Digital Signal Processing Lab

Laboratory report submitted for the partial fulfillment
of the requirements for the degree of

Bachelor of Technology

in

Electronics and Communication Engineering

by

Name: Riya Bora   Roll no: 22ucc087

Course Coordinator

Dr. Navneet Upadhyay

Department of Electronics and Communication Engineering

The LNM Institute of Information Technology, Jaipur

8 September 2024

Contents

Chapter *1*

Experiment -8

1.1 Aim

a) Radix-2 FFT Algorithm.

1.2 Software Used

MATLAB

1.3 Theory

The N-point Discrete fourier transform(DFT) calculation generally requires N^2 complex multiplications.On the other hand Radix-2 FFT algorithm: Desimation in time(DIT) or Decimation in frequency(DIF) are computationally efficient and requires only $\frac{N}{2}log_2N$ complex multiplications.
The DFT of x[n] is given by

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}$$

where k=0,1,2,..... $\frac{N}{2} - 1$ and W_N= exp $\frac{-j2\pi}{N}$

The N length sequence can be divided into two N/2 point data sequence $f_1[n]$ and $f_2[n]$, corresponding to the even numbered and odd numbered samples of x[n].

$$X(k) = \sum_{n=even} x(n)W_N^{kn} + \sum_{n=odd} x(n)W_N^{kn}$$

$$X(k) = \sum_{m=o}^{\frac{N}{2}-1} x(2m)W_N^{2mk} + \sum_{m=0}^{\frac{N}{2}-1} x(2m+1)W_N^{(2m+1)k}$$

$$X(k)=\sum_{m=o}^{\frac{N}{2}-1} f_1(m)W_{N/2}^{mk}+\sum_{m=0}^{\frac{N}{2}-1} f_2(m)W_{N/2}^{mk}$$

since $W_N^2=W_{N/2}$

$$X(k)=F_1(k)+F_2(k)W_N^k$$

for k=0,1,2....N-1, where $F_1[k]$ and $F_2[k]$ are N/2 point DFT sequence of $f_1[m]$ and $f_2[m]$
Direct computation of $F_1[k]$ requires $\frac{N^2}{2}$ complex multiplications, same for $F_2(k)$. Additional $\frac{N}{2}$ for $F_2[k]W_N^k$. Thus calculating N point DFT of x[n] using above method total requires Total = $(\frac{N}{2})^2+(\frac{N}{2})^2+\frac{N}{2}$ complex multiplication.
At the same time, the direct calculation of N point DFT of x[n] requires total=N^2 complex multiplications. So reduction in Total = $N^2-((\frac{N}{2})^2+(\frac{N}{2})^2+\frac{N}{2})$.

1.4 Code and Result

1. **RADIX 2**

```
clc; clear all;
close all;

N=2; X=randn(1,N); k=0:(N/2)-1;
W=exp(1i*2*pi*k/N);
Y(1)=X(1)+X(2)*W(1); Y(2)=X(1)-X(2)*W(1);  z=fft(X,N);
disp(Y)
disp(z)
```

```
-1.3967   -3.1210

-1.3967   -3.1210
```

ii. **RADIX 8** clc;

```
clear all;
close all;


x=[1,2,3,4,5,6,7,8]; F1 =
x (1 : 2 : length(x));
F2 = x (2 : 2 : length(x));  N=8;
for k = 1 : N/2     temp1 = 0;     temp2 = 0;     for n = 1 : N/2
temp1 = temp1 + (F1(n)*exp (-1j*2*pi*(n-1)*(k-1)/(N/2)));        temp2
= temp2 + (F2(n)*exp (-1j*2*pi*(n-1)*(k-1)/(N/2)));

    end
    F1cal(k)=temp1;
    F2cal(k)=temp2*(exp(-1j*2*pi*(k-1)/N));
end for k=1 : N/2
    Xc(k)=F1cal(k)+F2cal(k);
Xc(k+N/2)=F1cal(k)-F2cal(k);
end z=fft(x); disp(Xc);
disp(z);
```

```
Columns 1 through 5

36.0000 + 0.0000i  -4.0000 + 9.6569i  -4.0000 + 4.0000i  -4.0000 + 1.6569i  -4.0000 + 0.0000i

Columns 6 through 8

-4.0000 - 1.6569i  -4.0000 - 4.0000i  -4.0000 - 9.6569i

Columns 1 through 5
```

iii. **RADIX 4** clc;

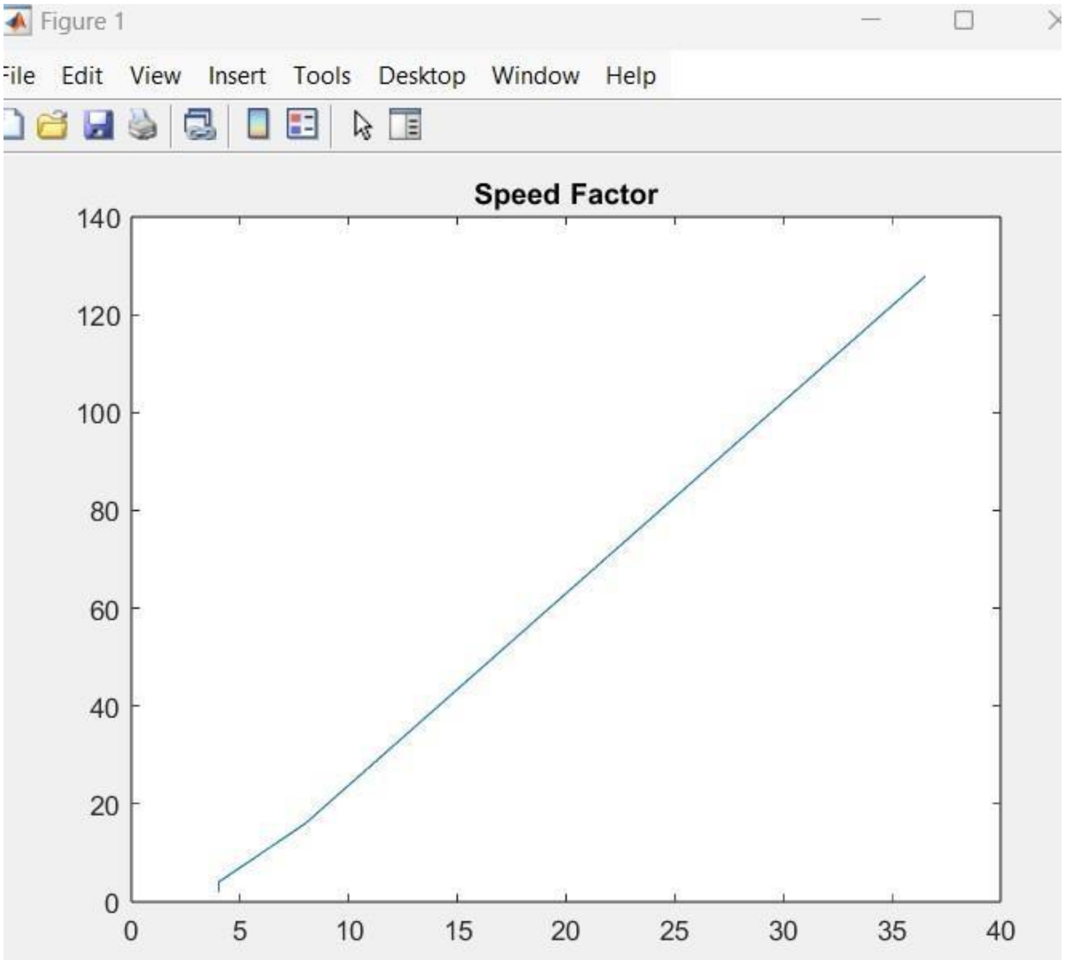
```
clear all;
close all;
```

```
x=[1,2,3,4];
F1 = x (1 : 2 : length(x));
F2 = x (2 : 2 : length(x));  N=4;
for k = 1 : N/2     temp1 = 0;     temp2 = 0;     for n = 1 : N/2
temp1 = temp1 + (F1(n)*exp (-1j*2*pi*(n-1)*(k-1)/(N/2)));         temp2
= temp2 + (F2(n)*exp (-1j*2*pi*(n-1)*(k-1)/(N/2)));

    end
    F1cal(k)=temp1;
    F2cal(k)=temp2*(exp(-1j*2*pi*(k-1)/N)); end
for k=1 : N/2
    Xc(k)=F1cal(k)+F2cal(k);
Xc(k+N/2)=F1cal(k)-F2cal(k); end
z=fft(x); disp(Xc);
disp(z);
```

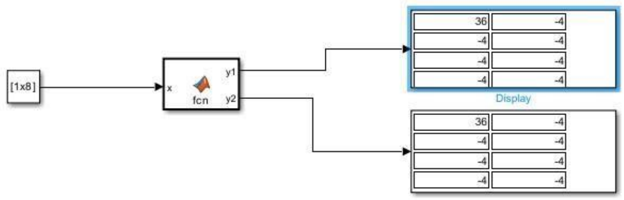
```
10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 + 0.0000i  -2.0000 - 2.0000i

10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 + 0.0000i  -2.0000 - 2.0000i
```

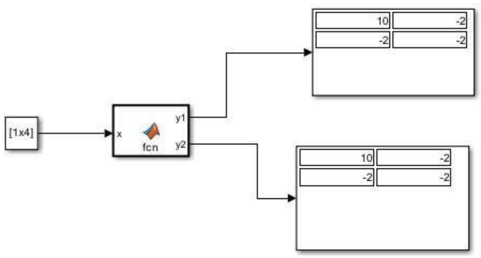
iv. **SPEED FACTOR**

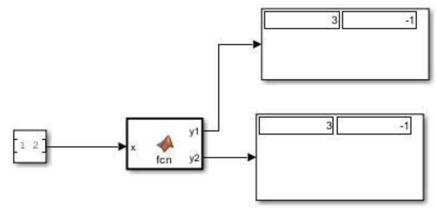
```
clc; clear all; close all;

N=[2,4,8,16,128];
x=zeros(1,5); for i
= 1 :5
   x(i) = (N(i)*N(i))/((N(i)/2)*log2(N(i)));
end plot (x,N); title("Speed Factor"); disp(x);
```



SIMULINK RESULT:


[1x8]
x
fcn
y1
y2
36
-4
-4
-4
-4
-4
-4
-4
Display
36
-4
-4
-4
-4
-4
-4
-4


[1x4]
x
fcn
y1
y2
10
-2
-2
-2
10
-2
-2
-2


1 2
x
fcn
y1
y2
3
-1
3
-1