dominates the mixer's output, in spite of being attenuated by the mixer's band pass filter.

10. Tune in to a strong broadcast station again and note that the monitored signal shows little, if any, sign of modulation. This is because the wanted component, which is now at the IF frequency of 455 KHz, is still very small in component, which is now at the IF frequency of 455 KHz, is still very small in comparison to the local oscillator component. What we need to do now is to preferentially amplify frequencies around 455 KHz, without amplifying the higher-frequency local oscillator and SUM components. This selective amplification is achieved by using two IF amplifier stages, IF amplifier 1 and IF amplifier 2, which are designed to amplify strongly a narrow band of frequencies around 455 KHz, without amplifying frequencies on either side of this narrow band. These IF amplifiers are basically tuned amplifiers which have been pre-tuned to the IF frequency-they have a bandwidth just wide enough to amplify the 455 KHz carrier and the AM sidebands either side of it. Any frequencies outside this narrow frequency band will not be amplified. Examine the output of IF amplifier 1 (at. t.p. 24) with an a.c.-coupled oscilloscope channel, and note that:

a. The overall amplitude of the signal is much larger than the signal amplitude at the mixer's output, indicating that voltage amplification has occurred.

b. The dominant component of the signal is now at 455 KHz, irrespective of any particular station you have tuned into. This implies that the wanted signal, at the IF frequency, has been amplified to a level where it dominates over the unwanted components.

c. The envelope of the signal is modulated in amplitude, according to the sound information being transmitted by the station you have tuned into.

11. Examine the output of IF amplifier 2 (t.p.28) with an a.c.-coupled oscilloscope channel, noting that the amplitude of the signal has been further amplified by this second IF amplitude of the signal has been further amplified by this second IF amplifier stage. IF amplifier 2 has once again preferentially amplified signals around the IF frequency (455 KHz), so that:

- a. The unwanted local oscillator and sum components from the mixer are now so small in comparison, that they can be ignored totally,
- b. Frequencies close to the I F frequency, which are due to stations close to the wanted station, are also strongly attenuated.

The resulting signal at the output of IF amplifier 2 (t.p.28) is therefore composed almost entirely of a 455 KHz carrier, and the A.M. sidebands either side of it carrying the wanted audio information.

12. The next step is extract this audio information from the amplitude variations of the signal at the output of IF amplifier 2. This operation is performed by the diode detector block, whose output follows the changes in the amplitude of the signal at its input. To see how this works, examine the output of the diode detector block (t.p.31), together with the output from IF amplifier 2 (at t.p.28). Note that the signal at the diode detector's output:

- Follows the amplitude variations of the incoming signal as required:
- Contains some ripple at the IF frequency of 455 KHz, and
- The signal has a positive DC offset, equal to half the average peak to peak amplitude of the incoming signal. We will see how we make use of this offset later on, when we look at automatic gain control (AGC).

13. The final stage of the receiver is the audio amplifier block contains a simple low-pass filter which passes only audio frequencies, and removes the high frequency ripple from the diode detector's output signal. This filtered audio signal is applied to the input of an audio power amplifier, which drives on board loudspeaker (and the headphones, if these are used). The final result is the sound you are listening to the audio signal which drives the loudspeaker can be monitored at t.p. 39 (providing that the audio amplifier block's volume pot is not in its minimum volume position). Compare this signal with that at the diode detector's output (t.p. 31), and note how the audio amplifier block's low pass filter has 'cleaned up' the audio signal. You may notice that the output from the audio amplifier block (t.p. 39) is inverted with respect to the signal at the output of the diode detector (t.p. 31) this inversion is performed by the audio power amplifier IC, and in no way affects the sound produced by the receiver.

14. Now that we have examined the basic principles of operation of the ST2202 receiver for the reception and demodulation of AM broadcast signals, we will try receiving the AM signal from the ST2201 transmitter. Presently, the gain of ST2201's output amplifier block is zero, so that there is no output from the Transmitter. Now turn the gain pot in ST2201's output amplifier block to its fully clockwise (maximum gain) position, so that the transmitter generates an AM signal. On the ST2201 module, examine the transmitter's output signal (t.p.13), together with the audio modulating signal (t.p.1), triggering the 'scope with the signal'. Since ST2201 TX output select switch is in the ANT position, the AM signal at t.p.13 is fed to the transmitter's antenna. Prove this by touching ST2201's antenna, and nothing that the loading caused by your hand reduces the amplitude of the AM waveform. at t.p.13. The antenna will propagate this AM signal over a maximum distance of about 1.4 feet. We will now attempt to receive the propagated AM waveform with the ST2201/ ST2202 board, by using the receiver's on board antenna.

Note: If more than one ST2201 transmitter/receiver system is in use at one time, it is possible that there may be interference between nearby transmitters if antenna propagation is used. To eliminate this problem, use a cable between each

transmitter/receiver pair, connecting it between ST2201's TX output socket and ST2201/ST2202's RX input socket. If you do this, make sure that the transmitter's TX output select switch, and the receiver's RX input select switch, are both in the SKT position, then follow the steps below as though antenna propagation were being used.

15. On the ST2201 module, turn the volume pot (in the audio amplifier block) clockwise, until you can hear the tone of the audio oscillator's output signal, from the loudspeaker on the board.

Note: If desired, headphones may be used instead of the loudspeaker on the board. To use the headphones, simply plug the headphone jack into the audio amplifier block's headphones socket, and put the speaker switch in the OFF position. The volume from the headphones is still controlled by the block's volume pot. Turn the volume pot to the full counter-clockwise (minimum volume) position.

16. On the ST2201/ST2202 receiver, adjust the volume pot so that the receiver's output can be clearly heard. Then adjust the receiver's tuning dial until the tone generated at the transmitter is also clearly audible at the receiver (this should be when the tuning dial is set to about 55-65 and adjust the receiver's volume pot until the tone is at a comfortable level. Check that you are tuned into the transmitter's output signal, by varying ST2201's frequency pot in the audio oscillator block, and nothing that the tone generated by the receiver changes.

The ST2201/2202 receiver is now tuned into AM signal generated by the ST2201 transmitter. Briefly check that the waveforms, at the outputs of the following receiver blocks, are as expected:

R. F. Amplifier (t.p.12)
Mixer (t.p.20)
I.F. Amplifier 1 (t.p.24)
I.F. Amplifier 2 (t.p.28)
Diode Detector (t.p.31)
Audio Amplifier (t.p.39)

17. By using the microphone, the human voice can be used as transmitter's audio modulating signal, instead of using ST2201's audio oscillator block. Use DSB and not DSBSC. Connect the microphone's output to the external audio input on the ST2201 board, and put the audio input select switch in the EXT position.

18. In the output of diode detector peak diagonal clipping can be observed at low values of time constant of tuning circuitry.

RESULT:-

AM signal has been demodulated using envelope detector and peak diagonal clipping effect has been observed.

PRECAUTIONS:-

1. Do not use open ended wires for connecting 230 V power supply.
2. Before connecting the power supply plug into socket, ensure power supply should be switched off.
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.

5. Power supply should be switched off after completion of experiment.

QUIZ / ANSWERS:-

Q. 1. What is amplitude modulation?

Ans. Amplitude Modulation is a process in which the amplitude of the carrier is made proportional to the instantaneous amplitude of the modulating signal.

Q. 2. Which are the three discrete frequencies in AM?

Ans. (1) Carrier frequency (2) lower sideband frequency (3) upper sideband frequency

Q. 3 How many sidebands in AM?

Ans. There are two sidebands in AM i.e. LSB and USB

Q. 4. Which circuit is used as LPF?

Ans. R-C circuit.

Q. 5. Which are the two methods of AM generation?

Ans. (1) single sideband (2) double sideband

Q.6. What is diagonal clipping?

Ans. Distortion caused because of small value of time constant of tuned circuit is called diagonal clipping.

Q.7. What is the unit of modulation index in AM?

Ans. It is unit less.

Q. 8. Where the modulation index lies?

Ans. modulation index always lies between 0 and 1. More than 1 is over modulation.

Q. 9. What happens in case of over modulation?

Ans. The wave will get distorted.

Q. 10. How DSBSC can be converted into conventional AM?

Ans. By carrier reinsertion.

EXPERIMENT No.4

AIM:- To generate Frequency modulated signal using Voltage Control Oscillator.

APPARATUS REQUIRED:- (i) C.R.O. (ii) CRO Probe (iii) FM Modulation and Demodulation Trainer (ST 2203) (iv) Connecting leads

THEORY:-

Frequency modulation is a form of angle modulation in which the amplitude of the modulated carrier is kept constant while its frequency and its rate of change are varied by the modulating signal. In FM the instantaneous angular frequency ω_i is varied linearly in accordance with the instantaneous magnitude of base band signal $X(t)$, about an un-modulated carrier frequency (also called as resting frequency) ω_c and the rate at which the carrier shifts from its resting point to its non resting point is determined by the frequency of modulating signal while keeping the amplitude of the carrier wave constant.

$$\text{Carrier signal } C(t) = A \sin(\omega_c t + \theta_0) = A \sin \Phi \dots\dots\dots(1)$$

where ω_c is the frequency of Carrier wave in radians/second and

$$\Phi \text{ in radians} = \text{Total phase angle of the unmodulated carrier} = (\omega_c t + \theta_0) \dots\dots(2)$$

In FM while the amplitude A remains constant, instantaneous value of Φ changes. If $\omega_i(t)$ = Instantaneous value of angular velocity, and

and Φ_i = Instantaneous phase angle of FM wave,

$$\text{then } \omega_i(t) = d \Phi_i / dt, \dots\dots\dots(3)$$

$$\text{and } \Phi_i = \int \omega_i(t) dt \dots\dots\dots(4)$$

$$\text{Therefore FM wave can be represented as } S(t) = A \sin \Phi_i \dots\dots\dots(5)$$

$$\text{Modulating voltage Signal} = X(t) \text{ volts} \dots\dots\dots(6)$$

Then instantaneous angular frequency of an FM signal is given by

$$d \Phi_i / dt = \omega_i(t) = \omega_c + K_f X(t) \dots\dots\dots(7)$$

where K_f = Constant of proportionality = frequency sensitivity of the modulator
in Hertz per volt

$$\text{Therefore Frequency variation} = |K_f X(t)| \dots\dots\dots(8)$$

Since the value of ω_c is assumed to be fixed,

$$\Phi_i = \int \omega_i(t) dt = \int [\omega_c + K_f X(t)] dt = \omega_c t + K_f \int X(t) dt \dots\dots\dots(9)$$

Frequency Deviation It is the amount by which carrier frequency is varied from its unmodulated value and it is same as frequency variation.

$$\text{Max Frequency deviation } \Delta \omega = |K_f X(t)|_{\text{max}} \dots\dots\dots(10)$$

Very often we write $\Delta \omega = \delta$;

Maximum allowed deviation = 75 kHz

Frequency Modulation Index m_f - It is the ratio of frequency deviation $\Delta \omega$ in rad/sec to the angular frequency of modulating signal ω_m or frequency deviation in Hertz/sec to the modulating frequency in Hertz/sec.

$$\text{Thus } m_f = \Delta \omega / \omega_m = \delta / \omega_m \text{ if } \delta \text{ is given in rad /Sec} \dots\dots\dots(11)$$

$$\text{If } \delta \text{ is given in Hertz/Sec then } m_f = \delta / f_m \dots\dots\dots(12)$$

Mathematical expression for FM wave

$$S(t) = A \sin \Phi_i = A \sin [W_{ct} + K_f \int X(t) dt] \dots\dots\dots(13)$$

For Single tone FM

$$X(t) = V_m \cos W_{mt} \dots\dots\dots(14)$$

Thus $\Phi_i = W_{ct} + K_f \int V_m \cos W_{mt} dt = W_{ct} + K_f \frac{V_m}{W_m} \sin W_{mt}$

$$W_m = W_{ct} + \Delta W \sin W_{mt} = W_{ct} + m_f \sin W_{mt}$$

Thus $S(t) = A \sin [W_{ct} + K_f \int X(t) dt]$
 $= A \sin [W_{ct} + m_f \sin W_{mt}] = \dots\dots\dots(15)$

Deviation Ratio

It is the ratio of deviation in carrier frequency to the maximum modulating frequency.

In single tone FM, modulation index and the deviation ratio will be the same. If the modulating signal (AF) is 15 kHz at a certain amplitude and the carrier shift is 75 kHz, the transmitter will produce eight (8) significant sidebands as shown in the table above. The corresponding deviation ratio / modulation index is known as Maximum Deviation Ratio.

However in multi tone FM, the amplitude of highest frequency component may not necessarily be maximum. Modulation index will be different for each signal frequency component. The deviation ratio in this case will not be equal to any particular modulation index.

Frequency Spectrum

Analysis of equation (15) which is a sine function of another sine function shows:

$$S(t) = A \{ J_0(m_f) \sin W_{ct} + J_1(m_f) \{ \sin(W_{ct} + W_{mt}) + \sin(W_{ct} - W_{mt}) \} \\ + J_2(m_f) \{ \sin(W_{ct} + 2W_{mt}) + \sin(W_{ct} - 2W_{mt}) \} \\ + J_3(m_f) \{ \sin(W_{ct} + 3W_{mt}) + \sin(W_{ct} - 3W_{mt}) \} \\ + J_4(m_f) \{ \sin(W_{ct} + 4W_{mt}) + \sin(W_{ct} - 4W_{mt}) \} + \dots\dots\dots \}$$

The output consists of a carrier and an apparently infinite number of pairs of side bands having an amplitude coefficient $J_n(m_f)$, which is a Bessel function of m_f and of the order n denoted by the subscript. Values of these coefficients are available readily in table form as well as in graphic form as shown below.

Analysis of FM waveforms Wave forms of carrier, modulating signal, modulated signal as well as graphical form of plot of $J_n(m_f)$ versus values of m_f are shown below.

It can be seen that:

1. Unlike AM, FM output contains carrier component of frequency f_c as well as infinite number of side bands separated from the carrier frequency by f_m , $2f_c$, $3f_c$, and thus have a recurrence frequency of f_m .
2. The values of each J_n coefficient, which represent the amplitude of a pair of side bands, fluctuates on either side of zero, gradually diminishes with increasing value of m_f like damped oscillations. The values of J_n coefficients also eventually decrease, but

only past increased value of n . As the value of m_f increases, the value of J_0 decreases from 1 and the values of J_1 to J_n increases from 0 and fluctuate around mean value of 0.

3. The modulation index determines how many side band components have significant values.

4. Unlike AM, in FM, while the amplitude of modulated signal remains constant, the value of the carrier component decreases with increase in m_f like a damped oscillation. It means that while the total transmitted power remains constant in FM, the number side bands of significant amplitude (and therefore the effective band width) increase with increase in m_f . This increases the immunity to noise in FM unlike AM.

5. As the value of m_f increases, The carrier component becomes zero for certain values of modulation index, called eigen values which are approximately 2.4, 5.5, 8.6, 11.8 and so on. These disappearances of carrier for specific values of m_f , form a handy basis for measuring deviation.

BLOCK DIAGRAM:-

The audio oscillator supplies the information signal and could, if we wish, be replaced by a microphone and AF amplifier to provide speech and music instead of the sine wave signals that we are using with ST2203. The FM modulator is used to combine the carrier wave and the information signal in much the same way as in the AM transmitter. The only difference in this case is that the generation of the carrier wave and the modulation process is carried out in the same block. It is not necessary to have the two processes in same block, but in our case, it is. The output amplifier increases the power in the signal before it is applied to the antenna for transmission just as it did in the corresponding block in the FM transmitter.

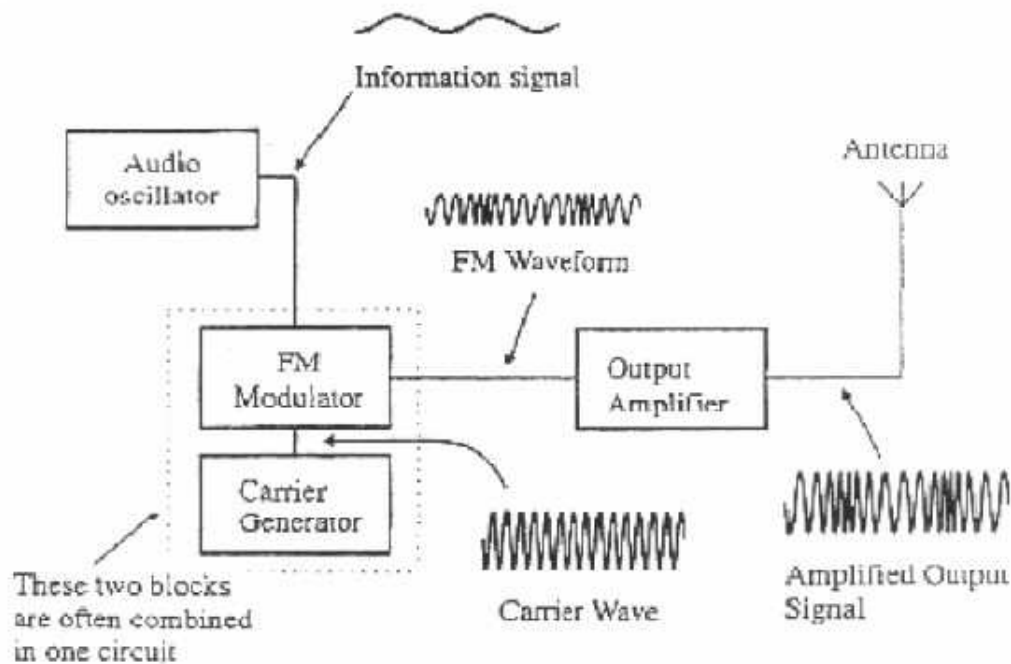


Figure 1: FM Transmitter

Controlling the VCO :

COMMUNICATION SYSTEMS LAB (EE-226-F)

To see how the VCO is actually controlled, let us assume that it is running at the same frequency as an un-modulated input signal. The input signal is converted into a square wave and, together with the VCO output, forms the two inputs to an Exclusive - OR gate. Remember that the Exclusive - OR gate provides an output whenever the two inputs are different in value and zero output whenever they are the same. The provided an output from the Exclusive -OR gate with an on-off ratio of unity and an average voltage at the output of half of the peak value. Now let us assume that the FM signal at the input decreases in frequency (see fig. 34). The period of the 'squared up' FM signal increases and the mean voltage level from the Exclusive -OR gate decreases. The mean voltage level is both the demodulated output and the control voltage for the VCO. The VCO frequency will decrease until its frequency matches the incoming FM signal.

PROCEDURE:--

1. Ensure that the following initial conditions exist on the ST2202 board.
 - a. All switched faults off.
 - b. Amplitude pot (in mixer amplifier block) in fully clockwise position.
 - c. VCO switch in 'ON' position.
2. Turn the audio oscillator block's amplitude pot to its fully clockwise position, and examine the block's output t.p.1 on an oscilloscope. This is the audio frequency sine wave, which will be used as our modulating signal. Note that the sine wave's frequency can be adjusted from about 300Hz to approximately 3.4 KHz, by adjusting the audio oscillator's frequency pot.
3. Connect the output of audio oscillator to VCO section's MOD In socket.
4. Turn ON the power supply.
5. Observe the modulating signal and modulated output at the VCO's MOD OUT socket by using CRO.
4. Calculate $m_f = \delta / f_m$.
5. Vary the modulating frequency keeping carrier freq constant and repeat steps 3 & 4.
6. Vary the carrier frequency keeping modulator freq constant and repeat steps 3 & 4.
7. Tabulate the results.

OBSERVATION TABLE:-

1.				
2.				
3.				

SAMPLE CALCULATION:-

$$\begin{aligned}m_f &= \delta / f_m \\&= 2 \times 8.3 \times 10^3 / 1000 \\&= 16.6\end{aligned}$$

WAVEFORMS:-

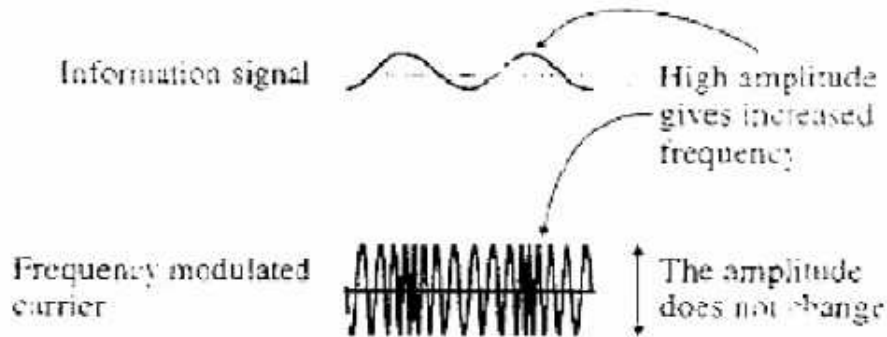


Figure 2: Modulating and FM Modulated signal

RESULT:-

Frequency modulated wave using VCO is obtained on CRO and m_f is calculated.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.
2. Before connecting the power supply plug into socket, ensure power supply should be switched off
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.
5. Power supply should be switched off after completion of experiment.

QUIZ / ANSWERS:-

- Q.1. How many types of FM are there? Write their names.
Ans. There two types of FM i.e. narrow band FM and wideband FM.
- Q.2. What frequency deviation in FM?
Ans. The maximum change in instantaneous frequency from the average is known as frequency deviation.
- Q.3. Which is the useful parameter for determination of bandwidth?
Ans. Frequency deviation is the useful parameter for determination of bandwidth.
- Q.4. How many sidebands are there in FM?
Ans. Theoretically, number sidebands in FM are infinite.
- Q.5. Which sidebands are ignored in FM?
Ans. The sidebands with small amplitude are ignored in FM.
- Q.6. Which are significant sidebands?
Ans. The sidebands having considerable amplitudes i.e. more than or equal to 1% of the carrier amplitude are known as significant sidebands.
- Q.7. what is CCIR?
Ans. CCIR stands for Consultative Committee for International Radio.
- Q.8. What is the indirect method of FM generation?
Ans. Armstrong method.

Q.9. What is the direct method of FM generation?

Ans. The parameter variation method.

Q.10. What is VCO?

Ans. VCO stands for voltage controlled oscillator whose frequency is controlled by modulating voltage.

EXPERIMENT No.5

AIM:- To generate FM signal using Varactor & reactance modulation.

APPARATUS REQUIRED:-(i)C.R.O. (ii) CRO Probe (ii) FM Modulation and Demodulation Trainer (ST 2203) (iv) Connecting leads

THEORY:

FM Using Varactor Modulator:

The variations in capacitance form part of the tuned circuit that is used to generate the FM signal to be transmitted. Varactor modulator is shown in fig 1.

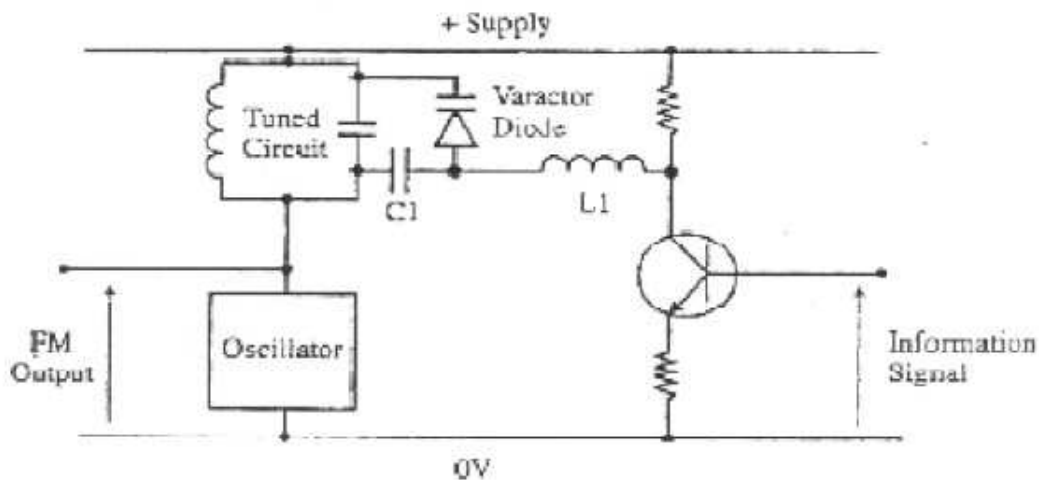


Figure 1: FM generation using Varactor Modulator

We can see the tuned circuit which sets the operating frequency of the oscillator and the varactor which is effectively in parallel with the tuned circuit. Two other components which may not be immediately obvious are C1 and L1. C1 is a DC blocking capacitor to provide DC isolation between the oscillator and the collector of the transmitter. L1 is an RF choke which allows the information signal through to the varactor but blocks the RF signals.

The operation of the varactor modulator:

1. The information signal is applied to the base of the input transistor and appears amplified and inverted at the collector.
2. This low frequency signal passes through the RF choke and is applied across the varactor diode.
3. The varactor diode changes its capacitance in sympathy with the information signal and therefore changes the total value of the capacitance in the tuned circuit.
4. The changing value of capacitance causes the oscillator frequency to increase and decrease under the control of the information signal. The output is therefore a FM signal. Before we start the study of varactor/ reactance modulation techniques we shall study a simple VCO circuit.

Simply connect the audio output to the socket labeled VCO modulation in and observe the FM modulated waveform on the oscilloscope at the VCO modulation out terminal. Keep the amplitude of audio output to approx 4 V p-p and frequency 2 kHz approx. Observe a stable FM modulated waveform on CRO Now turn the timebase speed of CRO little higher and you will observe the same waveforms as under (like Bessel function).

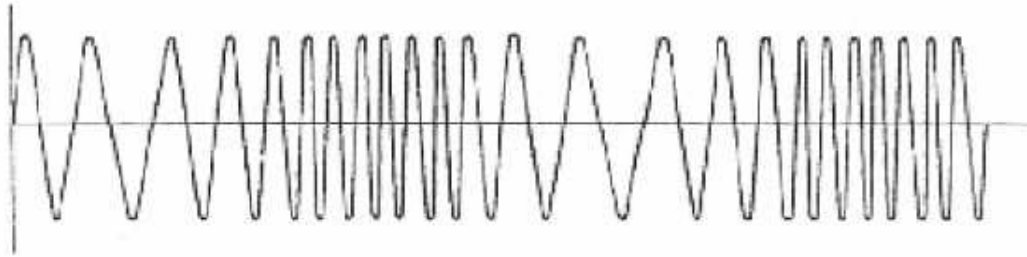


Figure 2: FM modulated wave

Now disconnect the audio amplifier's output from modulation IN and connect it to audio IN, keep the reactance/varactor switch in varactor position. Observe the output of mixer / amplifier circuit. Keep the oscilloscope in X10 position now observe the full waveform by shifting the X position. It is as shown in fig. Mark the resemblance between the output of VCO and the Varactor modulator. They are same. The freq. modulation in VCO was more because the Frequency difference between the carrier and the modulating signal was very less.

FM Using Reactance Modulator: In fig. 3, the left hand half is the previous varactor modulator simply an oscillator and a tuned circuit, which generates the un-modulated carrier. The capacitor C and the resistor R are the two components used for the phase shifting, and together with the transistor, form the voltage controlled capacitor. This voltage-controlled capacitor is actually in parallel with the tuned circuit. This is not easy to see but figure 18 may be helpful. In the first part of the figure, the capacitor and associated components have been replaced by the variable capacitor, shown dotted.

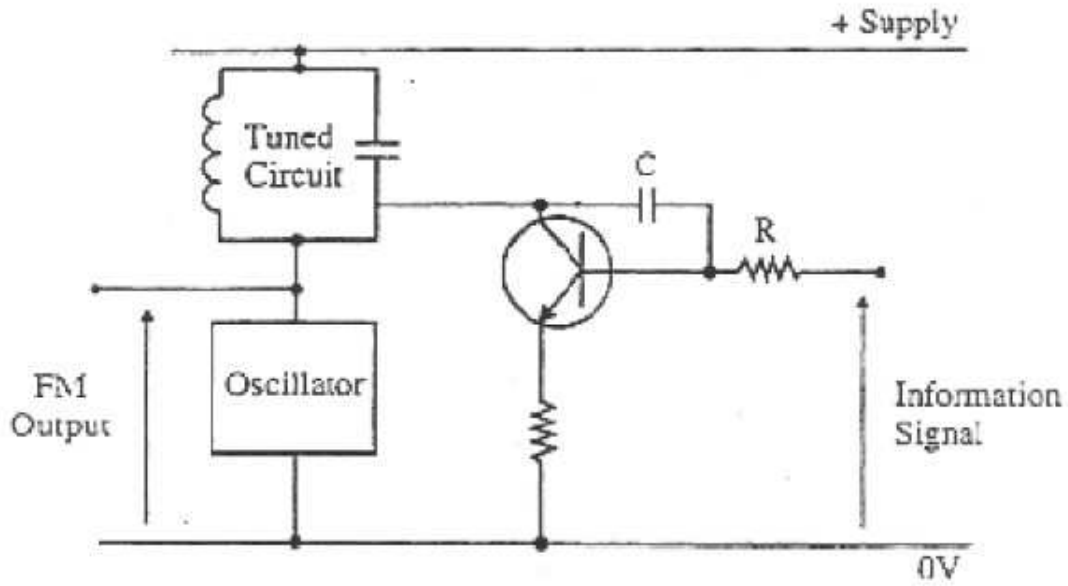


Figure 3: FM using Reactance Modulation.

In the next part, the two supply lines are connected together. We can justify this by saying that the output of the DC power supply always includes a large smoothing capacitor to keep the DC voltages at a steady value. This large capacitor will have a very low reactance at the frequencies being used in the circuit less than a milliohm. We can safely ignore this and so the two supply lines can be assumed to be joined together. Remember that this does not affect the DC potentials, which remain at the normal supply voltages. If the two supply voltages are at the same AC potential, the actual points of connection do not matter and so we can redraw the circuit as shown in the third part.

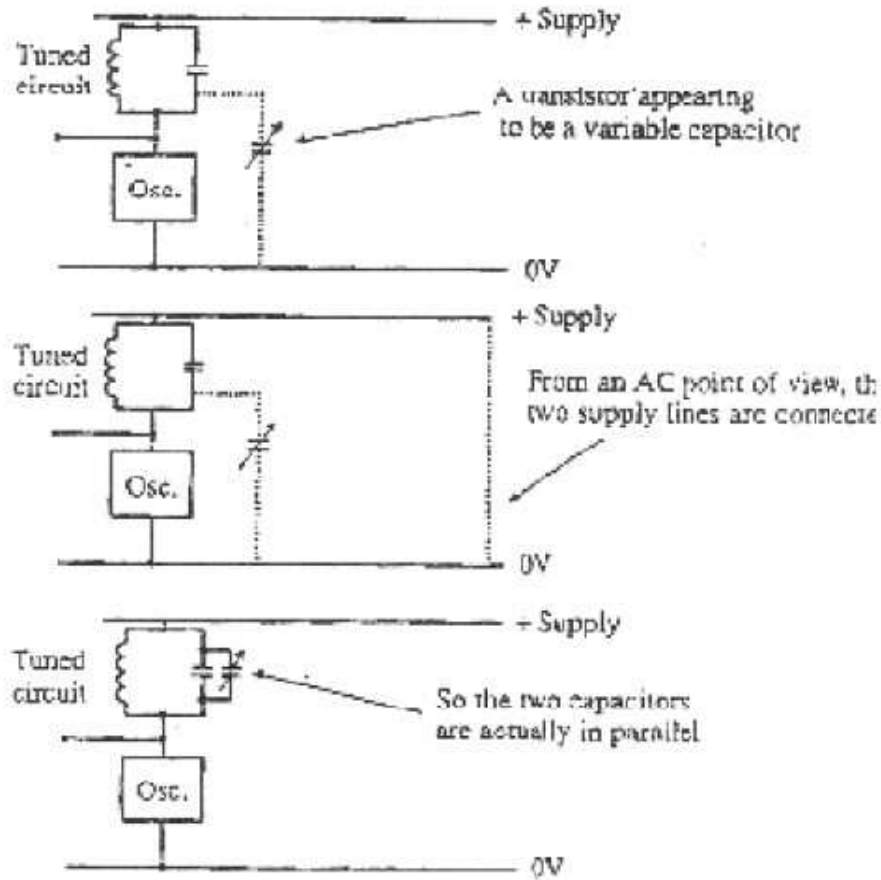


Figure 4: VCO using capacitor

Operation of the Reactance Modulator :

1. The oscillator and tuned circuit provide the un-modulated carrier frequency and this frequency is present on the collector of the transistor.
2. The capacitor and the resistor provide the 90° phase shift between the collector voltage and current. This makes the circuit appear as a capacitor.
3. The changing information signal being applied to the base has the same effect as changing the bias voltage applied to the transistor and, this would have the effect of increasing and decreasing the value of this capacitance.
4. As the capacitance is effectively in parallel with the tuned circuit the variations in value will cause the frequency of resonance to change and hence the carrier frequency will be varied in sympathy with the information signal input.

BLOCK DIAGRAM:-

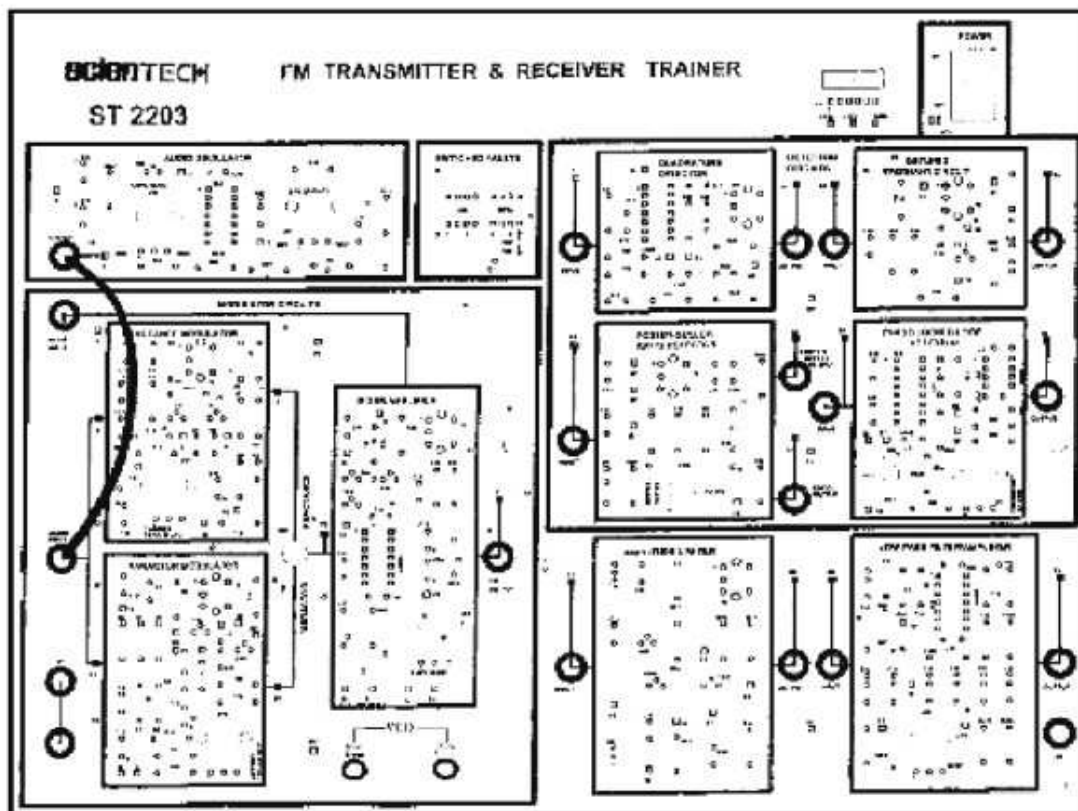


Figure 3: Block Diagram of FM Trainer kit

PROCEDURE:-

1. Ensure that the following initial conditions exist on the ST2202 board.
 - a. All switched faults off.
 - b. Amplitude pot (in mixer amplifier block) in fully clockwise position.
 - c. VCO switch (in phase locked loop detector block) in 'OFF' position.
 2. Make the connections as shown in fig 3.
 3. Switch 'on' the power.
 4. Turn the audio oscillator block's amplitude pot to its fully clockwise position, and examine the block's output t.p.1 on an oscilloscope. This is the audio frequency sine wave, which will be used as our modulating signal. Note that the sine wave's frequency can be adjusted from about 300Hz to approximately 3.4KHz, by adjusting the audio oscillator's frequency pot. Note also that the amplitude of this modulating signal is adjusted by audio oscillator amplitude pot. Leave the amplitude pot in min. position.
 5. Connect the output socket of the audio oscillator block to the audio input socket of the modulator circuit's block.
- For FM Varactor Modulator
6. Set the reactance / varactor switch to the varactor position. This switch selects the varactor modulator and also disables the reactance modulator to prevent any interference between the two circuits.
 7. The output signal from the varactor modulator block appears at t.p. 24 before being buffered and amplified by the mixer / amplifier block, any capacitive loading (e.g. due

to oscilloscope probe) may slightly affect the modulators output frequency. In order to avoid this problem we monitor the buffered FM output signal the mixer / amplifier block at t.p.34.

8. Put the varactor modulator's carrier frequency pot in its midway position, and then examine t.p. 34. Note that it is a sine wave of approximately 1.2 Vp-p, centered on 0V. This is our FM carrier, and it is un-modulated since the varactor modulators audio input signal has zero amplitude.

9. The amplitude of the FM carrier (at t.p.34) is adjustable by means of the mixer/amplifier block's amplitude pot, from zero to its pot level. Try turning this pot slowly anticlockwise, and note that the amplitude of the FM signal can be reduced to zero. Return the amplitude pot to its fully clockwise position.

10. Try varying the carrier frequency pot and observe the effects.

11. Also, see the effects of varying the amplitude and frequency pots in the audio oscillator block.

12. Turn the carrier frequency pot in the varactor modulator block slowly clockwise and note that in addition to the carrier frequency increasing there is a decrease in the amount of frequency deviation that is present.

13. Return the carrier frequency pot to its midway position, and monitor the audio input (at t.p.6) and the FM output (at p.34) triggering the oscilloscope on the audio input signal. Turn the audio oscillator's amplitude pot throughout its range of adjustment, and note that the amplitude of the FM output signal does not change. This is because the audio information is contained entirely in the signals frequency and not in its amplitude.

14. By using the optional audio input module ST2108 the human voice can be used as the audio modulating signal, instead of using ST2203's audio oscillator block. If you have an audio input module, connect the module's output to the audio input socket in the modulator circuit's block. The input signal to the audio input module may be taken from an external microphone be (supplied with the module) or from a cassette recorder, by choosing the appropriate switch setting on the module.

For FM Reactance Modulator:

6. Put the reactance /varactor switch in the reactance position. This switches the output of the reactance modulator through to the input of the mixer/amplifier block ~ and also switches off the varactor modulator block to avoid interference between the two modulators.

7. The output signal from the reactance modulator block appears at tp.13, before being buffered and amplified by the mixer/amplifier block. Although the output from the reactance modulator block can be monitored directly at tp.13, any capacitive loading affect this point (e.g. due to an oscilloscope probe) may slightly affect the modulator's output frequency. In order to avoid this problem we will monitor the buffered FM output signal from the mixer/amplifier block at t.p. 34.

8. Put the reactance modulator's pot in its midway position (arrow pointing towards top of PCB) then examine t.p. 34. Note that the monitored signal is a sine wave of approximately 1.2V peak/peak centered on 0 volts D.C. This is our FM carrier, and it is presently un-modulated since the reactance modulator's audio input signal has, zero amplitude.

9. The amplitude of the FM carrier (at t.p.34) is adjustable by means of the mixer/amplifier block's amplitude pot, from zero to its present level. Try turning this pot

slowly anticlockwise, and note that the amplitude of the FM signal can be reduced to zero. Return the amplitude pot to its fully clockwise position.

10. The frequency of the FM carrier signal (at t.p.34) should be approximately 455Khz at the moment This carrier frequency can be varied from 453Khz to 460Khz (approx.) by adjusting the carrier frequency pot in the reactance modulator block. Turn this pot over its range of adjustment and note that the frequency of the monitored signal can be seen to vary slightly. Note also that the carrier frequency is maximum when the pot is in fully clockwise position.

11. Try varying the amplitude & frequency pot in audio oscillators block, and also see the effect of varying the carrier frequency pot in the mixer/amplifiers block.

12. Monitor the audio input (at t.p.6) and the FM output (at t.p.34) triggering the oscilloscope on the audio input signal. Turn the audio oscillator's amplitude pot throughout its range of adjustment and note that the amplitude of the FM output signal does not change. This is because the audio information is contained entirely in the signal's frequency, and not in its amplitude.

RESULT:-

Frequency modulated signal is generated by using varactor and reactance modulator.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.
2. Before connecting the power supply plug into socket, ensure power supply should be switched off
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.
5. Power supply should be switched off after completion of experiment.

QUIZ / ANSWERS:-

Q.1. How many types of FM are there? Write their names.

Ans. There two types of FM i.e. narrow band FM and wideband FM.

Q.2. What frequency deviation in FM?

Ans. The maximum change in instantaneous frequency from the average is known as frequency deviation.

Q.3. Which is the useful parameter for determination of bandwidth?

Ans. Frequency deviation is the useful parameter for determination of bandwidth.

Q.4. How many sidebands are there in FM?

Ans. Theoretically, infinite numbers of sidebands are there in FM.

Q.5. Which sidebands are ignored in FM?

Ans. The sidebands with small amplitude are ignored in FM.

Q.6. Which are significant sidebands?

Ans. The sidebands having considerable amplitudes i.e. more than or equal to 1% of the carrier amplitude are known as significant sidebands.

Q.7. What is CCIR?

Ans. CCIR stands for Consultative Committee for International Radio.

Q.8. What is the indirect method of FM generation?

Ans. Armstrong method.

Q.9. Classify FM on the basis of bandwidth.

Ans. Narrowband and wideband FM.

Q.10. Which one is better in terms of noise immunity AM or FM?

Ans. FM.

EXPERIMENT No.6

AIM:- To Detect FM Signal using PLL & Foster-Seelay method.

APPARATUS REQUIRED:- (i) C.R.O. (ii) CRO Probe (ii) FM Modulation and Demodulation Trainer (ST 2203) (iv) Connecting leads

THEORY:-

A FM receiver is very similar to an AM receiver. The most significant change is that the demodulator must now extract the information signal from a frequency rather than amplitude modulated wave.

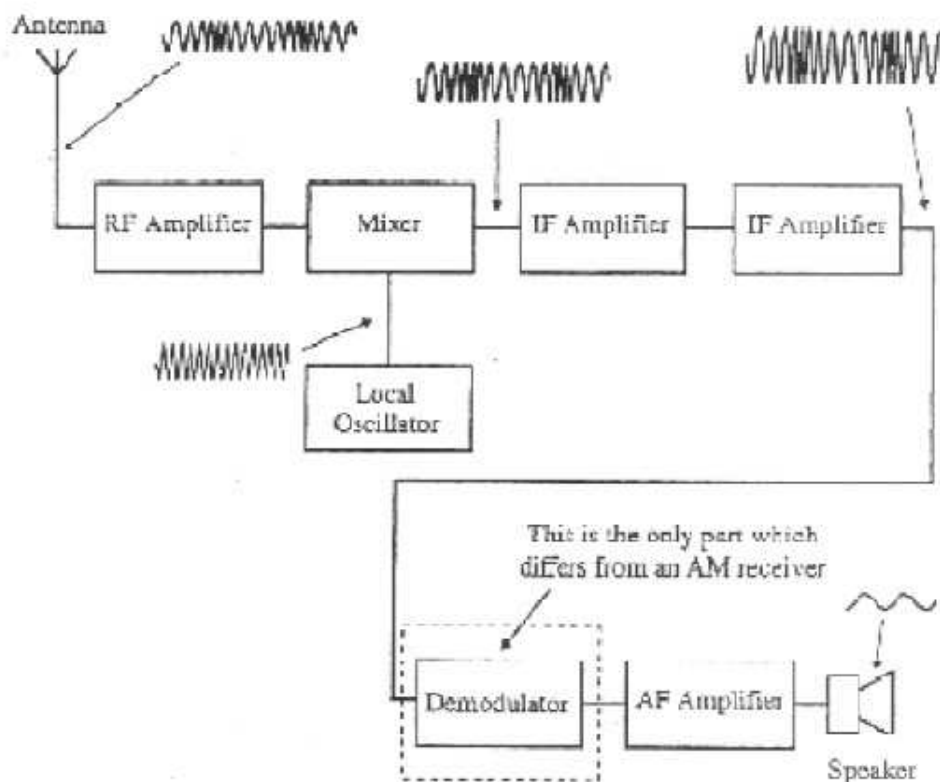


Figure 1: FM Receiver

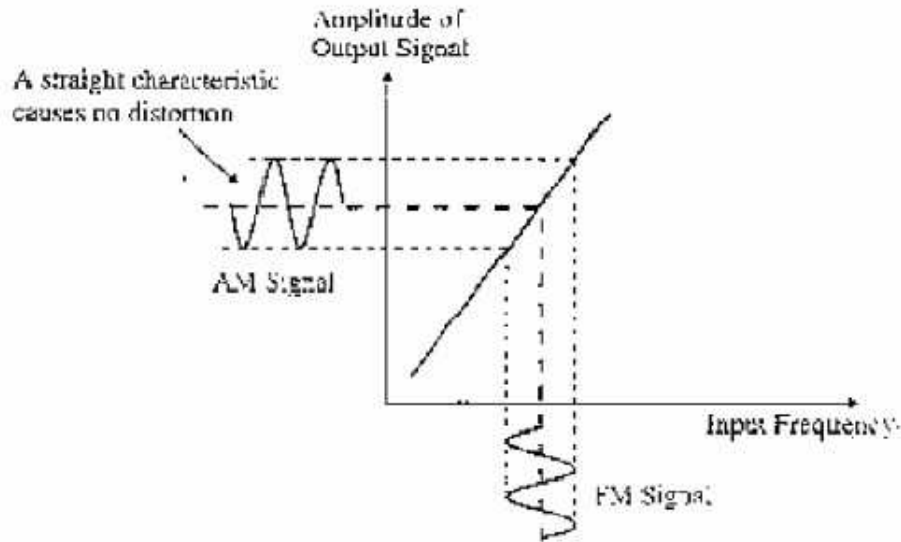


Figure 2: Voltage/Frequency Characteristics.

The basic requirement of any FM demodulator is therefore to convert frequency changes into changes in voltage, with the minimum amount of distortion. To achieve this, it should ideally have a linear voltage/frequency characteristic, similar to that shown in figure 2. A 'demodulator' can also be called a 'discriminator' or a 'detector'.

PHASE LOCK LOOP DETECTOR

~~This is a demodulator that employs~~ a phase comparator circuit. It is a very good demodulator and has the advantage that it is available, as a self-contained integrated circuit so there is no setting up required. You plug it in and it works. For these reasons, it is often used in commercial broadcast receivers. It has very low levels of distortion and is almost immune from external noise signals and provides very low levels of distortion. Altogether it is a very nice circuit.

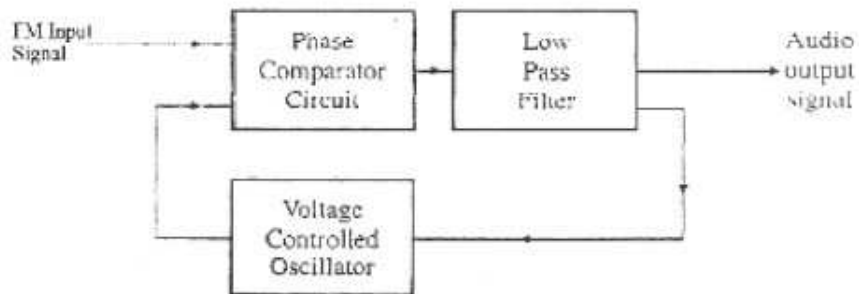


Fig. 3 Phase Lock Loop Detector

The overall action of the circuit may, at first, seem rather pointless. As we can see in fig 3, there is a voltage-controlled oscillator (VCO). The DC output voltage from the output of the low pass filters controls the frequency of this oscillator. Now this DC voltage keeps the oscillator running at the same frequency as the original input signal and 90° out of phase. And if we did, then why not just add a phase shifting circuit at the input to give the 90° phase shift? The answer can be seen by imagining what happens when the input frequency changes - as it would with a FM signal. If the input frequency increases

and decreases, the VCO frequency is made to follow it. To do this, the input control voltage must increase and decrease. These change of DC voltage level that forms the demodulated signal. The AM signal then passes through a signal buffer to prevent any loading effects from disturbing the VCO and then through an audio amplifier if necessary. The frequency response is highly linear as shown in figure 2.

FOSTER SEELEY DETECTOR

The foster Seeley circuit is shown in fig. 4. At first glance, it looks rather complicated but it becomes simpler if we consider it a bit at a time.

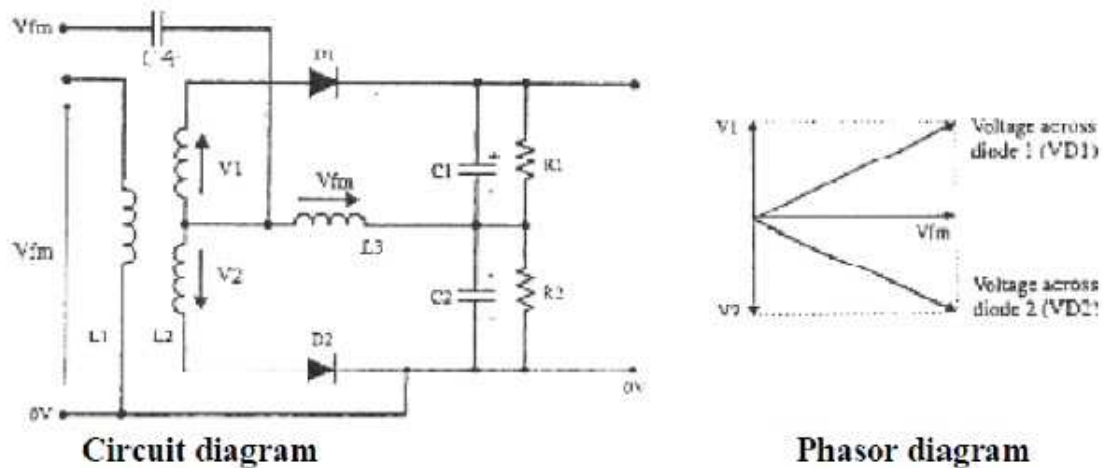


Figure 4: Foster –Seelay Detector

When the input signal is un-modulated: We will start by building up the circuit a little at a time. To do this, we can ignore many of the components we may recognize immediately that it consists of two envelope detectors like half wave rectifiers are fed from the center-tapped coil L2. With reference to the center-tap, the two voltages V1 and V2 are in anti-phase as shown by the arrows. The output voltage would be zero volts since the capacitor voltages are in anti-phase and are equal in magnitude. After adding two capacitors: The next step is to add two capacitors and see their effect on the phase of the signals. L1 and L2 are magnetically tightly coupled and by adding C3 across the centre-tapped coil, they will form a parallel tuned circuit with a resonance frequency equal to the un-modulated carrier frequency. Capacitor C5 will shift the phase of the input signal by 90° with reference to the voltage across L1 and L2. The voltages are shown as Va and Vb in the phasor diagram given in figure 39. Using the input signal Vfm as the reference, the phasor diagrams now look the way shown in figure 4. C4 is not important. It is only a DC blocking capacitor and has negligible impedance at the frequencies being used. But what it does do is to supply a copy of the incoming signal across L3. The entire incoming signal is dropped across L3 because C1 and C2 also have negligible impedance. If we return to the envelope detector section, we now have two voltages being applied to each diode. One is V1 or V2 and the other is the new voltage across L3, which is equal to Vfm. When the input Frequency changes: If the input frequency increased above its un-modulated value, the phasor of Va would fall below 90° due to the parallel tuned circuit becoming increasingly capacitive. This would result in a larger total voltage being applied across D1 and a reduced voltage across D2. Since the capacitor C1 would now charge to a higher voltage, the final output from the circuit would be a positive voltage. Conversely, if the frequency of the

FM input signal decreased below the unmodulated value, the phase shift due to capacitor C5 increases above 90° as the parallel tuned circuit becomes slightly inductive. This causes the voltage across diode D2 to increase and the final output from the demodulator becomes negative. The effect of noise is to change the amplitude of the incoming FM signal resulting in a proportional increase and decrease in the amplitude of diode voltages VD1 and VD2 and the difference in voltage is the demodulated output, the circuit is susceptible to noise interference and should be preceded by a noise limiter circuit.

BLOCK DIAGRAM:-

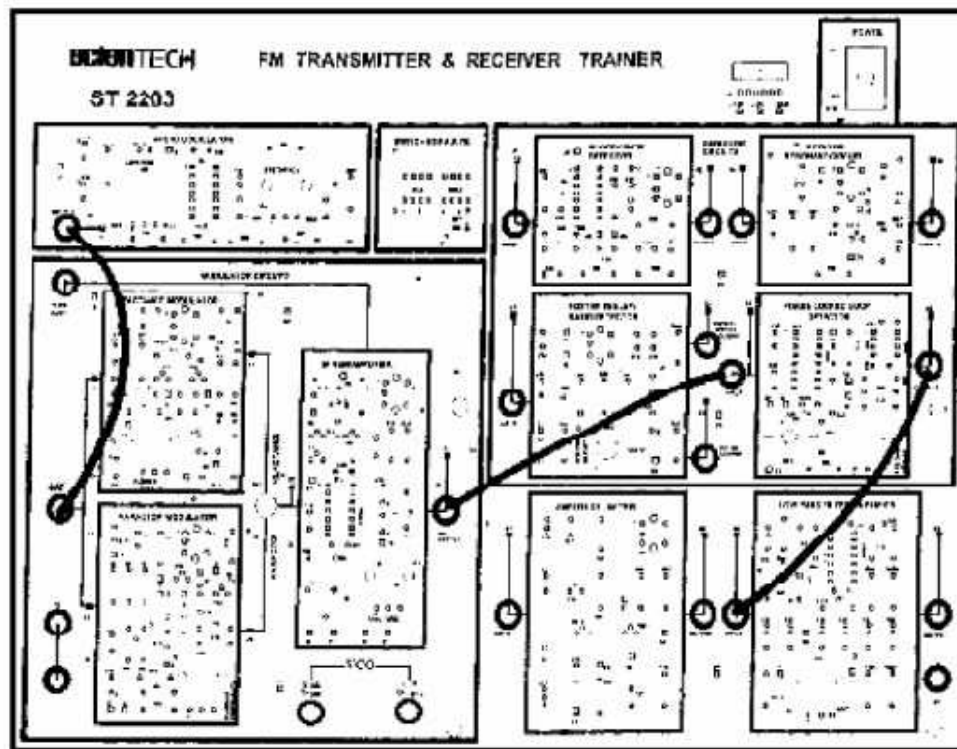


Figure 5: Connections for FM Demodulation using PLL

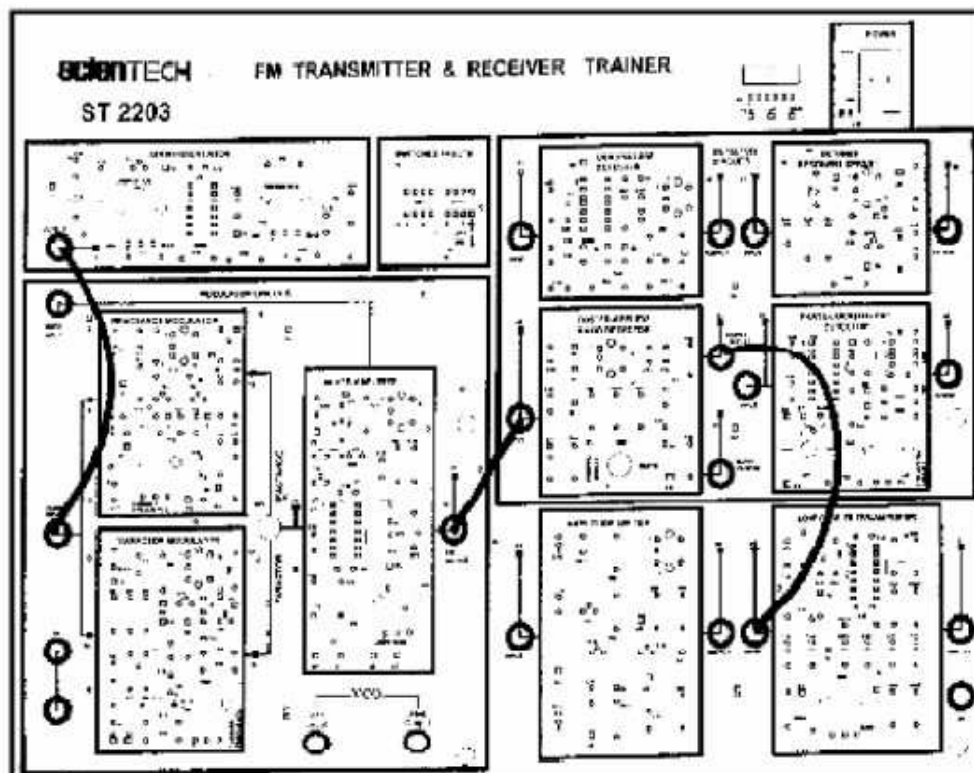


Figure 6: Connections for FM Demodulation using Foster-Seely Detector

PROCEDURE:-

FM Detection using PLL:

1. Ensure that the following initial conditions exist on the ST2203 module:
 - a. All switched faults OFF;
 - b. Audio amplifier block's amplitude pot in fully clockwise (MAX) position.
 - c. Audio amplifier block's frequency pot in fully counter clockwise.
 Ensure that the following initial conditions exist on the ST2203 clockwise (MIN) position.
 - d. Amplitude pot (in the mixer/amplifier block) in fully clockwise position;
 - e. VCO switch (in phase-locked loop detector block) in ON position.
2. Make the connections shown in figure 5.
3. Turn on power to the ST2203 module.
4. Now monitor the audio input signal to the varactor modulator block (at t.p.14) together with the output from the phase-locked loop detector block (at t.p.60), triggering the oscilloscope in t.p.14. The signal at t.p.68 should contain three components:
 - A positive D.C. offset voltage.
 - A sine wave at the same frequency as the audio signal at t.p.14.
 - A high - frequency ripple component.

5. The low pass filter/amplifier block strongly attenuates the high-frequency ripple component at the detector's output and also blocks the D.C. offset voltage. Consequently the signal at the output of the low-pass filter/amplifier block (at t.p.73) should be very closely resemble the original audio making signal, if not then slowly adjust the freq. adjust pot of PLL block.

6. Adjust the audio oscillator block's amplitude and frequency pots, and compare the original audio signal with the final demodulated signal.

FM Detection using Foster-Seelay Detector:

1. Ensure that the following initial conditions exist on the ST2203 module:

- a. All switched faults OFF;
- b. Audio amplifier block's amplitude pot in fully clockwise (MAX) position.
- c. Audio amplifier block's frequency pot in fully counter-clockwise (MIN) position.
- d. Amplitude pot (in the mixer/amplifier block) in fully clockwise position.
- e. VCO switch (in phase-locked loop detector block) in OFF position.

2. Make connection as shown in figure 42

3. Turn on power to the ST2203 module.

4. We will now investigate the operation of the foster-Seeley detector on the ST2203 module. In the Foster-Seeley / ratio detector block, select the Foster-Seeley detector by putting the switch in the Foster-Seeley position.

5. Initially, we will use the varactor modulator to generate our FM signal, since this is the more linear of the two modulators, as fast as its frequency/voltage characteristic is concerned. To select the varactor modulator, put the reactance/ varactor switch in the varactor position. Ensure that the varactor modulator's carrier frequency pot is in the midway position.

6. The audio oscillator's output signal (which appears at t.p.1) is now being used by the varactor modulator, to frequency-modulate a 455Khz carrier sine wave. As we saw earlier, this FM waveform appears at the FM output socket from the mixer/amplifier block. You will probably need to have an X-expansion control on your oscilloscope.

7. Now monitor the audio input signal to the varactor modulator block (at t.p. 14) together with the foster-seeley output from the foster-seeley/ratio detector block (at t.p. 52), triggering the oscilloscope on t.p. 14. The signal at t.p. 52 should contain two components:

- A sine wave at the same frequency as the audio signal at t.p. 14.
- A High frequency ripple component of small amplitude.

8. The low-pass filter/amplifier strongly attenuates this high-frequency ripple component, and blocks any small D.C. offset voltage that might exist at the detector's output. Consequently, the signal at the output of the low-pass filter/ amplifier block (at t.p. 73) should very closely resemble the original audio modulating signal.

9. Monitor the audio input to the varactor modulator (at t.p. 14) and the output of the low pass filter / amplifier block (at t.p. 73) and adjust the gain pot (in the low pass filter/ amplifier block) until the amplitudes of the monitored audio waveforms are the same.

10. Adjust the audio oscillator block's amplitude and frequency pots, and compare the original audio signal with the final demodulated signal.

RESULT:-

FM signal is being demodulated by using PLL and Foster-Seelay Method.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.
2. Before connecting the power supply plug into socket, ensure power supply should be switched off
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.
5. Power supply should be switched off after completion of experiment.

QUIZ / ANSWERS:-

Q.1. How many types of FM are there? Write their names.

Ans. There two types of FM i.e. narrow band FM and wideband FM.

Q.2. What frequency deviation in FM?

Ans. The maximum change in instantaneous frequency from the average is known as frequency deviation.

Q.3. Which is the useful parameter for determination of bandwidth?

Ans. Frequency deviation is the useful parameter for determination of bandwidth.

Q.4. What are different methods of FM detection?

Ans. 1. Slope detection method 2. Phase detection method.

Q.5. Which sidebands are ignored in FM?

Ans. The sidebands with small amplitude are ignored in FM.

Q.6. Which are significant sidebands?

Ans. The sidebands having considerable amplitudes i.e. more than or equal to 1% of the carrier amplitude are known as significant sidebands.

Q.7. What is basic principle of FM detection?

Ans. Conversion of frequency variations into amplitude variations.

Q.8. What is the indirect method of FM generation?

Ans. Armstrong method.

Q.9. What is the direct method of FM generation?

Ans. The parameter variation method.

Q.10. What is the function of amplitude limiter?

Ans. It suppresses the undesirable amplitude fluctuation in generated FM signal.

EXPERIMENT No.7

AIM:- To Study Super heterodyne AM receiver and measurement of receiver parameters viz. sensitivity, selectivity & fidelity.

APPARATUS REQUIRED:- (i) C.R.O. (ii) CRO Probe (iii) DSB/SSB Transmitter (ST 2201) and Receiver Trainer (ST 2202) (iv) Connecting leads

BLOCK DIAGRAM:-

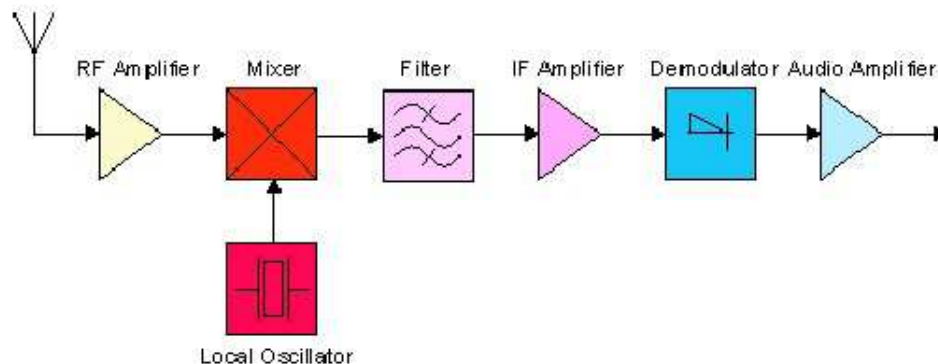


Figure 1: Superheterodyne Receiver

THEORY:-

The principle of operation of the superheterodyne receiver depends on the use of heterodyning or frequency mixing. The signal from the antenna is filtered sufficiently at least to reject the image frequency (see below) and possibly amplified. A local oscillator in the receiver produces a sine wave which mixes with that signal, shifting it to a specific intermediate frequency (IF), usually a lower frequency. The IF signal is itself filtered and amplified and possibly processed in additional ways. The demodulator uses the IF signal rather than the original radio frequency to recreate a copy of the original modulation (such as audio).

Fig.1. shows the minimum requirements for a single-conversion superheterodyne receiver design. The following essential elements are common to all superhet circuits: a receiving antenna, a tuned stage which may optionally contain amplification (RF amplifier), a variable frequency local oscillator, a frequency mixer, a band pass filter and intermediate frequency (IF) amplifier, and a demodulator plus additional circuitry to amplify or process the original audio signal (or other transmitted information). To receive a radio signal, a suitable antenna is required. This is often built into a receiver; especially in the case of AM broadcast band radios. The output of the antenna may be very small, often only a few microvolts. The signal from the antenna is tuned and may be amplified in a so-called radio frequency (RF) amplifier, although this stage is often omitted. One or more tuned circuits at this stage block frequencies which are far removed from the intended reception frequency. In order to tune the receiver to a particular station, the frequency of the local oscillator is controlled by the tuning knob

(for instance). Tuning of the local oscillator and the RF stage may use a variable capacitor, or varicap diode. The tuning of one (or more) tuned circuits in the RF stage must track the tuning of the local oscillator.

Mixer stage: The signal is then fed into a circuit where it is mixed with a sine wave from a variable frequency oscillator known as the local oscillator (LO). The mixer uses a non-linear component to produce both sum and difference beat frequencies signals, each one containing the modulation contained in the desired signal. The output of the mixer may include the original RF signal at f_d , the local oscillator signal at f_{LO} , and the two new frequencies $f_d + f_{LO}$ and $f_d - f_{LO}$. The mixer may inadvertently produce additional frequencies such as 3rd- and higher-order intermodulation products. The undesired signals are removed by the IF bandpass filter, leaving only the desired offset IF signal at f_{IF} which contains the original modulation (transmitted information) as the received radio signal had at f_d .

Intermediate frequency stage: The stages of an intermediate frequency amplifier are tuned to a particular frequency not dependent on the receiving frequency; this greatly simplifies optimization of the circuit.^[6] The IF amplifier (or IF strip) can be made highly selective around its center frequency f_{IF} , whereas achieving such a selectivity at a much higher RF frequency would be much more difficult. By tuning the frequency of the local oscillator f_{LO} , the resulting difference frequency $f_{LO} - f_d$ (or $f_d - f_{LO}$ when using so-called low-side injection) will be matched to the IF amplifier's frequency f_{IF} for the desired reception frequency f_d . One section of the tuning capacitor will thus adjust the local oscillator's frequency f_{LO} to $f_d + f_{IF}$ (or, less often, to $f_d - f_{IF}$) while the RF stage is tuned to f_d . Engineering the multi-section tuning capacitor (or varactors) and coils to fulfill this condition across the tuning range is known as tracking. Other signals produced by the mixer (such as due to stations at nearby frequencies) can be very well filtered out in the IF stage, giving the superheterodyne receiver its superior performance. However, if f_{LO} is set to $f_d + f_{IF}$, then an incoming radio signal at $f_{LO} + f_{IF}$ will also produce a heterodyne at f_{IF} ; this is called the image frequency and must be rejected by the tuned circuits in the RF stage. The image frequency is $2f_{IF}$ higher (or lower) than f_d , so employing a higher IF frequency f_{IF} increases the receiver's image rejection without requiring additional selectivity in the RF stage. Usually the intermediate frequency is lower than the reception frequency f_d , but in some modern receivers (e.g. scanners and spectrum analyzers) it is more convenient to first convert an entire band to a much higher intermediate frequency; this eliminates the problem of image rejection. Then a tunable local oscillator and mixer convert that signal to a second much lower intermediate frequency where the selectivity of the receiver is accomplished. In order to avoid interference to receivers, licensing authorities will avoid assigning common IF frequencies to transmitting stations. Standard intermediate frequencies used are 455 kHz for medium-wave AM radio, 10.7 MHz for broadcast FM receivers, 38.9 MHz (Europe) or 45 MHz (US) for television, and 70 MHz for satellite and terrestrial microwave equipment.

Bandpass filter: The IF stage includes a filter and/or multiple tuned circuits in order to achieve the desired selectivity. This filtering must therefore have a band pass equal to or less than the frequency spacing between adjacent broadcast channels. Ideally a filter would have a high attenuation to adjacent channels, but maintain a flat response across the desired signal spectrum in order to retain the quality of the received signal. This may be obtained using one or more dual tuned IF transformers or a multipole ceramic crystal filter.

Demodulation: The received signal is now processed by the demodulator stage where the audio signal (or other baseband signal) is recovered and then further amplified. AM demodulation requires the simple rectification of the RF signal (so-called envelope detection), and a simple RC low pass filter to remove remnants of the intermediate frequency. FM signals may be detected using a discriminator, ratio detector, or phase-locked loop. Continuous wave (morse code) and single sideband signals require a product detector using a so-called beat frequency oscillator, and there are other techniques used for different types of modulation. The resulting audio signal (for instance) is then amplified and drives a loudspeaker. When so-called high-side injection has been used, where the local oscillator is at a higher frequency than the received signal (as is common), then the frequency spectrum of the original signal will be reversed. This must be taken into account by the demodulator (and in the IF filtering) in the case of certain types of modulation such as single sideband.

RECEIVER CHARACTERISTICS:

The important characteristics of receivers are sensitivity, selectivity, & fidelity described as follows:

Sensitivity:

The sensitivity of radio receiver is that characteristic which determines the minimum strength of signal input capable of causing a desired value of signal output. Therefore, expressing in terms of voltage or power, sensitivity can be defined as the minimum voltage or power at the receiver input for causing a standard output. In case of amplitude-modulation broadcast receivers, the definition of sensitivity has been standardized as "amplitude of carrier voltage modulated 30% at 400 cycles, which when applied to the receiver input terminals through a standard dummy antenna will develop an output of 0.5 watt in a resistance load of appropriate value substituted for the loud speaker".

Selectivity:

The selectivity of a radio receiver is that characteristic which determines the extent to which it is capable of differentiating between the desired signal and signal of other frequencies.

Fidelity:

This is defined as the degree with which a system accurately reproduces at its output the essential characteristics of signals which is impressed upon its input.

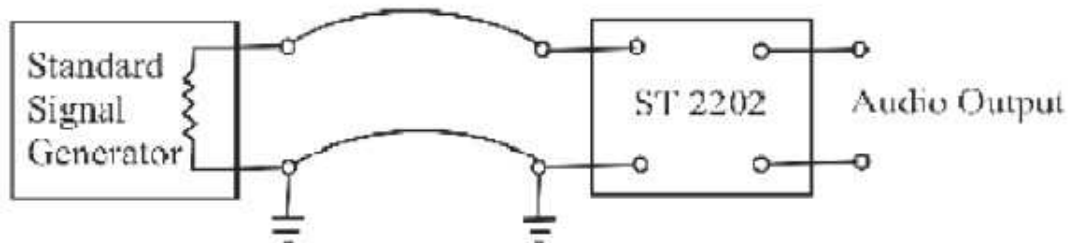


Figure 2: Setup for Determining Receiver Characteristics

Determination of receiver characteristics:

A laboratory method for the measurement of receiver characteristics is shown in Fig. 2. We use here an artificial signal to represent the voltage that is induced in the receiving antenna. This artificial signal is applied through 'dummy' antenna, which in association with the antenna with which the receiver is to be used. Substituting the resistance load of proper value for the loudspeaker and measuring the audio frequency power determine the receiver output.

Sensitivity:

Sensitivity is determined by impressing different RF voltages in series with a standard dummy antenna and adjusting the intensity of input voltage until standard outputs are obtained at resonance for various carrier frequencies. Sensitivity is expressed in microvolt.

Selectivity: Selectivity is expressed in the form of a curve that gives the carrier signal strength with standard modulation that is required to produce the standard test output plotted as a function of resonance of the test signal. The receiver is tuned to the desired frequency and manual volume control is set for maximum value. At standard modulation, the signal generator is set at the resonant frequency of the receiver. The carrier output of the signal generator is varied until the standard test output is obtained. At the same tuning of receiver, the frequency of signal generator is varied above and below the frequency to which the receiver is tuned. For every frequency, the signal generator voltage, applied to the receiver input, is adjusted to give the standard test output from the receiver.

Fidelity: Fidelity is the term expressing the behavior of receiver output with modulation frequency of input voltage. To obtain a fidelity curve, the carrier frequency of the signal generator is adjusted to resonance with the receiver, standard 400 cycles modulation is applied, the signal generator carrier level is set at a convenient arbitrary level and the manual volume control of the receiver is adjusted to give the standard test output. The modulation frequency is then varied over the audio range, keeping degree of modulation constant.

PROCEDURE:-

(a) To Plot Selectivity of Receiver:

1. Setting on ST2202

- Set the detector in diode mode.
- AGC on.
- Set the volume control full clockwise.

COMMUNICATION SYSTEMS LAB (EE-226-F)

2. Apply AM signal with 400 Hz modulating frequency and 30% modulation taken from AM generator into Rx input socket.
3. Set the input carrier frequency to suitable value that lies within the AM band (525 KHz - 1600 KHz). Also set signal level to 100mV.
4. Tune the Receiver using tuning control. Also adjust gain potentiometer provided in R.F. amplifier section of ST2202 so as to get unclipped demodulated signal at detector's output (output of audio amplifier).
5. Note the voltage level at receiver's final output stage i.e. audio amplifier's output on CRO (voltage at resonance (V_r)).
6. Now gradually offset the carrier frequency in suitable steps of 5 KHz or 10 KHz below and above the frequency adjusted in step 2 without changing the tuning of receiver while maintaining the input signal level.
7. Now record the signal level at output of audio amplifier for different input carrier frequency, on CRO (i.e. voltage off resonance (V_i)).
8. Tabulate the readings as under:

Carrier Frequency	Output Voltage	Ratio = $20 \log (V_i / V_r)$ dB

9. Plot the curve between ratio and carrier frequency.

(b) To Plot Sensitivity of Receiver:

1. Setting on ST2202 :
 - a. Set the detector in diode mode.
 - b. AGC on.
 - c. Set the volume control fully clockwise.
2. Apply AM signal, with 400 Hz modulating signal and 30% modulation, taken from AM generator into Rx input socket.
3. Set the input carrier frequency so as to lie within the AM Band (525 KHz – 1600 KHz). Also tune the detector to that carrier frequency using tuning control.(You will hear atone)
4. Set the input AM level to 100 mV. Also adjust the gain potentiometer provided in R.F. amplifier section of ST2202 so as to get unclipped demodulated signal at detectors output.
5. Record input carrier frequency & signal level at the final output stage i.e. output of audio amplifier (observed on CRO).
6. Change the input carrier frequency & also tune the receiver to that frequency & repeat step 4.
7. Tabulate the collected readings as under:

Carrier frequency	Output (pp)

8. Plot the graph between carrier frequency & output level.

(c) To Plot Fidelity of Receiver:

1. Setting on ST220:

- Set the detector in diode mode.
- AGC on.
- Set the volume control fully clockwise.

2. Apply AM signal of 100mV with 400Hz modulating signal and 30% modulation, into Rx input.

3. Select a suitable carrier frequency that lies within AM Band (525 KHz - 1600 KHz). Tune the ST2202 receiver to that frequency using tuning control. Also adjust gain potentiometer provided in R.F. amplifier section so as to get unclipped demodulated signal at detector's output.

4. Note the demodulated signal level (V_r) at the final output stage i.e. output of audio amplifier (on CRO) for the applied AM signal with 400Hz modulating signal.

5. Now vary the modulating signal frequency over audio range (300 Hz-3 KHz) in suitable steps say 100Hz. Note the corresponding output level (V_i) at the output of audio amplifier (on CRO).

6. Tabulate readings as under:

Carrier frequency	Modulating frequency	Output Voltage

$$\text{Relative response} = 20 \log (V_i / V_r) \text{ dB}$$

7. Plot the graph between modulating frequency and relative response.

RESULT:-

Superhetrodyne receiver has been studied and plot for receiver parameters viz. sensitivity, selectivity and fidelity has been drawn.

PRECAUTIONS:-

- Do not use open ended wires for connecting to 230 V power supply.
- Before connecting the power supply plug into socket, ensure power supply should be switched off
- Ensure all connections should be tight before switching on the power supply.
- Take the reading carefully.
- Power supply should be switched off after completion of experiment.

QUIZ / ANSWERS:-

Q. 1. What is amplitude modulation?

Ans. Amplitude Modulation is a process in which the amplitude of the carrier is made proportional to the instantaneous amplitude of the modulating signal.

Q. 2. Which are the three discrete frequencies in AM?

Ans. (1) Carrier frequency (2) lower sideband frequency (3) upper sideband frequency

Q. 3 What is receiver sensitivity?

Ans. The sensitivity of radio receiver is that characteristic which determines the minimum strength of signal input capable of causing a desired value of signal output.

Q. 4. What is receiver selectivity?

Ans. Selectivity is expressed in the form of a curve that give the carrier signal strength with standard modulation that is required to produce the standard test output plotted as a function off resonance of the test signal.

Q.6. What is receiver fidelity?

Ans. This is defined as the degree with which a system accurately reproduces at its output the essential characteristics of signals which is impressed upon its input.

Q.7. Which type of receiver is used in the experiment?

Ans. Superhetrodyne.

Q. 8. Where the modulation index lies?

Ans. modulation index always lies between 0 and 1. More than 1 is over modulation.

Q. 9. What happens in case of over modulation?

Ans. The wave will get distorted.

Q. 10. Mention a disadvantage of superhetrodyning?

Ans. Generation of Image frequency.

EXPERIMENT No.8 (a)

AIM:-To study Pulse Amplitude Modulation.

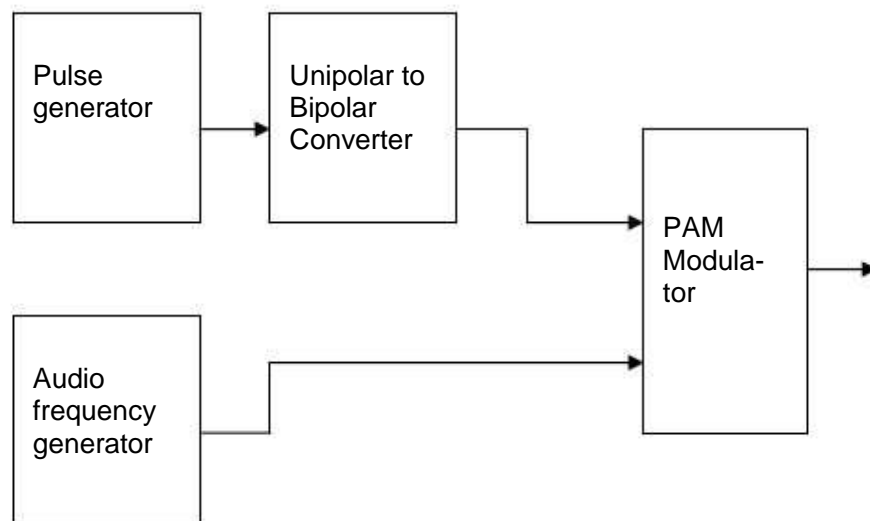
APPARATUS REQUIRED:-CRO, experimental kit, power supply, connecting leads.

THEORY:-

Pulse Modulation: We know that in Analog modulation systems, some parameter of a sinusoidal carrier (continuous in time domain) is varied according to the instantaneous value of the modulating signal. But in pulse modulation methods, the carrier is no longer a continuous signal but consists of a train of uniform pulses having a defined PRF (Pulse Repetition Frequency). The continuous modulating message signal waveforms are sampled at regular intervals. Information regarding the signal is transmitted only at the sampling times, together with any synchronizing pulses that may be required. At the receiving end, the original wave form may be reconstituted with negligible distortion from the information regarding the samples, if these samples are taken with minimum sufficient frequency.

In Pulse Modulation some parameter of the pulsed carrier is varied according to the instantaneous value of the modulating signal. Pulse modulation may be broadly subdivided into two categories: Analog & Digital. In the former, the indication of sample amplitude may be infinitely variable, while in the latter a code which indicates the sample amplitude to the nearest pre-determined level is sent. Pulse-Amplitude & Pulse-Time Modulation are both analog while the Pulse-code and Delta modulation are both digital. All the modulation systems have sampling in common, but they differ from each other in the manner of indicating the sample amplitude. In PAM the signal is sampled at a regular intervals and each sample is made proportional to the instant of sampling. In single polarity PAM is fixed, AC level is acted to ensure that all the pulse are +Ve going. The frequency spectrum is decaying but with decaying amplitude. The rate of decay depends upon the width of the pulses. As the pulses are made wider, the spectrum decays faster.

BLOCK DIAGRAM:-



WAVEFORM FORMS:-

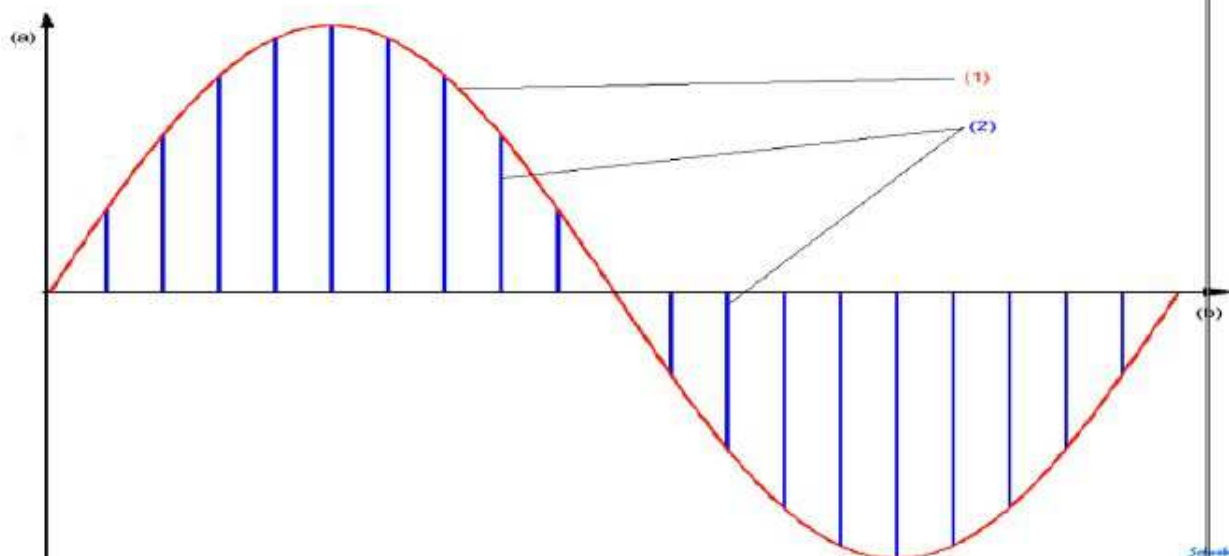


Fig 1: Principle of PAM; (1) original Signal, (2) PAM-Signal, (a) Amplitude of Signal, (b) Time

PROCEDURE:-

1. Make the connection according to the block diagram.
2. Connect pulse generator to the unipolar to bipolar converter

3. Connect the audio frequency of 2 KHz, 2V to modulator.
4. Connect the modulator output to CRO.
5. Observe output on CRO.

RESULT:-

Pulse modulated waveform is obtained on CRO.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.
2. Before connecting the power supply plug into socket, ensure power supply should be switched off.
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.
5. Power supply should be switched off after completion of experiment.

QUIZ / ANSWERS:-

Q.1. What is pulse amplitude modulation?

Ans. Amplitude of the sampled pulse is varied according to the modulating signal.

Q.2. How many types of pulse modulation?

Ans. There are two types of PM i.e. PAM and PTM.

Q.3. How many types of pulse time modulation?

Ans. There are two types of PTM i.e. PWM and PPM.

Q.4. What is base band system?

Ans. When the basic signal is transmitted without a frequency translation is known as base band signal.

Q.5. How many types of pulse amplitude modulation?

Ans. There are two types of PAM i.e. single polarity and double polarity.

Q.6. Write the demodulation method of Plato sample signal.

Ans. (1) using an equalizer (2) using holding circuit.

Q.7. Which filter is used in PAM demodulator circuit?

Ans. Second order low pass filter.

Q.8. Write the two method of multiplexing

Ans. (1) FDM (2) TDM

Q.9. What is direct method of PTM generation?

Ans. In the direct method of PTM generation wave form is generated without generating the PAM.

Q.10. What is the merit of flat top sampling?

Ans. The tops of pulses are flat thus the pulses have constant amplitude with in the pulse Interval.

EXPERIMENT No.8(b)

AIM:- To study the pulse width modulation.

APPARATUS REQUIRED:- CRO, experimental kit, power supply, connecting leads

THEORY:-

PWM is a part of PTM modulation. The PWM is also called PDM (pulse duration modulation) and sometimes it is also called PLM (pulse length modulation).

In PWM width of each pulse depends on the instantaneous value of the base band signal at the sampling instant. In pulse width modulation continuous waveform is sampled at regular intervals and the width of each pulse is kept proportional to the magnitude of signal at that instant in PWM. In pulse width modulation pulse is varied accordance with the modulating signal but the amplitude and starting time of each pulse is fixed. In PWM, the information about the base band signal lies in the trailing edge of the pulse

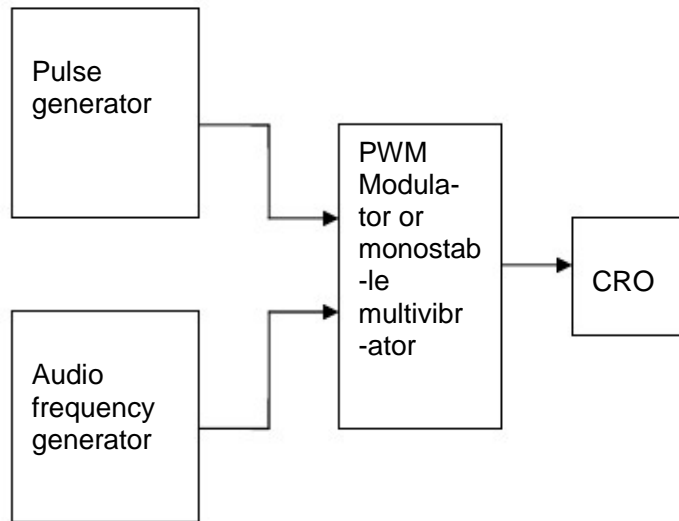
PWM has the disadvantage, when compared with PPM that its pulses are of varying width and therefore of varying power content. This means that transmitter must be powerful enough to handle the maximum-width pulses, although the average power transmitted is perhaps only half of the peak power. PWM still works if synchronization between transmitter and receiver fails.

Generation and Demodulation of PWM

PWM may be generated by applying trigger pulses to control the duration of these pulses. The emitter coupled mono-stable multi-vibrator is used as voltage to time converter, since its gate width is dependent on the voltage to which the capacitor C is charged. If this voltage is varied in accordance with a signal voltage, a series of rectangular pulses will be obtained, with widths varying as required. The demodulation of pulse width modulation is a simple process. PWM is fed to an integrating circuit from which a signal emerges whose amplitude at any time is proportional to the pulse width at that time.

BLOCK DIAGRAM:-

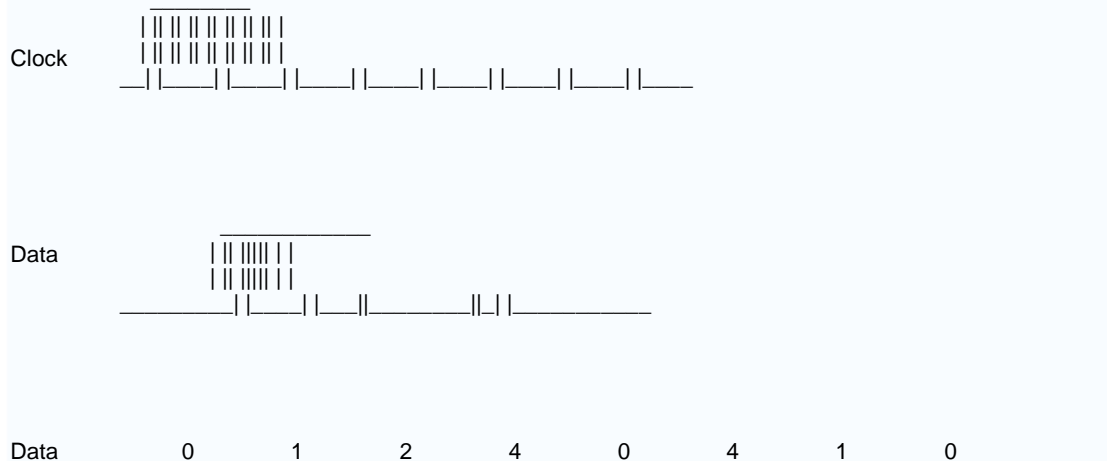
COMMUNICATION SYSTEMS LAB (EE-226-F)



PROCEDURE:-

1. Make the connection according to the block diagram.
2. Connect the audio frequency of 2 KHz, 2V to modulator.
3. Connect the modulator output to CRO.
4. Switch ON the power supply.
5. Observe output on CRO.

OUTPUT WAVE FORMS:-



RESULT:-

Pulses width modulated wave is obtained on CRO.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.

2. Before connecting the power supply plug into socket, ensure power supply should be switched off
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.
5. Power supply should be switched off after completion of experiment.

QUIZ / ANSWERS:-

Q.1 Which modulation is PWM?

Ans. Analog Modulation

Q.2 What are two categories of Pulse Modulation.?

Ans. 1. Analog Modulation 2. Digital Modulation

Q.3 What is the unit of signaling speed?

Ans. Baud

Q.4 What is the disadvantage of PWM?

Ans. Due to varying of pulses width power contents of PWM also varying

Q.5 Which multivibrator is used for PWM?

Ans. Monostable Multivibrator

Q.6 Which circuit is used for PWM demodulator?

Ans. Integrating circuit.

Q.7 What is difference bet. PAM and PWM ?

Ans. In PAM, amplitude of pulse is varied according to modulating signal and in PWM, width is varied of pulses.

Q.8 How PWM may be generated?

Ans. PWM may be generated applying trigger pulses to control the starting time of pulses from a monostable multivibrator.

Q.9 What is the use of sampling theorem?

Ans. Sampling Theorem is used to determine minimum sampling speed.

Q.10 What is the worldwide standard sampling rate?

Ans. Eight thousand samples per second.

EXPERIMENT No.8(c)

AIM:--To study the pulse position modulation.

APPARATUS REQUIRED:-- CRO, experimental kit, power supply, connecting leads.

THEORY:-

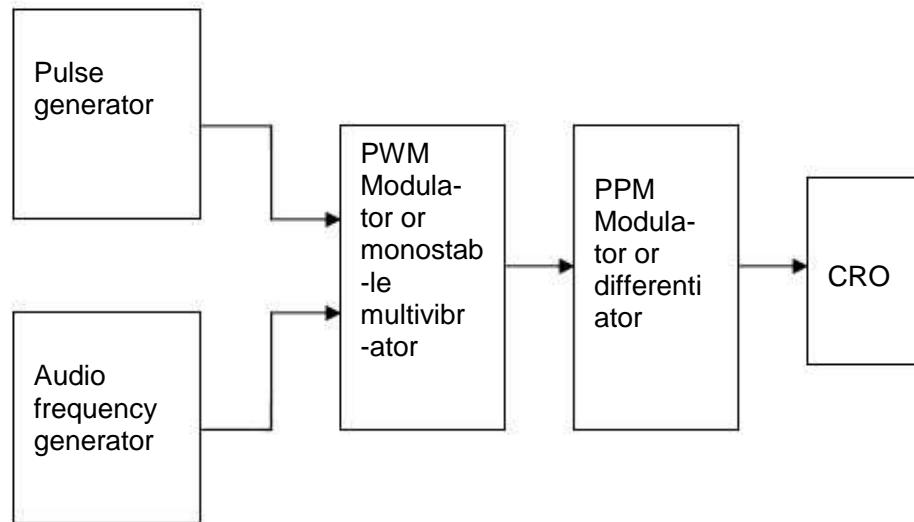
In pulse position modulating the amplitude of pulse is kept constant and position of the pulse in relation to the position of the reference pulse or synchronize pulse is varied by each sample value of modulating signal.

PPM may be obtained from PWM. In PWM each pulse has a leading edge and a trailing edge but the location of leading edge are fixed where trailing edge depends on the pulse width. Thus PPM may be obtained from PWM by simply getting rid of the leading edge and slots tops of PWM pulses. In pulse position modulating the amplitude of pulse is kept constant and position of the pulse in relation to the position of the reference pulse or synchronize pulse is varied by each sample value of modulating signal. In PWM each pulse has a leading edge and a trailing edge but the location of leading edge are fixed where trailing edge depends on the pulse width. The trailing edges of PWM pulses are in fact position modulated. Thus PPM may be obtained from PWM by simply getting rid of the leading edge and slots tops of PWM pulses. In comparison with PWM, PPM has the advantage of requiring constant transmitter power output, but the disadvantage of depending on transmitter –receiver synchronization.

Generation and demodulation of PPM:

PPM may be generated from PWM easily. First of all, PWM pulses are generated and then they are differentiated. The result is another pulse train which has positive going narrow pulses corresponding to leading edges and negative going narrow pulses corresponding to trailing edges. If the position corresponding to the trailing edges of an un-modulated PWM pulse is counted as zero displacement, then the trailing edges of a modulated pulse will arrive earlier or later. An unmodulated PWM pulse is one that is obtained when the instantaneous signal value is zero. The differentiated pulses corresponding to the leading edges are removed with a diode clipper and the remaining pulses are nothing but position modulated output. When the PPM is demodulated in the receiver, it is again first converted into PWM by using flip-flop or bistable multivibrator. One input of the multivibrator receives trigger pulses from a local generator which is synchronized by trigger pulses received from the transmitter, and these triggers are used to switch off one of the stages of the flip-flop. The PPM pulses are fed to the other base of the flip-flop and switch that stage ON. The period of time during which this particular stage is OFF, depends on the time difference between the two triggers, so that the resulting pulse has a width that depends on the time displacement of each individual PPM pulse. The PWM pulse train thus obtained is a demodulated output.

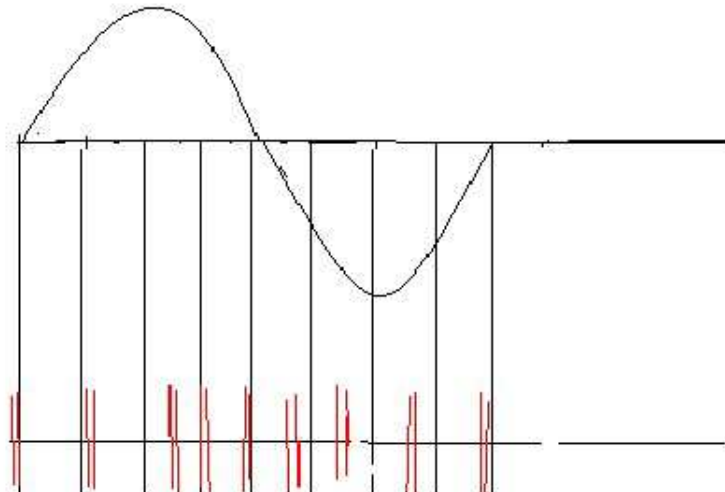
BLOCK DIAGRAM:-



PROCEDURE:--

1. Make the connection according to the block diagram.
2. Connect the audio frequency of 2 KHz, 2V to modulator.
4. Connect the PWM output to the PPM modulator.
4. Connect the PPM modulator output to CRO.
5. Switch ON the power supply.
6. Observe output on CRO

OUTPUT WAVEFORMS:-



RESULT:-

The Pulse Position Modulated wave is obtained on CRO.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.
2. Before connecting the power supply plug into socket, ensure power supply should be switched off.
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.
5. Power supply should be switched off after completion of experiment

QUIZ / ANSWERS:-

Q.1 What is advantage of PPM?

Ans. It has no varying width of pulse so power content are not varying.

Q.2 What is PPM?

Ans. In PPM the position of pulses is varied and width and amplitude are constant.

Q.3 Which Multivibrator is used for PPM De-modulator?

Ans. Bi-stable Multivibrator.

Q.4 What is the difference between PPM & PWM?

Ans. In PWM, the width is varied and in PPM, the position is varied according to modulating signal.

Q5. Which filter is used in PPM demodulator?

Ans. Second order low pass filter.

Q6. In which category of PM is PPM?

Ans. Analog Modulation.

Q7. Which modulation is similar to PDM?

Ans. Phase modulation.

Q8. At which factor the band-width of PPM depends?

Ans. Bandwidth of transmission channel depends on rising time of the pulse.

Q.9 What is the use of sampling theorem?

Ans. Sampling Theorem is used to determine minimum sampling speed.

Q.10 What is the world wide standard sampling rate?

Ans. Eight thousand samples per second.

EXPERIMENT No.9

AIM:--To study Time Division Multiplexing.

APPARATUS REQUIRED:- (i) C.R.O. (ii) CRO Probe (ii) TDM Pulse Code Modulation Transmitter Trainer (ST 2103) and TDM Pulse Code Modulation Receiver Trainer (ST 2104) (iv) Connecting leads.

THEORY:-

Time division multiplexing is a technique of transmitting more than one information on the same channel. As can be noticed from the fig. 11 below the samples consists of short pulses followed by another pulse after a long time intervals. This no-activity time intervals can be used to include samples from the other channels as well. This means that several information signals can be transmitted over a single channel by sending samples from different information sources at different moments in time. This technique is known as time division multiplexing or TDM. TDM is widely used in digital communication systems to increase the efficiency of the transmitting medium. TDM can be achieved by electronically switching the samples such that they interleave sequentially at correct instant in time without mutual interference. The basic 4 channel TDM is shown in fig. 2.

The switches S1 & S2 are rotating in the shown direction in a synchronized manner, where S1 is sampling channel to the transmission media. The timing of the two switches is very important to ensure that the samples of one channel are received only by the corresponding channel at the receiver. This synchronization between S1 & S2 must be established by some means for reliable communication. One such method is to send synchronization code (information) along itself to the transmitter all the time. In practice, the switches S1 & S2 are simulated electronically.

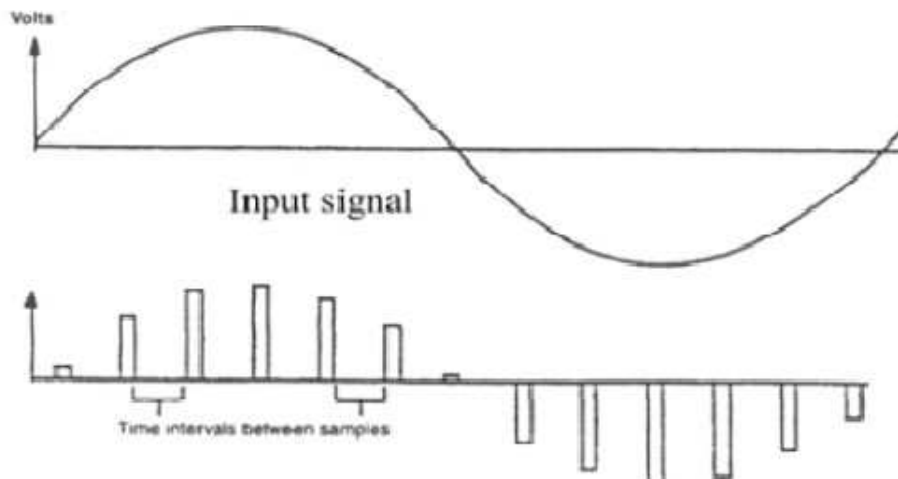


Figure 1: Pulse Amplitude Modulated wave with large time Intervals between samples

On ST2103, the sequence of operation is synchronized to the transmitter clock TX. clock (t.p.3). The time occupied by each clock pulse is called a Bit. The sequence of

operation is repeated after every 15 bits. The complete cycle of 15 bits is called as timing frame. The start of the timing frame is denoted by the TX.TO signal (t.p.4) which goes high during the bit time 0. The various bits reserved for the data appearing in the middle of each transmitter clock cycle is shown in fig. The fig.12 shows the complete timing frame .

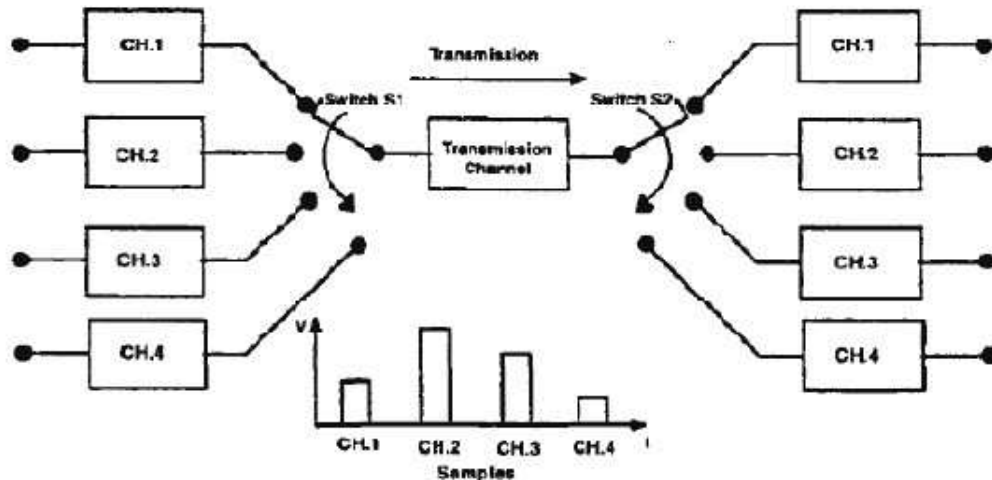


Figure 2: Principle of 4-Channel TDM System

Bit 0: This bit is reserved for the synchronization information generated by the Pseudo random sync code generator block. More about its operation in the later section. When the Pseudo Random Sync Code is switched OFF a '0' is transmitted.

Bit 1 to 7: These carry a 7 bit data word corresponding to the last sample taken from the analog channel CH.0. Remember that the trainer transmits lowest significant bit (LSB) first. This time interval during which the coded information regarding the analog information is transmitted is called as the timeslot. Since the present timeslot corresponds to channel 0 it is known as timeslot 0.

Bit 8 to 14: This timeslot termed as timeslot 1 contains the 7 bit word corresponding to the last sample taken of analog channel 1. As with channel 0 the least significant bit is transmitted first. The receiver requires two signals for its correct operation & reliable communication, namely.

- a. Receiver clock operating at the same frequency as that of the ST2103 clock.
- b. Synchronization signal, which allows the receiver to synchronise its clock/operation with the transmitter's clock operation. All these requirements can be achieved by transmitting two essential information signals:
 - I. A Transmit clock signal.
 - II. A Frame synchronization signal.

The simplest method is to transmit the synchronization information & the clock over a separate transmission link. This results in a simplest receiver. It is used in data communication LAN (Local Area Network) & in telemetry systems. However it is waste of media & is not economical for long distance communications. The ST2103 provides these two signals at TX. clock output (t.p.3) & TX.TO output (t.p.4). In this mode the Pseudo random sync code generator & detector (on ST2104) are switched

OFF. The second technique is to transmit the synchronization code along with transmitted data to be sufficiently different from the information samples. The ST2103 involves the use of a pseudo-random sync code generator. These codes are bit streams of '0's & '1's whose occurrence is detected by some rules. The Pseudo - Random Sync Code gets its name from the fact that the occurrence of '0's & '1's in the stream is random for a portion of sequence i.e. there is equal probability of occurrence of '0' and '1'. This portion of sequence is 15 bit long on ST2103. On the receiver the pseudo-random sync code detector recognizes the Pseudo random code & use it to identify, which incoming data bit is associated, with which transmitter timeslot. The advantage of this technique is that if the synchronization is temporarily lost, due to noise corruption, it can be re-established as the signal clears. Hence there is minimal loss of transmitted information. Also this technique reduces the separate link required for synchronization signal transmission.

Mode 1: Mode 1 is TDM system of three transmission links between transmitter & receiver. They are information, TX clock & TX.TO (synchronization) signal links. The Pseudo random sync code generator & Detector are switched OFF in this case.

Mode 2: Mode 2 is TDM system of two transmission links between transmitter & receiver. These are information & TX clock signal links. The synchronization is established by sync codes transmitted along with the data stream. No need to say that the pseudo random sync generator & detector are switched ON.

Mode 3: Mode 3 is TDM system of one link between transmitter & receiver, namely the link carrying information. Synchronization is again established by the sync codes. The clock signal is regenerated by the phase locked loop (PLL) circuit at the receiver from the transition of the information data bits.

BLOCK DIAGRAM:-

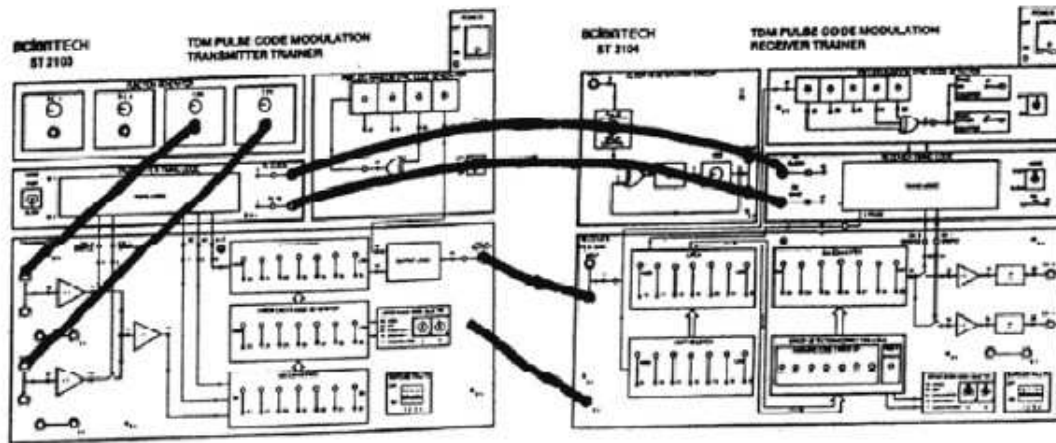


Figure 3: Block diagram to study TDM

PROCEDURE:-

1. Set up the following initial conditions on ST2103:
Mode Switch in fast position

COMMUNICATION SYSTEMS LAB (EE-226-F)

DC 1 & DC2 Controls in function generator block fully clockwise.
~ 1 KHz and ~2 KHz control levels set to give 10Vpp.
Pseudo - random sync code generator on/off switch in OFF Position.
Error check code generator switch A & B in A=0 & B=0 position (OFF Mode)
All switched faults off.

2. First, connect only the 1 KHz output to CH 0.
3. Turn ON the power. Check that the PAM output of 1 KHz sine wave is available at t.p. 15 of the ST2103.
4. Connect channel 1 of the oscilloscope to t.p.10 & channel 2 of the oscilloscope to t.p. 15. Observe the timing & phase relation between the sampling signal t.p.10 & the sampled waveform at t.p.15.
5. Turn OFF the power supply. Now connect also the 2 KHz supply to CH 1.
6. Connect channel 1 of the oscilloscope to t.p. 12 & channel 2 of the oscilloscope to t.p. 15.
7. Observe the individual signals, time division multiplexed and finally demodulated and demultiplexed signal.

RESULT:-

Time Division Multiplexing has been studied.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.
2. Before connecting the power supply plug into socket, ensure power supply should be switched off.
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.
5. Power supply should be switched off after completion of experiment.

QUIZ / ANSWERS:-

Q.1 What is multiplexing?

Ans. Multiplexing is the technique used to send information from a number of users through common channel.

Q.2 What is the advantage of multiplexing?

Ans. BW utilization of channel is efficient.

Q.3 Classify multiplexing techniques.

Ans. 1. Time Division Multiplexing 2. Frequency Division Multiplexing 3. Wavelength Division Multiplexing

Q.4 For what kind of systems TDM is more appropriate?

Ans. Digital Systems.

Q.5 For what kind of systems FDM is more appropriate?

Ans. Analog systems.

Q.6 Write down one example of Elastic store.

Ans. Tape-recorder.

Q.7 What is the BW available to each user in case of TDM?

Ans. Its same as that of channel BW.

Q.8 For what purpose commutator is used in PAM-TDM?

Ans. To allocate time slots to different users.

Q.9 Why PCM-TDM is used?

Ans. To use channel efficiently.

Q.10 What are strategies for time slot allocation in TDM?

Ans. 1.Uniform 2.Non-uniform.

EXPERIMENT No.10

AIM:--To study the pulse code modulation and demodulation with parity and Hamming codes.

APPARATUS REQUIRED:- (i) C.R.O. (ii) CRO Probe (iii) TDM Pulse Code Modulation Transmitter Trainer (ST 2103) and TDM Pulse Code Modulation Receiver Trainer (ST 2104) (iv) Connecting leads.

THEORY:-

PCM is a digital process. In this instead of sending a pulse train capable of continuously varying one of the parameters, the PCM generator produces a series of numbers. Each one of these digits, almost always in binary code, represents the approximate amplitude of the signal sample at that instant .

In pulse code modulation, the message signal is rounded to the nearest of a finite set of allowable values. So that both time and amplitude are discrete form. This allows the message to be transmitted by means of coded electrical signals. There by distinguishing PCM from all other methods. Modulations with increasing ability of wide band communication channel coupled with the emergence of required device technology. The use of PCM has become a practical reality. The essential operation in the transmitter of PCM system are sampling, quantizing, and encoding. The quantizing and encoding operation are performed in the same circuit called A / D converter. The essential operations in the receiver are regeneration of unpaired signal, decoding and demodulation of train of quantized.

Steps in Pulse Code Modulation :

Sampling: The analog signal is sampled according to the Nyquist criteria. The nyquist criteria states that for faithful reproduction of the band limited signal, the sampling rate must be at least twice the highest frequency component present in the signal. For audio signals the highest frequency component is 3.4 KHz.

So, Sampling Frequency $\geq 2 f_m$

$$\geq 2 \times 3.4 \text{ KHz}$$

$$\geq 6.8 \text{ KHz}$$

Practically, the sampling frequency is kept slightly more than the required rate. In telephony the standard sampling rate is 8 KHz. Sample quantifies the instantaneous value of the analog signal point at sampling point to obtain pulse amplitude output. Allocation of Binary Codes: Each binary word defines a particular narrow range of amplitude level. The sampled value is then approximated to the nearest amplitude level. The sample is then assigned a code corresponding to the amplitude level, which is then transmitted. This process is called as Quantization & it is generally carried out by the A/D converter.

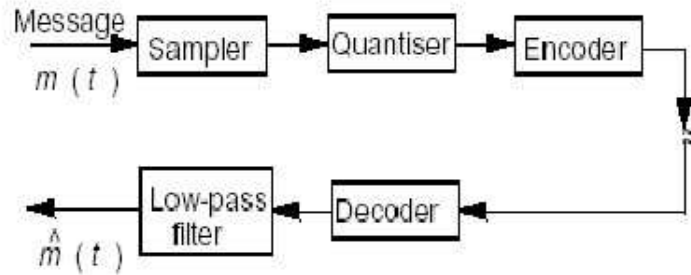


Figure 14.1 A single-channel PCM transmission system.

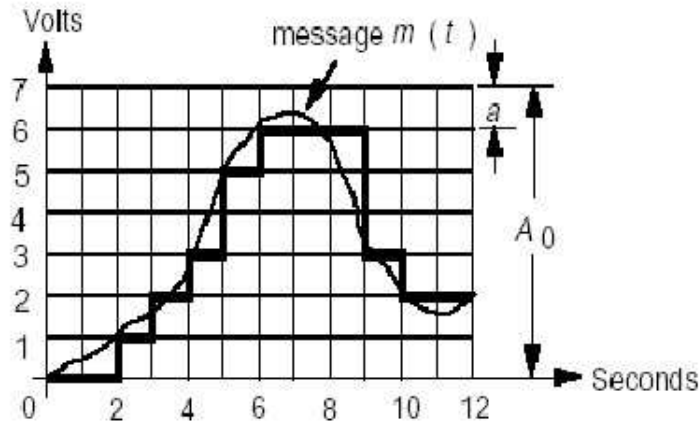


Figure 14.2 Message and quantised signal.

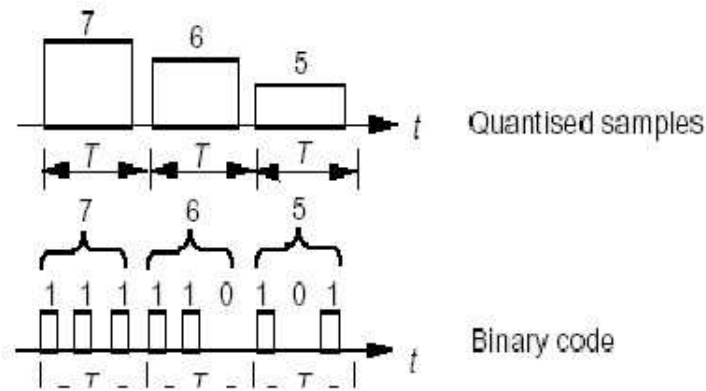


Figure1: Process of PCM

Many different types of codes have been developed and are in use to reduce error in digital communication. The commonly used Codes employed in ST2103 & ST2104 are:

Parity Coding:

It is the simplest method of error coding. Parity is a method of encoding such that the number of 1's in a codeword is either even or odd. Signal parity is established as follows. Each word is examined to determine whether it contains an odd or even number of '1' bits. If even parity is to be established (known as Even parity), a '1' bit is added to each

word containing odd '1' and a '0' bit is added to each word containing even '1' so the result is that all the code words contain an even number of 1 bits after encoding. Similarly, the parity coding can ensure that the total number of '1's in the encoded word is odd. In such number of '1's in the encoded word is odd. In such cases it is called as odd parity. Continuing with the example of even parity, after transmission, each code word is examined to see if it contains an even number of 1 bits. If it does not, the presence of an error is indicated. If it does, the parity bit remains and the data is passed to the user. Note that single bit parity code can detect single errors only and it cannot provide error correction because there is no way of knowing which bit is in error. It is for this reason that parity coding is normally only used on transmission systems where the probability of error occurring is deemed to be low.

Hamming Coding:

Hamming coding, decode each word at transmitter into a new code by stuffing the word with extra redundant bits. As the name suggests, the redundant bits do not convey information but also provides a method of allowing the receiver to decide when an error has occurred & which bit is in error since the system is binary, the bit in error is easily corrected.

Three bit hamming code provides single bit error detection and correction.

The ST2103 & ST2104 involves the use of 7 bit word. Therefore only four bits are used for transmitting data if hamming code is selected. The format becomes.

D6 D5 D4 D3 C2 C1 C0

Where C2, C1 & C0 are Hamming Code Bits.

Hamming code was invented by R.W. Hamming. It uses three redundant bits, as opposed to the single redundant bit needed by simple parity checking. But it provides a facility of single bit error detection & correction.

Code Generation on Trainer

The code on this trainer is generated by adding parity check bit to each group as shown below :

Group 1 D6, D5, D4 Parity Bit - C2

Group 2 D6, D5, D3 Parity Bit - C1

Group 3 D6, D4, D3 Parity Bit – C0

The Groups & Parity bit forms an even parity check group. If an error occurs in any of the digits, the parity is lost & can be detected at receiver e.g. Let us encode binary value

D6, D5, D4, D3 of '1101'

Group 1 D6 D5 D4 C2

1100

Group 2 D6 D5 D3 C1

1111

Group 3 D6 D4 D3 C0

1010

So, the data word after coding will be

D6 D5 D4 D3 C2 C1 C0

1101010

At the receiver, the four digits representing a particular quantized value are taken in as three groups. The Error Detection/ Correction Logic carries out even parity checks on the three groups.

COMMUNICATION SYSTEMS LAB (EE-226-F)

Group 1 D6 D5 D4 C2 .

Group 2 D6 D5 D3 C1

Group 3 D6 D4 D3 C0

If none of them fails, then no error has occurred in transmission & all bit values are valid.

BLOCK DIAGRAM:-

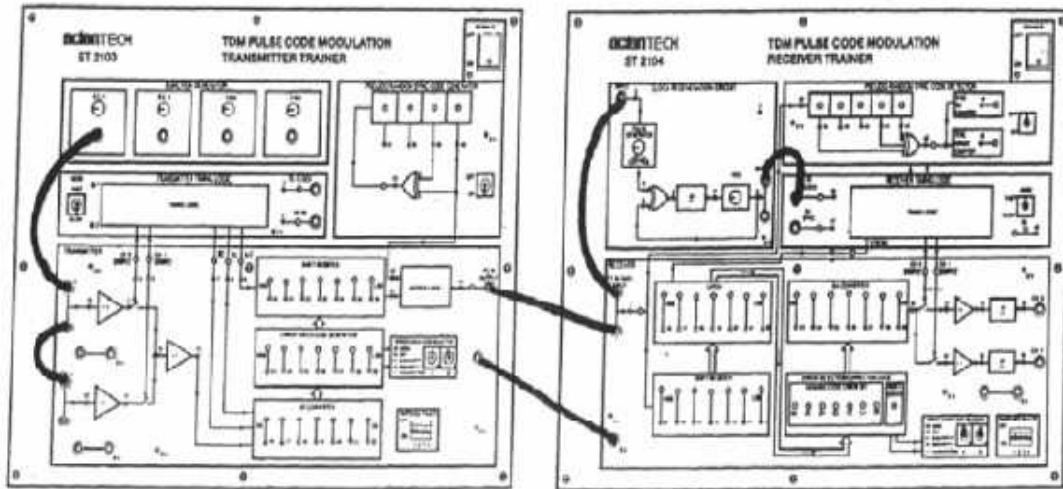


Figure 2: PCM with Error check codes

PROCEDURE:-

1. Make the connection according to the block diagram.
2. Observe PCM output on CRO at the PCM OUT tp. on the ST 2103.
3. Note the variations in the digital output as per variations in the value of DC1.
4. Observe the operation of error check codes by putting switches A & B respectively in positions 00, 01, 10 & 11.
5. Change input from DC1 to 1kHz and 2 kHz sinusoidal signals and repeat from step 2 to 4.
6. Observe the demodulated PCM output on ST 2104 output point.

RESULT:-

Pulse code modulation and demodulation is studied with error check codes.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.
2. Before connecting the power supply plug into socket, ensure power supply should be switched off.
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.

5. Power supply should be switched off after completion of experiment.

QUIZ / ANSWERS:-

Q.1 In which category of PM is PCM?

Ans. Digital Modulation

Q.2 Which noise is occurs in PCM?

Ans. Quantization Noise

Q.3 What is Quantization?

Ans. In PCM, the total amplitude range which is signal may be divided into number of standard level is called quantization.

Q.4 Which noise is occurs in PCM

Ans. Quantization noise.

Q.5 How analog signal can be encoded in to bits/

Ans. By delta modulation technique

Q.6 What is the advantage of DM over PCM?

Ans. DM needs a simple circuit as compared to PCM.

Q.7 What is the advantage of PCM?

Ans. In PCM, S/N ratio is more than DM

Q.8 At which factor bandwidth of PCM depends?

Ans. It depends upon the bit duration i.e. sampling period/total no. of bits.

Q.9 What is Elastic store?

Ans. A device which can store and reproduce data at different speed is Elastic store.

Q.10 Write down one example of Elastic store.

Ans. Tape-recorder.

EXPERIMENT No.11

AIM:- To Study pulse data coding and decoding formats.

APPARATUS:- Trainer Kit, Power supply, Connecting Wires.

THEORY:-

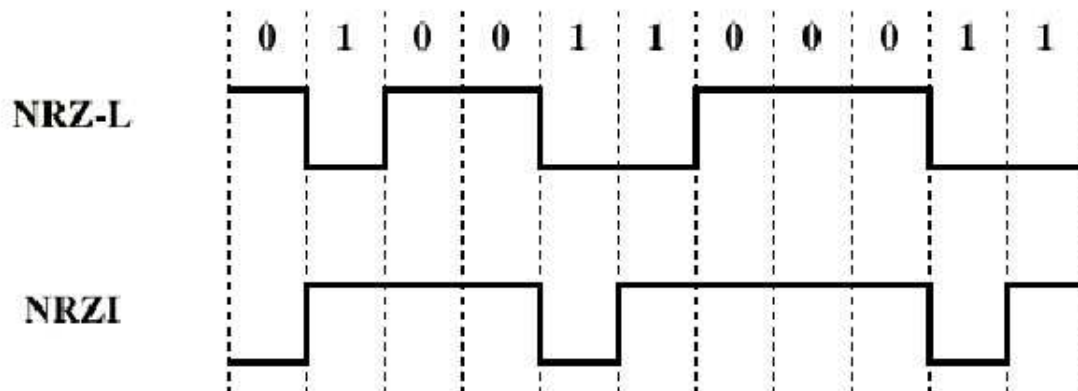
Encoding Schemes: Non-return to Zero-Level (NRZ-L), Nonreturn to Zero Inverted (NRZI), Manchester and Differential Manchester.

Nonreturn to Zero-Level (NRZ-L)

Two different voltages are there for 0 and 1 bits. Voltage constant during bit interval no transition i.e. no return to zero voltage e.g. Absence of voltage for zero, constant positive voltage for one. More often, negative voltage for one value and positive for the other, this is NRZ-L.

Nonreturn to Zero Inverted

Nonreturn to zero inverted on ones Constant voltage pulse for duration of bit
Data encoded as presence or absence of signal transition at beginning of bit time
Transition (low to high or high to low) denotes a binary 1 No transition denotes binary 0
An example of differential encoding



Differential Encoding

- Data represented by changes rather than levels
- More reliable detection of transition rather than level
- In complex transmission layouts it is easy to lose sense of polarity

TRANSISTOR SERIES VOLTAGE REGULATOR

A voltage Regulator generally employs some active devices such as zener, or a transistor or both to achieve its objective. A series voltage regular using a transistor and zener diode is as shown,

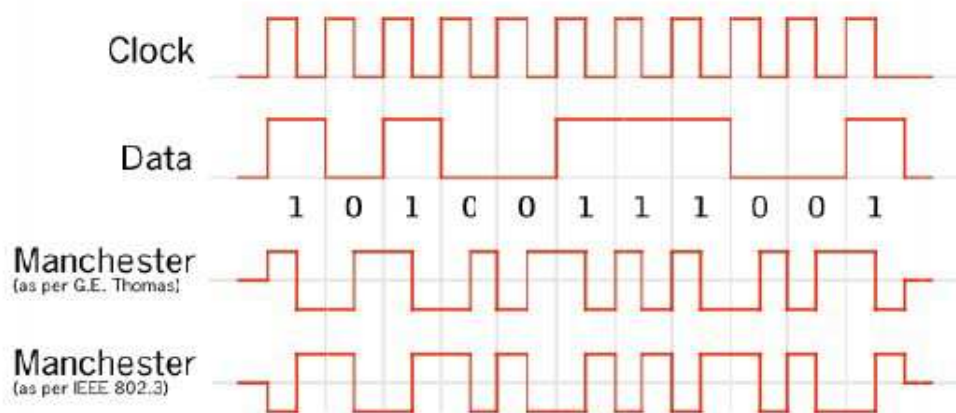
COMMUNICATION SYSTEMS LAB (EEE-226-F)

The circuit is called a series voltage regulator because the load current passes through the series transistor Q1.

The main drawback of series regulator is that the pass transistor can be destroyed by excessive load current.

Manchester code

In telecommunication, Manchester code (also known as Phase Encoding, or PE) is a waveform, form of data communication line code in which each bit of data is signified by at least one voltage level transition.



Manchester encoding is therefore considered to be self-clocking, which means that accurate synchronization of a data stream is possible. Each bit is transmitted over a predefined time period.

Manchester code provides a simple way to encode arbitrary binary sequences without ever having long periods without level transitions thus preventing the loss of clock synchronization, or bit errors from low-frequency or non-fully-equalized analog links (see ones-density). If transmitted as an AC signal it ensures that the DC component of

the encoded signal is zero, again preventing baseline drift of the repeated signal, making it easy to regenerate and preventing waste of energy. However, there are today many more sophisticated codes (8B/10B encoding) which accomplish the same aim with less bandwidth overhead, and less synchronization ambiguity in pathological cases.

Regardless of these losses, Manchester coding has been adopted into many efficient and

widely used telecommunication standards such as Ethernet.

An example of Manchester encoding showing both conventions

PROCEDURE:

- 1 Make the connection according to the block diagram. Power supply should be Switched off .
- 2 Connect frequency-modulated output to the AM De-Modulator input. connections should be tight.
- 3 Connect the De-Modulator output to CRO.
- 4 Observe output on CRO. Take output carefully.

RESULT: Different pulse data coding and decoding formats have been studied.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.
2. Before connecting the power supply plug into socket, ensure power supply should be switched off
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.
5. Power supply should be switched off after completion

QUIZ / ANSWERS:-

Q.1 What are the number of symbols available in binary?

Ans. 2.

Q.2 What is RZ?

Ans. Return-to-Zero.

Q.2 What is NRZ?

Ans. Non Return-to-Zero.

Q.4 Why encoding is used?

Ans. To represent quantized samples in appropriate digital format.

Q.5 What is the unit of data rate?

Ans. Bits/s.

Q.6 Why decoder is used?

Ans. To convert digital data into discrete sample values.

Q.7 What is the advantage of PCM?

Ans. In PCM, S/N ratio is more than DM

Q.8 At which factor bandwidth of PCM depends?

Ans. It depends upon the bit duration i.e. sampling period/total no. of bits.

Q.9 What is no. of bits required to represent a sample of a system of 7 symbols?

Ans. 3

Q.10 Why channel encoder is used?

Ans. To avoid errors.

EXPERIMENT No.12 (a)

AIM:- - Study of Amplitude Shift Keying .

APPARATUS REQUIRED:-ASK modulation kit, CRO and connecting leads.

THEORY:-

The binary ASK System was one of the earliest forms of digital modulation used in wireless telegraphy. This simplest form of digital modulation is no longer used widely in digital communication .Nevertheless it serves as a useful model which helps in understanding certain concepts. In an ASK system, binary symbol 1 is represented by transmitting a sinusoidal carrier wave of fixed amplitude A_c and fixed frequency f_c for the bit duration T_b seconds whereas binary symbol 0 is represented by switching off the carrier for T_b seconds. This signal can be generated by switching off the carrier of a sinusoidal oscillator on and off for the prescribed periods indicated by the modulating pulse train. For this reason the scheme is also known as on-off keying (OOK).

Let the sinusoidal carrier be represented by

$$e_c(t) = A_c \cos(2\pi f_c t)$$

Then, the binary ASK signal can be represented by a wave $s(t)$ given by

$$S(t) = A_c \cos(2\pi f_c t) \text{ symbol 1} \\ = 0, \text{ symbol 0}$$

A typical ASK waveform is illustrated in figure for a binary data represented by {10110101}

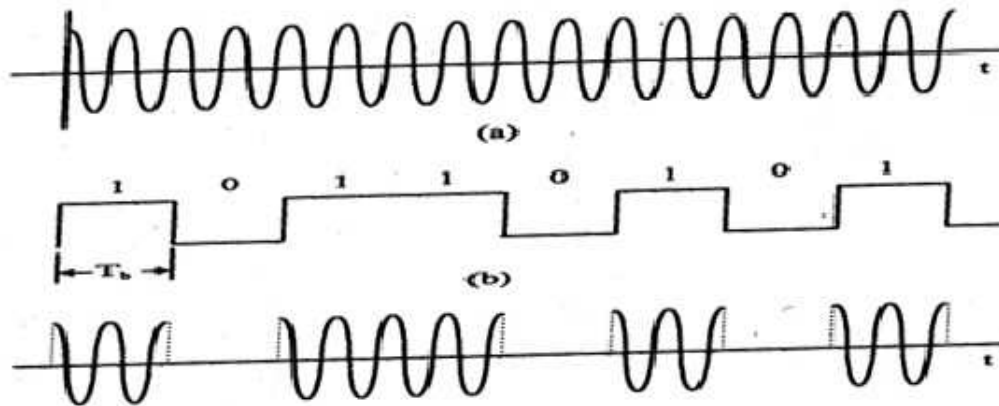


Figure1: ASK wave forms: (a) Unmodulated carrier (b) Unipolar bit sequence (c) ASK wave.

Generation Of ASK Signal

ASK signal can be generated by applying the incoming binary data (represented in unipolar form) and the sinusoidal carrier to the two inputs of a product modulator (balanced modulator) The resulting output is the ASK wave. This is illustrated in figure modulation causes a shift of the baseband signal spectrum.

The ASK signal which is basically the product of the binary sequence and the carrier signal .

BLOCK DIAGRAM:-

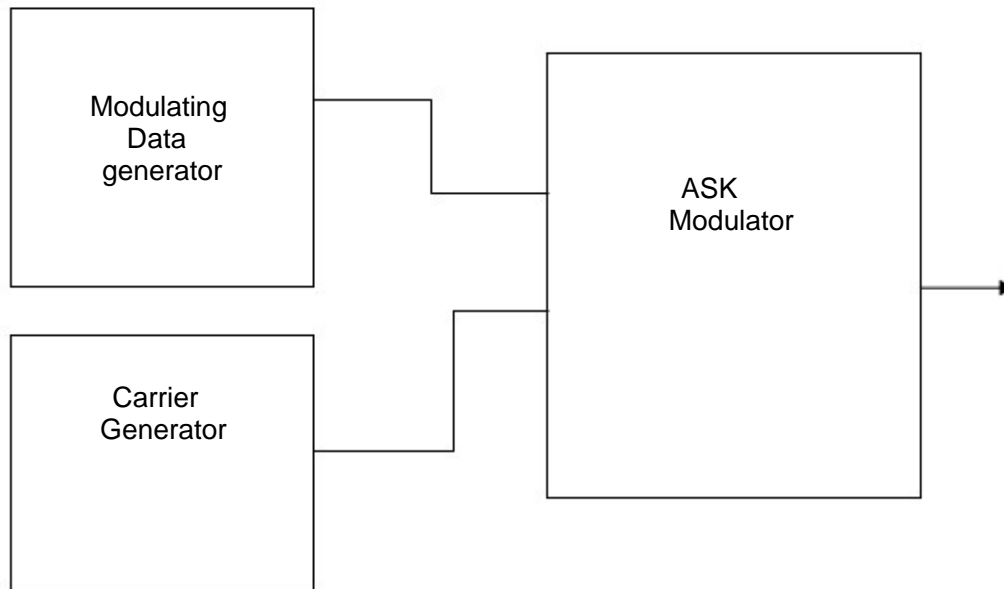


Figure 2: Block diagram for ASK Generation

PROCEDURE:-

1. Make the connection according to the circuit diagram.
2. Connect the modulator output to CRO.
3. Observe output on CRO.

RESULT:- ASK output is obtained on CRO.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.
2. Before connecting the power supply plug into socket, ensure power supply should be switched Off.
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.
5. Power supply should be switched off after completion of experiment.

EXPERIMENT No.12 (b)

AIM:--Study of Frequency Shift Keying.

APPARATUS REQUIRED:- Data generator, FSK modulation kit, CRO and connecting leads.

THEORY:-

FSK is one of the basic modulation techniques for the transmission of digital data .If the frequency of the sinusoidal carrier is switched depending upon the input digital signal , then it is known as frequency shift keying. As the amplitude remains constant in FSK, so the effect of non-linearities, noise interference is minimum on digital detection. So FSK is preferred over ASK.

Frequency shift keying consists of shifting of frequency of carrier from a mark frequency to a space frequency according to the base band digital signal. Frequency shift keying is identical to modulating an FM carrier with a binary digital signal

In an FSK system, two sinusoidal carrier waves of the same amplitude A_c but different frequencies f_{c1} and f_{c2} are used to represent binary symbols 1 and 0 respectively. It can be easily verified that binary FSK waveform is a superposition of two binary ASK waveforms , one with a frequency f_{c1} and other with a frequency f_{c2} . No discrete components appear in the signal spectrum of FSK signal. The main advantage of FSK lies in its easy hardware implementation.

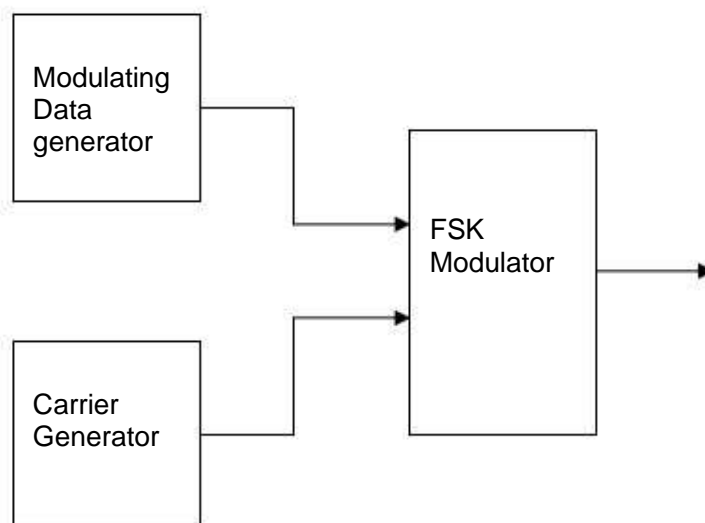
Generation of FSK signal:-

The FSK signal can be generated by applying the incoming binary data to a frequency modulator. To the other input a sinusoidal carrier wave of constant amplitude A_c and frequency f_c is applied. As the modulating voltages changes from one level to another, the frequency modulator output changes its frequency in the corresponding fashion.

Detection of FSK signal:-

FSK can be demodulated by using coherent and non-coherent detector. The detector based on coherent detection requires phase and timing synchronization. Non coherent detection can be done by using envelop detector.

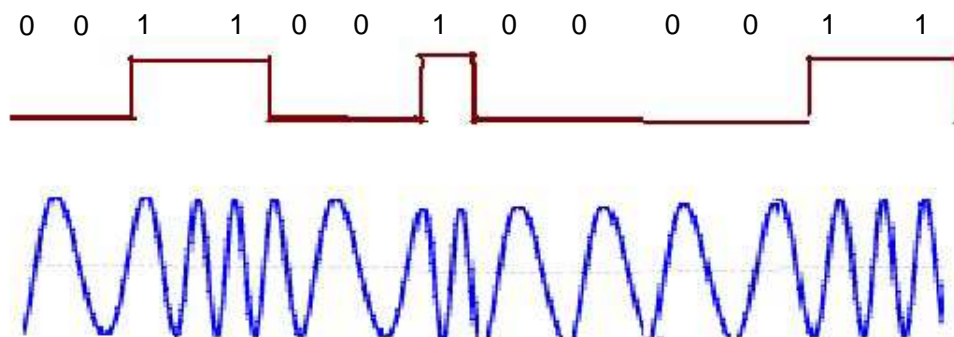
BLOCK DIAGRAM:-



PROCEDURE:-

1. Make the connection according to the block diagram.
2. Connect the modulator output to CRO.
3. Observe output on CRO.

WAVE FORMS:-



RESULT:- FSK output is obtained on CRO.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.

2. Before connecting the power supply plug into socket, ensure power supply should be switched off
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.
5. Power supply should be switched off after completion of experiment

QUIZ / ANSWERS:-

Q.1 What is FSK?

Ans. This is one of the basic modulation techniques for transmission of digital data. The frequency of carrier is switched on or off according to the input digital signal.

Q.2 Why FSK is preferred over ASK?

Ans. Because of constant amplitude of FSK the effect of non-linearity's and noise interference is minimum on signal detection.

Q.3 What are various components of FSK detector?

Ans. Two synchronous detector, differential amplifier, low-pass filter.

Q.4 What is BFSK?

Ans. In BFSK frequency of the carrier is sifted according to the binary symbol keeping the phase of the carrier unaffected.

Q.5 What is the difference between FM and FSK?

Ans. FM is an analog modulation technique where FSK is digital modulation technique.

Q.6 How BFSK signal is generated?

Ans. An input signal is processed in two paths each existing of level shifter and product modulator. One path uses directly and other path uses inverter signal. Orthogonal carrier signal are used as the other input for the product modulator. The output of the product modulator are added which generates a BFSK.

Q.7 What is the bandwidth of BFSK?

Ans. $4f_b$ where f_b - bandwidth of the input signal.

Q.8 Compare bandwidth of BFSK and BPSK.

Ans. Bandwidth of BFSK = 2(bandwidth of BPSK)

Q.9 What is the disadvantage of BFSK?

Ans. The error rate of BFSK is more as compared to BPSK.

Q.10 How can you detect FSK by non-coherent method?

Ans. BFSK waves may be demodulated coherently using envelop detectors.

EXPERIMENT No.13

AIM:- To study the PSK and QPSK.

APPARATUS REQUIRED CRO, experimental kit power supply, connecting leads.

THEORY:-

PSK involves the phase change at the carrier sine wave between 0 to 180 degrees in accordance with the data stream to be transmitted.

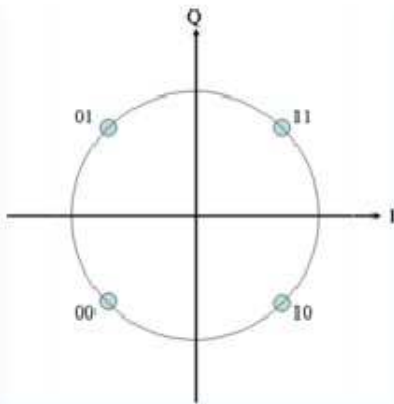
PSK modulator is similar to AS modulator both use balanced modulator to multiply the carrier with a balanced modulator signal. The digital signal with applied to the modulation input for PSK generation is bipolar i.e. equal positive and negative voltage levels.

When the modulating input is positive the output of the modulator is a sine wave in phase with the carrier input whereas for negative voltage level, the output of the modulator is out of phase with the carrier input.

When the modulating input is a sine wave which is switched out of phase by 180 degrees from the carrier input.

Quadrature Phase-shift Keying (QPSK)

QPSK:- In QPSK each pair of consecutive data bits is treated as a two-bit code which is used to switch the phase of the carrier sine wave between four discrete phases 90° apart. The four possible combinations of two bits are 00, 01, 10, and 11 each code represents either a phase of 45°, 135°, 225°, or 315° lagging, relative to the reference phase at the original unmodulated carrier. QPSK offers an advantage over PSK as a carrier through which each phase represents a two-bit code rather than a single bit. This means that either we can have twice the data rate or the same amount of data can be transmitted with half the bandwidth.



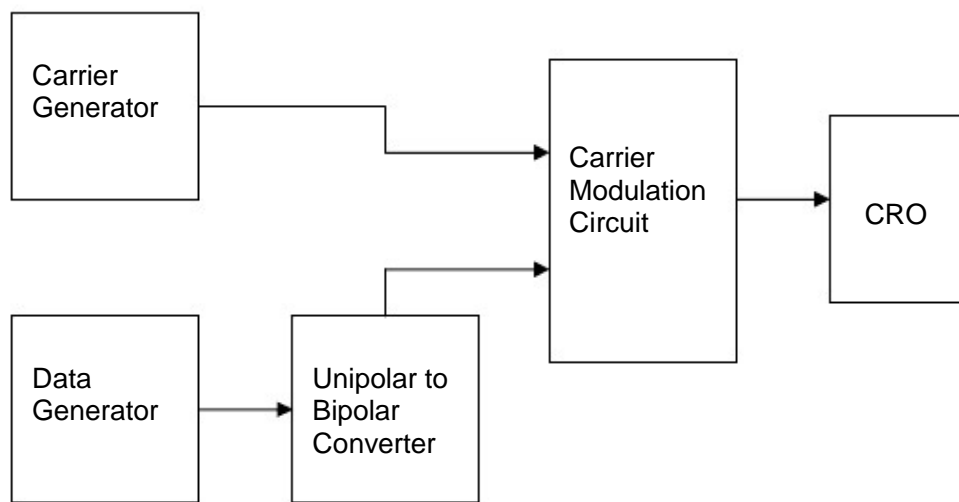
Constellation diagram for QPSK with Gray coding. Each adjacent symbol only differs by one bit.

Sometimes known as quaternary or quadriphase PSK or 4-PSK, QPSK uses four points on the constellation diagram equispaced around a circle. With four phases, QPSK can encode two bits per symbol as shown in the diagram with Gray coding to minimize the BER — twice the rate of BPSK. Analysis shows this may be used either to double the data rate or to halve the bandwidth.

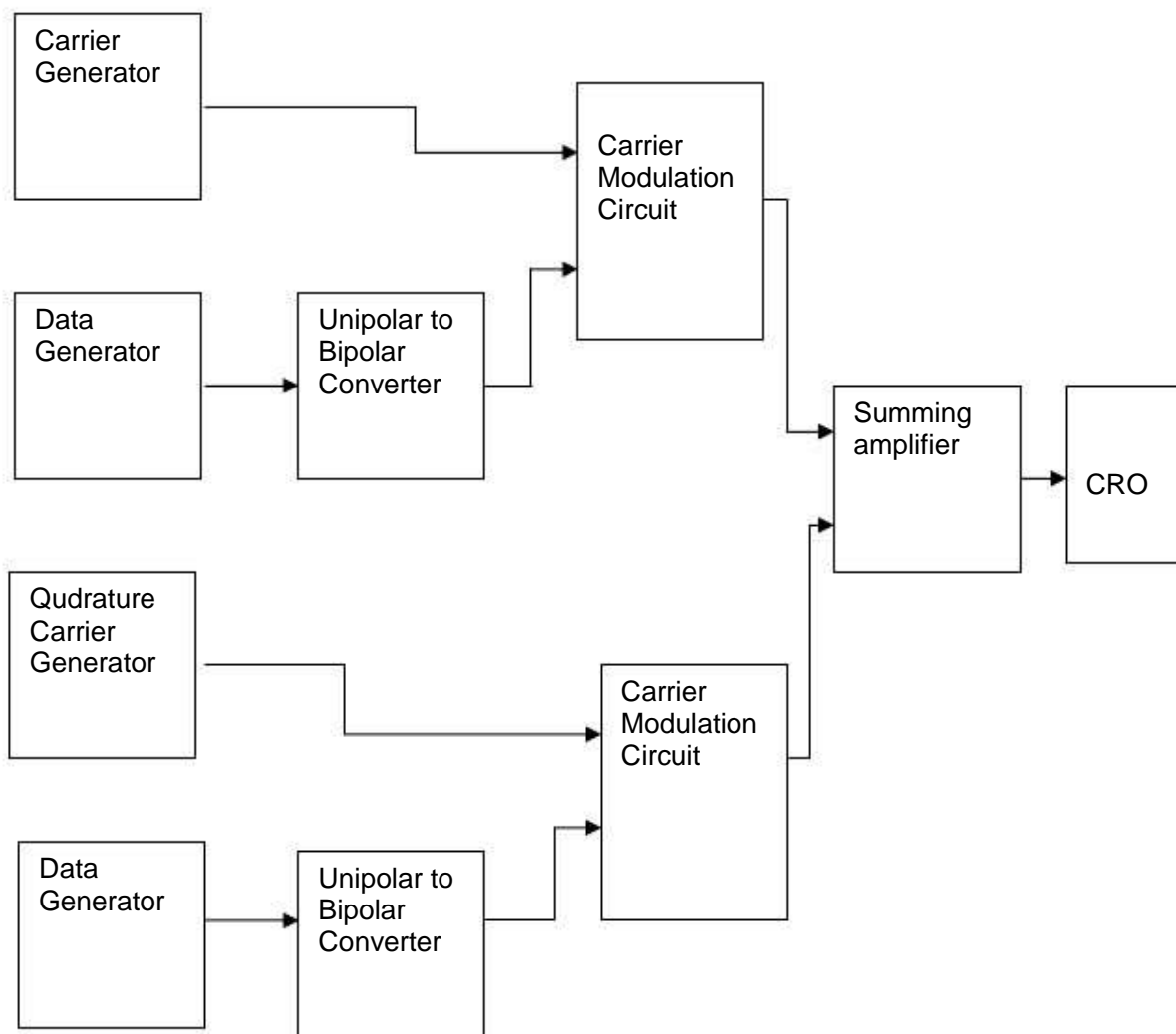
the data rate compared to a BPSK system while maintaining the bandwidth of the signal or to maintain the data-rate of BPSK but halve the bandwidth needed.

Although QPSK can be viewed as a quaternary modulation, it is easier to see it as two independently modulated quadrature carriers. With this interpretation, the even (or odd) bits are used to modulate the in-phase component of the carrier, while the odd (or even) bits are used to modulate the quadrature-phase component of the carrier. BPSK is used on both carriers and they can be independently demodulated

BLOCK DIAGRAM:-



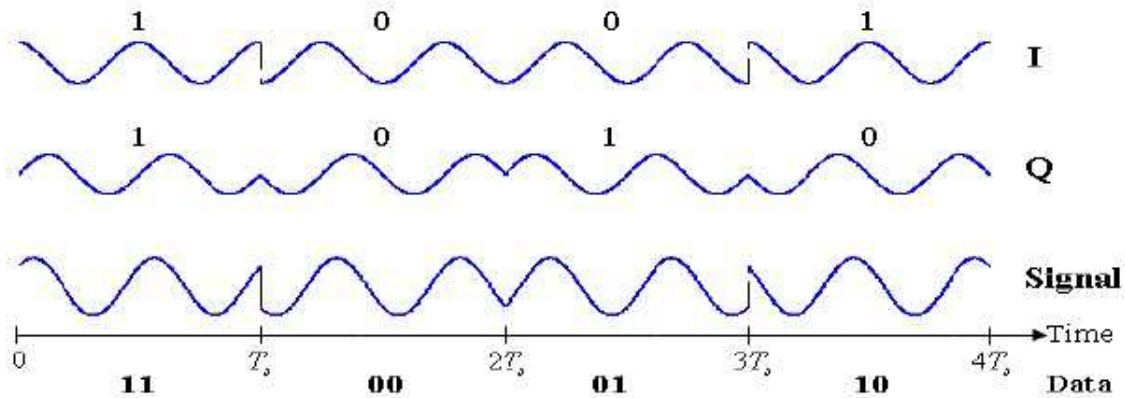
(Block diagram of PSK)



PROCEDURE:--

1. Make the connection according to the block diagram.
2. Connect the modulator output to CRO.
3. Observe output on CRO.

WAVE FORM:-



RESULT:-

PSK and QPSK output is obtained on CRO.

PRECAUTIONS:-

1. Do not use open ended wires for connecting the power supply plug into socket, ensure power supply should be plugged.
2. Before connecting to 23 V power supply, ensure the power supply is switched off.
3. Ensure all connections should be tight before switching on the power supply.
4. Take the readings carefully.
5. Power supply should be switched off after completion of experiment.

QUIZ/ANSWERS:-

Q.1 What is PSK?

Ans. It is one of the basic digital modulation techniques. Here the phase of the carrier is shifted depending upon the input digital signal. This is similar to the Phase Shift Keying Modulation and has constant Amplitude envelope.

Q.2 What is the disadvantage of PSK?

Ans. It needs a complicated synchronizing circuit of the receiver.

Q.3 What is BPSK?

Ans. BPSK is Binary Phase Shift Keying. Here binary symbol 1 and 0 modulate the phase of the carrier. The phase of the carrier changes by 180 degrees.

Q.4 How BPSK is generated?

Ans. It can be generated by applying carrier signal and baseband signal as modulating and carrier signals to a balanced modulator.

Q.5 What is the advantage of PSK?

Ans. Error rate is less than DPSK.

Q.6 What is the difference between QPSK and BPSK?

Ans. In BPSK phase shift is 180 degrees whereas in PSK phase shift is 45 degrees.

Q.7 What is QPSK?

Ans. In QPSK two successive bits are combined. This combination of two bits forms four distinguishing symbols. When the symbol is changed to next symbol the phase of carrier is changed by 45° .

Q.8 How QPSK is generated?

Ans. The input binary sequence is first converted to a bipolar NRZ type of signal called $b(t)$ than it is divided by demultiplexer and added together after insertion of carrier. The generates QPSK signal.

Q.9 How QPSK is detected?

Ans. Basically QPSK receiver uses a synchronous reception. A coherent carrier applied to the two synchronous demodulator, each consists of a multiplier and an integrator.

The output is detected original signal.

Q.10 What is DPSK?

Ans. DPSK is differential phase shift keying and is a non-coherent versus of PSK. DPSK does not need a coherent carrier at the demodulator. The input sequence of binary bits is modified such that the next bit depends upon the previous bit.

EXPERIMENT No.14

AIM:- To Study Differential pulse code modulation and Demodulation.

APPARATUS:- Trainer Kit, Power supply, Connecting Wires.

THEORY:-

Meaning of DPCM – “Differential Pulse Code Modulation”, is a modulation technique invented by the British Alec Reeves in 1937. It is a digital representation of an analog signal where the magnitude of the signal is sampled regularly at uniform intervals. Every sample is quantized to a series of symbols in a digital code, which is usually a binary code. PCM is used in digital telephone systems. It is also the standard form for digital audio in computers and various compact disc formats. Several PCM streams may be multiplexed into a larger aggregate data stream. This technique is called Time-Division Multiplexing. TDM was invented by the telephone industry, but today the technique is an integral part of many digital audio workstations such as Pro Tools. In conventional PCM, the analog signal may be processed (e.g. by amplitude compression) before being digitized. Once the signal is digitized, the PCM signal is not subjected to further processing (e.g. digital data compression). Some forms of PCM combine signal processing with coding. Older versions of these systems applied the processing in the analog domain as part of the A/D process, newer implementations do so in the digital domain. These simple techniques have been largely rendered obsolete by modern transform-based signal compression techniques.

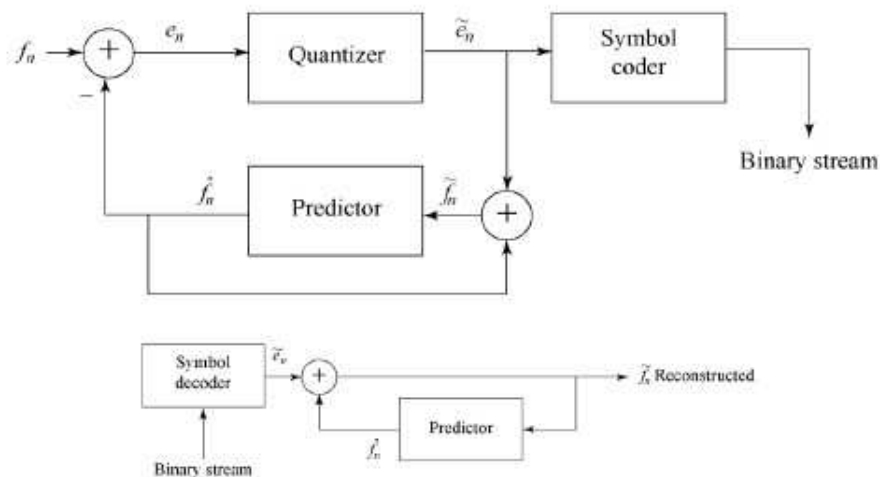


Figure 1: DPCM System

In practical system bandwidth requirement for the transmission of information is very important aspect, since if bandwidth requirement is less more number of channels can be multiplexed on a single line and full utility of transmitting media is extracted out.

In a system in which a baseband signal $m(t)$ is transmitted by sampling, there is available a scheme of transmission which is an alternative to transmitting the sample values at each sampling time. We can instead, at each sampling time, say the K_{th}

COMMUNICATION SYSTEMS LAB (EE-226-F)

sampling time, transmit the difference between the sample value $m(k)$ at sampling time K and the sample value $m(K-1)$ at time $k-1$. If such changes are transmitted, then simply by adding up these changes we shall generate at the receiver a waveform identical in form to $m(t)$.

PROCEDURE:-

1. Make the connection according to the circuit diagram.
2. Observe output on CRO.

RESULT:- DPCM modulation and demodulation has been studied.

PRECAUTIONS:-

1. Do not use open ended wires for connecting to 230 V power supply.
2. Before connecting the power supply plug into socket, ensure power supply should be switched off
3. Ensure all connections should be tight before switching on the power supply.
4. Take the reading carefully.
5. Power supply should be switched off after completion of experiment

QUIZ / ANSWERS:-

Q.1 What is DPCM?

Ans. Differential Pulse Code Modulation.

Q.2 What is the advantage of DPCM?

Ans. It require less BW as compared to PCM.

Q.3 What is quantizer?

Ans. It converts the sample values to some fixed finite levels.

Q.4 What is the use of predictor?

Ans. To estimate previous sample.

Q.5 Which one is better PCM or DPCM?

Ans. DPCM.

Q.6 Is DPCM analog modulation technique?

Ans. It belong to the class of pulse digital modulation.

Q.7 Which one has less BW requirement DPCM or Delta modulation?

Ans. Delta modulation.

Q.8 DPCM is suitable for which kind of input signals?

Ans. Where dynamic changes in signal are small, DPCM is very useful.

Q.9 Why DPCM is preferred over PCM?

Ans. Because of low BW.

Q.10 DPCM is preferably used for.....

Ans. Voice or picture Communication.

NM LAB

MATH-204-F

IV SSEMESSTER

NUMERICAL METHODS LAB
IV SEM.

LIST OF EXPERIMENTS

SR. NO.	NAME OF EXPERIMENT	PAGE NO.
1	TO FIND THE ROOTS OF NON-LINEAR EQUATION USING BISECTION METHOD.	3-5
2	TO FIND THE ROOTS OF NON-LINEAR EQUATION USING NEWTON'S METHOD.	6-9
3	CURVE FITTING BY LEAST – SQUARE APPROXIMATIONS.	10-15
4	TO SOLVE THE SYSTEM OF LINEAR EQUATIONS USING GAUSS - ELIMINATION METHOD.	16-20
5	TO SOLVE THE SYSTEM OF LINEAR EQUATIONS USING GAUSS - SEIDAL ITERATION METHOD	21-25
6	TO SOLVE THE SYSTEM OF LINEAR EQUATIONS USING GAUSS - JORDEN METHOD.	26-27
7	TO INTEGRATE NUMERICALLY USING TRAPEZOIDAL RULE.	28-30
8	TO INTEGRATE NUMERICALLY USING SIMPSON'S RULES.	31-33
9	TO FIND THE LARGEST EIGEN VALUE OF A MATRIX BY POWER - METHOD.	34-37
10	TO FIND NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS BY EULER'S METHOD.	38-40
11	TO FIND NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS BY RUNGE- KUTTA METHOD.	41-43
12	TO FIND NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS BY MILNE'S METHOD.	44-47
13	TO FIND THE NUMERICAL SOLUTION OF LAPLACE EQUATION	48-53
14	TO FIND THE NUMERICAL SOLUTION OF WAVE EQUATION	54-58
15	TO FIND THE NUMERICAL SOLUTION OF HEAT EQUATION.	59-63

Program 1

OBJECTIVES: To find the roots of non linear equations using
Bisection method.

SOURCE CODE:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

/*    define prototype for USER-SUPPLIED function f(x)                                */

double ffunction(double x);

/*    EXAMPLE for "ffunction"                                                        */

double ffunction(double x)

{
    return (x * sin(x) - 1);
}

/* ----- */

/*    Main program for algorithm 2.2                                                */

void main()

{
    double Delta = 1E-6; /* Tolerance for width of interval                        */
    int Satisfied = 0; /* Condition for loop termination                          */
    double A, B; /* Endpoints of the interval [A,B]                             */
    double YA, YB; /* Function values at the interval-borders                    */
    int Max; /* Calculation of the maximum number of iterations                  */
    int K; /* Loop Counter                                                         */
    double C, YC; /* Midpoint of interval and function value there               */

    printf("-----\n");
    printf("Please enter endpoints A and B of the interval [A,B]\n");
    printf("EXAMPLE : A = 0 and B = 2. Type:0 2 \n");
    scanf("%lf %lf", &A, &B);
    printf("The interval ranges from %lf to %lf\n", A,B);

    YA = ffunction(A); /* compute function values */
    YB = ffunction(B);
    Max = (int) ( 1 + floor( ( log(B-A) - log(Delta) ) / log(2) ) );
    printf("Max = %d\n",Max);

    /* Check to see if the bisection method applies */
```

```

if( ( (YA >= 0) && (YB >=0) ) || ( (YA < 0) && (YB < 0) ) ) {
    printf("The values ffunction(A) and ffunction(B)\n");
    printf("do not differ in sign.\n");
    exit(0);/* exit program */
}

for(K = 1; K <= Max ; K++) {

    if(Satisfied == 1)                break;

    C = (A + B) / 2;                    /* Midpoint of interval */
    YC = ffunction(C);                 /* Function value at midpoint */

    if( YC == 0) {/* first 'if'*/
        A = C;/* Exact root is found */
        B = C;
    }
    else if( ( (YB >= 0) && (YC >=0) ) || ( (YB < 0) && (YC < 0) ) ) {
        B = C;/* Squeeze from the right */
        YB = YC;
    }
    else {
        A = C;/* Squeeze from the left */
        YA = YC;
    }

    if( (B-A) < Delta ) Satisfied = 1; /* check for early convergence */

} /* end of 'for'-loop */

printf("-----\n");
printf("The maximum number of iterations is : %d\n",Max);
printf("The number of performed iterations is : %d\n",K - 1);
printf("-----\n");
printf("The computed root of f(x) = 0 is : %lf\n",C);
printf("-----\n");
printf("The accuracy is +- %lf\n", B-A);
printf("-----\n");
printf("The value of the function f(C) is %lf\n",YC);

} /* End of main program */

```

Quiz

Q.1) What is use of Bisection Method?

Answer: Bisection method is used to find the root of non-linear equation.

Q.2) What is Bisection method?

Answer: The bisection method in mathematics is a root-finding method which repeatedly bisects an interval and then selects a subinterval in which a root must lie for further processing. It is a very simple and robust method, but it is also relatively slow. Because of this, it is often used to obtain a rough approximation to a solution which is then used as a starting point for more rapidly converging methods

Q.3) What are the observations of Bisection method?

Answer: There are two observation of Bisection method. They are

Obs. 1: Since the new interval containing the root, is exactly half the length of the previous one, the interval width is reduced by a factor of $\frac{1}{2}$ at each step. At the end of the n th step, the new interval will therefore, be of length $(b-a)/2^n$

Obs. 2: As the error decreases with each step by a factor of $\frac{1}{2}$, the convergence in the bisection method is linear.

Q.4) What are the disadvantages of Bisection Method?

Answer: The convergence process in the bisection method is very slow. It depends only on the choice of end points of the interval $[a, b]$. The function $f(x)$ does not have any role in finding the point c (which is just the mid-point of a and b). It is used only to decide the next smaller interval $[a, c]$ or $[c, b]$.

Q.5) Which property is used in Bisection method?

Answer: Bisection method uses intermediate value property.

Program 2

OBJECTIVES: To find the roots of non linear equations using Newton's method.

Source code:

Algorithm 2.5 (Newton-Raphson Iteration). To find a root
 $f(x) = 0$ given one initial approximation p_0 and using the iteration

$$p_k = p_{(k-1)} - \frac{f(p_{(k-1)})}{f'(p_{(k-1)})} \quad \text{for } k = 1, 2, \dots$$

```
-----
/*
/* User has to supply a function named : ffunction
   and its first derivative : dffunction

   An example is included in this program */

#include<stdio.h>
#include<stdlib.h>
#include<math.h>

/*   define prototype for USER-SUPPLIED function f(x)                               */

double ffunction(double x);
double dffunction(double x);

/*   EXAMPLE for "ffunction"                                                         */

double ffunction(double x)

{
    return ( pow(x,3) - 3 * x + 2 );
}

/*   EXAMPLE for "dffunction" , first derivative of ffunction. */

double dffunction(double x)

{
    return ( 3 * pow(x,2) - 3 );
}

/* ----- */
```

NM LAB (MATH-204-F)

```
/*      Main program for algorithm 2.5      */

void main()

{
    double Delta = 1E-6;          /* Tolerance      */
    double Epsilon = 1E-6;        /* Tolerance      */
    double Small = 1E-6;          /* Tolerance      */

    int Max = 99;                  /* Maximum number of iterations */
    int Cond = 0;                  /* Condition fo loop termination */
    int K;                         /* Counter for loop*/

    double P0; /* INPUT : Must be close to the root */
    double P1; /* New iterate*/
    double Y0; /* Function value */
    double Y1; /* Function value */
    double Df; /* Derivative*/
    double Dp;
    double RelErr;

    printf("-----\n");
    printf("Please enter initial approximation of root !\n");
    scanf("%lf",&P0);
    printf("-----\n");
    printf("Initial value for root: %lf\n",P0);

    Y0 = dffunction(P0);

    for ( K = 1; K <= Max ; K++) {

        if(Cond) break;

        Df = dffunction(P0);          /* Compute the derivative      */

        if( Df == 0 ) {                /* Check division by zero */
            Cond = 1;
            Dp= 0;
        }

        else Dp = Y0/Df;

        P1 = P0 - Dp; /* New iterate */
        Y1 = ffunction(P1); /* New function value */

        RelErr = 2 * fabs(Dp) / ( fabs(P1) + Small );          /* Relative
error */

        if( (RelErr < Delta)          && (fabs(Y1) < Epsilon) ) { /* Check for
*/

            if( Cond != 1) Cond = 2;          /*
convergence */

        }
    }
}
```



```
        P0 = P1;
        Y0 = Y1;
    }

    printf("-----\n");
    printf("The current %d -th iterate is %lf\n",K-1, P1);
    printf("Consecutive iterates differ by %lf\n",Dp);
    printf("The value of f(x) is %lf\n",Y1);
    printf("-----\n");

    if(Cond == 0) printf("The maximum number of iterations was exceeded
!\n");

    if(Cond == 1) printf("Division by zero was encountered !\n");

    if(Cond == 2) printf("The root was found with the desired tolerance
!\n");

    printf("-----\n");
}    /* End of main program */
```

Quiz

Q.1)What is the use of Newton's Raphsons method?

Answer: Newton Raphsons method is used to find the root of non Linear equation.

Q2)Explain Newton Raphson method in detail.

Answer: The Newton-Raphson method uses an iterative process to approach one root of a function. The specific root that the process locates depends on the initial, arbitrarily chosen x-value.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Here, x_n is the current known x-value, $f(x_n)$ represents the value of the function at x_n , and $f'(x_n)$ is the derivative (slope) at x_n . x_{n+1} represents the next x-value that you are trying to find. Essentially, $f'(x)$, the derivative represents $f(x)/dx$ ($dx = \Delta x$). Therefore, the term $f(x)/f'(x)$ represents a value of dx .

$$\frac{f(x)}{f'(x)} = \frac{f(x)}{f(x)/\Delta x} = \Delta x$$

The more iterations that are run, the closer dx will be to zero (0).

Q.3)What are the observation of Newton's Raphsons method?

Obs. 1: Newton's method is useful in cases of large values of $f'(x)$ i.e. when the graph of $f(x)$ while crossing the x-axis is nearly vertical.

Obs. 2: This method replace the part of curve between the point A_0 and x-axis by means of the tangent to the curve at A_0 .

Obs. 3: Newton method is used to improve the result obtained by other method. It is applicable to the solution of both algebraic and transcendental equations.

Obs.4: Newton's formula converges provided the initial approximation x_0 is chosen sufficiently close to the root.

Obs. 5: Newton's method has a quadratic converges.

Q.4) Which of the following method converges faster-Regula Falsi method or Newton-Raphson method?

Answer: Newton Raphson method.

Q.5)What is the other name for Newton-Raphson method

Answer: The other name for Newton Raphson method is Newton Iteration method

Program 3

OBJECTIVES: Curve fitting by least square approximations

Algorithm 5.2 (Least-Squares Polynomial).

To construct the least-squares polynomial of degree

M of the form

$$P_M(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + \dots + c_Mx^M + c_{(M+1)}x^{2M-1M}$$

that fits the N data points $(x_1, y_1), \dots, (x_N, y_N)$.

```

-----
*/

#include<stdio.h>
#include<stdlib.h>
#include<math.h>

/* ----- */

/*    Main program for algorithm 5.2    */

/*    remember : in C the fields begin with element 0    */

#define DMAX      15    /* Maximum degree of polynomial */
#define NMAX      20    /* Maximum number of points*/

void main(void)
{
    extern void FactPiv();

    int R,      K, J;                /* Loop counters */
    double X[NMAX-1], Y[NMAX-1];    /* Points (x,y) */
    double A[DMAX][DMAX];          /* A */
    double B[DMAX];                /* B */
    double C[DMAX];
    double P[2*DMAX];
    int N;
    int M;
    double      /* Number of points :INPUT */
    int p;      /* Degree of polynomial : INPUT */
    x, y;

    printf("Try the examples on page 277 or 281 of the book !\n");
    printf("-----\n");

```

```

do /* force proper input */
{
    printf("Please enter degree of polynomial [Not more than
%d]\n",DMAX);
    scanf("%d", &M);
} while( M > DMAX);

printf("-----\n");
do /* force proper input */
{
    printf("Please enter number of points [Not more than
%d]\n",NMAX);
    scanf("%d", &N);
} while( N > NMAX);

printf("You say there are %d points.\n", N);
printf("-----\n");
printf("Enter points in pairs like : 2.4, 4.55:\n");

for (K = 1; K <= N; K++)
{
    printf("Enter %d st/nd/rd pair of points:\n", K);
    scanf("%lf, %lf", &X[K-1], &Y[K-1]);
    printf("You entered the pair (x,y) = %lf, %lf\n", X[K-1], Y[K-
1]);
}

/* Zero the array */

for (R = 1; R <= M+1; R++) B[R-1] = 0;

/* Compute the column vector */

for (K = 1; K <= N; K++)
{
    y = Y[K-1];
    x = X[K-1];
    p = 1;

    for( R = 1; R <= M+1; R++ )
    {
        B[R-1] += y * p;
        p = p*x;
    }
}

/* Zero the array */

for (J = 1; J <= 2*M; J++) P[J] = 0;

P[0] = N;

/* Compute the sum of powers of x_(K-1) */

for (K = 1; K <= N; K++)
{

```

```

        x = X[K-1];
        p = X[K-1];

        for (J = 1; J <= 2*M; J++)
        {
            P[J] += p;
            p = p * x;
        }
    }

    /* Determine the matrix entries */

    for (R = 1; R <= M+1; R++)
    {
        for( K = 1; K <= M+1; K++) A[R-1][K-1] = P[R+K-2];
    }

    /* Solve the linear system of M + 1 equations : A*C = B
       for the coefficient vector C = (c_1,c_2,...,c_M,c_(M+1)) */

    FactPiv(M+1, A, B);
}    /* end main */

/*-----*/

void FactPiv(N, A, B)
int N;
double A[DMAX][DMAX];
double *B;
{

    int K,      P, C, J;                /* Loop counters          */
    int      Row[NMAX];                /* Field with row-number   */
    double   X[DMAX], Y[DMAX];
    double   SUM, DET = 1.0;

    int      T;

    /* Initialize the pointer vector */

    for (J = 1; J<= N; J++) Row[J-1] = J - 1;

    /* Start LU factorization */

    for (P = 1; P <= N - 1; P++)
    {

        /* Find pivot element */

```

```

for (K = P + 1; K <= N; K++)
{
    if ( fabs(A[Row[K-1]][P-1]) > fabs(A[Row[P-1]][P-1]) )
    {
        /* Switch the index for the p-1 th pivot row if necessary */
        T= Row[P-1];
        Row[P-1] = Row[K-1];
        Row[K-1] = T;
        DET= - DET;
    }

}

} /* End of simulated row interchange */

    if (A[Row[P-1]][P-1] == 0)
    {
        printf("The matrix is SINGULAR !\n");
        printf("Cannot use algorithm --> exit\n");
        exit(1);
    }

/* Multiply the diagonal elements */

DET = DET * A[Row[P-1]][P-1];

/* Form multiplier */

for (K = P + 1; K <= N; K++)
{
    A[Row[K-1]][P-1]= A[Row[K-1]][P-1] / A[Row[P-1]][P-1];

    /* Eliminate X_(p-1) */

    for (C = P + 1; C <= N + 1; C++)
    {
        A[Row[K-1]][C-1] -= A[Row[K-1]][P-1] * A[Row[P-1]][C-1];
    }
}

} /* End of          L*U    factorization routine */

DET = DET * A[Row[N-1]][N-1];

/* Start the forward substitution */

for(K = 1; K <= N; K++) Y[K-1] = B[K-1];

Y[0] = B[Row[0]];
for ( K = 2; K <= N; K++)
{
    SUM =0;
    for ( C = 1; C <= K -1; C++) SUM += A[Row[K-1]][C-1] * Y[C-1];
    Y[K-1] = B[Row[K-1]] - SUM;
}

```

```
if( A[Row[N-1]][N-1] == 0)
{
    printf("The matrix is SINGULAR !\n");
    printf("Cannot use algorithm --> exit\n");
    exit(1);
}

/* Start the back substitution */

X[N-1] = Y[N-1] / A[Row[N-1]][N-1];

for (K = N - 1; K >= 1; K--)
{
    SUM = 0;
    for (C = K + 1; C <= N; C++)
    {
        SUM += A[Row[K-1]][C-1] * X[C-1];
    }

    X[K-1] = ( Y[K-1] - SUM) / A[Row[K-1]][K-1];
} /* End of back substitution */

/* Output */

printf("-----:\n");
printf("The components of the vector with the solutions are:\n");
for( K = 1; K <= N; K++) printf("X[%d] = %lf\n", K, X[K-1]);
}

/*
-----
---

");
} /* End of main programm */
```

Quiz

Q.1) Write down working procedure of Least square method

Answer: (a) To fit the straight line $y=a+bx$

i) Substitute the observed the set of n values in this equation.

ii) Form normal equation for each constant i.e. $\sum y=na =b\sum x$, $\sum xy=a\sum x+b\sum x^2$

iii) Solve these normal equations as simultaneous equation for a & b .

iv) Substitute the values of a & b in $y=a+bx$, which is required line of best fit

b) To fit the parabola: $y=a+bx+cx^2$

i) Form the normal equation $\sum y=n+b\sum x+c\sum x^2$

ii) Solve these equation simultaneous equation for a , b , c .

iii) Substitute the values of a , b , c in $y=a+bx+cx^2$

c) In general, the curve $y=a+bx+cx^2+\dots+kx_{m-1}$ can be fitted to a given data by writing m normal equations.

Q.2) What is observation of least square method?

Answer: On calculating $\partial^2 E/\partial a^2$, $\partial^2 E/\partial b^2$, $\partial^2 E/\partial c^2$ and substituting the values of a , b , c just obtained, we will observe that each is positive i.e. E is a minimum.

Q.3) What is the use of Least square method?

Answer: Least square method is used to fit a unique curve to a given data.

Q.4) What is the error for the curve $y= a+bx+cx^2$?

Answer: The error for the curve $y= a+bx+cx^2$ is $e_i=y_i-n_i$

Q.5) Whether E value is minimum or maximum in Least square method?

Answer: E value is minimum in Least Square method

Program 4

OBJECTIVES: To solve the system of linear equations using gauss elimination method

Source Code:

(Gauss-Elimination-Iteration).

To solve the linear system $AX = B$ by starting with $P_0 = 0$ and generating a sequence $\{P_K\}$ that converges to the solution P (i.e., $AP = B$). A sufficient condition for the method to be applicable is that A is diagonally dominant.

```
-----
---
*/

#include<stdio.h>
#include<stdlib.h>
#include<math.h>

/* ----- */

/*    Main program for algorithm 3.5                */

/*    remember : in C the fields begin with element 0                */

#define N 4

void main(void)
{
    Float a[N][N+1],x[N],t,s;
    Int i,j,k;
    Printf("Enter the element of the                argumented matrix row
wise\n");

    For(i=0;i<N;i++)
    For(j=0;j<N+1;j++)
        Scanf("%f",&a[i][j]);
    For(j=0;j<N-1;j++)
    For(i=j+1;i<N;i++)
    {
        t=a[i][j]/a[j][j];
        for(k=0;k<N+1;k++)
            a[i][k]=a[j][k]*t;
    }

    Printf("The upper triangular matrix is:\n");

    For(i=0;i<N;i++)
```

```
For(j=0;j<N+1;j++)
    printf("%8.4f%",a[i][j]);

    Printf("\n");

For(i=N-1;i>=0;i--)
{

s=0;

for(j=i+1;j<N;j++)

    s+=a[i][j]*x[j];

    x[i]=(a[i][N]-s)/a[i][i];

}

Printf("The solution is :\n");

For(i=0;i<N;i++)

    Printf("x[%3d]=%7.4f\n", i+1,x[i]);

}
```

Q.1)What is Gauss Elimination method?

Answer: Gaussian elimination is a method of solving a linear system $Ax = b$ (consisting of equations in unknowns) by bringing the augmented matrix

$$[A \ b] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{array} \right]$$

to an upper triangular form

$$\left[\begin{array}{cccc|c} c_{11} & c_{12} & \cdots & c_{1n} & d_1 \\ 0 & c_{22} & \cdots & c_{2n} & d_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & c_{nn} & d_n \end{array} \right].$$

This elimination process is also called the forward elimination method.

Q.2) Explain the procedure of Gauss elimination method.

Answer: Consider a linear system.

1. Construct the augmented matrix for the system;
2. Use elementary row operations to transform the augmented matrix into a triangular one;
3. Write down the new linear system for which the triangular matrix is the associated augmented matrix;
4. Solve the new system. You may need to assign some parametric values to some unknowns, and then apply the method of back substitution to solve the new system.

Q.3)Solve the following system via Gaussian elimination

$$\begin{cases} 2x - 3y - z + 2w + 3v = 4 \\ 4x - 4y - z + 4w + 11v = 4 \\ 2x - 5y - 2z + 2w - v = 9 \\ \quad 2y + z + 4v = -5 \end{cases}$$

Answer:The augmented matrix is

$$\left(\begin{array}{ccccc|c} 2 & -3 & -1 & 2 & 3 & 4 \\ 4 & -4 & -1 & 4 & 11 & 4 \\ 2 & -5 & -2 & 2 & -1 & 9 \\ 0 & 2 & 1 & 0 & 4 & -5 \end{array} \right).$$

We use elementary row operations to transform this matrix into a triangular one. We keep the first row and use it to produce all zeros elsewhere in the first column. We have

$$\left(\begin{array}{ccccc|c} 2 & -3 & -1 & 2 & 3 & 4 \\ 0 & 2 & 1 & 0 & 5 & -4 \\ 0 & -2 & -1 & 0 & -4 & 5 \\ 0 & 2 & 1 & 0 & 4 & -5 \end{array} \right).$$

Next we keep the first and second row and try to have zeros in the second column. We get

$$\left(\begin{array}{ccccc|c} 2 & -3 & -1 & 2 & 3 & 4 \\ 0 & 2 & 1 & 0 & 5 & -4 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 & -1 \end{array} \right).$$

Next we keep the first three rows. We add the last one to the third to get

$$\left(\begin{array}{ccccc|c} 2 & -3 & -1 & 2 & 3 & 4 \\ 0 & 2 & 1 & 0 & 5 & -4 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right).$$

This is a triangular matrix. Its associated system is

$$\begin{cases} 2x - 3y - z + 2w + 3v = 4 \\ 2y + z + 5v = -4 \\ v = 1 \end{cases}$$

Clearly we have $v = 1$. Set $z=s$ and $w=t$, then we have

$$y = -2 - \frac{1}{2}s - \frac{5}{2}t = -\frac{9}{2} - \frac{1}{2}s.$$

The first equation implies

$$\frac{3}{2}x = \frac{1}{2}z - \frac{3}{2}t$$

$$x = 2 + y + z - w - v.$$

Using algebraic manipulations, we get

$$x = -\frac{25}{4} - \frac{1}{4}s - t.$$

Putting all the stuff together, we have

$$\begin{pmatrix} x \\ y \\ z \\ w \\ v \end{pmatrix} = \begin{pmatrix} -\frac{25}{4} - \frac{1}{4}s - t \\ -\frac{9}{2} - \frac{1}{2}s \\ s \\ t \\ 1 \end{pmatrix}.$$

Q.4) What is use of Gauss elimination method?

Answer: Gauss elimination is used to solve linear equation of the form $Ax = b$.

Q.5) What is the order of matrix in Gauss elimination method?

Answer: The matrix order is same as given matrix A.

0.000000.000000.000000
0.850000-1.0275001.010875
1.978588-1.1462440.880205
2.084265-1.1686290.866279
2.105257-1.1724750.863603
2.108835-1.1731450.863145
2.109460-1.1732620.863065
2.109568-1.1732820.863051

$x_1 = 2.110$

$x_2 = -1.173$

$x_3 = 0.863$

*/

Quiz

Q.1) Write down convergence property of Gauss Seidel method.

Answer: The convergence properties of the Gauss–Seidel method are dependent on the matrix A. Namely, the procedure is known to converge if either:

- A is symmetric positive-definite, or
- A is strictly or irreducibly diagonally dominant.

The Gauss–Seidel method sometimes converges even if these conditions are not satisfied.

Q.2) Explain Gauss Seidel method.

Answer: Given a square system of n linear equations with unknown x:

$$Ax = b$$

where:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

Then A can be decomposed into a lower triangular component L , and a strictly upper triangular component U:

$$A = L + U \quad \text{where} \quad L = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad U = \begin{bmatrix} 0 & a_{12} & \cdots \\ 0 & 0 & \cdots \\ \vdots & \vdots & \ddots \\ 0 & 0 & \cdots \end{bmatrix}$$

The system of linear equations may be rewritten as:

$$Lx = b - Ux$$

The Gauss–Seidel method is an iterative technique that solves the left hand side of this expression for x, using previous value for x on the right hand side. Analytically, this may be written as:

$$x^{(k+1)} = L^{-1}(b - Ux^{(k)}).$$

However, by taking advantage of the triangular form of L^* , the elements of $x_{(k+1)}$ can be computed sequentially using forward substitution:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^n a_{ij} x_j^{(k+1)} \right), \quad i = 1, 2, \dots, n.$$

Note that the sum inside this computation of $x_{i(k+1)}$ requires each element in $x_{(k)}$ except $x_{i(k)}$ itself.

The procedure is generally continued until the changes made by an iteration are below some tolerance.

Q.3) What is the use of Gauss Seidal method.

Answer: In numerical linear algebra, the Gauss–Seidel method, also known as the Liebmann method or the method of successive displacement, is an iterative method used to solve a linear system of equations. It is named after the German mathematicians Carl Friedrich Gauss and Philipp Ludwig von Seidel, and is similar to the Jacobi method. Though it can be applied to any matrix with non-zero elements on the diagonals, convergence is only guaranteed if the matrix is either diagonally dominant, or symmetric and positive definite.

Q.4) Solve the following equations using Gauss Seidal Method.

$$\begin{aligned} 10x_1 - x_2 + 2x_3 &= 6, \\ -x_1 + 11x_2 - x_3 + 3x_4 &= 25, \\ 2x_1 - x_2 + 10x_3 - x_4 &= -11, \\ 3x_2 - x_3 + 8x_4 &= 15. \end{aligned}$$

Solving for x_1, x_2, x_3 and x_4 gives:

$$\begin{aligned} x_1 &= x_2/10 - x_3/5 + 3/5, \\ x_2 &= x_1/11 + x_3/11 - 3x_4/11 + 25/11, \\ x_3 &= -x_1/5 + x_2/10 + x_4/10 - 11/10, \\ x_4 &= -3x_2/8 + x_3/8 + 15/8. \end{aligned}$$

Suppose we choose (0, 0, 0, 0) as the initial approximation, then the first approximate solution is given by

$$x_1 = 3/5 = 0.6,$$

$$x_2 = (3/5)/11 + 25/11 = 3/55 + 25/11 = 2.3272,$$

$$x_3 = -(3/5)/5 + (2.3272)/10 - 11/10 = -3/25 + 0.23272 - 1.1 = -0.9873,$$

$$x_4 = -3(2.3272)/8 + (-0.9873)/8 + 15/8 = 0.8789.$$

Using the approximations obtained, the iterative procedure is repeated until the desired accuracy has been reached. The following are the approximated solutions after four iterations.

x_1	x_2	x_3	x_4
0.6	2.32727	-0.987273	0.878864
1.03018	2.03694	-1.01446	0.984341
1.00659	2.00356	-1.00253	0.998351
1.00086	2.0003	-1.00031	0.99985

method is used to find it.

The exact solution of the system is (1, 2, -1, 1).

Q.5) Which method is similar to Jacobi method?

Answer: Gauss Seidal method is similar to Jacobi method.

Program 6

OBJECTIVES: To Solve The System Of Linear Equations Using Gauss - Jordan Method.

(Code Gauss Jordan method)

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{float a[6][6],b[6],x[6],t,s;
int i,j,n,k;
clrscr();
cout<<"Enter the maximum no. of matrix"<<endl;
cin>>n;
cout<<"Enter th elements of matrix"<<endl;
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
cin>>a[i][j];
}
}
cout<<"enter the right constant"<<endl;
for(i=0;i<n;i++)
{
cin>>a[i][n];
}
for(k=0;k<n;k++)
{
for(i=0;i<n;i++)
if(i!=k)
{
for(j=k+1;j<n+1;j++)
{
a[i][j]=a[i][j]-(a[i][k]/a[k][k])*a[k][j];
}}}
cout<<"the solution is"<<endl;
for(i=0;i<n;i++)
{
x[i]=(a[i][n]/a[i][i]);
cout<<"x["<<i<<"]="<<x[i]<<endl;
}
getch();
}
```

QUIZ

Q.1) How Gauss Jordan method is different from Gauss Elimination Method?

Answer: Gauss Jordan method is similar method except that instead of first converting A into upper triangular form, it is directly converted into the unit matrix.

Q.2) What happens when A reduced to I?

Answer: As soon as A reduces to I, the other matrix represents A^{-1}

Q.3) What are the numerical method of gauss elimination and gauss Jordan method?

Answer: In Computer Programming, Algebra, Linear Algebra

Q.4) Explain Gauss Jordan method.

Answer: Here two matrices A and I are written side by side and the same transformations are performed on both. As soon as A is reduced to I, the other matrix represents A^{-1}

Q.5) What is the Gauss Jordan method?

Answer: Gauss Jordan method is used to solve linear equation of the form $Ax = b$.

Program 7

OBJECTIVES: To integrate numerically using trapezoidal rule.

(Trapezoidal Rule).

Composite Trapezoidal

$$\int_a^b f(x) dx \approx \frac{b-a}{2} \left[f(A) + f(B) \right] + h \sum_{k=1}^{M-1} f(x_k)$$

by sampling $f(x)$ at the $M + 1$ equally spaced points

$x_k = A + h \cdot k$ for $k = 0, 1, 2, \dots, M$.

Notice that $x_0 = A$ and $x_M = B$.

 */

/* User has to supply a function named : ffunction
 An example is included in this program */

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
```

/* define prototype for USER-SUPPLIED function f(x) */

```
double ffunction(double x);
```

/* EXAMPLE for "ffunction" */

```
double ffunction(double x)
```

```
{
    return ( 1 / ( 1 + pow(x,2) ) );
}
```

/* ----- */

/* Main program for algorithm 7.1 */

```
void main()
```

```
{
    int K; /* loop counter*/
    int M; /* INPUT : number of subintervals */
```

NM LAB (MATH-204-F)

```
double    A,B;                                /* INPUT : boundaries of integral */
double    H;                                  /* Subinterval width*/
double    SUM = 0;                            /* approx. integral value*/
double    X;

printf("-----\n");
printf("Please enter the boundaries of the integral [A,B]\n");
printf("EXAMPLE: A = -1 and B = 1, so type: -1 1\n");
scanf("%lf %lf",&A, &B);
printf("The boundaries of the integral are: %lf %lf\n",A, B);
printf("-----\n");
printf("Please enter the number of SUBINTERVALS.\n");
scanf("%d",&M);

printf("You say : %d subintervals.\n",M);

H = (B - A)/M;                                /* Subinterval width */

for ( K = 1; K <= M-1; K++ ) {
    X = A + H*K;
    SUM = SUM + ffunction(X);
}

SUM = H * ( ffunction(A) + ffunction(B) + 2*SUM )/2;

printf("-----\n");

printf(" The approximate value of the integral of f(X)\n");
printf(" on the interval %lf %lf\n", A,B);
printf(" using %d subintervals computed using the\n");
trapezoidal\n",M);
printf(" rule is : %lf\n",SUM);

printf("-----\n");

} /* End of main program */
```

Quiz

Q.1)What is use of Trapezoidal rule?

Answer: Trapezoidal rule is used to solve numerical differentiation.

Q.2)What is numerical differentiation?

Answer: The process of evaluating a definite integral from a set of tabulated values of the integrand $f(x)$ is called numerical integration. This process when applied to a function of a single variable, is known as quadrate.

Q.3) What is observation of Trapezoidal rule?

Answer: Obs. : The area of each strip(trapezium) is found separately. Then the are under the curve and ordinates at x_0 and x_0+nh is approximately equal to the sum of the areas of the n trapeziums.

Q.4) Write an equation of Trapezoidal rule.

Answer: $\int_{x_0}^{x_0+h} f(x) dx = h/2 [(y_0 + y_n) + 2(y_1 + y_2 + \dots + y_{n-1})]$

Q.5) Which formula is used in Trapezoidal rule?

Answer: Newton-Cotes formula is used in Trapezoidal rule.

Program 8

OBJECTIVES: To Integrate Numerically Using Simpson's Rules.

(Code for SIMPSON'S 1/3 RULE)

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
    float x[10],y[10],sum=0,h,temp;
    int i,n,j,k=0;
    float fact(int);
    clrscr();
    printf("\nhow many record you will be enter: ");
    scanf("%d",&n);
    for(i=0; i<n; i++)
    {
        printf("\n\nenter the value of x%d: ",i);
        scanf("%f",&x[i]);
        printf("\n\nenter the value of f(x%d): ",i);
        scanf("%f",&y[i]);
    }
    h=x[1]-x[0];
    n=n-1;
    sum = sum + y[0];
    for(i=1;i<n;i++)
    {
        if(k==0)
        {
            sum = sum + 4 * y[i];
            k=1;
        }
        else
        {
            sum = sum + 2 * y[i];
            k=0;
        }
    }
    sum = sum + y[i];
    sum = sum * (h/3);
    printf("\n\n I = %f ",sum);
    getch();
}
```

/*

OUT PUT

how many record you will be enter: 5

NM LAB (MATH-204-F)

enter the value of x0: 0

enter the value of f(x0): 1

enter the value of x1: 0.25

enter the value of f(x1): 0.8

enter the value of x2: 0.5

enter the value of f(x2): 0.6667

enter the value of x3: 0.75

enter the value of f(x3): 0.5714

enter the value of x4: 1

enter the value of f(x4): 0.5

$$I = 0.693250$$

*/

QUIZ

Q.1) What is use of Simpsons one-third rule?

Answer: Simpsons one-third rule is used to find numerical integration of given equation.

Q.2) List Simpsons rule.

Answer: Simpsons one-third rule and Simpsons three eight rule.

Q.3) Write an observation about Simpsons one third rule.

Answer: While applying Simpsons one third rule, the given integral must be divided into even number of equal subintervals, since we find the area of two strips at a time.

Q.4) Write down Simpsons one-third rule.

Answer: $\int_{x_0}^{x_0+h} f(x) dx = h/3 [(y_0 + y_n) + 4(y_1 + y_3 + \dots + y_{n-1}) + (y_2 + y_{n-2})]$

Q.5) Simpsons one-third rule is applied on which point of parabola?

Answer: Putting $n=2$ in Newtons quadrate formula and taking the curve through (x_0, y_0) , (x_1, y_1) , (x_2, y_2) as a parabola.

Program 9

OBJECTIVES: To find the largest eigen value of matrix by power method.

/*To compute the dominant value Lambda_1 and its associated eigenvector V_1 for the n x n matrix A. It is assumed that the n eigenvalues have the dominance property

$|\text{Lambda}_1| > |\text{Lambda}_2| \geq |\text{Lambda}_3| \geq \dots \geq |\text{Lambda}_n| > 0$

```
-----
----
*/
```

Source code:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
```

```
#define MaxOrder 50
```

```
#define MAX(a,b) a > b ? a : b
```

```
/* ----- */
```

```
/* Main program for algorithm 11.1 */
```

```
void main(void)
```

```
{
    int i, j; /* Loop counter */
    int N; /* Order of Matrix */
    double Epsilon = 1E-7; /* Tolerance */
    double Max = 100; /* Maximum number of iterations */
    double X[MaxOrder], Y[MaxOrder];
    double A[MaxOrder][MaxOrder]; /* Matrix */
    double C1, DC, DV, Lambda = 0;
    int Count = 0, Iterating = 1;
    double Sum, Err = 1;
    double MaxElement;
```

```
    printf("-----Power Method-----\n");
    printf("----- Example 11.5 on page 550 ----- \n");
    printf("-----\n");
```

```
    printf("Please enter order of Matrix          A ( < %d !)\n", MaxOrder +
1);
    scanf("%d",&N);
    if( N > MaxOrder )
    {
```

```

printf(" Number of steps must be less than %d\n",MaxOrder+1);
printf(" Terminating. Sorry\n");
exit(0);
}
printf("Please enter elements of matrix A row by row:\n");

for ( i = 0; i < N; i++)
{
    for ( j = 0; j < N; j++)
    {
        printf(" Enter Element No. %d of row %d\n", j+1, i+1);
        scanf("%lf", &A[i][j]);
        printf("You entered A[%d][%d]= %lf\n", i+1, j+1, A[i][j] );
        printf("-----\n");
    }
}

printf("\n");

/* Initialize vector X */

for ( i = 0; i < N; i++)          X[i] = 1.0;

while( (Count <= Max) && (Iterating == 1) )
{

    /* Perform Matrix-Vector multiplication */

    for ( i = 0; i < N; i++ )
    {
        Y[i] = 0;
        for ( j = 0; j < N; j++ ) Y[i] += A[i][j] * X[j];
    }

    /* Find largest element of vector Y */
    /* Do what function MaxElement(X,N) in the book does */

    MaxElement = 0;
    for ( j = 0; j < N; j++ )
    {
        if( fabs(Y[j]) > fabs(MaxElement) ) MaxElement = Y[j];
    }

    C1 = MaxElement;

    DC = fabs(Lambda - C1);

    for ( i = 0; i < N; i++) Y[i] *= 1.0/C1;

    /* Do what function DIST(X,Y,N) in the book does */

    Sum = 0;
    for ( i = 0; i < N; i++) Sum += pow( ( Y[i] - X[i] ), 2.0);
    DV = sqrt(Sum);
    Err = MAX(DC,DV);

```

```
/* Update vector X and scalar Lambda */

for ( i = 0; i < N; i++) X[i] = Y[i];
Lambda = C1;

Iterating = 0;

if( Err > Epsilon) Iterating = 1;

Count++;

} /* End of while loop */

/* Output vector X and scalar Lambda */

printf("-----\n");

for ( j = 0; j < N; j++) printf("X[%d] = %lf\n", j, X[j]);

printf("-----\n");
printf("Lambda = %lf\n", Lambda);

} /* End of main program */
```

Quiz

Q.1)What is the use of Power method?

Answer: In many engineering application, it is required to compute the numerically largest eigen value and the corresponding eigen vector. Power method is used to find it.

Q.2)What is the observation of Power method?

Answer: OBS: Rewritting $AX=YX$ as $A^{-1}AX=Y A^{-1}X$

We have $A^{-1}X=1/YX$

If we use this equation, then power method yields the smallest eigen values.

Q.3)Does power method require normalization?

Answer: Yes

Q.4)When does power method yields the smallest eigen value?

Answer: We have $A^{-1}X=1/YX$

If we use this equation, then power method yields the smallest eigen values.

Q.5)Write down two properties of Eigen values.

Answer:1)The sum of eigen values of matrix A, is the sum of the elements of its principal diagonal.

2)Any similarity transformation applied to a matrix leaves its eigen values unchanged.

Program 10

OBJECTIVES: TO FIND NUMERICAL SOLUTION OF ORDINARY
DIFFERENTIAL EQUATIONS BY EULER'S METHOD.

(Code for Program of EULER'S METHOD)

```
#include<stdio.h>
#include <math.h>
#include<conio.h>
//dy/dx = xy#define F(x,y) (x)*(y)
void main()
{
    double y1,y2,x1,a,n,h;
    int j;
    clrscr();
    printf("\nEnter the value of range: ");
    scanf("%lf %lf",&a,&n);
    printf("\nEnter the value of y1: ");
    scanf("%lf",&y1);
    printf("\nEnter the h: ");
    scanf("%lf",&h);
    printf("\n\n y1 = %.3lf ",y1);
    for(x1=a,j=2; x1<=n+h; x1=x1+h,j++)
    {
        y2= y1 + h * F(x1,y1);
        printf("\n\n x = %.3lf => y%d = %.3lf ",x1,j,y2);
        y1=y2;
    }
    getch();
}
```

```
/*
OUT PUT
-----
```

Enter the value of range: 1 1.5

Enter the value of y1: 5

Enter the h: 0.1

y1 = 5.000

x = 1.000 => y2 = 5.500

x = 1.100 => y3 = 6.105

x = 1.200 => y4 = 6.838

x = 1.300 => y5 = 7.726

$$x = 1.400 \Rightarrow y_6 = 8.808$$

$$x = 1.500 \Rightarrow y_7 = 10.129$$

*/

QUIZ

Q.1)What is the use of Eulers method?

Answer: Euler's method is used to find out the solution of ordinary differential equation.

Q.2)What happens when h is small?

Answer:When the h is small the error is bound to be quite significant.

Q.3)What are the disadvantages of Eulers method?

Answer: This method is very slow.

Q.4)How the equation is solved using Eulers method?

Answer: In Eulers method we approximate the curve of the solution by the tangent in each interval.

Q.5)Write the equation for Eulers method for finding approximate solution?

Answer: $y_n = y_{n-1} + hf(x_{n-1}, y_{n-1})$

Program 11

OBJECTIVES: To Find Numerical Solution Of Ordinary
Differential Equations By Runge- Kutta Method.

```
/* Runge Kutta for a set of first order differential equations */

#include <stdio.h>
#include <math.h>

#define N 2 /* number of first order equations */
#define dist 0.1 /* stepsize in t */
#define MAX 30.0 /* max for t */

FILE *output; /* internal filename */

void runge4(double x, double y[], double step); /* Runge-Kutta function */
double f(double x, double y[], int i); /* function for derivatives */

void main()
{
    double t, y[N];
    int j;

    output=fopen("osc.dat", "w"); /* external filename */

    y[0]=1.0; /* initial position */
    y[1]=0.0; /* initial velocity */

    fprintf(output, "0\t%f\n", y[0]);

    for (j=1; j*dist<=MAX ;j++) /* time loop */
    {
        t=j*dist;
        runge4(t, y, dist);
        fprintf(output, "%f\t%f\n", t, y[0]);
    }

    fclose(output);
}

void runge4(double x, double y[], double step)
{
    double h=step/2.0, /* the midpoint */
    t1[N], t2[N], t3[N], /* temporary storage arrays */
```

```
k1[N], k2[N], k3[N],k4[N]; /* for Runge-Kutta */
int i;

for (i=0;i<N;i++)
{
    t1[i]=y[i]+0.5*(k1[i]=step*f(x,y,i));
}

for (i=0;i<N;i++)
{
    t2[i]=y[i]+0.5*(k2[i]=step*f(x+h, t1, i));
}

for (i=0;i<N;i++)
{
    t3[i]=y[i]+ (k3[i]=step*f(x+h, t2, i));
}

for (i=0;i<N;i++)
{
    k4[i]= step*f(x+step, t3, i);
}

for (i=0;i<N;i++)
{
    y[i]+=(k1[i]+2*k2[i]+2*k3[i]+k4[i])/6.0;
}
}

double f(double x, double y[], int i)
{
    if (i==0)
        x=y[1]; /* derivative of first equation */
    if (i==1)
        x= -0.2*y[1]-y[0]; /* derivative of second equation */

    return x;
}
```

QUIZ

Q.1)What is observation of RK method?

Answer: Here the operation is identical whether differential equation is linear or non-linear.

Q.2)Write down RK rule.

Answer: $k=1/6(k_1+2k_2+2k_3+k_4)$

Q.3)What is working rule of RK method?

Answer: Working rule for finding k of y corresponding to an increment h of x by RK method from

$dy/dx=f(x, y)$, $y(x_0)=y_0$

is as follows:

Calculate successively $k_1=hf(x_0, y_0)$, $k_2=hf(x_0+1/2h, y_0+1/2k_1)$

$k_3= hf(x_0+1/2h, y_0+1/2k_2)$ and $k_4= hf(x_0+h, y_0+k_3)$

Finally compute $k=1/6(k_1+2k_2+2k_3+k_4)$

Q.4)What is the use of RK method?

Answer: RK method is used to solve differential equation.

Q.5)What are the advantages of RK method?

Answer:1)RK method do not require the calculation of higher order derivatives.

2) This method gives greater accuracy.

3)It requires only the functions values at some selected point.

Program 12

OBJECTIVES: To find the numerical solution of differential equation using milne method.

(Milne-Simpson Method)

To approximate the solution of the initial value problem with $y(a) = y_0$ over $[a,b]$ by using the predictor

$$y' = f(t,y)$$

$$p_{k+1} = y_{k-3} + \frac{4h}{3} [2f_{k-2} - f_{k-1} + 2f_k]$$

and the corrector :

$$y_{k+1} = y_{k-1} + \frac{h}{3} [f_{k-1} + 4f_k + f_{k+1}].$$

User has to supply functions named : ffunction
An example is included in this program.

```
-----
*/

#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define MAX 500

/*    define prototype for USER-SUPPLIED function f(x)                                */

double ffunction(double t, double y);

/*    EXAMPLE for "ffunction"                                                        */

double ffunction(double t, double y)
{
    return ( (t - y) / 2.0 );
}

/* ----- */

/*    Main program for algorithm 9.7                                                */

void main(void)
{
```

NM LAB (MATH-204-F)

```

int l,      K;          /* Loop counter          */
int N;      /* Number of steps > 3  */
double A, B, Y[MAX];   /* Endpoints and initial value */
double H;    /* Compute the step size */
double T[MAX];
double F1, F2, F3, F4;  /* Function values       */
double Hmin, Hmax;     /* Minimum and maximum step size */
double Pold, Yold;     /* Predictor and Corrector */
double Pmod, Pnew;     /* Modifier              */

```

```

printf("-----Milne-Simpson Method-----\n");
printf("----- Example 9.13 on page 468 -----\n");
printf("-----\n");
printf("Please enter endpoints of the interval [A,B]:\n");
printf("For used Example type : 0, 3.0\n");
scanf("%lf, %lf", &A, &B);
printf("You entered [%lf, %lf]\n", A, B);
printf("-----\n");
printf("Please enter number of steps: ( > 3 and < %d !)\n", MAX+1);
scanf("%d",&N);
if( (N > MAX) || (N < 4) )
{
printf(" Number of steps must be greater than 3 and less than
%d\n",MAX+1);
printf(" Terminating. Sorry\n");
exit(0);
}
printf("-----\n");

printf("You need all together FOUR initial values. You can\n");
printf("use the Runge-Kutta method to compute the 2nd, 3rd and\n");
printf("4th from the 1st one.\n");
printf("Please enter initial values Y[0], Y[1], Y[2], Y[3] :\n");
printf("Example 9.13 page 468: 1, 0.94323919, 0.89749071,
0.86208736\n");
scanf("%lf, %lf, %lf, %lf", &Y[0], &Y[1], &Y[2], &Y[3]);
printf("You entered Y[0] = %lf\n", Y[0]);
printf("You entered Y[1] = %lf\n", Y[1]);
printf("You entered Y[2] = %lf\n", Y[2]);
printf("You entered Y[3] = %lf\n", Y[3]);

```

```

/* Compute the step size and initialize */

```

```

H= (B - A) / N;
T[0] = A;

```

```

for( K = 1; K <= 3; K++) T[K] = A + K * H;

```

```

F1 = ffunction(T[1],Y[1]);
F2 = ffunction(T[2],Y[2]);
F3 = ffunction(T[3],Y[3]);

```

```

Pold = 0;
Yold = 0;

```

```

for( K = 3; K <= N-1; K++)

```

```
{
    /* Milne Predictor */
    Pnew = Y[K-3] + 4.0 * H * (2.0*F1 - F2 + 2.0*F3) / 3.0;
    Pmod = Pnew + 28.0 * (Yold - Pold) / 29.0;
    /* Next mesh point */
    T[K+1] = A + H * (K+1);
    /* Evaluate f(t,y) */
    F4 = ffunction(T[K+1],Pmod);
    /* Corrector */
    Y[K+1] = Y[K-1] + H * (F2 + 4.0 * F3 + F4) / 3.0;
    /* Update the values */
    F1 = F2;
    F2 = F3;
    F3 = ffunction(T[K+1], Y[K+1]);

}

/* Output */

for ( K = 0; K <= N; K++)
{
    printf("K = %d, T[K] = %lf, Y[K] = %lf\n", K, T[K], Y[K]);
}

} /* End of main program */
```

QUIZ

Q.1)How to get greater accuracy in Mline method?

Answer:To insure greater accuracy, we must first improve the accuracy of the starting value and then subdivide the interval.

Q.2)What is the use of Mline method?

Answer:Mline method used to solve the differential equation over an interval.

Q.3)What is the equation of predictor in Mline method

Answer: $Y_{4(p)} = y_0 + 4h/3(2f_1 - f_2 + 2f_3)$

Q.4) What is the equation of a corrector in Mline method?

Answer: $Y_{4(c)} = y_2 + h/3(f_2 - 4f_3 + f_4)$

Q.5) What is the better approximation to the value of y_5 ?

Answer: $Y_{5(c)} = y_3 + h/3(f_3 + 4f_4 + f_5)$

Program 13

OBJECTIVES: To find the numerical solution of differential equation using laplace equation.

(Dirichlet Method for Laplace's Equation)

To approximate the solution of $u_{xx}(x,y) + u_{yy}(x,y) = 0$

over $R = \{(x,y): 0 \leq x \leq a, 0 \leq y \leq b\}$ with

$u(x,0) = f_1(x), u(x,b) = f_2(x)$ for $0 \leq x \leq a$ and

$u(0,y) = f_3(y), u(a,y) = f_4(y)$ for $0 \leq y \leq b$.

It is assumed that $\Delta(x) = \Delta(y) = h$ and that integers

n and m exist so that $a = nh$ and $b = mh$.

User has to supply functions F1, F2, F3, F4 with gridpoints. as arguments. An example is given in this program.

---*/

Source code:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
```

```
#define Max 50
```

```
/* Global Variables */
```

```
double H; /* Step size */
```

```
/* Prototypes */
```

```
double F1i(int i);
double F2i(int i);
double F3i(int i);
double F4i(int i);
```

```
/* Grid function for amplitude */
```

```
double F1i(int i)
{
    extern double H;
    double arg;
```

```
    arg = H * ( i - 1.0 );

    if( arg >= 0 ) && ( arg <= 4.0 ) return ( 180.0 );
    else
    {
        printf(" F1i() :Argument not in range ! Exiting !\n");
        printf(" arg : %lf\n", arg);
        exit(0);
    }
}

double F2i(int i)
{
    extern double H;
    double arg;

    arg = H * ( i - 1.0 );

    if( arg >= 0 ) && ( arg <= 4 ) return ( 20.0 );
    else
    {
        printf(" F2i() :Argument not in range ! Exiting !\n");
        printf(" arg : %lf\n", arg);
        exit(0);
    }
}

double F3i(int i)
{
    extern double H;
    double arg;

    arg = H * ( i - 1.0 );

    if( arg >= 0 ) && ( arg <= 4 ) return ( 80.0 );
    else
    {
        printf(" F3i() :Argument not in range ! Exiting !\n");
        printf(" arg : %lf\n", arg);
        exit(0);
    }
}

double F4i(int i)
{
    extern double H;
    double arg;

    arg = H * ( i - 1.0 );

    if( arg >= 0 ) && ( arg <= 4 ) return ( 0.0 );
    else
    {
        printf(" F4i() :Argument not in range ! Exiting !\n");
        printf(" arg : %lf\n", arg);
        exit(0);
    }
}
```

```

}

/* ----- */

void main(void)
{
    int i, j;                /* Loop Counters
*/
    double A, B;            /* INPUT : Width and height of Rectangle
*/
    double Ave;             /* INPUT : Initial approximation
*/
    int N, M;               /* dimensions of the grid
*/
    double U[Max][Max];     /* Grid-Amplitudes
*/
    int Count;              /* counter for while loop
*/
    double Tol;             /* stop criteria
*/
    double w, Pi;
    double H;               /* Grid spacing
*/
    double Relax, temp;

    printf("-----\n");
    printf("----- Dirichlet Method for Laplace's Equation -----
\n");
    printf("----- Try Example 10.6 on page 528/529-----
\n");
    printf("-----\n");

    printf("Please enter A (interval boundary of x)\n");
    printf("For present example type 4\n");
    scanf("%lf",&A);
    printf("Please enter B (interval boundary of y)\n");
    printf("For present example type 4\n");
    scanf("%lf",&B);
    printf("You entered A = %lf and B = %lf \n", A, B);
    printf("-----\n");
    printf("Please enter Grid-Spacing H\n");
    printf("For present example type 0.5\n");
    scanf("%lf",&H);
    printf("You entered H = %lf\n", H);
    printf("-----\n");
    printf("\n");
    printf("Please enter dimension of grid in x-direction :\n");
    printf("For present example type 9\n");
    scanf("%d",&N);
    printf("Please enter dimension of grid in y-direction :\n");

```

```

printf("For present example type 9\n");
scanf("%d",&M);
printf("You entered N = %d and M = %d\n", N, M);
printf("-----\n");
printf("\n");
printf("Please enter the initial approximation :\n");
printf("For present example type 70\n");
scanf("%lf",&Ave);
printf("You entered = %lf\n", Ave);
printf("-----\n");
printf("\n");

/* Compute step sizes */

Pi = 3.1415926535;

/* Only to make the 0-indexes save we initialize them to zero.
   This is only because in C fields begin with index 0.
   In this program we ignore the index 0. */

for ( i = 0; i < N; i++ ) U[i][0] = 0.0;
for ( i = 1; i < M; i++ ) U[0][i] = 0.0;

/* Initialize starting values at the interior points */

for ( i = 2; i <= N-1; i++ )
{
    for ( j = 2; j <= M-1; j++ )
        U[i][j] = Ave;

/* Store boundary values in the solution matrix */

for ( j = 1; j <= M ; j++ )
{
    U[1][j] = F3i(j);
    U[N][j] = F4i(j);
}

for ( i = 1; i <= N ; i++ )
{
    U[i][1] = F1i(i);
    U[i][M] = F2i(i);
}

/* The SQR parameter */

temp =      cos( Pi/(N-1) ) + cos( Pi/(M-1) );

w = 4.0 / ( 2.0 + sqrt( 4.0 - temp * temp ) );

/* Initialize the loop control parameters */

Tol= 1.0;
Count = 0.0;

while ( (Tol > 0.001) && (Count <= 140) )
{

```

```
Tol = 0.0;
for ( j = 2; j <= M - 1; j++ )
{
    for ( i = 2; i <= N - 1; i++ )
    {
        Relax = w * ( U[i][j+1] + U[i][j-1] + U[i+1][j] +
                     U[i-1][j] - 4.0 * U[i][j] ) / 4.0;
        U[i][j] += Relax;
        if( fabs(Relax) > Tol ) Tol = fabs(Relax);
    }
    Count++;
}

}

/* Output the solution */

for ( j = 1; j <= M; j++ )
{
    for ( i = 1; i <= N; i++ ) printf("%8.4lf ", U[i][j]);
    printf("\n");
}

} /* End of main program */
```

QUIZ

Q.1)What is the use of Laplace method?

Answer: Laplace method is used to find out the solution of partial differential equation.

Q.2)What is the boundary of Rectangular region R.

Answer: $u(x,y)$ is the boundary of rectangular region R.

Q.3)What is the first step in Laplace equation?

Answer:Divide the given into network of square mesh of side h.

Q.4)What is the four mesh point?

Answer:The four mesh point is the average of its values at four neighbouring point to the left, right, above and below.

Q.5)How can we find the interior points?

Answer: Interior points are computed by the standard five point formula.

Program 14

OBJECTIVES: To find the numerical solution of differential equation using wave equation

(Finite-Difference Solution for the Wave Equation)

To approximate the solution of $u_{tt}(x,t) = c^2 u_{xx}(x,t)$

over $R = \{(x,t): 0 \leq x \leq a, 0 \leq t \leq b\}$ with

$u(0,t) = 0, \quad u(a,t) = 0 \quad \text{for} \quad 0 \leq t \leq b$ and

$u(x,0) = f(x), \quad u_t(x,0) = g(x) \quad \text{for} \quad 0 \leq x \leq a.$

User has to supply function $F_i(\text{gridpoint})$ and $G_i(\text{gridpoint})$: boundary functions at the grid points. An examples is implemented in this program.

Source code:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define Max 50

/* Global Variables */

double H;          /* Step size */

/* Prototypes */

double Fi(int i);
double Gi(int i);

/* Grid function for amplitude */

double Fi(int i)
{
    extern double H;
    double arg;

    arg = H * (i - 1);

    if ( arg >= 0 && (arg <= 1.0) )
        return ( sin ( 3.1415926 * arg ) );
    else
    {
        printf(" Fi() :Argument not in range ! Exiting !\n");
    }
}
```

```

        printf(" arg          : %lf\n", arg);
        exit(0);
    }
}

/* Grid function for velocity */

double Gi(int i)
{
    extern double H;
    double arg;

    arg = H * (i - 1);

    if( (arg >= 0) && (arg <= 1) ) return ( 0.0 );
    else
    {
        printf(" Gi() :Argument not in range ! Exiting !\n");
        printf(" arg : %lf\n", arg);
        exit(0);
    }
}

/* ----- */

/*   Main program for algorithm 10-1           */

void main(void)
{
    int i, j;                                /* Loop Counters
*/
    double A, B;                             /* INPUT :Width and height of Rectangle
*/
    double C;                                /* INPUT : Wave equation const.
*/
    int N, M;                                /* Dimensions of the grid
*/
    double K, R, R2, R22, S1, S2;
    double U[Max][Max];/* Grid-Amplitudes
*/

    printf("-- Finite-Difference Solution to the Wave Equation ----
\n");
    printf("----- Example Problem No 4 on page 508 -----
\n");
    printf("-----
\n");

    printf("Please enter A (interval boundary of x)\n");
    printf("For present example type : 1.0\n");
    scanf("%lf",&A);
    printf("Please enter B (interval boundary of t)\n");
    printf("For present example type : 0.5\n");
    scanf("%lf",&B);
    printf("You entered A = %lf and B = %lf\n", A, B);

```



```

printf("-----
\n");
printf("\n");
printf("Please enter dimension of grid in x-direction :\n");
printf("For present example type : 6\n");
scanf("%d",&N);
printf("Please enter dimension of grid in y-direction :\n");
printf("For present example type : 6\n");
scanf("%d",&M);
printf("You entered N = %d and M = %d\n", N, M);
printf("-----
\n");
printf("\n");
printf("Please enter the wave equation constant C :\n");
printf("For present example type : 2\n");
scanf("%lf",&C);
printf("You entered C = %lf\n", C);
printf("-----\n");
printf("\n");

/* Compute step sizes */

H   = A /   ( N - 1 );
K   = B /   ( M - 1 );
R   = C *   K / H;
R2  = R *   R;
R22 = R *   R / 2.0;
S1  = 1.0 - R * R;
S2  = 2.0 - 2.0 * R * R;

/* Boundary conditions */

for ( j = 1; j <= M; j++ )
{
    U[1][j] = 0;
    U[N][j] = 0;
}

/* First and second rows */

for ( i = 2; i <= N - 1; i++ )
{
    U[i][1] = Fi(i);
    U[i][2] = S1 * Fi(i) + K * Gi(i) + R22 * ( Fi(i+1) + Fi(i-1) );
}

/* Generate new waves */

for ( j = 3; j <= M; j++ )
{
    for ( i = 2; i <= N - 1; i++ )
    {
        U[i][j] = S2 * U[i][j-1] + R2 * ( U[i-1][j-1] + U[i+1][j-1]
                                           - U[i][j-2];
    }
}

```

```
/* Output the solution */  
  
for ( j = 1; j <= M; j++ )  
{  
    printf("%8.6lf ", K * (j - 1));  
  
    for ( i = 1; i <= N; i++ ) printf(" %8.6lf ", U[i][j]);  
  
    printf("\n");  
}  
}  
/* End of main program */
```

Quiz

Q.1) What is the use of wave equation?

Wave equation is used to find out the partial solution of differential equation.

Q.2) What happens when $\alpha = 1/c$?

Answer: The solution is stable.

Q.3) What happens when $\alpha < 1/c$?

Answer: The solution is stable but inaccurate.

Q.4) What happens when $\alpha > 1/c$?

Answer: The solution is unstable.

Q.5) Give an example of hyperbolic partial differential equation.

Answer: Wave equation is the simplest example.

Program 15

OBJECTIVES: To find the numerical solution of differential equation using heat equation.

Source code:

(Forward-Difference Method for the Heat Equation).

```
-----
*/
/*
-----
```

(Forward-Difference Method for the Heat Equation)

To approximate the solution of $u_{tt}(x,t) = c^2 u_{xx}(x,t)$

over $R = \{(x,t): 0 \leq x \leq a, 0 \leq t \leq b\}$ with

$u(x,0) = f(x)$ for $0 \leq x \leq a$ and

$u(0,t) = c_1, \quad u(a,t) = c_2$ for $0 \leq t \leq b$.

User has to supply function $F_i(\text{gridpoint})$.
An example is implemented in this program.

```
-----
*/

#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define Max 50

/* Global Variables */

double H; /* Step size */

/* Prototypes */

double Fi(int i);

/* Grid function for amplitude */

double Fi(int i)
{
```

```

extern double H;
double arg;

arg = H * (i - 1);

if( (arg >= 0) && (arg <= 1.0) ) return (4.0 * arg - 4.0 * arg *
arg);
else
{
    printf(" Fi() :Argument not in range ! Exiting !\n");
    printf(" arg : %lf\n", arg);
    exit(0);
}
}

/* ----- */

/*    Main program for algorithm 10-2                                */

/*    Remember : in C the fields start with index 0.
    In this program this field is ignored and when you
    change the program be aware of the index 0 in the U[Max][Max]
    array ! Here - as said before - it is ignored, so don't
    access it if you do not initialize it. */

void main(void)

{
    int i, j;                                /* Loop Counters
*/
    double A, B;                             /* INPUT :Width and height of Rectangle
*/
    double C;                                /* INPUT : Wave equation const.
*/
    int N, M;                                /* Dimensions of the grid
*/
    double K, S, C1, C2, R;
    double U[Max][Max];                     /* Grid-Amplitudes
*/

    printf("-- Forward-Difference Method for the Heat Equation ----
\n");
    printf("----- Example 10.3 on page 511 -----
\n");
    printf("-----
\n");

    printf("Please enter A (interval boundary of x)\n");
    printf("For present example type : 1.0\n");
    scanf("%lf",&A);
    printf("Please enter B (interval boundary of t)\n");
    printf("For present example type : 0.2\n");
    scanf("%lf",&B);
    printf("You entered A = %lf and B = %lf \n", A, B);
    printf("-----
\n");
    printf("\n");

```

```

    printf("Please enter dimension of grid in x-direction ( < 50
):\n");
    printf("For present example type : 6\n");
    scanf("%d",&N);
    if ( N >= 50 )
    {
        printf("MUST BE SMALLER THAN 50. ... terminating.\n");
        exit(0);
    }
    printf("Please enter dimension of grid in y-direction ( < 50
):\n");
    printf("For present example type : 11\n");
    scanf("%d",&M);
    if ( M >= 50 )
    {
        printf("MUST BE SMALLER THAN 50. ... terminating.\n");
        exit(0);
    }
    printf("You entered N = %d and M = %d\n", N, M);
    printf("-----
\n");
    printf("\n");
    printf("Please enter the heat-equation constant C :\n");
    printf("For present example type : 1\n");
    scanf("%lf",&C);
    printf("You entered C = %lf\n", C);
    printf("-----\n");
    printf("\n");
    printf("Please enter constant C1 = u(0,t) :\n");
    printf("For present example type : 0\n");
    scanf("%lf",&C1);
    printf("You entered C1 = %lf\n", C1);
    printf("-----\n");
    printf("\n");
    printf("Please enter constant C2 = u(A,t) :\n");
    printf("For present example type : 0\n");
    scanf("%lf",&C2);
    printf("You entered C2 = %lf\n", C2);
    printf("-----\n");
    printf("\n");

```

/* Compute step sizes */

```

H   = A /  ( N - 1 );
K   = B /  ( M - 1 );
R   = C *  C * K /  ( H * H );
S   = 1.0 - 2.0 * R;

```

/* Boundary conditions */

```

for ( j = 1; j <= M; j++ )
{
    U[1][j] = C1;
    U[N][j] = C2;
}

```

```
/* First row */

for ( i = 2; i <= N - 1 ; i++ ) U[i][1] = Fi(i);

/* Generate new waves */

for ( j = 2; j <= M; j++ )
{
    for ( i = 2; i <= N - 1; i++ )
    {
        U[i][j] = S * U[i][j-1] + R * ( U[i-1][j-1] + U[i+1][j-1]
);
    }
}

/* Output the solution */

printf("The grid amplitudes look like this :\n");
printf("\n");

for ( j = 1; j <= M; j++ )
{
    printf("%8.6lf ", K * (j - 1));

    for ( i = 1; i <= N; i++ ) printf(" %8.6lf ", U[i][j]);

    printf("\n");
}

/* End of main program */
```

QUIZ

Q.1)What is the use of heat method?

Answer: Heat method is used to find out solution of partial differential equation.

Q.2)What are the different method for heat equation?

Answer:

- 1)Schmidt method
- 2)Crank –Nicolson method
- 3)Iterative method of solution.

Q.3)Which method is used for non restrict α ?

Answer: Crank –Nicolson method

Q.4)What is implicit scheme?

Answer: A method in which the calculation of an unknown mesh value necessitates the solution of a set of simultaneous equation is known as an implicit scheme.

Q.5)Which is 3-level method?

Answer:Richardson scheme is 3-level method.

REFERENCES:

1. Numerical methods by B.S.Grewal
2. Numerical method :E. Balagurusamy T.M.H

DIGITAL ELECTRONICS

(EE--224-F))

LAB MANUAL

IV SEMESTER

LIST OF EXPERIMENTS

SR. NO.	NAME OF EXPERIMENT	PAGE NO.
1	Introduction to Digital Electronics lab- nomenclature of digital ICS, specifications, study of the data sheet, concept of V_{CC} and ground, verification of the truth tables of logic gates using TTL ICS.3-6	
2	Implementation of the given Boolean function using logic gates in both sop and pos forms.	7-8
3	Verification of state tables of RS, JK, T and D flip-flops using NAND & nor gates.	9-11
4	Implementation and verification of decoder/de-multiplexer and encoder using logic gates.	12-15
5	Implementation of 4x1 multiplexer using logic gates.	16-18
6	Implementation of 4-bit parallel adder using 7483 IC.	19-20
7	Design and verify the 4-bit synchronous counter.	21-24
8	Design and verify the 4-bit asynchronous counter.	25-27
9	To design and verify operation of half adder and full adder.	28-29
10	To design and verify operation of half subtractor.	30-31
11	To design & verify the operation of magnitude comparator.	32-33
12	To study and verify NAND as a universal gate.	34-35
13	To study 4 bit ALU(IC 74181).	36-38
14	Mini project	39-41

EXPERIMENT NO: 1

Aim: - Introduction to Digital Electronics Lab- Nomenclature of Digital Ics, Specifications, Study of the Data Sheet, Concept of V_{cc} and Ground, Verification of the Truth Tables of Logic Gates using TTL Ics.

APPARATUS REQUIRED: Power Supply, Digital Trainer Kit., Connecting Leads, IC's (7400, 7402, 7404, 7408, 7432, and 7486)

BRIEF THEORY:

AND Gate: The AND operation is defined as the output as (1) one if and only if all the inputs are (1) one. 7408 is the two Inputs AND gate IC. A&B are the Input terminals & Y is the Output terminal.

$$Y = A.B$$

OR Gate: The OR operation is defined as the output as (1) one if one or more than 0 inputs are (1) one. 7432 is the two Input OR gate IC. A&B are the input terminals & Y is the Output terminal.

$$Y=A+B$$

NOT GATE: The NOT gate is also known as Inverter. It has one input (A) & one output (Y). IC No. is 7404. Its logical equation is,

$$Y = A \text{ NOT } B, Y = A'$$

NAND GATE: The IC no. for NAND gate is 7400. The NOT-AND operation is known as NAND operation. If all inputs are 1 then output produced is 0. NAND gate is inverted AND gate.

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

NOR GATE: The NOR gate has two or more input signals but only one output signal. IC 7402 is two I/P IC. The NOT- OR operation is known as NOR operation. If all the inputs are 0 then the O/P is 1. NOR gate is inverted OR gate.

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

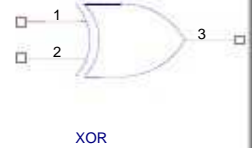
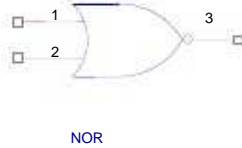
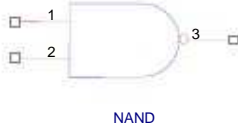
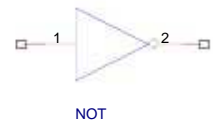
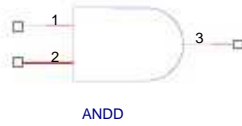
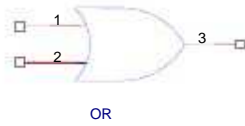
EX-OR GATE: The EX-OR gate can have two or more inputs but produce one output. 7486 is two inputs IC. EX-OR gate is not a basic operation & can be performed using basic gates.

$$Y=A \oplus B$$

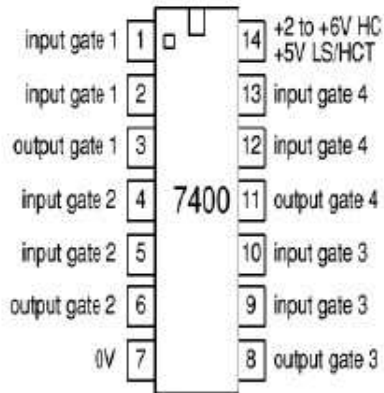
LOGIC SYMBOL:

. Logic Symbol of Gates

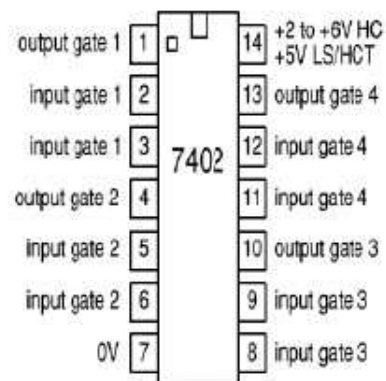
DIGITAL ELECTRONICS LAB (EE-2224-F)



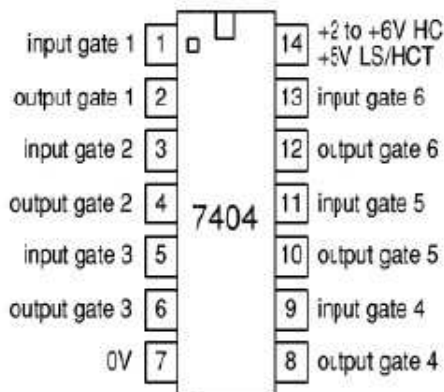
CRATION:PIN CONFIGUR
7400(NAND)



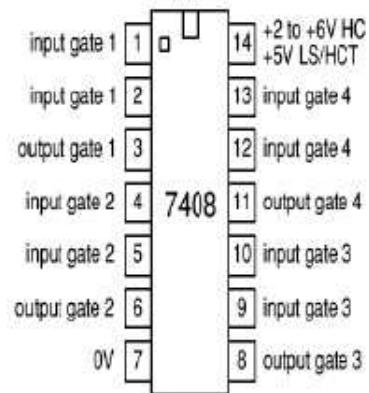
7402(NNOR)



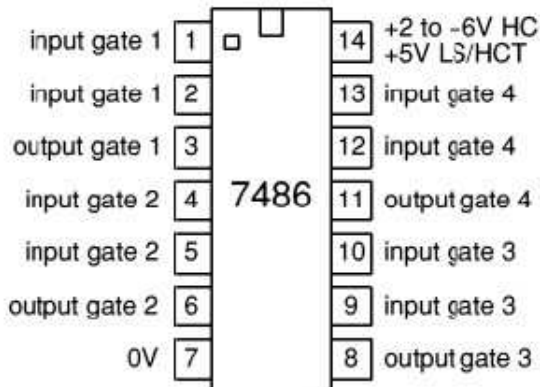
04(NOT)740



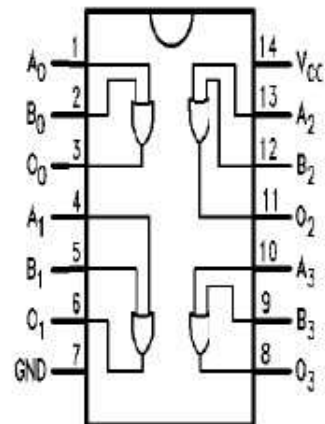
7408 (AAND)



7486(EX-OR)



7432(OR)7



PROCEDURE:

- Fix the IC's on breadboard & give the supply.
- Connect the +ve terminal of supply to pin 14 & -ve to pin 7.
- Give input at pin 1, 2 & take output from pin 3. It is same for all except NOT & NOR IC.
- For NNOR, pin 1 is output & pins 2 & 3 are inputs.
- For NNOT, pin 1 is input & pin 2 is output.
- Note the values of output for different combinations of inputs & draw the TRUTH TABLE.

OBSERVATION TABLE:

INPUTS				OUTPUTS			
A	B	A'	A+B	(A+B)'	(A*B)	(A*B)'	(A B)
			NOR	NOR	AND	NAND	Ex-ORR
0	0	1	0	1	0	1	0
0	1	1	1	0	0	1	1
1	0	0	1	0	0	1	1
1	1	0	1	0	1	0	0

ULT: We have learnt all the gates IC according to the IC pin diagram.

PRECAUTIONS

- Make the connections according to the IC pin diagram. Connections should be tight.
- The connection of V_{cc} & ground should be applied carefully at the specified pin only.

Quiz Questions with answer.

Q.1 Define gates ?

Ans. Gates are the digital circuits, which perform a specific type of logical operation.

Q.2 Define IC?

Ans. IC means integrated circuit. It is the integration of no. of components on a common substrate.

Q.3 Give example of Demorgan's theorem.

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

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

Q.4 $(A+A) A = ?$

Ans. A.

Q5 Define Universal gates.

Ans. Universal gates are those gates by using which we can design any type of logical expression.

Q6. Write the logical equation for AND gate.

Ans. $Y = A.B$

Q7 How many no. of input variables can a NOT Gate have?

Ans. One.

Q8. Under what conditions the output of a two input AND gate is one?

Ans. Both the inputs are one.

Q9. $1+0 = ?$

Ans. 1

Q10. When will the output of a NAND Gate be 0?

Ans. When all the inputs are 1.

EXPERIMENT NO: 2

Aim: Implementation of the Given Boolean Function using Logic Gates in Both Sop and Pos Forms.

APPARATUS REQUIRED: Power Supply, Digital Trainer, IC's (7404, 7408, 7432) Connecting leads.

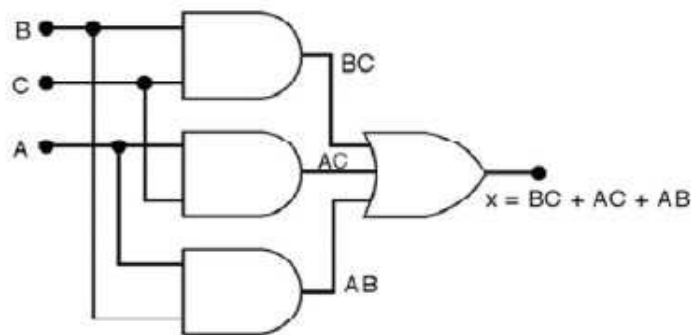
BRIEF THEORY: Karnaugh maps are the most extensively used tool for simplification of Boolean functions. It is mostly used for functions having up to six variables beyond which it becomes very cumbersome. In an n-variable K-map there are 2^n cells. Each cell corresponds to one of the combination of n variable, since there are 2^n combinations of n-variables. Gray code has been used for the identification of cells.

Example- $f(A, B, C, D) = A'BC + AB'C + ABC' + ABC$ (SOP)

Reduced form is $BC + AC + AB$ and POS form is $f(X, Y, Z) = Y' (X' + Y + Z') (X + Z)$

LOGIC DIAGRAM

SOP form



POS Form

PROCEDURE:

- (a) With given equation in SOP/POS forms first of all draw a K-map.
- (b) Enter the values of the O/P variable in each cell corresponding to its Min/Max term.
- (c) Make group of adjacent ones.
- (d) From group write the minimized equation.
- (e) Design the ckt. of minimized equation & verify the truth table.

RESULT/CONCLUSION: Implementation of SOP and POS form is obtained with AND and OR gates.

PRECAUTIONS:

- 1) Make the connections according to the IC pin diagram.
- 2) The connections should be tight.
- 3) The V_{cc} and ground should be applied carefully at the specified pin only.

Quiz Questions with answer.

Q.1 Define K-map ?

Ans. It is a method of simplifying Boolean Functions in a systematic mathematical way.

Q.2 Define SOP ?

Ans. Sum of Product.

Q.3 Define POS ?

Ans. Product of Sum.

Q.4 What are combinational circuits?

Ans. These are those circuits whose output depends upon the inputs present at that instant of time.

Q.5 What are sequential circuits?

Ans. These are those circuits whose output depends upon the input present at that time as well as the previous output.

Q.6 If there are four variables how many cells the K-map will have?

Ans. 16.

Q.7 When two min-terms can be adjacent?

Ans. 2 to the power n.

Q.8 Which code is used for the identification of cells?

Ans. Gray Code.

Q.9 Define Byte?

Ans. Byte is a combination of 8 bits.

Q.10 When simplified with Boolean Algebra $(x + y)(x + z)$ simplifies to

Ans. $x + yz$

EXPERIMENT NO: 3

Aim: Verification of State Tables of Rs, J-k, T and D Flip-Flops using NAND & NOR Gates

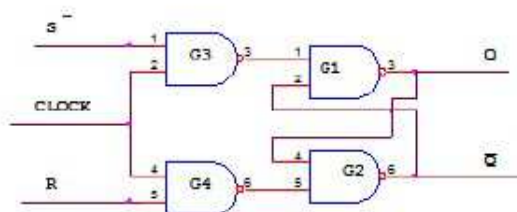
APPARATUS REQUIRED: IC' S 7400, 7402 Digital Trainer & Connecting leads.

BRIEF THEORY:

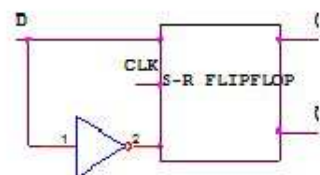
- **RS FLIP-FLOP:** There are two inputs to the flip-flop defined as R and S. When I/Ps $R = 0$ and $S = 0$ then O/P remains unchanged. When I/Ps $R = 0$ and $S = 1$ the flip-flop is switches to the stable state where O/P is 1 i.e. SET. The I/P condition is $R = 1$ and $S = 0$ the flip-flop is switched to the stable state where O/P is 0 i.e. RESET. The I/P condition is $R = 1$ and $S = 1$ the flip-flop is switched to the stable state where O/P is forbidden.
- **JK FLIP-FLOP:** For purpose of counting, the JK flip-flop is the ideal element to use. The variable J and K are called control I/Ps because they determine what the flip-flop does when a positive edge arrives. When J and K are both 0s, both AND gates are disabled and Q retains its last value.
- **D FLIP –FLOP:** This kind of flip flop prevents the value of D from reaching the Q output until clock pulses occur. When the clock is low, both AND gates are disabled D can change value without affecting the value of Q. On the other hand, when the clock is high, both AND gates are enabled. In this case, Q is forced to equal the value of D. When the clock again goes low, Q retains or stores the last value of D. a D flip flop is a bistable circuit whose D input is transferred to the output after a clock pulse is received.
- **T FLIP-FLOP:** The T or "toggle" flip-flop changes its output on each clock edge, giving an output which is half the frequency of the signal to the T input. It is useful for constructing binary counters, frequency dividers, and general binary addition devices. It can be made from a J-K flip-flop by tying both of its inputs high.

CIRCUIT DIAGRAM:

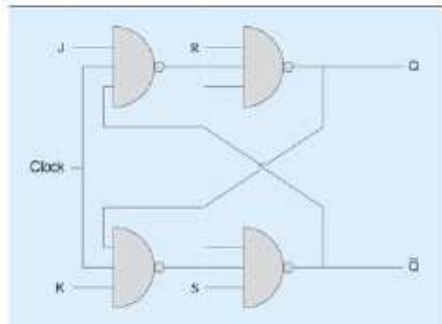
SR Flip Flop



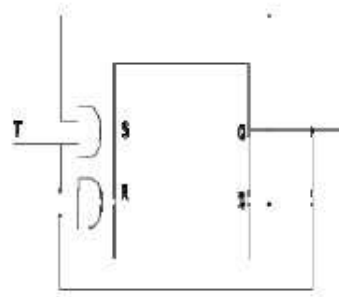
D Flip Flop



JK Flip Flop



T Flip Flop



PROCEDURE:

1. Connect the circuit as shown in figure.
2. Apply V_c & ground signal to every IC.
3. Observe the input & output according to the truth table.

& o

TRUTH TABLE:
SR FLIP FLOP:

CLOCK	S	R	Q _{n+1}
1	0	0	NO CHANGE
1	0	1	0
1	1	0	1
1	1	1	?

LIPFLOP:D FL

INPUT	OUTPUT
0	0
1	1

FLIPFLOPJK F

CLOCK	S	R	Q _{n+1}
1	0	0	NO CHANGE
1	0	1	0
1	1	0	1
1	1	1	Q _{nn'}

LIPFLOP T FL

CLOCK	S	R	Q _{n+1}
1	0	1	NO CHANGE
1	1	0	Q _{nn'}

RESULT: Truth table is verified on digital trainer.

PRECAUTIONS:

- 1) Make the connections according to the IC pin diagram.
- 2) The connections should be tight.
- 3) The V_{cc} and ground should be applied carefully at the specified pin only.

Quiz Questions with answer.

Q 1. Flip flop is Astable or Bistable?

Ans. Bistable.

Q2. What are the I/Ps of JK flip-flop where this race round condition occurs?

Ans. Both the inputs are 1.

Q3. When RS flip-flop is said to be in a SET state?

Ans. When the output is 1.

Q4. When RS flip-flop is said to be in a RESET state?

Ans. When the output is 0.

Q5. What is the truth table of JK flip-flop?

JK Q_{n+1}		
00 Q_n		
010		
101		
11 Q_n		

Q6. What is the function of clock signal in flip-flop?

Ans. To get the output at known time.

Q7. What is the advantage of JK flip-flop over RS flip-flop?

Ans. In RS flip-flop when both the inputs are 1 output is undetermined.

Q8. In D flip-flop I/P = 0 what is O/P?

Ans. 0

Q9. In D flip-flop I/P = 1 what is O/P?

Ans. 1

Q10. In T flip-flop I/P = 1 what is O/P?

Ans. Q_n

EXPERIMENT NO: 4

Aim:- Implementation and Verification of Decoder/De-Multiplexer and Encoder using Logic Gates.

APPARATUS REQUIRED: IC 7447, 7-segment display, IC 74139 and connecting leads.

BRIEF THEORY:

ENCODER: An encoder is a device, circuit, transducer, software program, algorithm or person that converts information from one format or code to another, for the purposes of standardization, speed, secrecy, security, or saving space by shrinking size. An encoder has M input and N output lines. Out of M input lines only one is activated at a time and produces equivalent code on output N lines. If a device output code has fewer bits than the input code has, the device is usually called an encoder. For example Octal-to-Binary Encoder take 8 inputs and provides 3 outputs, thus doing the opposite of what the 3-to-8 decoder does. At any one time, only one input line has a value of 1. The figure below shows the truth table of an Octal-to-binary encoder.

For an 8-to-3 binary encoder with inputs I0-I7 the logic expressions of the outputs Y0-Y2 are:

$$Y0 = I1 + I3 + I5 + I7$$

$$Y1 = I2 + I3 + I6 + I7$$

$$Y2 = I4 + I5 + I6 + I7$$

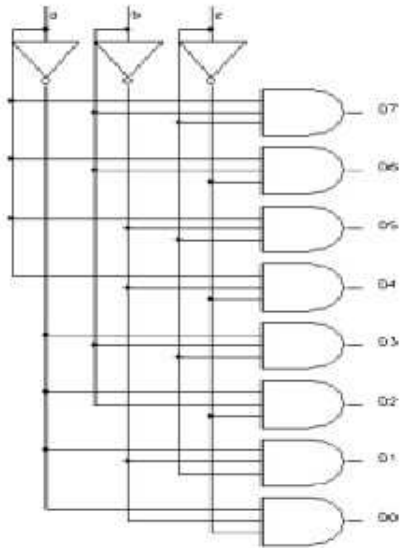
DECODER: A decoder is a device which does the reverse operation of an encoder, undoing the encoding so that the original information can be retrieved. The same method used to encode is usually just reversed in order to decode. It is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines. In digital electronics, a decoder can take the form of a multiple-input, multiple-output logic circuit that converts coded inputs into coded outputs, where the input and output codes are different. e.g. n-to- 2^n , binary-coded decimal decoders. Enable inputs must be on for the decoder to function, otherwise its outputs assume a single "disabled" output code word. In case of decoding all combinations of three bits eight ($2^3=8$) decoding gates are required. This type of decoder is called 3-8 decoder because 3 inputs and 8 outputs. For any input combination decoder outputs are 1.

DEMULTIPLEXER: Demultiplexer means generally one into many. A demultiplexer is a logic circuit with one input and many outputs. By applying control signals, we can steer the input signal to one of the output lines. The ckt. has one input signal, m control

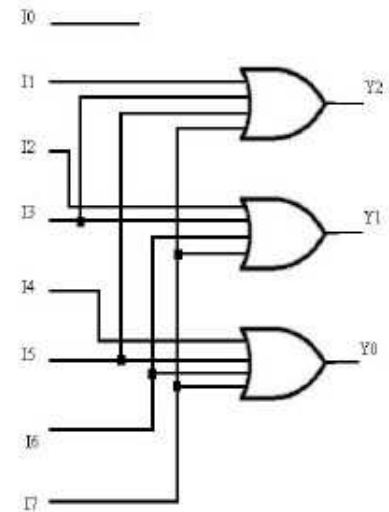
signal and n output signals. Where $2^n = m$. It functions as an electronic switch to route an incoming data signal to one of several outputs.

LOGIC DIAGRAM:

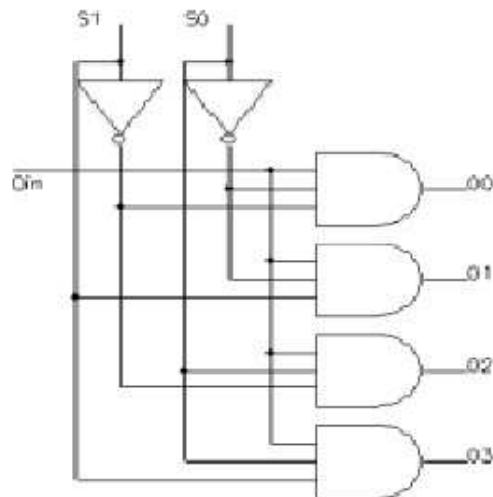
3:8 Decoder



Octal to Binary Encoder



1:4 Demux



PROCEDURE:

- 1) Connect the circuit as shown in figure.
- 2) Apply V_{cc} & ground signal to every IC.
- 3) Observe the input & output according to the truth table.

OBSERVATION TABLE:

Truth table for Decoder

Inputs			Outputs							
a	b	c	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1
Output function			$\bar{a}\bar{b}\bar{c}$	$\bar{a}\bar{b}c$	$\bar{a}b\bar{c}$	$\bar{a}bc$	$a\bar{b}\bar{c}$	$a\bar{b}c$	$ab\bar{c}$	abc

Truth table for Encoder

I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	Y ₂	Y ₁	Y ₀
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Truth table for Demux

Output select Lines		Output selected
S ₁	S ₀	
0	0	O ₀
0	1	O ₁
1	0	O ₂
1	1	O ₃

RESULT: Encoder/ decoder and demultiplexer have been studied and verified.

PRECAUTIONS:

- 1) Make the connections according to the IC pin diagram.
- 2) The connections should be tight.
- 3) The V_{cc} and ground should be applied carefully at the specified pin only.

Quiz Questions with answer.

Q. 1 What do you understand by decoder?

Ans. A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2ⁿ unique output lines. Most IC decoders include one or more enable inputs to control the circuit operation.

Q. 2 What is demultiplexer?

Ans. The demultiplexer is the inverse of the multiplexer, in that it takes a single data input and n address inputs. It has 2ⁿ outputs. The address input determine which data

output is going to have the same value as the data input. The other data outputs will have the value 0.

Q. 3 What do you understand by encoder?

Ans. An encoder or multiplexer is therefore a digital IC that outputs a digital code based on which of its several digital inputs is enabled.

Q. 4 What is the main difference between decoder and demultiplexer?

Ans. In decoder we have n input lines as in demultiplexer we have n select lines.

Q. 5 Why Binary is different from Gray code?

Ans. Gray code has a unique property that any two adjacent gray codes differ by only a single bit.

Q. 6 Write down the method of Binary to Gray conversion.

Ans. Using the Ex-Or gates.

Q. 7 Convert 0101 to Decimal.

Ans. 5

Q. 8 Write the full form of ASCII Codes?

Ans. American Standard Code for Information Interchange.

Q.9. If a register containing 0.110011 is logically added to register containing 0.101010 what would be the result?

Ans.111011

Q10.Binary code is a weighted code or not?

Ans. Yes

EXPERIMENT NO : 5

Aim: Implementation of 4x1 Multiplexer using Logic Gates.

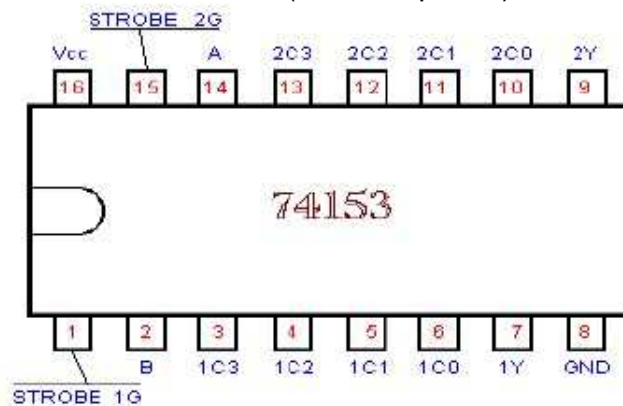
APPARATUS REQUIRED: Power Supply, Digital Trainer, Connecting Leads, IC's 74153(4x1 multiplexer).

BRIEF THEORY:

MULTIPLEXER: Multiplexer generally means many into one. A multiplexer is a circuit with many Inputs but only one output. By applying control signals we can steer any input to the output. The fig. (1) Shows the general idea. The ckt. has n-input signal, control signal & one output signal. Where $2^n = m$. One of the popular multiplexer is the 16 to 1 multiplexer, which has 16 input bits, 4 control bits & 1 output bit.

PIN CONFIGURATION;—

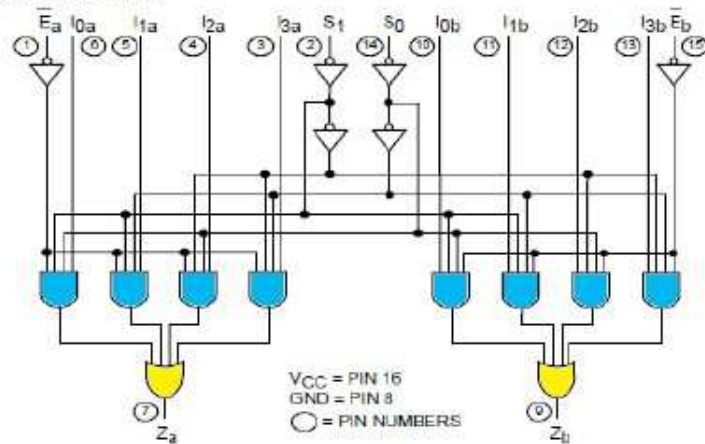
IC 74153 (4x1 multiplexer)



LOGIC DIAGRAM:

Multiplexer (4x1) IC 74153

LOGIC DIAGRAM



PROCEDURE:

1. Fix the IC's on the bread board & give the input supply.
2. Make connection according to the circuit.
3. Give select signal and strobe signal at respective pins.
4. Connect +5 v Vcc supply at pin no 24 & GND at pin no 12.
5. Verify the truth table for various inputs.

OBSERVATION TABLE:

Truth Table of multiplexer (4x1) IC 74153

INPUT							OUTPUT
A	B	C0	C1	C2	C3	G	Y
X	X	X	X	X	X	1	0
0	0	0	X	X	X	0	0
0	0	1	X	X	X	0	1
0	1	X	0	X	X	0	0
0	1	X	1	X	X	0	1
1	0	X	X	0	X	0	0
1	0	X	X	1	X	0	1
1	1	X	X	X	0	0	0
1	1	X	X	X	1	0	1

RESULT: Verify the truth table of multiplexer for various inputs.

PRECAUTIONS:

- 1) Make the connections according to the IC pin diagram.

- 2) The connections should be tight.
- 3) The V_{cc} and ground should be applied carefully at the specified pin only.

Quiz Questions with answer.

Q.1 Why is MUX called as "Data Selector"?

Ans. This selects one out of many inputs.

Q.2 What do you mean by Multiplexing?

Ans. Multiplexing means selecting only a single input out of many inputs.

Q.3 What is Digital Multiplexer?

Ans. The multiplexer which acts on digital data.

Q.4 What is the function of Enable input to any IC?

Ans. When this enable signal is activated.

Q.5 What is demultiplexer?

Ans. A demultiplexer transmits the data from a single source to various sources.

Q.6 Can a decoder function as a D'MUX?

Ans. Yes

Q.7 What is the role of select lines in a Demultiplexer?

Ans. Select line selects the output line.

Q.8 Differentiate between functions of MUX & D'MUX?

Ans. Multiplexer has only single output but demultiplexer has many outputs.

Q.9 The number of control lines required for a 1:8 demultiplexer will be

Ans. 3

Q.10 How many 4:1 multiplexers will be required to design 8:1 multiplexer?

Ans. 2

EXPERIMENT NO - 6

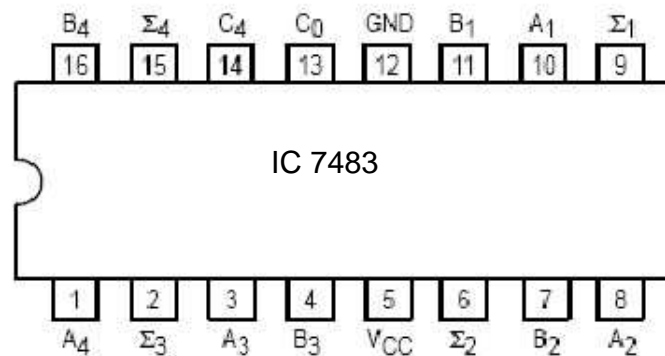
Aim – Implementation of 4-Bit Parallel Adder Using 7483 Ic.

APPRATUS REQUIRED – Digital trainer kit, IC 7483 (4-bit parallel adder).

BRIEF THEOR - A 4-bit adder is a circuit which adds two 4-bits numbers, say, A and B. In addition, a 4-bit adder will have another single-bit input which is added to the two numbers called the carry-in (C_{in}). The output of the 4-bit adder is a 4-bit sum (S) and a carry-out (C_{out}) bit.

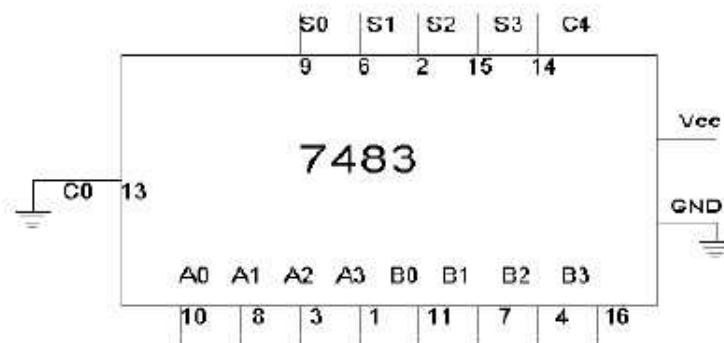
PIN CONFIGURATION–

Pin Diagram of IC 7483



LOGIC DIAGRAM:-

7483 4-bit Parallel Adder



OBSERVATION TABLE –

Truth table of 4-bit parallel adder

A3	A2	A1	A0	B3	B2	B1	B0	C4 (V)	S3(V)	S2(V)	S1(V)	S0(V)
0	0	0	1	0	0	1	0	0	0	0	1	1
0	1	0	1	1	0	1	1	1	1	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	0	0	1	1	0	1	0	1	0

PROCEDURE –

- Make the connections as per the logic diagram.
- Connect +5v and ground according to pin configuration.
- Apply diff combinations of inputs to the i/p terminals.
- Note o/p for summation.
- Verify the truth table.

RESULT- Binary 4-bit full adder is studied and verified.

PRECAUTIONS:

- Make the connections according to the IC pin diagram.
- The connections should be tight.
- The V_{cc} and ground should be applied carefully at the specified pin only.

Quiz Questions with answer.

Q 1 What do you understand by parallel adder?

Ans. If we place full adders in parallel, we can add two- or four-digit numbers or any other size desired i.e. known as parallel adder.

Q2 What happens when an N-bit adder adds two numbers whose sum is greater than or equal to 2^N

Ans. Overflow.

Q3 Is Excess-3 code is weighted code or not?

Ans. Excess-3 is not a weighted code.

Q4 What is IC no. of parallel adder?

Ans. IC 7483.

Q5 What is the difference between Excess-3 & Natural BCD code?

Ans. Natural BCD code is weighted code but Excess-3 code is not weighted code.

Q6. What is the Excess-3 code for $(396)_{10}$

Ans. $(396)_{10} = (011011001001)_{EX-3}$

Q7 Can we obtain 1's complement using parallel adder?

Ans. Yes

Q8 Can we obtain 2's complement using parallel adder?

Ans. yes

Q9 How many bits can be added using IC7483 parallel adder?

Ans. 4 bits.

Q10 Can you obtain subtractor using parallel adder?

Ans. Yes

EXPERIMENT NO. 7

Aim – Design and Verify the 4-Bit Synchronous Counter

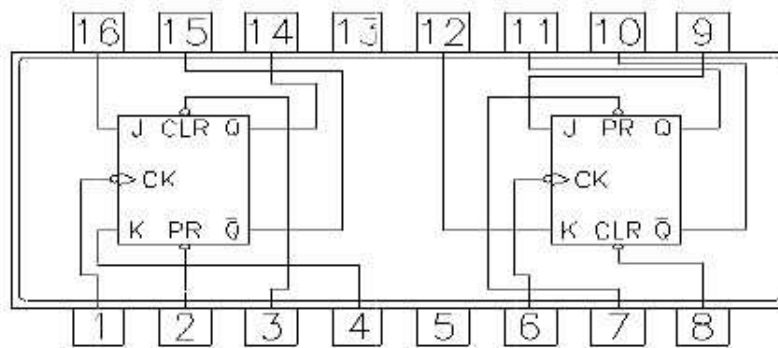
APPARATUS REQUIRED: Digital Trainer kit and 4 JK flip flop each IC 7476 (i.e. dual in-line package), two AND gates IC 7408.

BRIEF THEORY: Counter is a circuit which cycles through a sequence of states. Two types of counters are: Synchronous counter (e.g. parallel) and Asynchronous counter (e.g. ripple). In a synchronous counter, the flip-flop output is used as a clock signal for other flip-flops. In a ripple counter, the flip-flop output is used as a clock signal for other flip-flops.

Synchronous counter uses the same clock signal for all flip-flops.

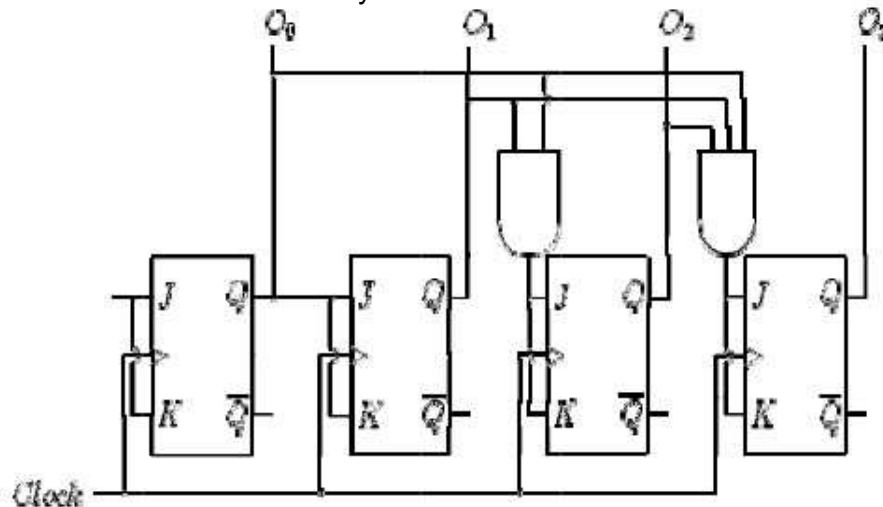
FUNCTION: PIN CONFIGURATION

Dual JK Master-Slave Flip-Flop with clear & preset



LOGIC DIAGRAM

4-Bit Synchronous counter



Pin Number	Description
1	Clock 1 Input
2	Preset 1 Input
3	Clear 1 Input
4	J1 Input
5	Vcc
6	Clock 2 Input
7	Preset 2 Input
8	Clear 2 Input
9	J2 Input
10	Complement Q2 Output
11	Q2 Output
12	K2 Input
13	Ground
14	Complement Q1 Output
15	Q1 Output
16	K1 Input

OBSERVATION TABLE:

Truth Table

States				Count
O ₄	O ₃	O ₂	O ₁	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6

0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

PROCEDURE:

- Make the connections as per the logic diagram.
- Connect +5v and ground according to pin configuration.
- Apply diff combinations of inputs to the i/p terminals.
- Note o/p for summation.
- Verify the truth table.

RESULT: 4-bit synchronous counter studied and verified.

PRECAUTIONS:

- Make the connections according to the IC pin diagram.
- The connections should be tight.
- The V_{cc} and ground should be applied carefully at the specified pin only.

Quiz Questions with answer.

Q.1 What do you understand by counter?

Ans. Counter is a register which counts the sequence in binary form.

Q.2 What is asynchronous counter?

Ans. Clock input is applied to LSB FF. The output of first FF is connected as clock to next FF.

Q.3 What is synchronous counter?

Ans. Where Clock input is common to all FF.

Q.4 Which flip flop is used in asynchronous counter?

Ans. All Flip-Flops are toggling FF.

Q.5 Which flip flop is used in synchronous counter?

Ans. Any FF can be used.

Q.6 What do you understand by modulus?

Ans. The total no. of states in counter is called as modulus. If counter is modulus-n, then it has n different states.

Q.7 What do you understand by state diagram?

Ans. State diagram of counter is a pictorial representation of counter states directed by arrows in graph.

Q.8 What do you understand by up/down counter?

Ans. Up/Down Synchronous Counter: two way counter which able to count up or down.

Q.9 Why Asynchronous counter is known as ripple counter?

Ans. Asynchronous Counter: flip-flop doesn't change condition simultaneously because it doesn't use single clock signal Also known as ripple counter because clock signal input as ripple through counter.

Q.10 which type of counter is used in traffic signal?

Ans. Down counters.

EXPERIMENT NO: 8

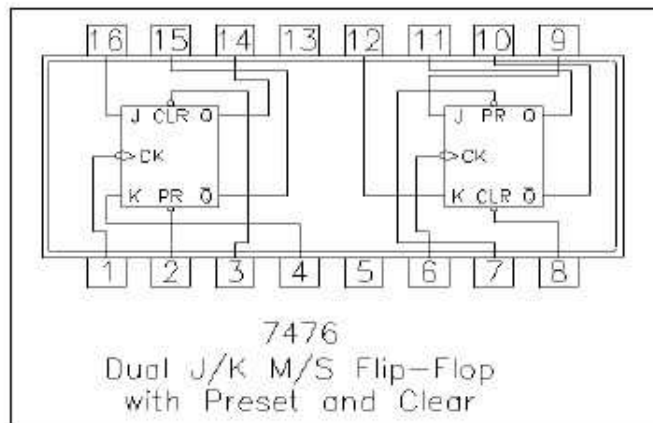
Aim: – Design, and Verify the 4-Bit Asynchronous Counter.

APPARATUS REQUIRED: Digital trainer kit and 4 JK flip flop each IC 7476 (i.e dual JK flip flop) and two AND gates IC 7408.

BRIEF THEORY: Counter is a circuit which cycle through state sequence. Two types of counter, Synchronous counter (e.g. parallel) and Asynchronous counter (e.g. ripple). In Ripple counter same flip-flop output to be used as clock signal source for other flip-flop. Synchronous counter use the same clock signal for all flip-flop.

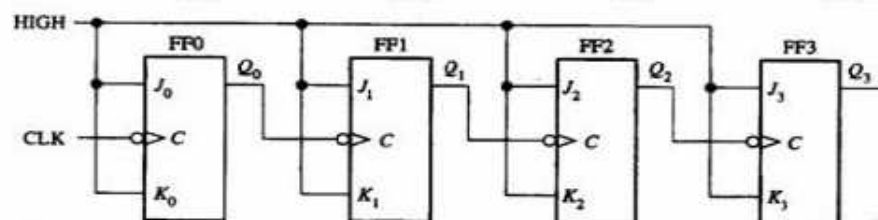
PIN CONFIGURATION:

Pin diagram of JK M/S Flip Flop



LOGIC DIAGRAM:

4-Bit Asynchronous counter



Pin Number	Description
1	Clock 1 Input
2	Preset 1 Input
3	Clear 1 Input
4	J1 Input
5	Vcc
6	Clock 2 Input
7	Preset 2 Input
8	Clear 2 Input
9	J2 Input
10	Complement Q2 Output
11	Q2 Output
12	K2 Input
13	Ground
14	Complement Q1 Output
15	Q1 Output
16	K1 Input

PROCEDURE:

- Make the connections as per the logic diagram.
- Connect +5v and ground according to pin configuration.
- Apply diff combinations of inputs to the i/p terminals.
- Note o/p for summation.
- Verify the truth table.

RESULT: 4-bit asynchronous counter studied and verified.

PRECAUTIONS:

- Make the connections according to the IC pin diagram.
- The connections should be tight.
- The V_{cc} and ground should be applied carefully at the specified pin only.

Quiz Questions with answer.

Q.1 How many flip-flops are required to make a MOD-32 binary counter?

Ans. 5.

Q.2 The terminal count of a modulus-11 binary counter is _____.

Ans.1010.

Q.3 Synchronous counters eliminate the delay problems encountered with asynchronous counters because the:

Ans. Input clock pulses are applied simultaneously to each stage.

Q4. Synchronous construction reduces the delay time of a counter to the delay of:

Ans. a single flip-flop and a gate.

Q5. What is the difference between a 7490 and a 7492?

Ans. 7490 is a MOD-10, 7492 is a MOD-12.

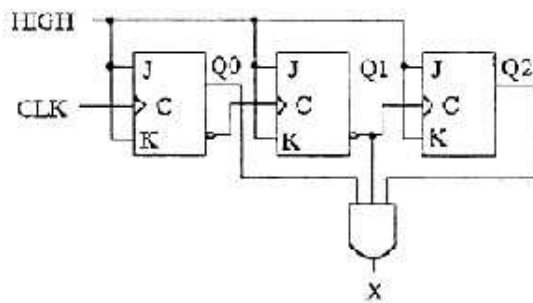
Q6. When two counters are cascaded, the overall MOD number is equal to the _____ of their individual MOD numbers.

Ans. Product.

Q7. A BCD counter is a _____.

Ans. decade counter.

Q8. What decimal value is required to produce an output at "X" ?



Ans. 5.

Q9. How many AND gates would be required to completely decode ALL the states of a MOD-64 counter, and how many inputs must each AND gate have?

Ans. 64 gates, 6 inputs to each gate.

Q.10 A ring counter consisting of five Flip-Flops will have

Ans. 5 states.

EXPERIMENT NO: 9

Aim:- To Design & Verify Operation of Half Adder & Full Adder.

APPARATUS REQUIRED: Power supply, IC's, Digital Trainer, Connecting leads.

BRIEF THEORY: We are familiar with ALU, which performs all arithmetic and logic operation but ALU doesn't perform/ process decimal no's. They process binary no's.

Half Adder: It is a logic circuit that adds two bits. It produces the O/P, sum & carry.
The Boolean equation for sum & carry are:

$$\text{SUM} = A + B$$

$$\text{CARRY} = A \cdot B$$

Therefore, sum produces 1 when A&B are different and carry is 1 when A&B are 1.
Application of Half adder is limited.

Full Adder: It is a logic circuit that can add three bits. It produces two O/P sum & carry.
The Boolean Equation for sum & carry are:

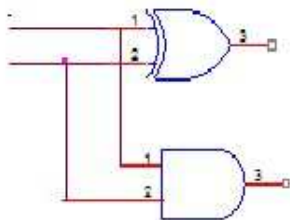
$$\text{SUM} = A + B + C$$

$$\text{CARRY} = A \cdot B + (A+B) \cdot C$$

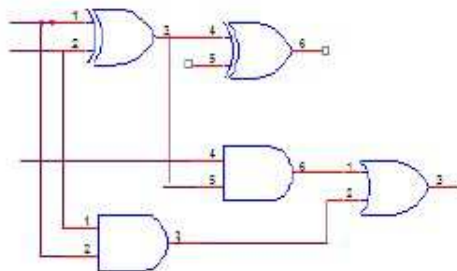
Therefore, sum produces one when I/P is containing odd no's of one & carry is one when there are two or more one in I/P.

LOGIC DAIGRAM:

Half Adder



Full Adder



PROCEDURE:

- Connect the ckt. as shown in fig. For half adder.
- Apply diff. Combination of inputs to the I/P terminal.
- Note O/P for Half adder.
- Repeat procedure for Full wave.
- The result should be in accordance with truth table.

OBSERVATION TABLE:

HALF ADDER:

INPUTS		OUTPUT	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

FULL ADDER:

INPUTS			OUTPUTS	
A	B	C	S	CARRY
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

RESULT: The Half Adder & Full Adder ckts. are verified.

PRECAUTIONS:

- 1) Make the connections according to the IC pin diagram.
- 2) The connections should be tight.
- 3) The V_{cc} and ground should be applied carefully at the specified pin only.

Quiz Questions with answer.

Q.1 Give the basic rules for binary addition?

Ans. $0+0 = 0$; $0+1 = 1$; $1+1 = 1\ 0$; $1+0 = 1$.

Q.2 Specify the no. of I/P and O/P of Half adder?

Ans2. Two inputs & one output.

Q.3 What is the drawback of half adder?

Ans. We can't add carry bit from previous stage.

Q.4 Write the equation for sum & carry of half adder?

Ans. Sum = A XOR B; carry = A.B.

Q.5 Write the equation for sum & carry of full adder?

Ans. SUM = $A'B'C + A'BC' + AB'C' + ABC$; CARRY = $AB + BC + AC$.

Q.6 How many half adders will be required for Implementing full adder?

Ans. Two half adders and a OR gate.

Q7 Define Bit?

Ans. Bit is an abbreviation for binary digit.

Q8.What is the difference b/w half adder& half sub tractor?

Ans. Half adder can add two bits & half sub tractor can subtract two bits.

Q9. Half subtractor logic circuit has one extra logic element. Name the element?

Ans. Inverter.

Q10. Define Nibble?

Ans. Combination of four bits.

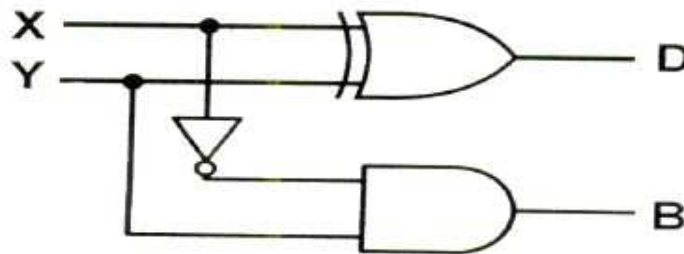
EXPERIMENT NO :10

Aim:- To Study & Verify Half Subtractor.

APPARATUS REQUIRED: Digital trainer kit,
IC 7486 (EX-OR)
IC 7408 (AND gate)
IC 7404 (NOT gate)

BRIEF THEORY: A logic circuit for the subtraction of B(subtrahend) from A (minuend) where A & B are 1 bit numbers is referred as half- sub tractor.

LOGIC DIAGRAM :



TRUTH TABLE:

INPUT 1 (X)	INPUT 2 (Y)	BORROW (B)	DIFFERENCE (D)
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

PROCEDURE:

1. Make the connections as per the logic diagram.
2. Connect +5v to pin 14 & ground to pin 7.
3. Apply 0 to input X & Y as per the truth table.
4. Switch on the instrument.
5. Observe the reading on 8 bits LED display.
6. Repeat steps 3 & 5 for different input as per truth table.
7. Verify the truth table.

RESULT: Half sub tractor circuit is studied and verified.

Quiz Questions with answer.

Q.1 What is half subtractor?

Ans. Performs subtraction of two bits.

Q.2 For implementing half subtractor how many EX-OR, AND gates and Not gates are required?

Ans. One EX-OR, one –AND gate, one- Not gate.

Q.3 What are the logical equations for difference & borrow?

Ans. $D = \bar{A}B + A\bar{B}$

$B = \bar{A}.B$

Q.4 How full subtractor is different from half subtractor.

Ans. Full subtractor performs subtraction of three bits but half subtractor Performs subtraction of two bits.

Q.5 If inputs of half subtractor are $A=0$, and $B=1$ then Borrow will be?

Ans. $B=1$

Q.6 Is 2's complement method appropriate for subtraction?

Ans. 2's complement method is appropriate method for subtraction.

Q.7 How many bits we use in half subtractor for subtraction?

Ans. only two bits.

Q.8 Can we use parallel adder for subtraction?

Ans. We can use parallel adder using 2's complement method.

Q.9 Which one is better subtractor or parallel adder for subtraction?

Ans. Parallel adder is the best option using 1's complement or 2's complement

Q.10 Which adder is used for addition of BCD numbers?

Ans. BCD adder.

EXPERIMENT NO: 11

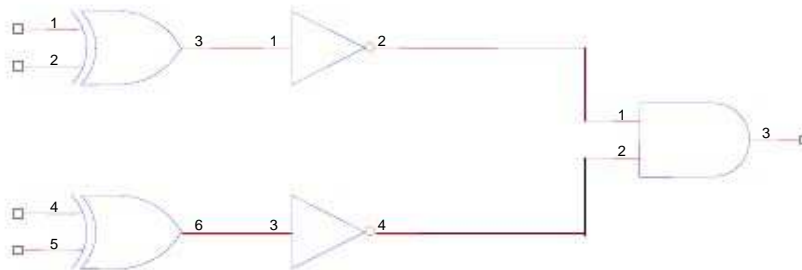
Aim: - To Design & Verify the Operation of Magnitude Comparator

APPARATUS REQUIRED: Power Supply, Digital Trainer Kit., Connecting Leads, and IC's (7404, 7408, and 7486).

BRIEF THEORY: Comparator compares the value of signal at the input. It can be designed to compare many bits. The adjoining figure shows the block diagram of comparator. Here it receives two 2-bit numbers at the input & the comparison is at the output.

CIRCUIT DIAGRAM:

Comparator



PROCEDURE:

- Make the connections according to the circuit diagram.
- The output is high if both the inputs are equal.
- Verify the truth table for different values.

OBSERVATION TABLE:

P0	Q0	P1	Q1	LOW IF P IS NOT EQUAL TO Q	HIGH IF Q IS EQUAL TO Q
0	0	0	0		HIGH
1	1	0	0		HIGH
0	1	0	1	LOW	
1	0	1	0	LOW	

RESULT: The comparator is designed & verified.

PRECAUTIONS:

- 1) Make the connections according to the IC pin diagram.
- 2) The connections should be tight.
- 3) The V_{cc} and ground should be applied carefully at the specified pin only.

Quiz Questions with answer.

Q1.What is comparator?

Ans. Comparator compares the inputs (bits).

Q2. What are universal gates?

Ans. NAND, NOR.

Q3. What is the full form of BCD?

Ans. Binary Coded decimal.

Q4. What is the base of binary number system?

Ans. 2

Q5How many bits are there in one byte?

Ans. 8

Q6. How many digits are there in octal number system?

Ans. 8

Q7. What is the binary no. equivalent to decimal no. 20?

Ans. 10100

Q8. How decimal no. minus 7 can be represented by 4 bit signed binary no's?

Ans. 1111

Q9.Convert the octal no 67 into binary no.?

Ans. 110111

Q10.A binary digit is called?

Ans. Bit.

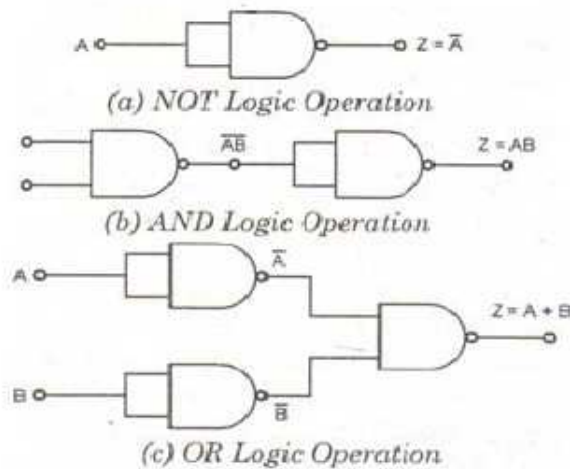
EXPERIMENT NO:12

Aim: - To Study and Verify NAND as a Universal Gate.

APPARATUS REQUIRED: Digital trainer kit, IC 7400 (NAND gate)

BRIEF THEORY: NAND OR NOR sufficient for the realization of any logic expression. because of this reason, NAND and NOR gates are known as UNIVERSAL gates.

LOGIC DIAGRAM:



TRUTH TABLE:

NAND GATE AS INVERTER: The circuit diagram of implementation of NAND gate as inverter.

A	Y
0	1
1	0

NAND GATE AS AND GATE:

The circuit diagram of implementation of NAND Gate as AND Gate.

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

NAND GATE AS OR GATE:

The circuit diagram of implementation of NAND Gate as OR Gate.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

PROCEDURE:

1. Make the connections as per the logic diagram.
2. Connect +5v to pin 14 & ground to pin 7.
3. Apply diff combinations of inputs to the i/p terminals.
4. Note o/p for NAND as universal gate.
5. Verify the truth table.

Quiz Questions with answer.

Q.1 Define Gates.

Ans. Gates are digital circuit, which perform a specific type of logical operation.

Q.2 Define IC?

Ans. IC means Integrated Circuit It is the integration of no. of components on a common substrate.

Q.3 $(A+A)$ A=?

Ans. A.

Q.4. Define universal gates

Ans. We can design any type of logical expression by using universal gates.

Q.5 Will the output of a NAND Gate be 0.

Ans. When all the inputs are 1.

Q.6 Which IC is used for NAND GATE?

Ans. IC 7400.

Q.7 Why NAND is called as universal gate?

Ans. Because all gates can be made using circuits.

Q.8 Name any other universal gate?

Ans. NOR Gate.

Q.9 Which type of TTL gates can drive CMOS Gate?

Ans. TTL with open collector can drive CMOS.

Q.10 What is meant by literal?

Ans. A logical variable in a complemented or Un-complemented form is called a literal.

EXPERIMENT NO:13

AIM: - To Study 4 Bit ALU(IC 74181).

APPARATUS REQUIRED: IC 74181, etc.

BRIEF THEORY:

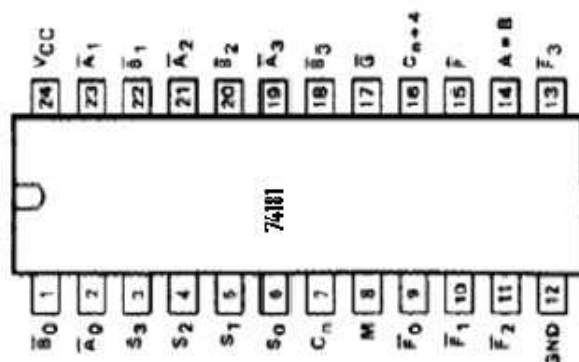
The 74181 is a 7400 series medium-scale integration (MSI) TTL integrated circuit, containing the equivalent of 75 logic gates and most commonly packaged as a 24-pin DIP. The 4-bit wide ALU can perform all the traditional add / subtract / decrement operations with or without carry, as well as AND / NAND, OR / NOR, XOR, and shift. Many variations of these basic functions are available, for a total of 16 arithmetic and 16 logical operations on two four-bit words. Multiply and divide functions are not provided but can be performed in multiple steps using the shift and add or subtract functions. Shift is not an explicit function but can be derived from several available functions including (A+B) plus A.

PIN DETAIL & FUNCTION TABLE:

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 22, 20, 18	\bar{B}_0 to \bar{B}_3	operand inputs (active LOW)
2, 23, 21, 19	\bar{A}_0 to \bar{A}_3	operand inputs (active LOW)
6, 5, 4, 3	S_0 to S_3	select inputs
7	C_n	carry input
8	M	mode control input
9, 10, 11, 13	F_0 to F_3	function outputs (active LOW)
12	GND	ground (0 V)
14	A=B	comparator output
15	\bar{P}	carry propagate output (active LOW)
16	C_{n+4}	carry output
17	\bar{G}	carry generate output (active LOW)
24	V_{CC}	positive supply voltage

IC 74181



FUNCTION TABLES

MODE SELECT INPUTS				ACTIVE HIGH INPUTS AND OUTPUTS	
S ₃	S ₂	S ₁	S ₀	LOGIC (M=H)	ARITHMETIC ⁽²⁾ (M=L; C _n =H)
L	L	L	L	\overline{A}	A
L	L	L	H	$\overline{A+B}$	A + B
L	L	H	L	\overline{AB}	A + \overline{B}
L	L	H	H	logical 0	minus 1
L	H	L	L	\overline{AB}	A plus \overline{AB}
L	H	L	H	\overline{B}	(A + B) plus \overline{AB}
L	H	H	L	$A \oplus B$	A minus B minus 1
L	H	H	H	\overline{AB}	\overline{AB} minus 1
H	L	L	L	$\overline{A+B}$	A plus AB
H	L	L	H	$\overline{A \oplus B}$	A plus B
H	L	H	L	B	(A + \overline{B}) plus AB
H	L	H	H	AB	AB minus 1
H	H	L	L	logical 1	A plus A ⁽¹⁾
H	H	L	H	$A + \overline{B}$	(A + B) plus A
H	H	H	L	A + B	(A + \overline{B}) plus A
H	H	H	H	A	A minus 1

Notes to the function tables

- Each bit is shifted to the next more significant position.
- Arithmetic operations expressed in 2s complement notation.

H = HIGH voltage level

L = LOW voltage level

MODE SELECT INPUTS				ACTIVE LOW INPUTS AND OUTPUTS	
S ₃	S ₂	S ₁	S ₀	LOGIC (M=H)	ARITHMETIC ⁽²⁾ (M=L; C _n =L)
L	L	L	L	\overline{A}	A minus 1
L	L	L	H	\overline{AB}	AB minus 1
L	L	H	L	$\overline{A+B}$	\overline{AB} minus 1
L	L	H	H	logical 1	minus 1
L	H	L	L	$\overline{A+B}$	A plus (A + \overline{B})
L	H	L	H	\overline{B}	AB plus (A + B)
L	H	H	L	$\overline{A \oplus B}$	A minus B minus 1
L	H	H	H	A + B	A + B
H	L	L	L	\overline{AB}	A plus (A + B)
H	L	L	H	$A \oplus B$	A plus B
H	L	H	L	B	\overline{AB} plus (A + B)
H	L	H	H	A + B	A + B
H	H	L	L	logical 0	A plus A ⁽¹⁾
H	H	L	H	AB	AB plus A
H	H	H	L	AB	\overline{AB} plus A
H	H	H	H	A	A

Notes to the function tables

- Each bit is shifted to the next more significant position.
- Arithmetic operations expressed in 2s complement notation.

H = HIGH voltage level

L = LOW voltage level

PROCEDURE:

- Connections are made as shown in the Circuit diagram.
- Change the values of the inputs and verify at least 5 functions given in the function table

PRECAUTIONS:

- Make the connections according to the IC pin diagram.
- The connections should be tight.
- The V_{cc} and ground should be applied carefully at the specified pin only.

Quiz Questions with answer

Q.1 What is the difference between a mnemonic code and machine code?

Ans. Machine codes are in binary, mnemonic codes are in shorthand English.

Q.2 Which bus is a bidirectional bus?

Ans. Data Bus.

Q. Which of the following buses is primarily used to carry signals that direct other 3 ICs to find out what type of operation is being performed?

Ans. control bus.

Q.4 Which of the following are the three basic sections of a microprocessor unit?

Ans. control and timing, register, and arithmetic/logic unit (ALU).

Q.5 The 8085A is a(n):

Ans. A 8-bit 8-bit parallel CPU.

Q.6 A register in the microprocessor that keeps track of the answer or results of any arithmetic or logic operation is the:

Ans. Accumulator.

Q.7 When was the first 8-bit microprocessor introduced?

Ans. 1971

Q.8 What type of circuit is used at the interface point of an output port?

Ans. Latch.

Q.9 What type of circuit is used at the interface point of an input port?

Ans. Tri-state buffer.

Q.10 The register in the 8085A that is used to keep track of the memory address of the next op-code to be run in the program is the:

Ans. program counter.

EXPERIMENT NO:14

LIST OF MINI PROJECTS

1. IR Remote Switch
2. Clap Switch
3. Water-Level Controller
4. LED-Based Message Display
5. Ultra-Bright LED Lamp
6. Ding-Dong Bell Infrared Cordless Headphone
7. Mobile Phone Battery Charger
8. Telephone Number Display
9. Automatic Night Lamp With Morning Alarm
10. Three-Colour Display Using Bi-Colour LEDs
11. Remote-Operated Musical Bell
12. Simple Telephone Ring Tone Generator
13. Anti-Theft Alarm For Bikes
14. Automatic Speed-Controller For Fans and Coolers
15. Digital Stop Watch
16. Power-Supply Failure Alarm
17. Dark Room Timer
18. Remote-Controlled Power-Off Switch
19. Simple Low-Cost Digital Code Lock
20. Number Guessing Game
21. Fire Alarm Using Thermistor
22. Simple Analogue To Digital Converter
23. PC-Based 7-Segment Rolling Display
24. IR Burglar Deterrent
25. Variable Power Supply Using a Fixed-Voltage Regulator IC
26. Digital Speedometer
27. Heat-Sensitive Switch
28. Fully Automatic Emergency Light
29. Running Message Display
30. School/College Quiz Buzzer
31. Digital Dice With Numeric Display
32. Dancing Lights
33. Ready -To-Use Object Counter Laptop Protector
34. PC Based Digital Clock
35. Fancy Christmas Light

AN EXAMPLE

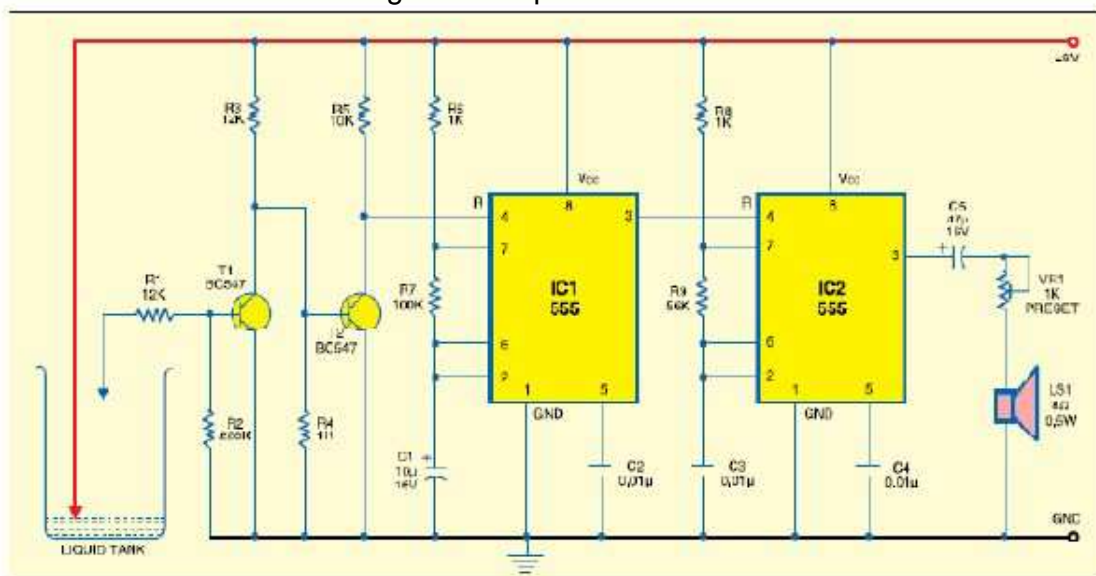
AIM: - Liquid Level Alarm

APPARATUS REQUIRED: Components as shown in the circuit diagram (such as 555IC,, soldering iron and solder flux and PCB board .

BRIEF THEORY: Here is a simple circuit for (T1 and T2) and two timers 555 ICs (IC1 and IC2). Both IC1 and IC2 are wired in astable multivibrator mode. Timer IC1 produces low frequency, while timer IC2 produces high frequency. As a result, a beeping tone is generated when the liquid tank is full. Initially, when the tank is empty, transistor T1 does not conduct. Consequently, transistor T2 conducts and pin 4 of IC1 is low. This low voltage disables IC1 and it does not oscillate. The low output of IC1 disables IC2 and it does not oscillate. As a result, no sound is heard from the speaker. But when the tank gets filled up, transistor T1 conducts. Consequently, transistor T2 is cut off and pin 4 of IC1 becomes high. This high voltage enables IC1 and it oscillates to produce low frequencies at pin 3. This low-frequency output enables IC2 and it also oscillates to produce high frequencies. As a result, sound is produced from the speaker. Using preset VR1 you can control the volume of the sound from the speaker. The circuit can be powered from a 9V battery or from mains by using a 9V power adaptor.

CIRCUIT DIAGRAM:

Circuit Diagram of Liquid Level Alarm



PROCEDURE: Assemble the circuit on a general purpose PCB and enclose in a suitable cabinet. Install two water-level probes using metal strips such that one touches the bottom of the tank and the other touches the maximum level of the water in the tank. Interconnect the sensor and the circuit using a flexible wire.

PRECAUTIONS:

- 1) Make the connections according to the Circuit diagram using soldering iron
- 2) The connections should be tight.
- 3) The V_{cc} and ground should be applied carefully at the specified pin only