Efficient Computation of the DFT: Fast Fourier Transform Algorithms

Dr. Sampath Kumar

Associate Professor

Department of ECE

MIT, Manipal

DFT plays an important role in many applications of Digital Signal Processing

- Linear Filtering
- Correlation Analysis
- Spectrum analysis

Efficient algorithms are available for DFT computation

Complexity in Direct computation of DFT:

Each DFT point direct computation of X(k) involves

- N complex multiplications (**4N real multiplications**)
- N-1 complex additions (**4N-2 real additions**)
- To compute All N points of the DFT we require N^2 **complex multiplications and** $N^2 - N$ **complex additions**

Two approaches for efficient computation:

1. Divide and Conquer approach (for composite N)
    Decomposition of an N-point DFT into successively smaller DFTs
        Decimation In Time FFT (DITFFT) Algorithm
        Decimation In Frequency (DIFFFT) Algorithm

2. Formulation of the DFT as a linear filtering operation on the data
    Goertzel Algorithm
    Chirp-z Transform

Radix-2 FFT Algorithms

Decimation In Time FFT (DITFFT) Algorithm

- Decompose the computation into successively smaller DFT computations.
- We exploit both the symmetry and periodicity property of the complex exponential $W_N^{nk} = e^{-j\left(\frac{2\pi}{N}\right)nk}$
- In this algorithm the decomposition is based on the sequence $x(n)$ into successively smaller subsequences

Consider N an integer power of 2, $N = 2^{\nu}$

We know that $X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \quad k = 0, 1, \ldots N-1$ —— (1)

Let us now separate $x(n)$ into even-and odd-numbered points

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

let us switch to new variables, $n = 2r$ (even) and $n = (2r+1)$ (odd)

$$X(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r) W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1) W_N^{(2r+1)k} \quad \text{—— (2)}$$

$$X(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r)\, W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)\, W_N^{(2r+1)k} \quad \text{——(2)}$$

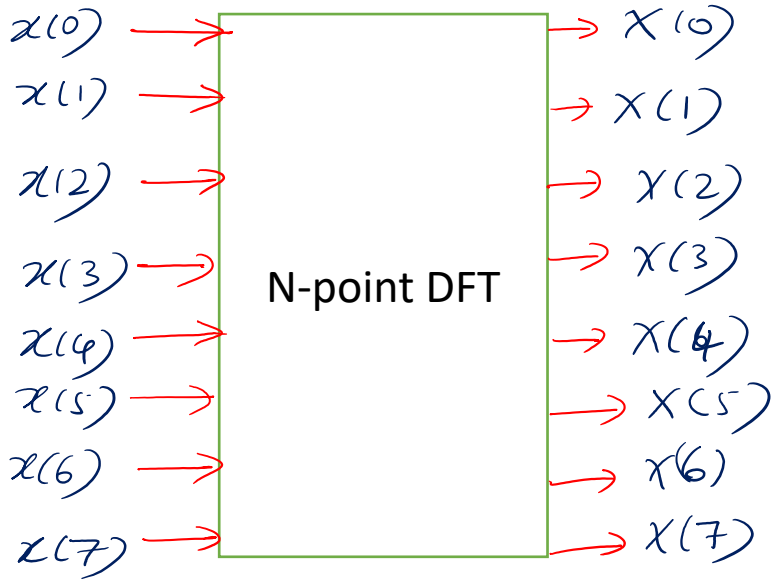
$$= \sum_{r=0}^{\frac{N}{2}-1} x(2r)\cdot W_{\frac{N}{2}}^{rk} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)\, W_{\frac{N}{2}}^{rk}\cdot W_N^k \quad \text{——(3)}$$

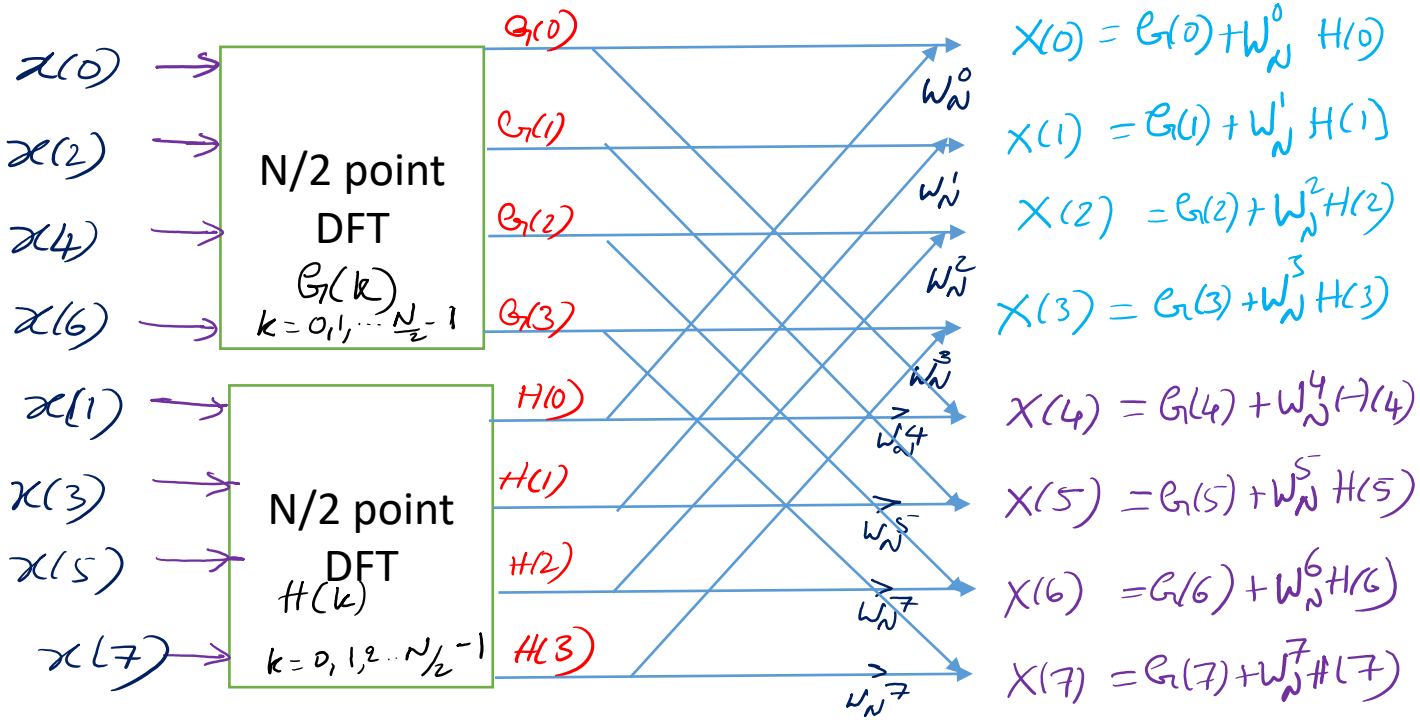
$G_1(k) \Rightarrow N/2$ pt DFT

$H(k) \Rightarrow \frac{N}{2}$ pt DFT

$$W_N^2 = W_{N/2} = e^{-j2\left(\frac{2\pi}{N}\right)} = e^{-j\frac{2\pi}{N/2}}$$

$$X(k) = G_1(k) + W_N^k\, H(k) \quad \text{——(4)}$$





$$X(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r) \cdot W_{\frac{N}{2}}^{rk} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)\, W_{\frac{N}{2}}^{rk} \cdot W_N^k \quad \text{—(3)}$$

Now letus consider first term in the eqn(3)

$$G_1(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r)\, W_{\frac{N}{2}}^{rk} \quad \text{—— (5)}$$

We can rewrite it as

$$G_1(k) = \sum_{r=0}^{\frac{N}{2}-1} g(r)\, W_{N/2}^{rk} \qquad k = 0, 1, \cdots N/2 - 1$$

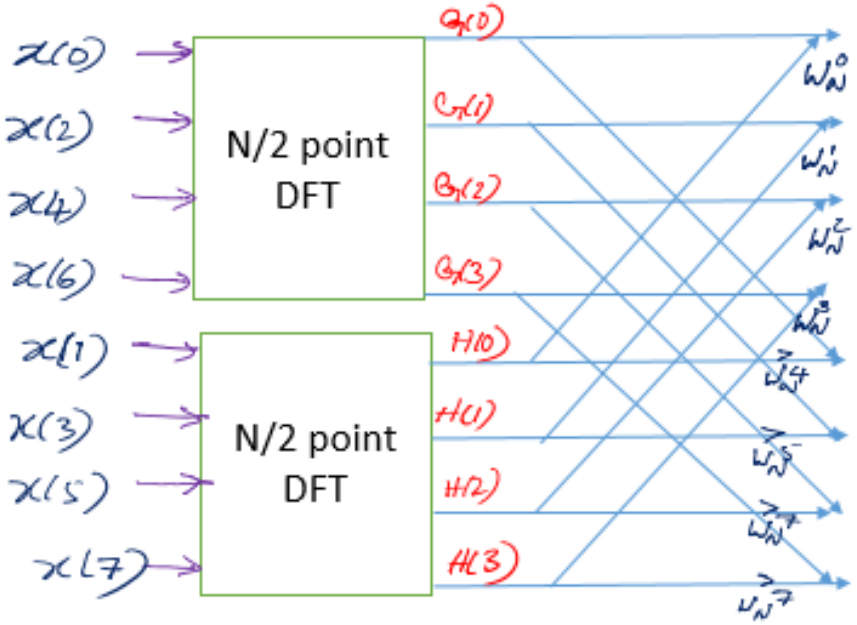
We can again decompose $g(r)$ into even and odd indexed sequence

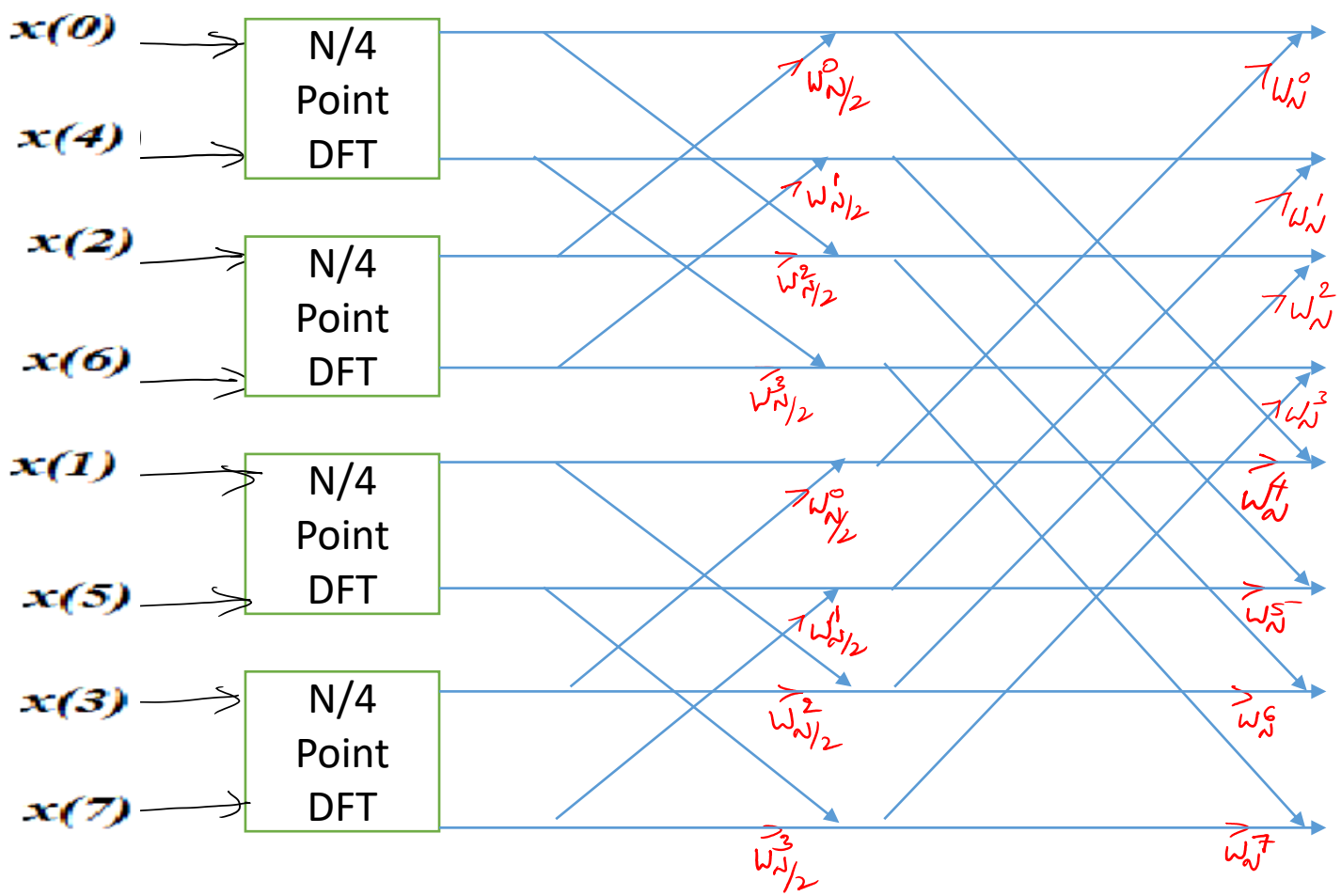
$$G(k) = \sum_{l=0}^{N/4-1} g(2l)\, W_{N/2}^{2lk} + \sum_{l=0}^{\frac{N}{4}-1} g(2l+1)\, W_{N/2}^{(2l+1)k}$$

$$= \sum_{l=0}^{N/4-1} g(2l)\, W_{N/4}^{lk} + W_{N/2}^{k} \sum_{l=0}^{\frac{N}{4}-1} g(2l+1)\, W_{\frac{N}{4}}^{lk}$$

Similarly

$$H(k) = \sum_{l=0}^{\frac{N}{2}-1} h(l)\, W_{\frac{N}{2}}^{lk} = \sum_{l=0}^{N/4-1} h(2l)\, W_{\frac{N}{4}}^{lk} + W_{\frac{N}{2}}^{k} \sum_{l=0}^{\frac{N}{4}-1} h(2l+1)\, W_{\frac{N}{4}}^{lk}$$



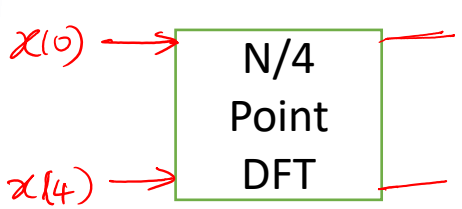


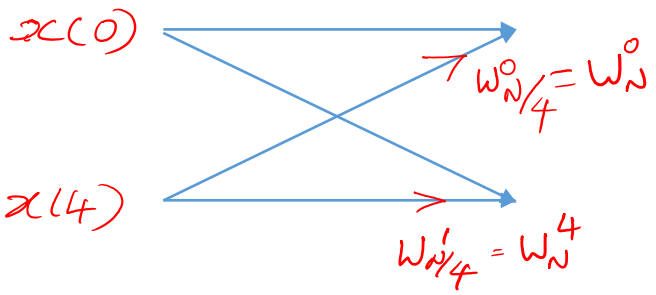
We can keep on decomposing $x(n)$ in this fashion

One final stage we reach is a two point DFT computation.

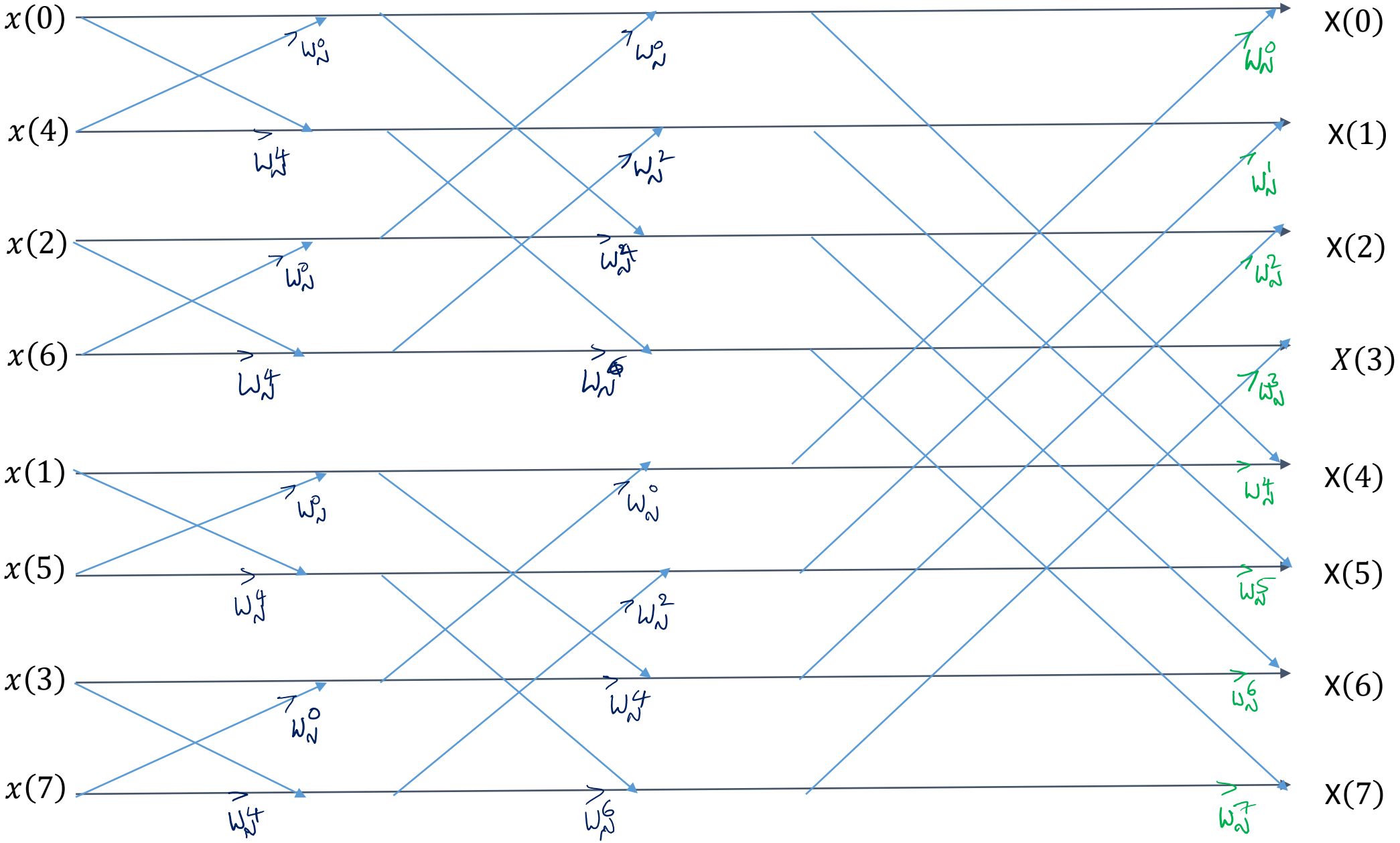
So far we have seen the decomposition of $N=8$ -pt DFT

and we have arrived at $\frac{N}{4}$-pt DFT i.e, 2-pt DFT computation

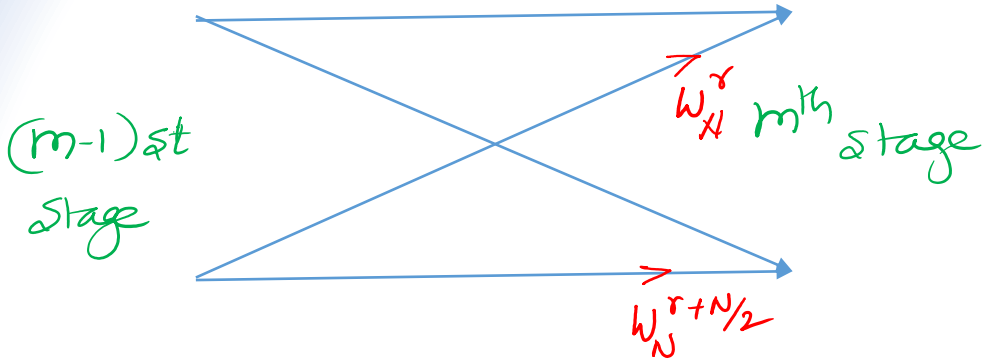


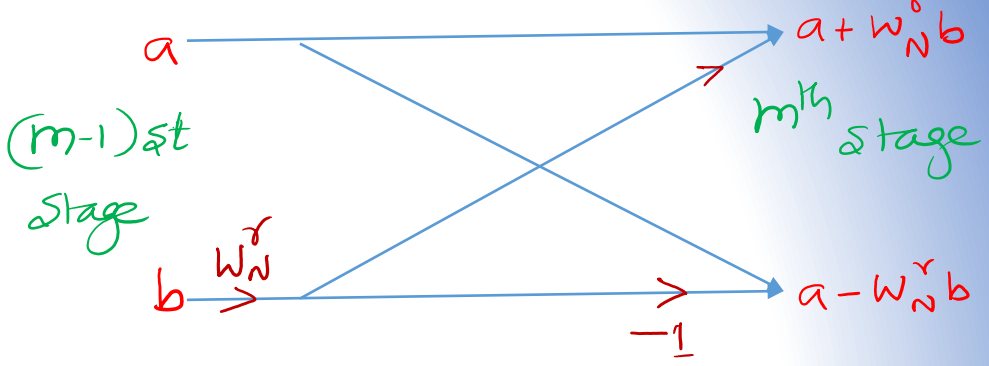


Simplified signal flow graph for $N=8$-point DFTs



Butterfly Computation





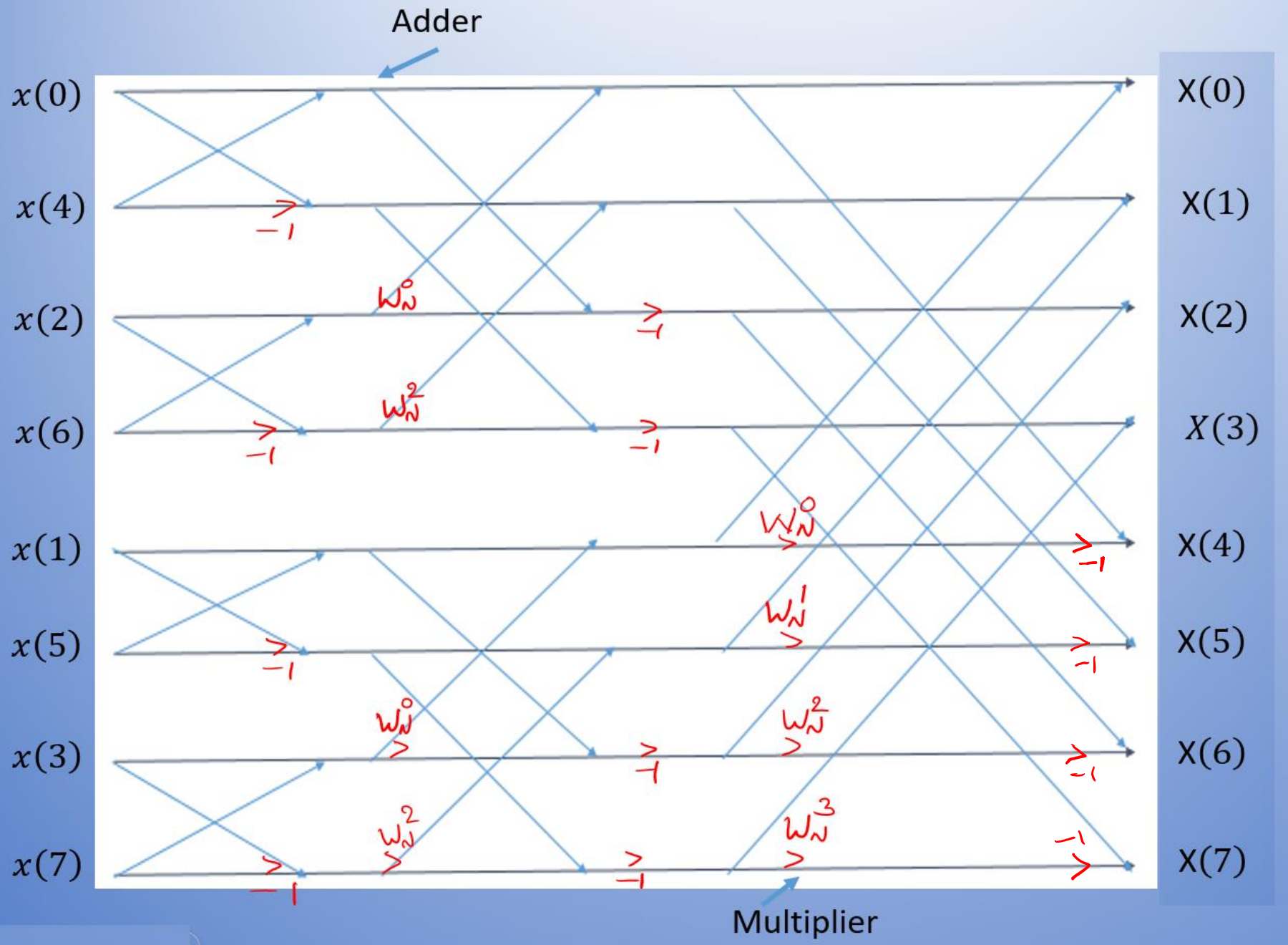
We can write $W_N^{r+N/2} = W_N^{N/2} \cdot W_N^r$

$= e^{\frac{-j2\pi N/2}{N}} \cdot W_N^r$

$= e^{-j\pi} W_N^r$

$\therefore W_N^{r+N/2} = -W_N^r$

Complete signal flow graph for DIT FFT Algorithm for 8-pt DFT




Thank
you