\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Title:Implementation of Algorithm for generating convolutional code using**

**a)Analytical method**

**b)code tree**

**c)Trellis code**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**a)Code for Analytical method:**

clc;clear all;close all;

k=input('Enter the length of msg vector');

m=input('Enter the msg vector');

g1=input('Enter the first generator');

g2=input('Enter the second generator');

a=conv(m,g1);

b=conv(m,g2);

a=mod(a,2);

b=mod(b,2);

c=size(a);%for rows and columns

d=zeros(1,c(2)\*2);%create all zero elements as c(2)=5

j=1;

for i=1:c(2);

d(j)=a(i);j=j+1;

d(j)=b(i);j=j+1;

end;

disp('o/p sequence');

disp(d);

**OUTPUT:**

Enter the length of msg vector3

Enter the msg vector[1 0 1]

Enter the first generator[1 0 1]

Enter the second generator[1 1 1]

o/p sequence: 1 1 0 1 0 0 0 1 1 1

**B)Code for code tree method:**

clc;clear all;close all;

d=0;

c=0;

e=0;

i=input('enter message');

i=[i,zeros(1,2)]

[m,n]=size(i);

for j=1:n

c=i(j);

a(j)=c+d+e;

b(j)=c+e;

e=d;

d=c;

w=rem(a(j),2);

if(w==1)

a(j)=1;

else

a(j)=0;

end

x=rem(b(j),2);

if(x==0)

b(j)=0;

else

b(j)=1;

end

end

g=zeros(1,2\*n)

k=1;

for z=1:n

g(k)=b(z);

k=k+1;

g(k)=a(z);

k=k+1;

end

disp('output seq.');

disp(g);

**OUTPUT:**

enter message[1 0 1]

i =

1 0 1 0 0

g =

0 0 0 0 0 0 0 0 0 0

output seq.

1 1 0 1 0 0 0 1 1 1

**a)Code for Trellis method:**

clc;clear all;close all;

n=input('Enter the no. of bits');

L=input('Enter the no. of bits shift register can store:');

m=input('Enterthe length of data code:');

M=input('Enter the no. of flip-flops');

g1=input('Enter the 1st polynomial:');

g2=input('Enter the 2nd polynomial:');

ms=input('Enter the msg vector');

k=M+1;%constraint length

l=n\*(L+k-1);%O/p length

ms=[ms,zeros(1,k-1)];

a1=num2str(g1);%convert number to string

b1=num2str(g2);

cd1=str2num(dec2base(bin2dec(a1),8));%to get number in decimal form for dc1

cd2=str2num(dec2base(bin2dec(b1),8));%to get number in decimal form for dc2

cd=[cd1,cd2];%to store both decimal values of cd1,cd2 in cd

trel=poly2trellis(k,cd);%to get trellis code

code=convenc(ms,trel);

disp(code);

**OUTPUT:**

Enter the no. of bits2

Enter the no. of bits shift register can store:2

Enterthe length of data code:3

Enter the no. of flip-flops2

Enter the 1st polynomial:[1 0 1]

Enter the 2nd polynomial:[1 1 1]

Enter the msg vector[1 0 1]

1 1 0 1 0 0 0 1 1 1