DFT

- N-point signals

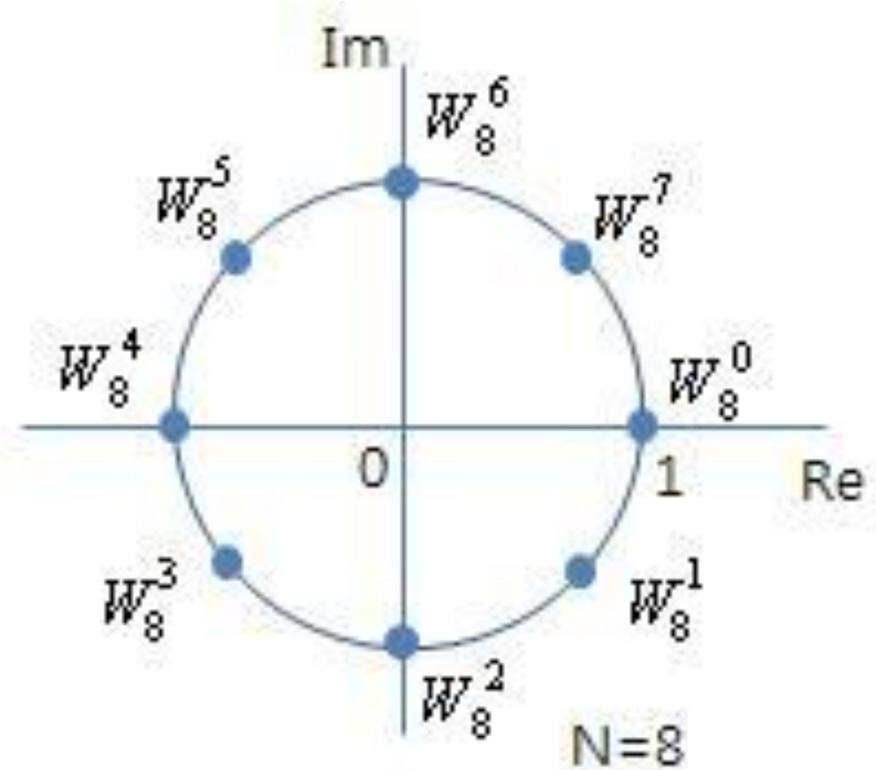
DFT	$X[k]=\sum_{n=0}^{N-1} x[n]W_N^{kn},$	$k=0,1\ldots,N-1$
IDFT (inverse DFT)	$x[n]=\frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn},$	$n=0,1\ldots,N-1$

where $W_N = e^{-j(2\pi/N)}$, and W_N^n are the roots of the polynomial $W^N = 1$.

$N = 8$

What is W_N^n? n-th Root of **1**

- W_N^n is the n-th root of the equation $\boldsymbol{W}^N = 1$.
- Eg., when N=8,



Fast Fourier Transform (FFT)

- **An important characteristic of DFT is that it can be computed very fast, referred to as FFT.**
- **So, the spectrogram are often computed using FFT.**

DFT pairs:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \qquad k = 0,1\ldots, N-1$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, \qquad n = 0,1\ldots, N-1$$

$W_N = e^{-j2\pi/N}$ is a root of the equation $\mathbf{W}^N=1$.

Originally, it requires N^2 complex multiplications and $(N-1)N$ complex additions for direct computation.

Fast Fourier Transform (FFT)

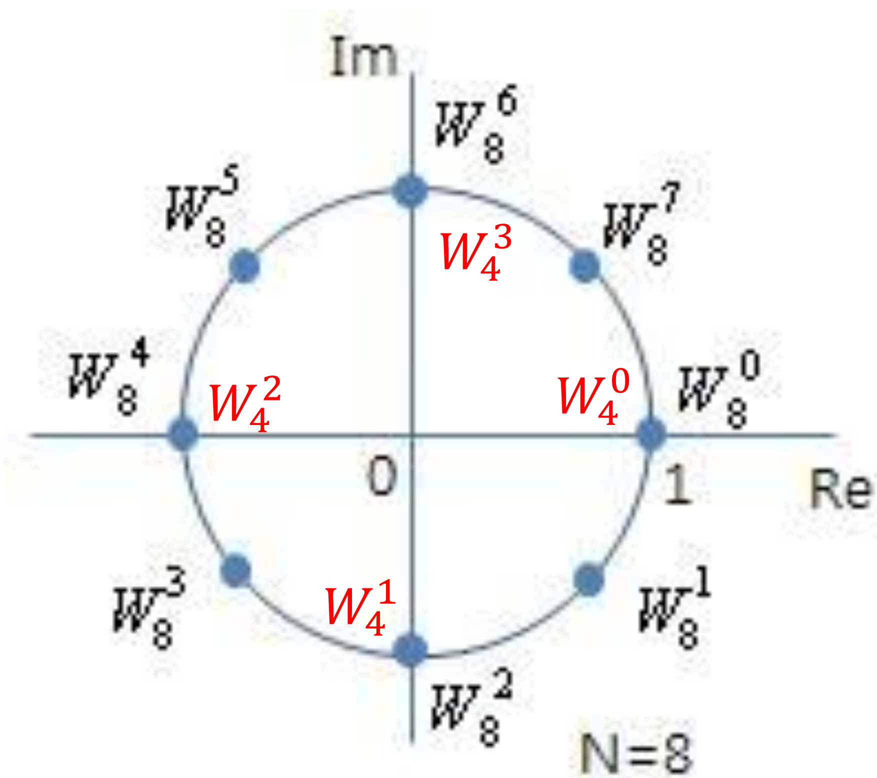
- One of the important algorithms for FFT is the Cooley-Tukey algorithm.
- The key principle used in the Cooley-Tukey algorithm is
  - $W_N^2 = W_{N/2}$

    (eg., $W_8^2 = W_4$)
  - Divide and concur (iteratively)

$N = 8$

What is W_N^n? n-th Root of **1**

- W_N^n is the n-th root of the equation $\boldsymbol{W}^N = 1$.
- Eg., when n=8,



Decimation-in-time FFT algorithm

Most conveniently illustrated by considering the special case of N an integer power of 2, i.e, $N = 2^v$.

Since N is an even integer, we can compute $X[k]$ by separating $x[n]$ into two $(N/2)$-point sequences consisting of the even numbered point in $x[n]$ and the odd-numbered points in $x[n]$, respectively

$$X[k] = \sum_{n\,even} x[n] W_N^{nk} + \sum_{n\,\mathrm{odd}} x[n] W_N^{nk}$$

Decimation-in-time FFT algorithm

With the substitution of variable n=2r for n even and n=2r+1 for n odd:

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r] W_N^{2rk} + \sum_{r=0}^{(N/2)-1} x[2r+1] W_N^{(2r+1)k}$$

By the key property: $W_N^2 = e^{-2j(2\pi/N)} = e^{-j2\pi/(N/2)} = W_{N/2}$

We have

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r](W_N^2)^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x[2r+1](W_N^2)^{rk}$$

$$= \sum_{r=0}^{(N/2)-1} x[2r](W_{N/2})^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x[2r+1](W_{N/2})^{rk}$$

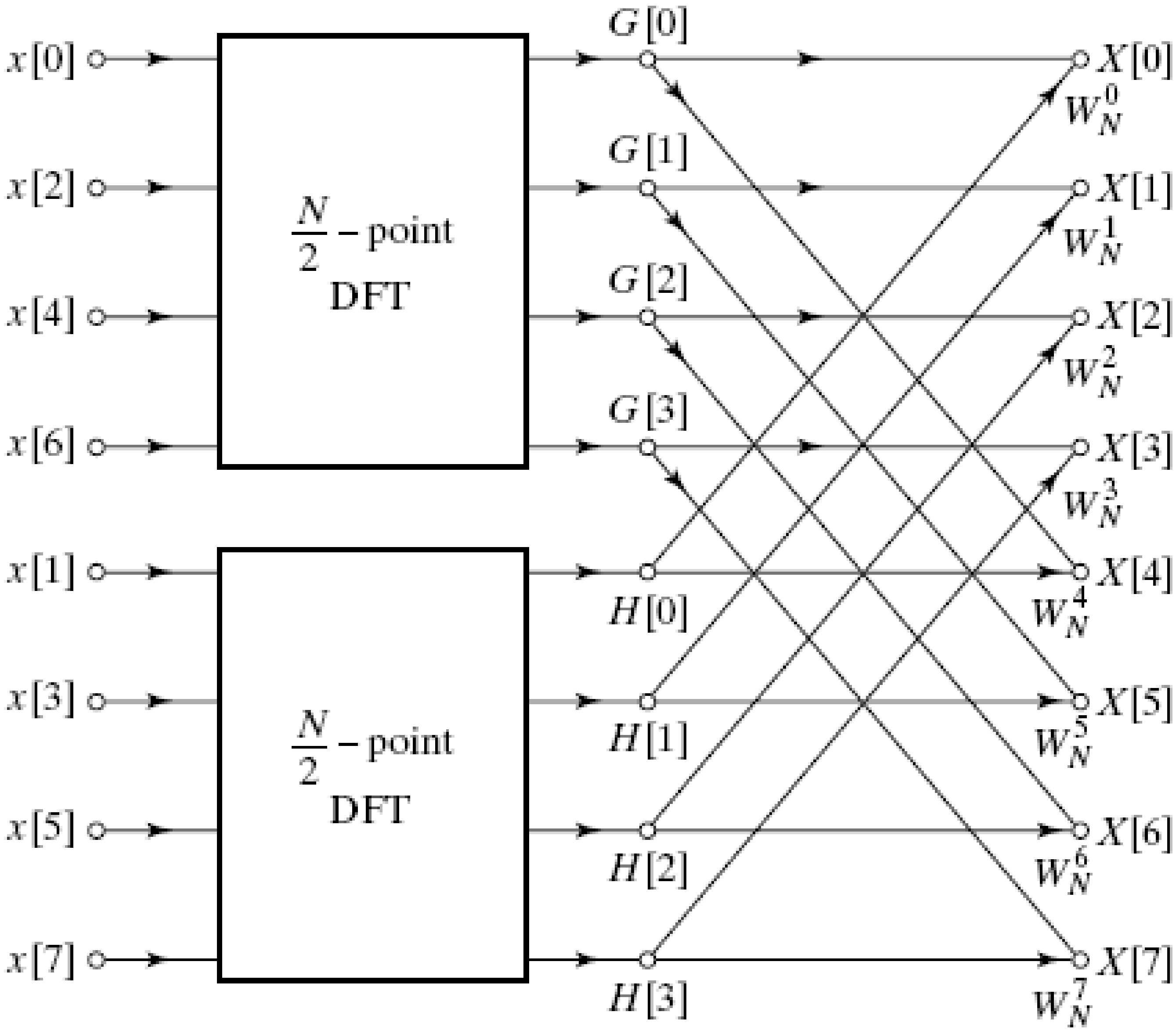
In sum,

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r](W_{N/2})^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x[2r+1](W_{N/2})^{rk}$$

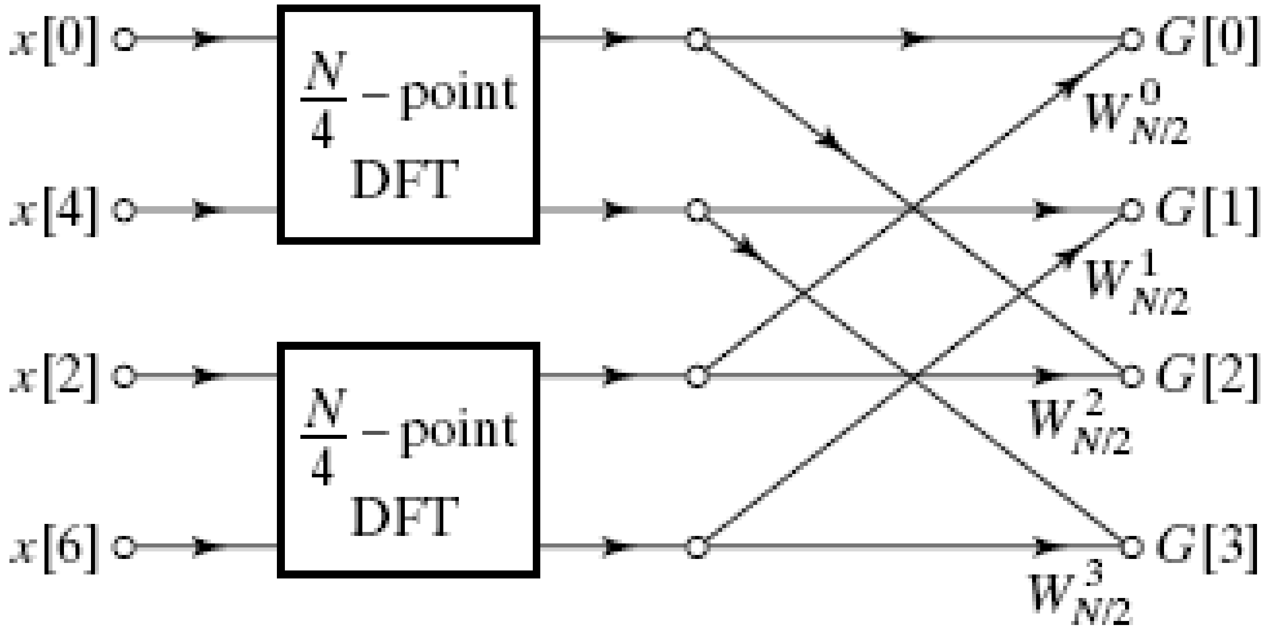
$$= G[k] + W_N^k H[k], \quad k = 0,1,\ldots, N-1$$

- Both G[k] and H[k] can be computed by (N/2)-point DFT
- G[k]: the (N/2)-point DFT of the even numbered points of the original sequence
- H(k): the (N/2)-point DFT of the odd-numbered points of the original sequence.
- Although the index ranges over N values, k = 0, 1, …, N-1, they must be computed only for k between 0 and (N/2)-1, since G[k] and H[k] are each periodic in k with period N/2.

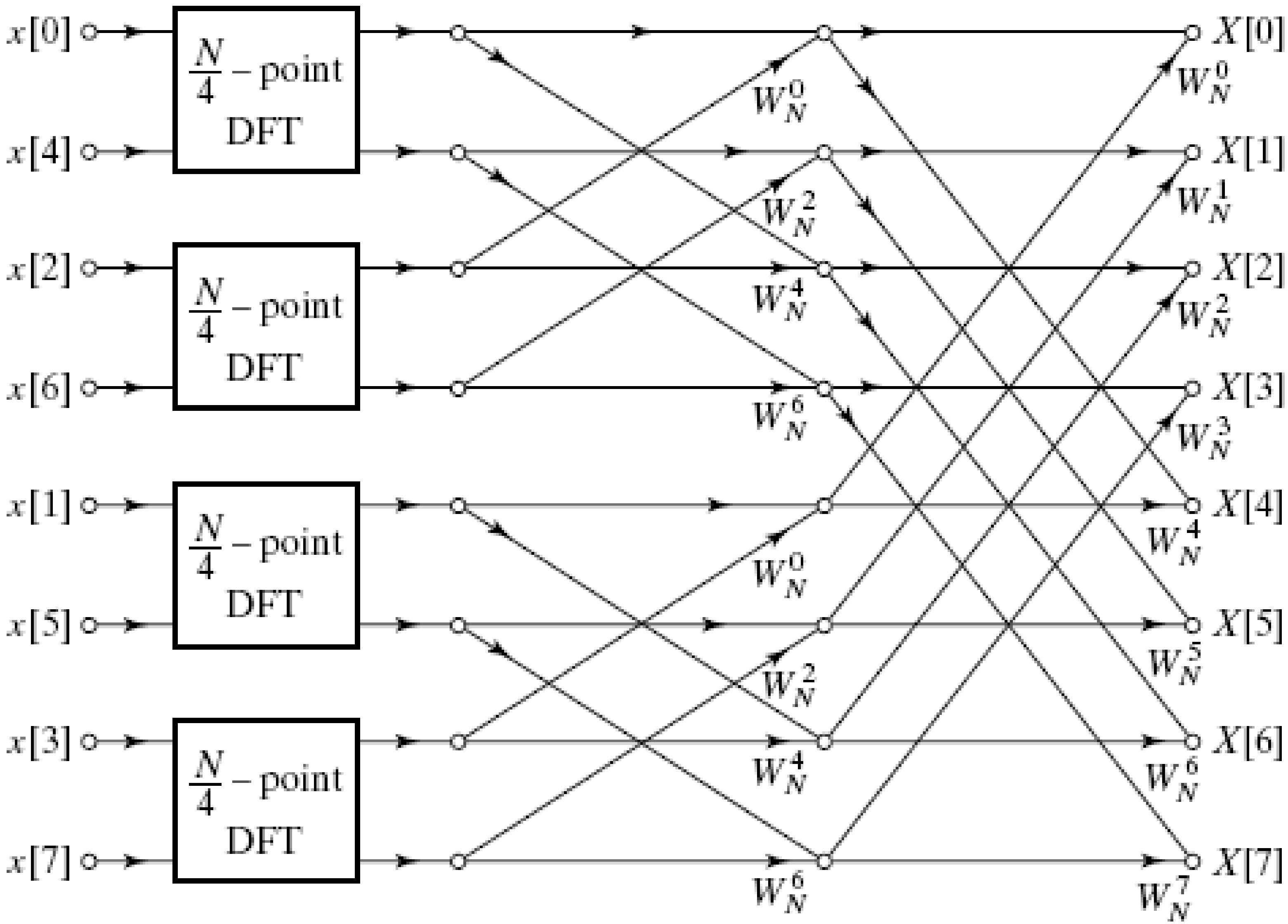
Decomposing N-point DFT into two (N/2)-point DFTs for the case of N=8



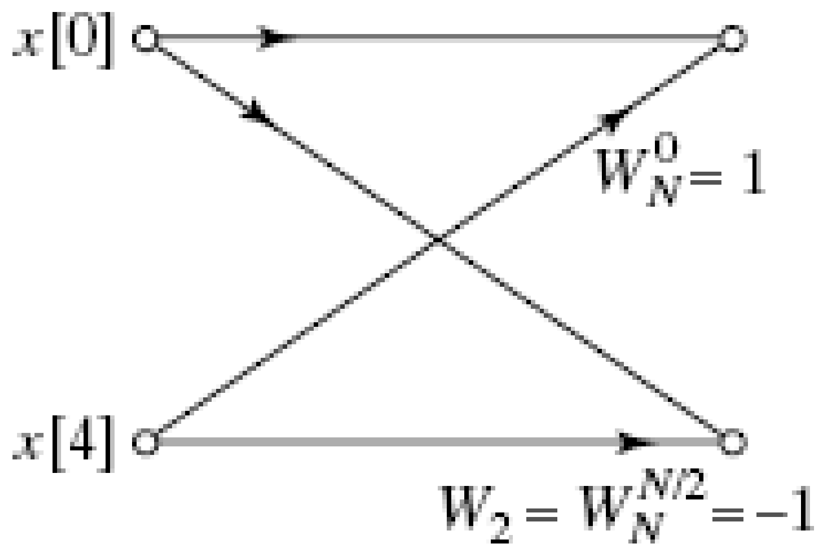
We can further decompose the (N/2)-point DFT into two (N/4)-point DFTs. For example, the upper half of the previous diagram can be decomposed as



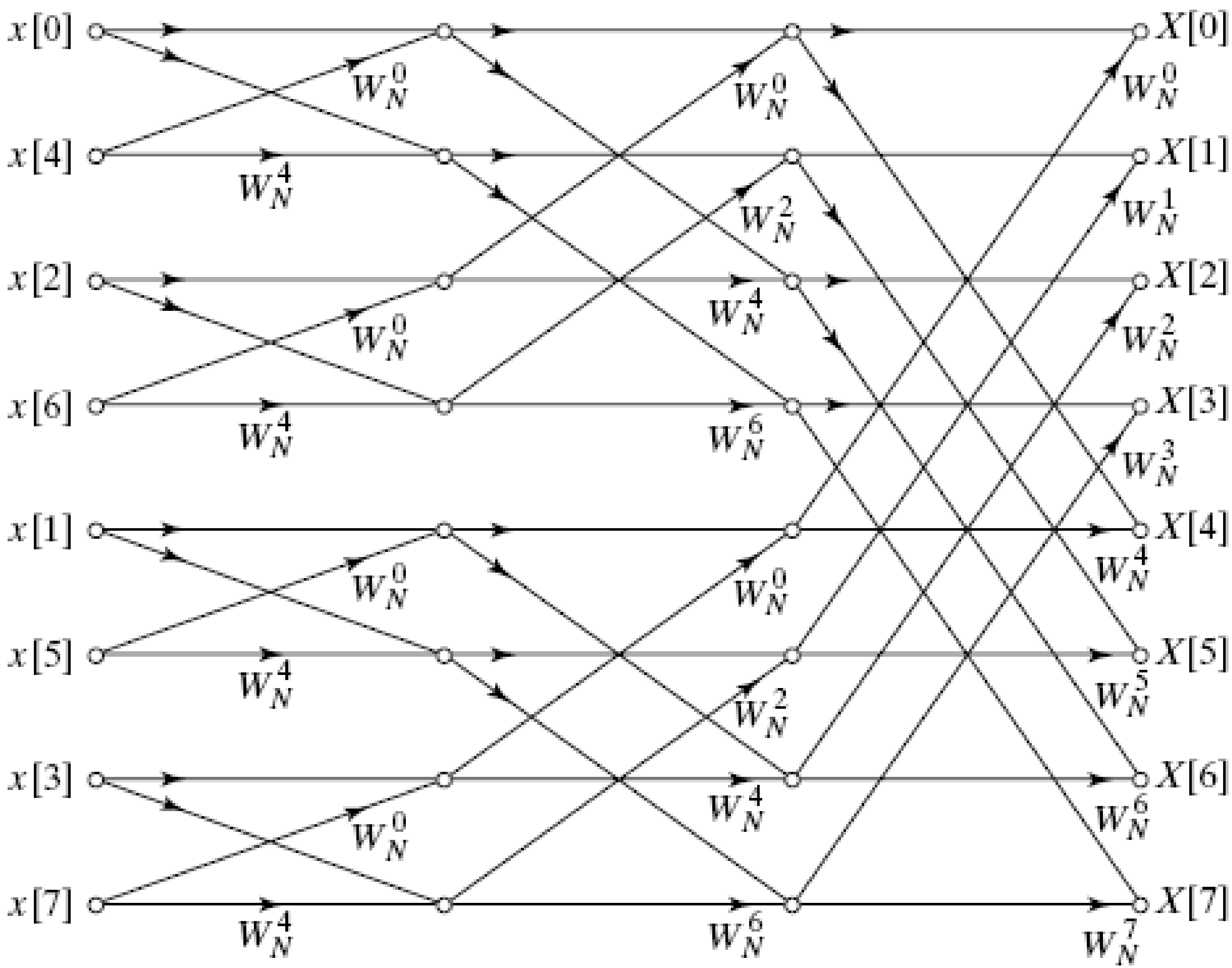
Hence, the 8-point DFT can be obtained by the following diagram with four 2-point DFTs.



Finally, each 2-point DFT can be implemented by the following signal-flow graph, where no multiplications are needed.



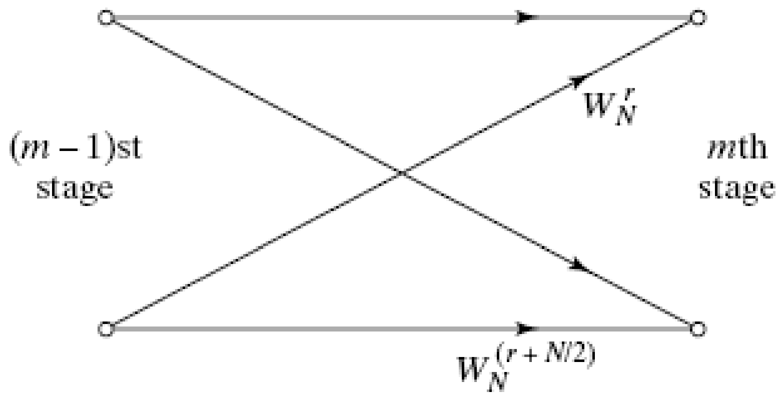
Flow graph of a 2-point DFT



Flow graph of complete decimation-in-time decomposition of an 8-point DFT.

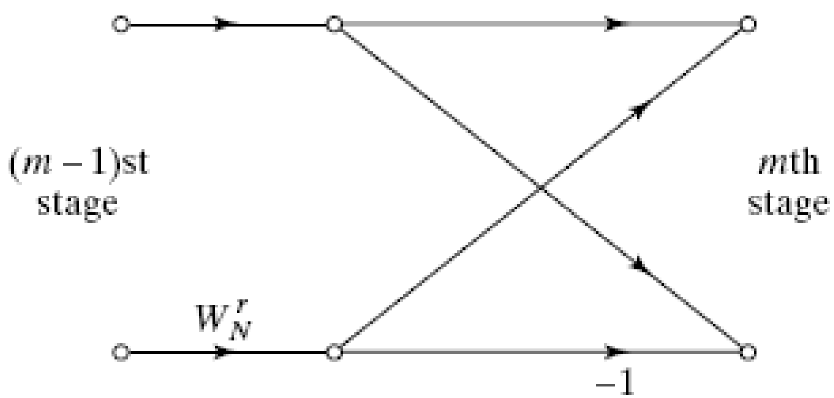
In each stage of the decimation-in-time FFT algorithm, there are a basic structure called the butterfly computation:

$$X_m[p] = X_{m-1}[p] + W_N^r X_{m-1}[q]$$
$$X_m[q] = X_{m-1}[p] - W_N^r X_{m-1}[q]$$

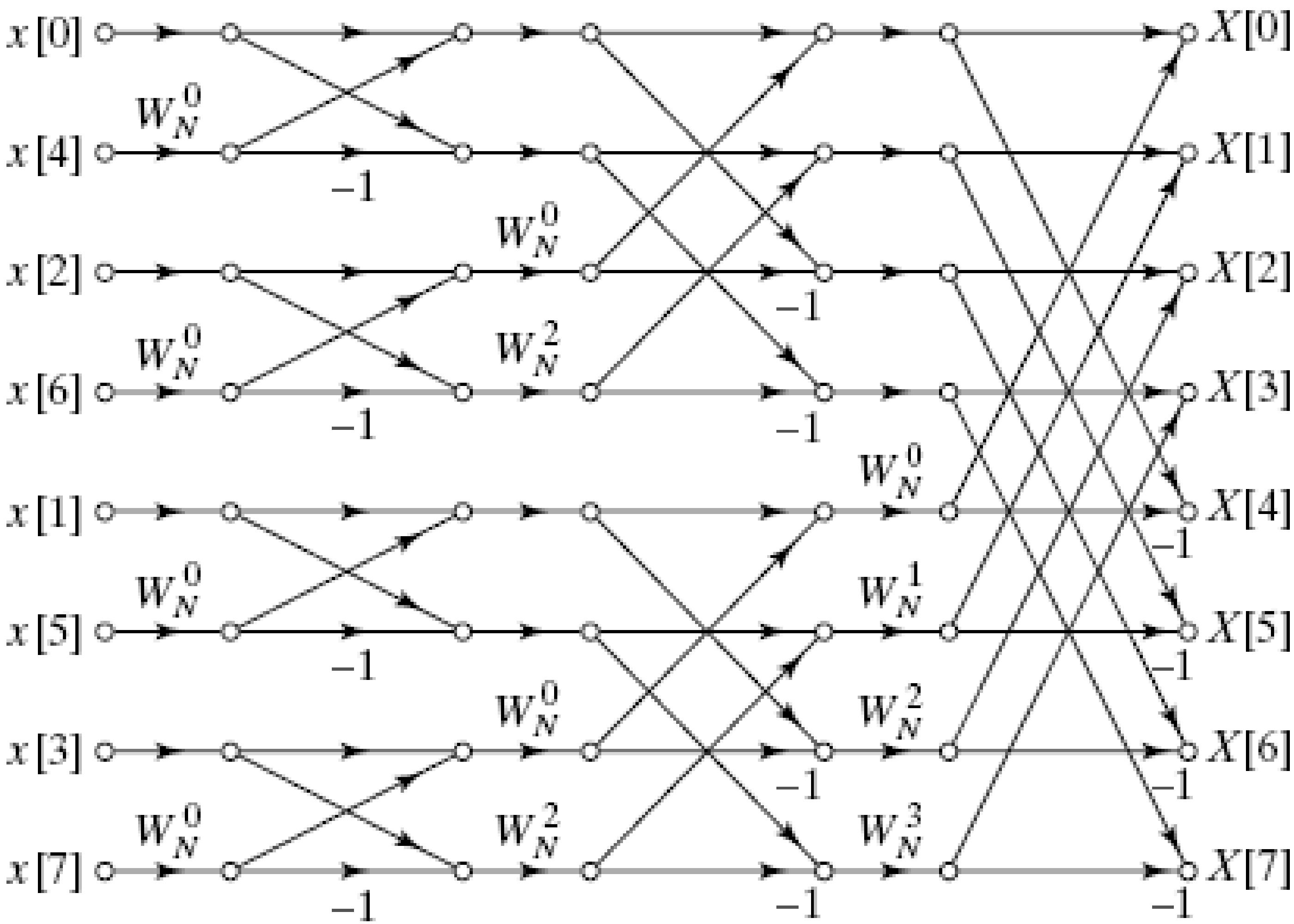


Flow graph of a basic butterfly computation in FFT.

The butterfly computation can be simplified as follows:



Simplified butterfly computation.



Flow graph of 8-point FFT using the simplified butterfly computation

- In the above, we have introduced the decimation-in-time algorithm of FFT.
- Here, we assume that N is the power of 2. For $N=2^v$, it requires $v=\log_2 N$ stages of computation.
- The number of complex multiplications and additions required was $N+N+\ldots N = Nv = N \log_2 N$.

- In practice, if N is not the power of 2, we can use zero padding to complement zeros to its rear to result in an length-M sequence, where M is the smallet poaer-of-2 integer that is larger than N. Then, we compute the M-point FFT instead.

- When N is not the power of 2, we can also apply the same principle that were applied in the power-of-2 case when N is a composite integer. For example, if N=RQ, it is possible to express an N-point DFT as either the sum of R Q-point DFTs or as the sum of Q R-point DFTs.
- The FFT algorithm of power-of-two is also called the Cooley-Tukey algorithm since it was first proposed by them.