- **The Fourier transform of an analogue signal x(t) is given by:**

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt$$

- **The Discrete Fourier Transform (DFT) of a discrete-time signal x(nT) is given by:**

$$X(k) = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}nk}$$

Where:

$$k = 0, 1, \ldots N-1$$

$$x(nT) = x[n]$$

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

x[n] = input

X[k] = frequency bins

W = twiddle factors

$$\mathbf{X(0) = x[0]W_N^{0} + x[1]W_N^{0*1} + \ldots + x[N-1]W_N^{0*(N-1)}}$$

$$\mathbf{X(1) = x[0]W_N^{0} + x[1]W_N^{1*1} + \ldots + x[N-1]W_N^{1*(N-1)}}$$

$$\vdots$$

$$\mathbf{X(k) = x[0]W_N^{0} + x[1]W_N^{k*1} + \ldots + x[N-1]W_N^{k*(N-1)}}$$

$$\vdots$$

$$\mathbf{X(N-1) = x[0]W_N^{0} + x[1]W_N^{(N-1)*1} + \ldots + x[N-1]W_N^{(N-1)(N-1)}}$$

Note: For N samples of x we have N frequencies representing the signal.

Performance of the DFT Algorithm

- **The DFT requires N^2 (NxN) complex multiplications:**
  - **Each X(k) requires N complex multiplications.**
  - **Therefore to evaluate all the values of the DFT (X(0) to X(N-1)) N^2 multiplications are required.**
- **The DFT also requires (N-1)*N complex additions:**
  - **Each X(k) requires N-1 additions.**
  - **Therefore to evaluate all the values of the DFT (N-1)*N additions are required.**

The Discrete Fourier Transform (DFT)

- DFT of an N-point sequence x_n, $n = 0, 1, 2, \ldots, N-1$ is defined as

$$X_k = \sum_{n=0}^{N-1} x_n \, e^{-j\frac{2\pi k}{N} n} \qquad k = 0, 1, 2, \cdots, N-1$$

- An N-point sequence yields an N-point transform
- X_k can be expressed as an *inner product*:

$$X_k = \begin{bmatrix} 1 & e^{-j\frac{2\pi k}{N}} & e^{-j\frac{2\pi k}{N} 2} & \ldots & e^{-j\frac{2\pi k}{N}(N-1)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix}$$

The Discrete Fourier Transform (DFT)

- Notation: $W_N = e^{-j\frac{2\pi}{N}}$. Hence,

$$X_k = \begin{bmatrix} 1 & W_N^k & W_N^{2k} & \dots & W_N^{(N-1)k} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix}$$

- By varying k from 0 to $N-1$ and combining the N inner products, we get the following:

$$\mathbf{X} = \mathbf{W}\mathbf{x}$$

- $\mathbf{W}$ is an $N \times N$ matrix, called as the "DFT Matrix"

The DFT Matrix

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ & & & \vdots & \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}_{N \times N}$$

- The notation $\mathbf{W}_N$ is used if we want to make the size of the DFT matrix explicit

How Many Complex Multiplications Are Required?

- Each inner product requires N complex multiplications
  - There are N inner products
- Hence we require N^2 multiplications
- However, the first row and first column are all 1s, and *should not be counted as multiplications*
  - There are $2N - 1$ such instances
- Hence, the number of complex multiplications is $N^2 - 2N + 1$, i.e., $(N-1)^2$

How Many Complex Additions Are Required?

- Each inner product requires $N-1$ complex additions
  - There are N inner products
- Hence we require $N(N-1)$ complex additions

Total Operation Count

- No. of complex multiplications: $(N-1)^2$
- No. of complex additions: $N(N-1)$
- The operation count for multiplications and additions assumes that W_N^k has been computed offline and is available in memory
  - If pre-computed values of W_N^k are not available, then the operation count will increase
- We will assume that all the required W_N^k have been pre-computed and are available

Operation Count Makes DFT Impractical

- For large N,

  $(N-1)^2 \approx N^2$

  $N(N-1) \approx N^2$
- Hence both multiplications and additions are $O(N^2)$
- If $N = 10^3$, then $O(N^2) = 10^6$, i.e., a million!
- This makes the straightforward method slow and impractical even for a moderately long sequence

The Decimation in Time (DIT) Algorithm

- From $\{x_n\}$ form two sequences as follows:

$$\{g_n\} = \{x_{2n}\} \qquad \{h_n\} = \{x_{2n+1}\}$$

- $\{g_n\}$ contains the even-indexed samples, while $\{h_n\}$ contains the odd-indexed samples
- The DFT of $\{x_n\}$ is

$$\begin{aligned}
X_k &= \sum_{n=0}^{N-1} x_n\, W_N^{nk} \\
&= \sum_{r=0}^{\frac{N}{2}-1} x_{2r}\, W_N^{(2r)k} + \sum_{r=0}^{\frac{N}{2}-1} x_{2r+1}\, W_N^{(2r+1)k} \\
&= \sum_{r=0}^{\frac{N}{2}-1} g_r\, W_N^{(2r)k} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} h_r\, W_N^{(2r)k}
\end{aligned}$$

The Decimation in Time (DIT) Algorithm

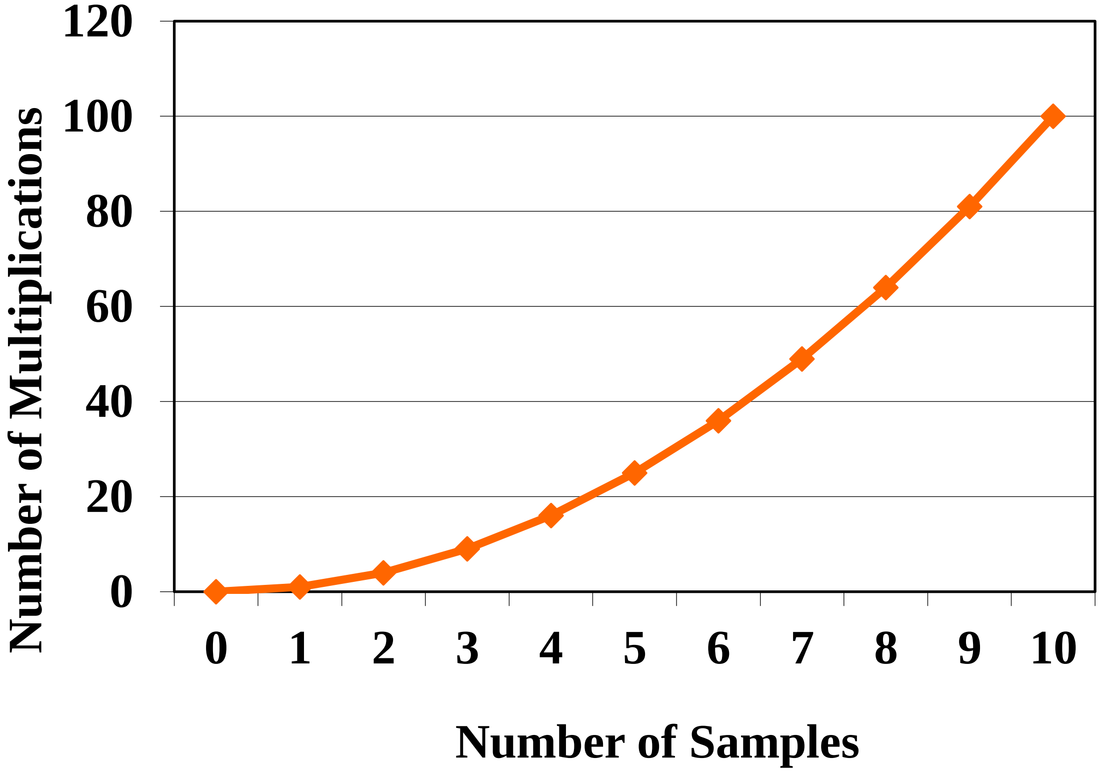
- But,

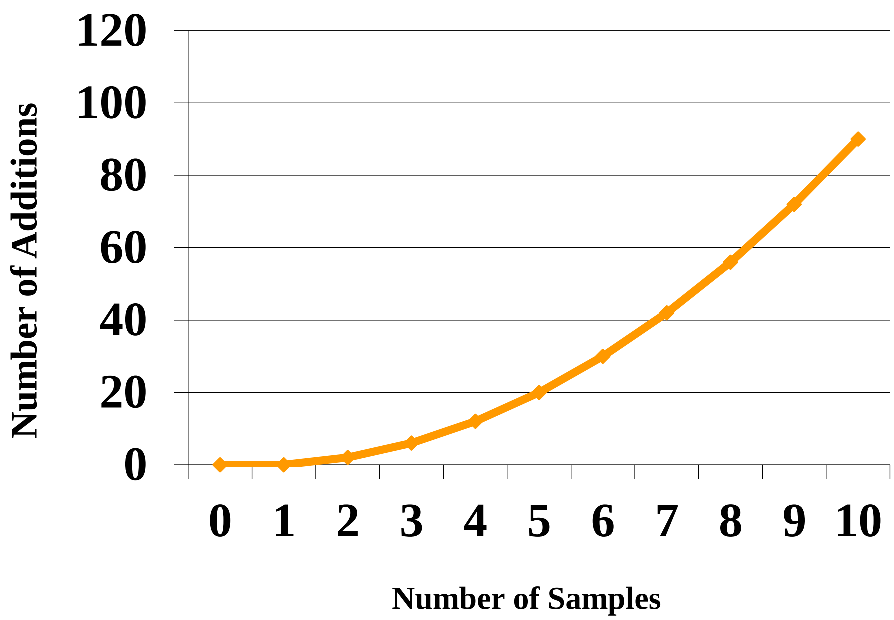
$$W_N^{2rk} = e^{-j\frac{2\pi}{N}(2rk)} = e^{-j\frac{2\pi}{N/2}(rk)} = W_{N/2}^{rk}$$

  and hence

$$\begin{aligned} X_k &= \sum_{r=0}^{\frac{N}{2}-1} g_r\, W_{N/2}^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} h_r\, W_{N/2}^{rk} \\ &= G_k + W_N^k H_k \qquad k = 0, 1, \ldots, N-1 \end{aligned}$$

- $\{G_k\}$ and $\{H_k\}$ are $\frac{N}{2}$ point DFTs
- The overhead for combining the two $\frac{N}{2}$ point DFTs is the multiplicative factor W_N^k for $k = 0, 1, \ldots, N-1$
    - W_N^k is called "twiddle factor"





- **Can the number of computations required be reduced?**

DFT $\rightarrow$ FFT

- **A large amount of work has been devoted to reducing the computation time of a DFT.**
- **This has led to efficient algorithms which are known as the Fast Fourier Transform (FFT) algorithms.**

$$X(k) = \sum_{n=0}^{N-1} x[n] W_N^{nk}; \quad 0 \le k \le N-1 \tag{1}$$

x[n] = x[0], x[1], …, x[N-1]

- **Lets divide the sequence x[n] into even and odd sequences:**
  - **x[2n] = x[0], x[2], …, x[N-2]**
  - **x[2n+1] = x[1], x[3], …, x[N-1]**

- **Equation 1 can be rewritten as:**

$$X\left(k\right)=\sum_{n=0}^{\frac{N}{2}-1}x\left[2n\right]W_N^{2nk}+\sum_{n=0}^{\frac{N}{2}-1}x\left[2n+1\right]W_N^{\left(2n+1\right)k} \tag{2}$$

- **Since:**

$$\begin{aligned}W_N^{2nk}&=e^{-j\frac{2\pi}{N}2nk}=e^{-j\frac{2\pi}{N/2}nk}\\&=W_{\frac{N}{2}}^{nk}\end{aligned}$$

$$W_N^{(2n+1)k}=W_N^k\cdot W_{\frac{N}{2}}^{nk}$$

- **Then:**

$$\begin{aligned}X\left(k\right)&=\sum_{n=0}^{\frac{N}{2}-1}x\left[2n\right]W_{\frac{N}{2}}^{nk}+W_N^k\sum_{n=0}^{\frac{N}{2}-1}x\left[2n+1\right]W_{\frac{N}{2}}^{nk}\\&=Y\left(k\right)+W_N^kZ\left(k\right)\end{aligned}$$

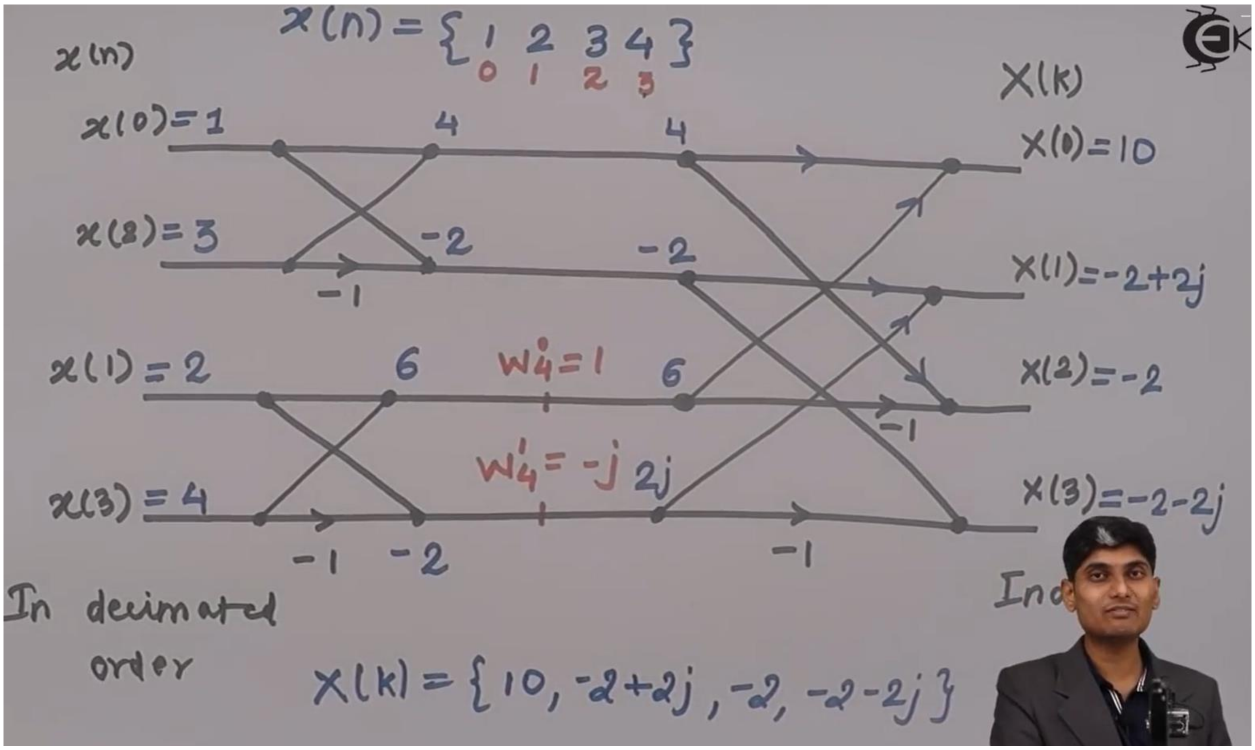
- **The result is that an N-point DFT can be divided into two N/2 point DFT's:**

$$X\left(k\right)=\sum_{n=0}^{N-1}x\left[n\right]W_N^{nk};\quad 0\le k\le N-1$$ **N-point DFT**

- **Where Y(k) and Z(k) are the two N/2 point DFTs operating on even and odd samples respectively:**

$$\begin{aligned}X\left(k\right)&=\sum_{n=0}^{\frac{N}{2}-1}x_1\left[n\right]W_{\frac{N}{2}}^{nk}+W_N^k\sum_{n=0}^{\frac{N}{2}-1}x_2\left[n\right]W_{\frac{N}{2}}^{nk}\\&=Y\left(k\right)+W_N^kZ\left(k\right)\end{aligned}$$ **Two N/2-point DFTs**

4 point DIT-FFT



Example 1: Consider a sequence $x[n] = \{\, 1, 1, -1, -1, -1, 1, 1, -1\}$

Determine DFT X[k] of x[n] using the **decimation-in-time FFT** algorithm.

$$f[n] = x[2n] = \{\, x[0], x[2], x[4], x[6]\} \quad = \{1, -1, -1, 1\}$$

$$g[n] = x[2n+] = \{\, x[1], x[3], x[5], x[7]\} = \{1, -1, 1, -1\}$$

X= W_x

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{bmatrix} = \begin{bmatrix} w_4^0 & w_4^0 & w_4^0 & w_4^0 \\ w_4^0 & w_4^1 & w_4^2 & w_4^3 \\ w_4^0 & w_4^2 & w_4^4 & w_4^6 \\ w_4^0 & w_4^3 & w_4^6 & w_4^9 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix}$$

$$=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -j & -1 & j\\ 1 & -1 & 1 & -1\\ 1 & j & -1 & -j\end{bmatrix}\begin{bmatrix}1\\ -1\\ -1\\ 1\end{bmatrix}$$

$$=\begin{bmatrix}0\\ 2+j2\\ 0\\ 2-j2\end{bmatrix}$$

$$\begin{bmatrix}G(0)\\ G(1)\\ G(2)\\ G(3)\end{bmatrix}=\begin{bmatrix}w_4^0 & w_4^0 & w_4^0 & w_4^0\\ w_4^0 & w_4^1 & w_4^2 & w_4^3\\ w_4^0 & w_4^2 & w_4^4 & w_4^6\\ w_4^0 & w_4^3 & w_4^6 & w_4^9\end{bmatrix}\begin{bmatrix}g(0)\\ g(1)\\ g(2)\\ g(3)\end{bmatrix}$$

$$=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -j & -1 & j\\ 1 & -1 & 1 & -1\\ 1 & j & -1 & -j\end{bmatrix}\begin{bmatrix}1\\ -1\\ 1\\ -1\end{bmatrix}$$

$$=\begin{bmatrix}0\\ 0\\ 4\\ 0\end{bmatrix}$$

$X[0] = F[0] + W_8^0 G[0] = 0$

$X[1] = F[1] + W_8^1 G[1] = 2 + j2$

$X[2] = F[2] + W_8^2 G[2] = -j4$

$X[3] = F[3] + W_8^3 G[3] = 2 - j2$

$X[4] = F[0] - W_8^0 G[0] = 0$

$X[5] = F[1] - W_8^1 G[1] = 2 + j2$

$X[6] = F[2] - W_8^2 G[2] = j4$

$X[7] = F[3] - W_8^3 G[3] = 2 - j2$

Example 1: Consider a sequence $x[n] = \{\, 1, 1, -1, -1, -1, 1, 1, -1\}$

Determine DFT X[k] of x[n] using the **decimation-in-frequency FFT algorithm**.

$$p[n] = x[n] + x\left[n + \frac{N}{2}\right]$$

$$= \{\, (1-1), (1+1), (-1+1), (-1-1)\} \quad = \{0, 2, 0, 2\}$$

$$q[n] = \left(x[n] - x\left[n + \frac{N}{2}\right]\right) W_8^n$$

$$= \{ (1+1)W_8^0, (1-1)W_8^1, (-1-1)W_8^2, (-1+1)W_8^3 \}$$

$$= \{2, 0, j2, 0\}$$

$X = Wx$

$$\begin{bmatrix} \mathrm{P}(0) \\ \mathrm{P}(1) \\ \mathrm{P}(2) \\ \mathrm{P}(3) \end{bmatrix} = \begin{bmatrix} \mathrm{w}_4^0 & \mathrm{w}_4^0 & \mathrm{w}_4^0 & \mathrm{w}_4^0 \\ \mathrm{w}_4^0 & \mathrm{w}_4^1 & \mathrm{w}_4^2 & \mathrm{w}_4^3 \\ \mathrm{w}_4^0 & \mathrm{w}_4^2 & \mathrm{w}_4^4 & \mathrm{w}_4^6 \\ \mathrm{w}_4^0 & \mathrm{w}_4^3 & \mathrm{w}_4^6 & \mathrm{w}_4^9 \end{bmatrix} \begin{bmatrix} \mathrm{p}(0) \\ \mathrm{p}(1) \\ \mathrm{p}(2) \\ \mathrm{p}(3) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 0 \\ -2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ -j4 \\ 0 \\ j4 \end{bmatrix}$$

$$\begin{bmatrix} \mathrm{Q}(0) \\ \mathrm{Q}(1) \\ \mathrm{Q}(2) \\ \mathrm{Q}(3) \end{bmatrix} = \begin{bmatrix} \mathrm{w}_4^0 & \mathrm{w}_4^0 & \mathrm{w}_4^0 & \mathrm{w}_4^0 \\ \mathrm{w}_4^0 & \mathrm{w}_4^1 & \mathrm{w}_4^2 & \mathrm{w}_4^3 \\ \mathrm{w}_4^0 & \mathrm{w}_4^2 & \mathrm{w}_4^4 & \mathrm{w}_4^6 \\ \mathrm{w}_4^0 & \mathrm{w}_4^3 & \mathrm{w}_4^6 & \mathrm{w}_4^9 \end{bmatrix} \begin{bmatrix} \mathrm{q}(0) \\ \mathrm{q}(1) \\ \mathrm{q}(2) \\ \mathrm{q}(3) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ j2 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2+j2 \\ 2-j2 \\ 2+j2 \\ 2-j2 \end{bmatrix}$$

$X[0] = P[0] = 0$

$X[1] = Q[0] = 2 + j2$

$X[2] = P[1] + W_8^2 G[2] = -j4$

$X[4] = P[2] = 0$

$X[5] = Q[2] = 2 + j2$

$X[6] = P[3] = j4$

$X[3] = Q[1] + W_8^3 G[3] = 2 - j2$

$X[7] = Q[3] = 2 - j2$