Lecture 18 - FFT Algorithm (fast way to compute DFT)

Recall DFT_N

$$\{x[n]\}_{n=0}^{N-1} \xleftrightarrow{DFT} \{X[k]\}_{k=0}^{N-1}$$

$$X[k] = \sum_{n=0}^{N-1} x[n] \underbrace{e^{-j\frac{2\pi}{N}kn}}_{W_N}$$

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

$$= \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \underbrace{e^{-j2\pi kn}}_{1} = X[k]$$

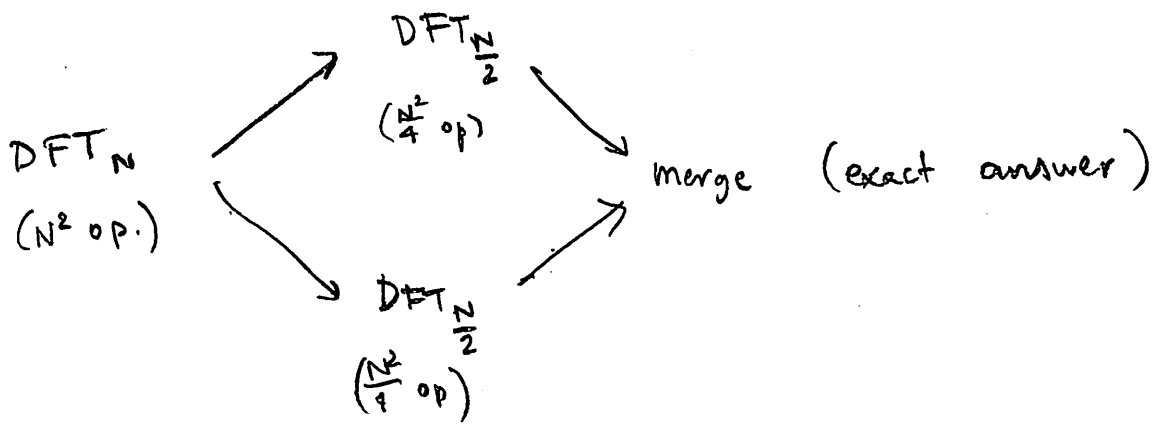
$\Rightarrow X[k]$ as N-periodic

How many operations do we need?

We need N^2 multiply-accumulate operations

→ bad scaling! Too slow.

Main idea of the FFT: divide and conquer!



FFT (radix-2, decimation in time)

DFT_N: $$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} = \sum_{n \text{ even}} + \sum_{n \text{ odd}}$$

assume N is even

$$= \sum_{m=0}^{\frac{N}{2}-1} \underbrace{x[2m]}_{a[m]} W_N^{k2m} + \sum_{m=0}^{\frac{N}{2}-1} \underbrace{x[2m+1]}_{b[m]} W_N^{k(2m+1)}$$

Observe: ① $$W_N^{2m} = e^{-j\frac{2\pi}{N}\cdot 2m} = e^{-j\frac{2\pi}{\frac{N}{2}}\cdot m} = W_{\frac{N}{2}}^{m}$$

② $$W_N^{2m+1} = W_N \cdot W_N^{2m} = W_N \cdot W_{\frac{N}{2}}^{m}$$

$$\Rightarrow X[k] = \underbrace{\sum_{m=0}^{\frac{N}{2}-1} a[m] W_{\frac{N}{2}}^{km}}_{A[k]} + W_N^k \cdot \underbrace{\sum_{m=0}^{\frac{N}{2}-1} b[m] W_{\frac{N}{2}}^{km}}_{B[k]}$$

For $k=0,\ldots,\frac{N}{2}-1$, this formula is fine: $X[k] = A[k] + W_N^k B[k]$

For other values, we use periodicity of DFT.

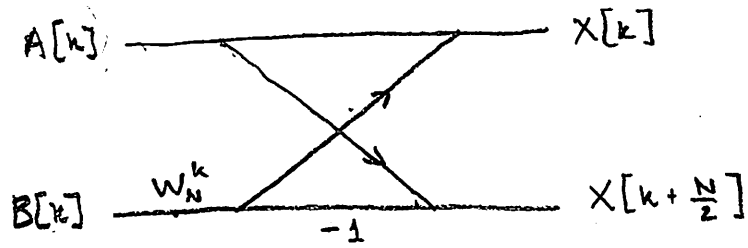
$$X[k+\tfrac{N}{2}] = A[k+\tfrac{N}{2}] + W_N^{k+\frac{N}{2}} B[k+\tfrac{N}{2}] = A[k] + W_N^k \cdot W_N^{N/2} B[k]$$

$$e^{-j\frac{2\pi}{N}\cdot\frac{N}{2}} = -1$$

For $k = 0, \ldots, \frac{N}{2} - 1$

$$X[k] = A[k] + W_N^k B[k]$$

$$X[k + \tfrac{N}{2}] = A[k] - W_N^k B[k]$$



butterfly operation

This holds as long as N is even.

$\Rightarrow$ We choose $N = 2^v$ so that we can apply this recursively

$$\{A[k]\}_{n=0}^{\frac{N}{2}-1} = DFT_{\frac{N}{2}}\{x[0], x[2], \ldots, x[N-2]\}$$

recurse ↓

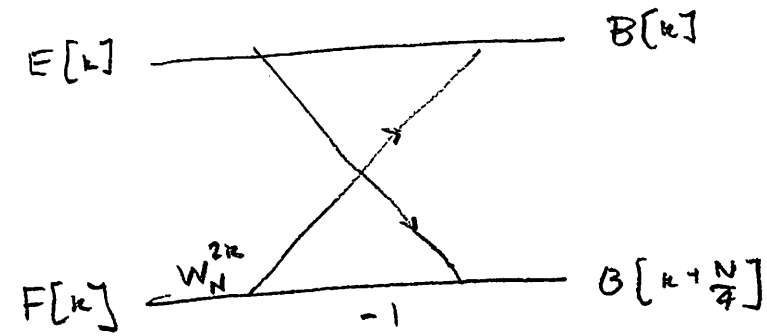
$$A[k] = C[k] + W_{\frac{N}{2}}^k D[k]$$

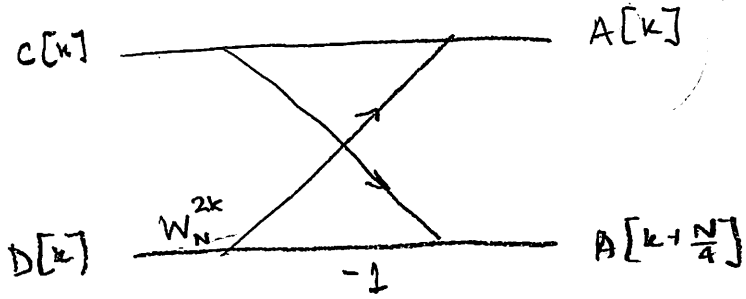
$$A[k + \tfrac{N}{4}] = C[k] - W_{\frac{N}{2}}^k D[k]$$

$$= W_N^{2k}$$

$$\{C[k]\}_{n=0}^{\frac{N}{4}} = DFT_{\frac{N}{4}}\{x[0], x[4], \ldots, x[N-4]\}$$

$$\{D[k]\}_{n=0}^{\frac{N}{4}} = DFT_{\frac{N}{4}}\{x[2], x[6], \ldots, x[N-2]\}$$





Ex: $N = 8$

Number of operations?

For $N = 2^{\upsilon}$, we have $\upsilon = \log_2 N$ stages

N multiply-accumulate operations per stage

$\Rightarrow N \log_2 N$ operations!

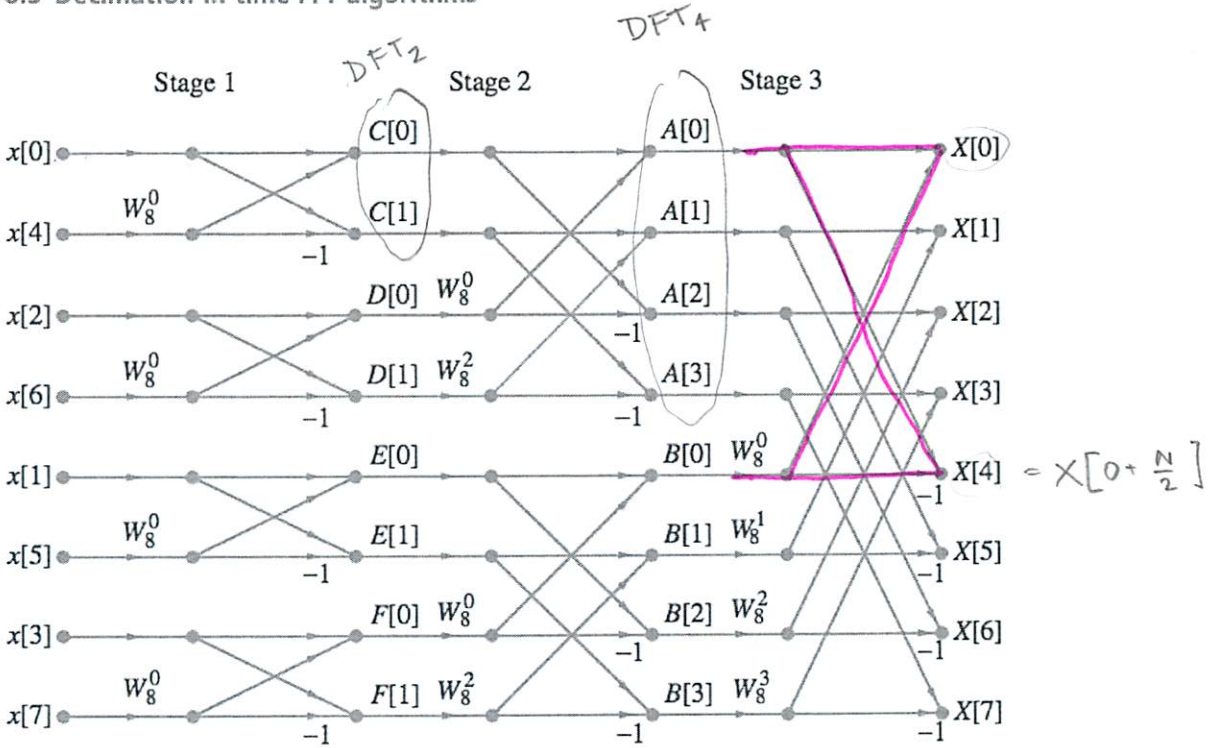


Figure 8.6 Flow graph of 8-point decimation-in-time FFT algorithm using the butterfly computation shown in Figure 8.4. The trivial twiddle factor $W_8^0 = 1$ is shown for the sake of generality.

Observations:

- Decimation in time: separating even and odd time indices (decimation in frequency separates even/odd frequency indices)
- Radix-2: we break DFT_N into two $DFT_{\frac{N}{2}}$
- inverse FFT can be derived in similar way

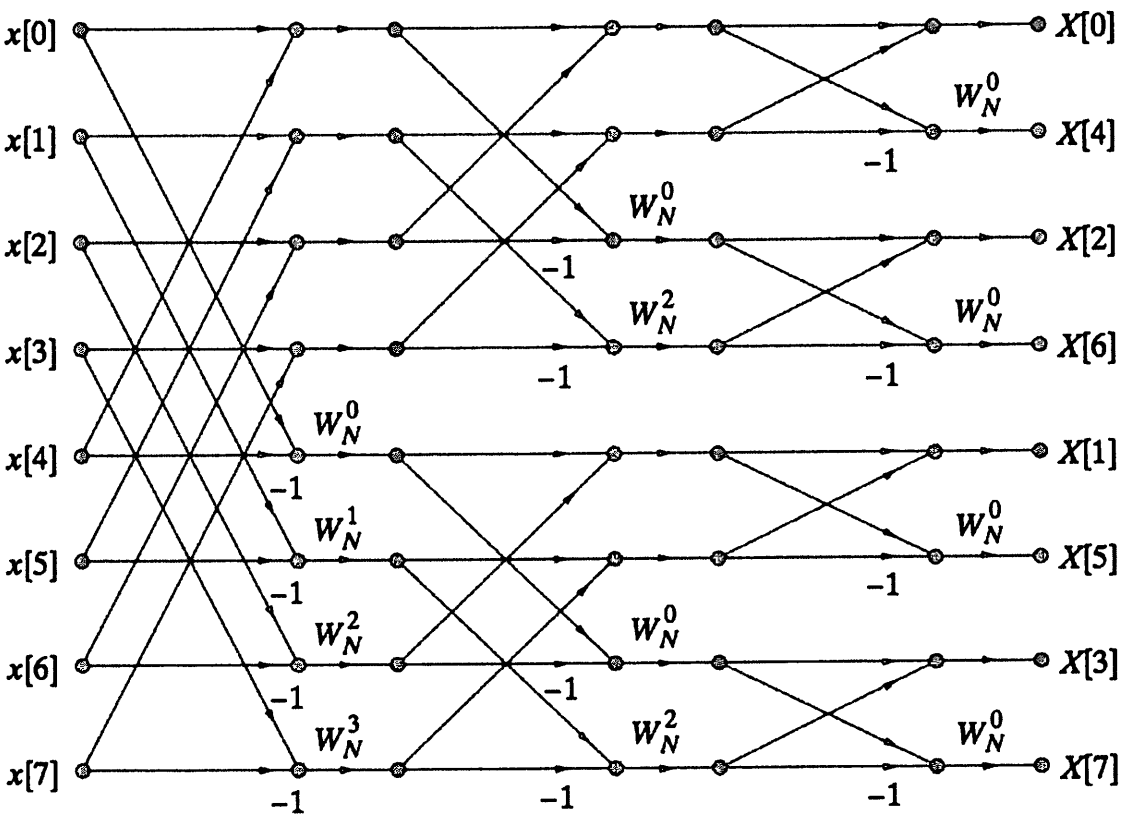


Figure 8.13 Flow graph for the decimation-in-frequency 8-point FFT algorithm. The input sequence is in natural order and the output sequence in bit-reversed order.

Fast convolution using FFT

Say we want to compute $\{x[n]\}_{n=0}^{L-1} * \{h[n]\}_{n=0}^{M-1}$ (Result should have length $L+M-1$)

Zero-pad to length N

$\{x[n]\}_{n=0}^{N-1}$ (L=1) $\xrightarrow{\text{DFT (FFT)}}$ $\{X[k]\}_{n=0}^{N-1}$

$\{h[n]\}_{n=0}^{N-1}$ (M=1) $\xrightarrow{\text{DFT (FFT)}}$ $\{H[k]\}_{n=0}^{N-1}$

$\{X[k]\}$, $\{H[k]\}$ → multiply

$\{Y[k] = H[k]X[k]\}_{n=0}^{N-1}$ $\xleftarrow{\text{IDFT (IFFT)}}$?

DFT property: $x[n] \circledast_N h[n] \overset{\text{DFT}}{\longleftrightarrow} X[k]H[k]$

$$\sum_{m=0}^{N-1} x[m]\, h[\langle n-m \rangle_N]$$

→ this is not what we wanted!

How can we get (linear) convolution from circular convolution?