NAME:	Anish Gade
UID:	2021700022
BRANCH:	T.Y. CSE Data Science
BATCH:	B
SUBJECT:	FOSIP
EXP. NO.:	4
DATE:	05/11/23

AIM: - To Perform FFT. To implement computationally Fast Algorithms

OBJECTIVES:

1. Develop a program to perform FFT of N point Signal.
2. Calculate FFT of a given DT signal and verify the results using mathematical formula.
3. Computational efficiency of FFT.

PROBLEM DEFINITION:

1. Take any four-point sequence x[n]. Find FFT of x[n] and IFFT of {X[k]}.
2. Calculate Real and Complex Additions & Multiplications involved to find X[k].

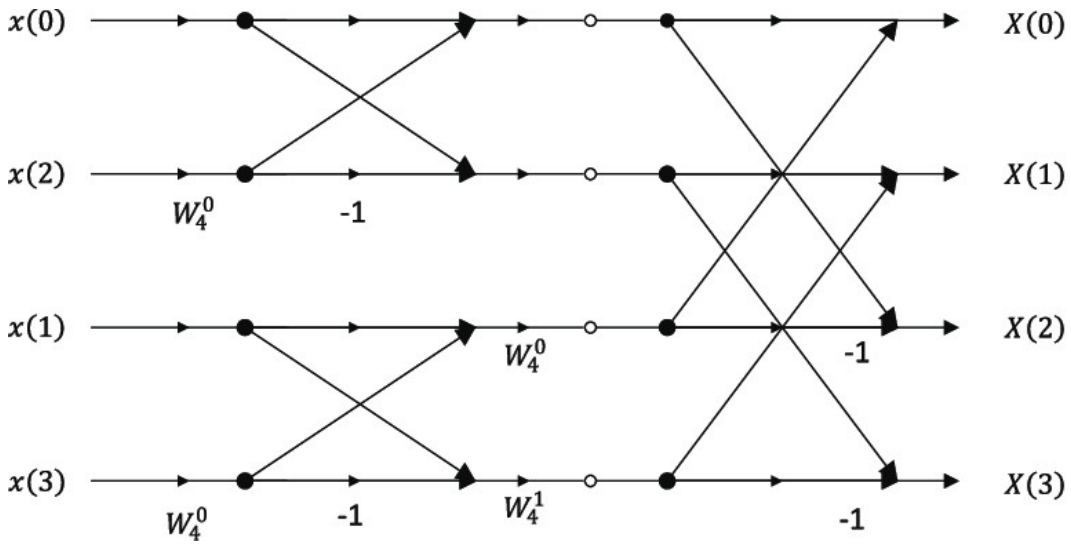
WRITE UP ON FFT:

The Fast Fourier Transform (FFT) is a powerful algorithm that has revolutionized the world of signal processing, mathematics, and a wide range of scientific disciplines. FFT is an essential tool for converting time-domain signals into their frequency-domain representations, enabling us to analyze, manipulate, and extract valuable information from various types of data. In this article, we will explore the concept of FFT, its history, applications, and its fundamental operation.

At its core, FFT is a mathematical algorithm that rapidly computes the discrete Fourier transform (DFT) of a sequence of data points. The DFT is a mathematical operation that transforms a signal from the time domain to the frequency domain. In simple terms, it breaks down a complex signal into its constituent sinusoidal components, revealing the frequencies and amplitudes that make up the original signal. This transformation is crucial for tasks like audio signal processing, image analysis, data compression, and more.

The FFT algorithm divides the DFT calculation into smaller, more manageable sub-problems. It leverages the periodicity and symmetry properties of complex exponentials to reduce the number of computations. By recursively breaking down the problem and recombining the results, the FFT significantly speeds up the computation, making it much more efficient than the naive approach, which requires O(N^2) operations, where N is the number of data points.

Four-Point FFT



Four Point FFT

1. To find FFT of 4-point sequence

   a. Input: [1,2,3,4]
   b. Output:

```
x = fft(x)
x
```

```
[(10+0j), (-2+2j), (-2+0j), (-1.9999999999999998-2j)]
```

IFFT of FFT of 4 Point Signal

```
x_iff = ifft(x)
np.abs(x_iff)
```

```
array([1., 2., 3., 4.])
```

   c.

Conclusion:

1. **Computational Efficiency in DFT:**

a) **Total Real Multiplications = $4N^2$**
b) **Total Real Additions = $4N^2-2N$**

2. **Computational Efficiency in FFT:**

a) **Total Real Multiplications = $2N \log_2 N$**
b) **Total Real Additions = $3N \log_2 N$**

3. **FFT produces fast results due to:**

Less Computations, Parallel implementations