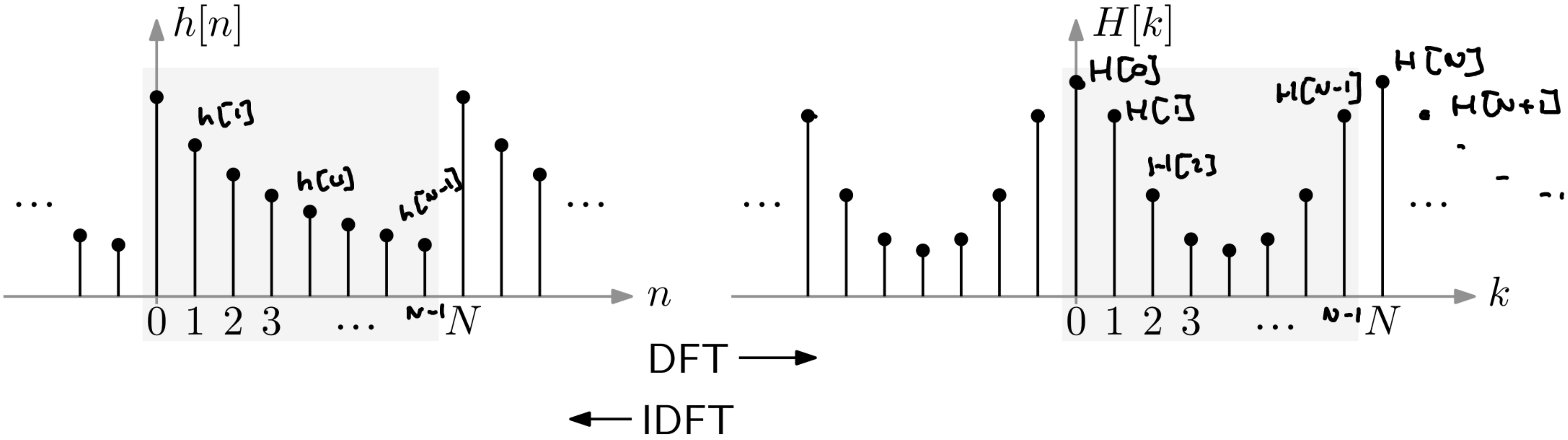
Fast Fourier transform (FFT)

And examples of how to compute things quickly

Herman Kamper



h[n]
h[1]
h[4]
h[N-1]
n
0 1 2 3 … N-1 N
H[k]
H[0]
H[1]
H[2]
H[N-1]
H[N]
H[N+1]
k
0 1 2 3 … N-1 N
DFT
IDFT

Fast Fourier transform (FFT)

N-point DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

$X[0] = \ldots$
$X[1] = \ldots$
$X[2] = \ldots$
$X[3] = x[0] \cdot e^{-j2\pi 3 \cdot 0/N}$
$+ x[1] \cdot e^{-j2\pi 3 \cdot 1/N}$
$+ \ldots$
$+ x[N-1] \cdot e^{-j2\pi 3 \cdot (N-1)/N}$
$X[4] = \ldots$
$\vdots$
$X[N-1] = \ldots$

Complex multiplications:

- for each k in $X[k]$: N complex mults
- for all N values of $X[k]$: N^2 complex mults

radix-2 FFT: Require N to be power of 2

The FFT and DFT lead to precisely the same result. They differ only in the way that they are calculated.

Decimation in time

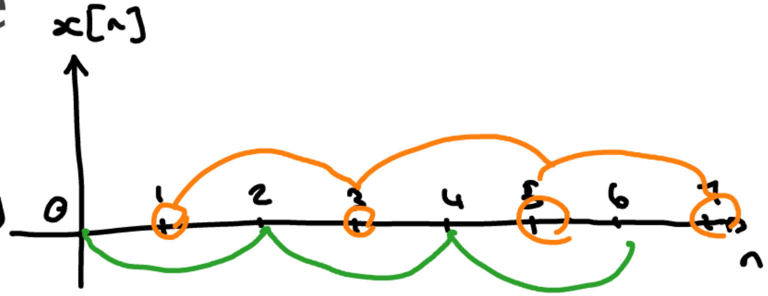
$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

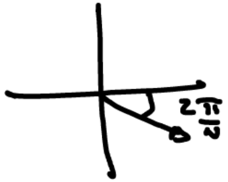
$$= \sum_{n=0}^{N/2-1} x[2n] \cdot e^{-j\frac{2\pi k(2n)}{N}} + \sum_{n=0}^{N/2-1} x[2n+1] \cdot e^{-j\frac{2\pi k(2n+1)}{N}}$$

$$= \sum_{n=0}^{N/2-1} x[2n] \cdot e^{-j\frac{2\pi k}{N/2} n} + e^{-j\frac{2\pi k}{N}} \cdot \sum_{n=0}^{N/2-1} x[2n+1] \cdot e^{-j\frac{2\pi k n}{N/2}}$$

$$= X_1[k] + e^{-j\frac{2\pi k}{N}} \cdot X_2[k]$$

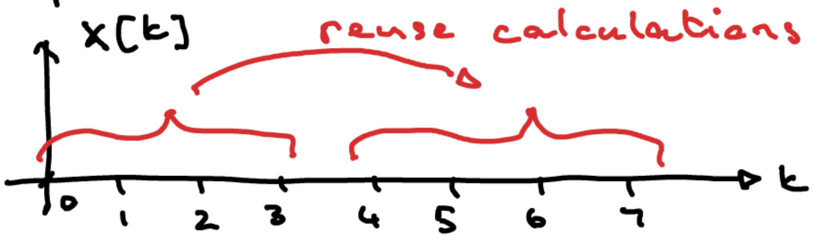
$$= X_1[k] + W_N^k \cdot X_2[k]$$

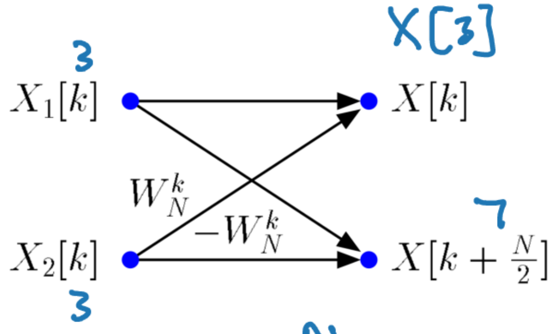




$$\boxed{W_N = e^{-j\frac{2\pi}{N}}}$$

Split $X[k]$ calculation in two:





$$X[k] = X_1[k] + W_N^k X_2[k] \quad \text{for } k=0,1,\dots,\tfrac{N}{2}-1$$

$\frac{N}{2}$-point DFTs

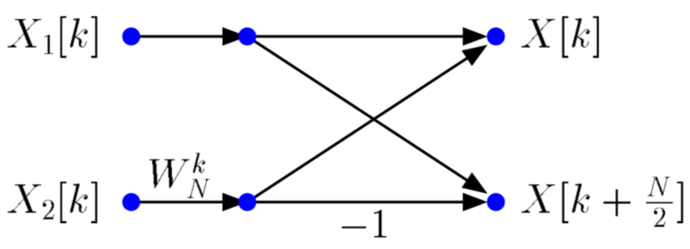
$$X[k+\tfrac{N}{2}] = X_1[k+\tfrac{N}{2}] + W_N^{k+\frac{N}{2}} X_2[k+\tfrac{N}{2}] \quad \text{for } k=0,1,\dots,\tfrac{N}{2}-1$$

$$= X_1[k] + W_N^{k+\frac{N}{2}} X_2[k]$$

$$(e^{-j\frac{2\pi}{\cancel{N}}})^{\cancel{N}/2} = e^{-j\pi} = -1$$

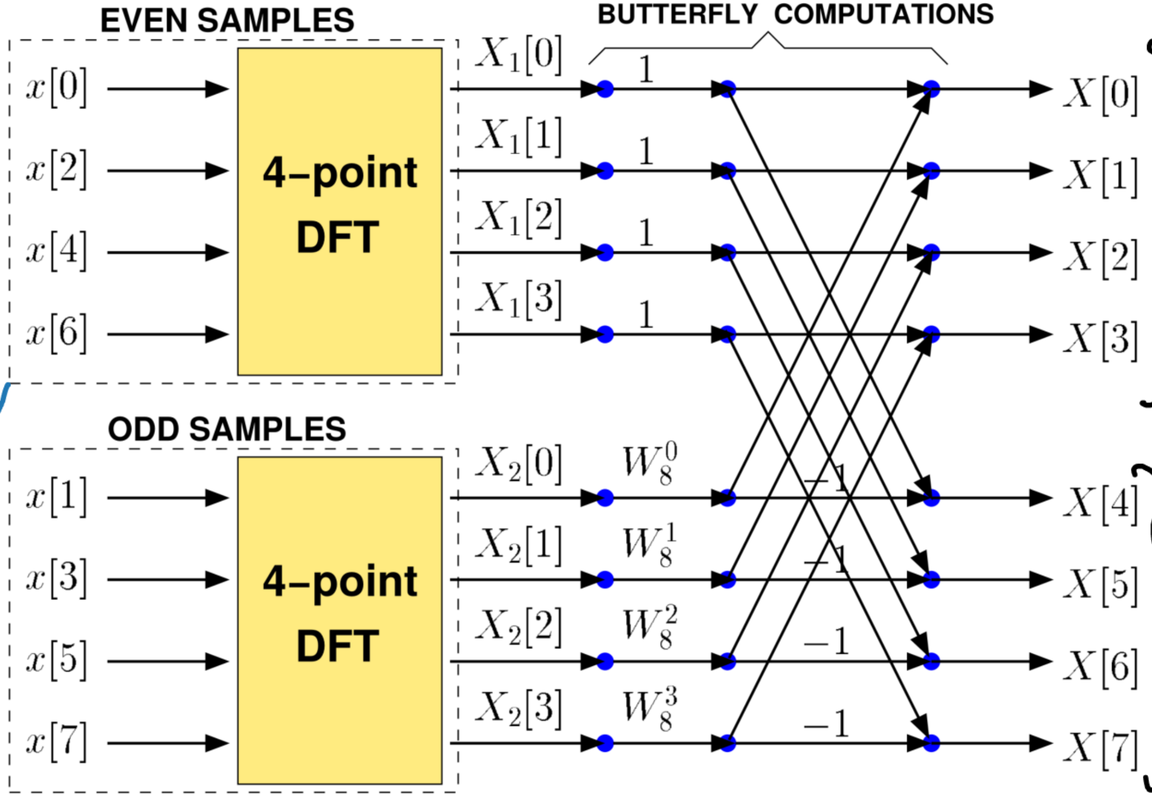
$$= X_1[k] + W_N^k \cdot W_N^{N/2} X_2[k]$$

$$= \underbrace{X_1[k]}_{(\frac{N}{2})^2} - W_N^k \underbrace{X_2[k]}_{(\frac{N}{2})^2}$$



Complex mults: $2\times\left(\frac{N}{2}\right)^2 = \frac{N^2}{2}$

Eight-point DFT

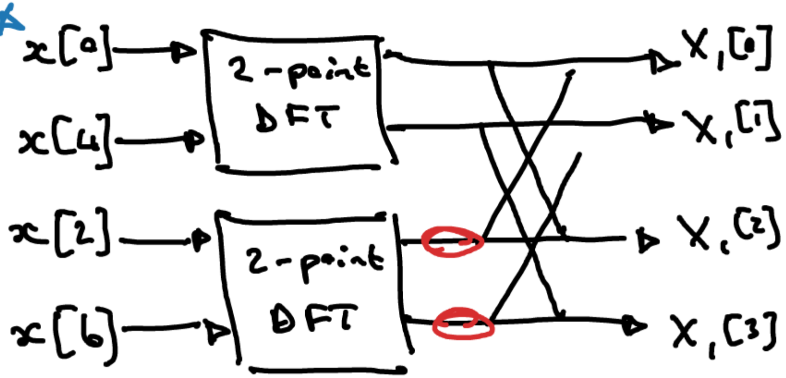


$X[k] = X_1[k] + W_N^k X_2[k]$

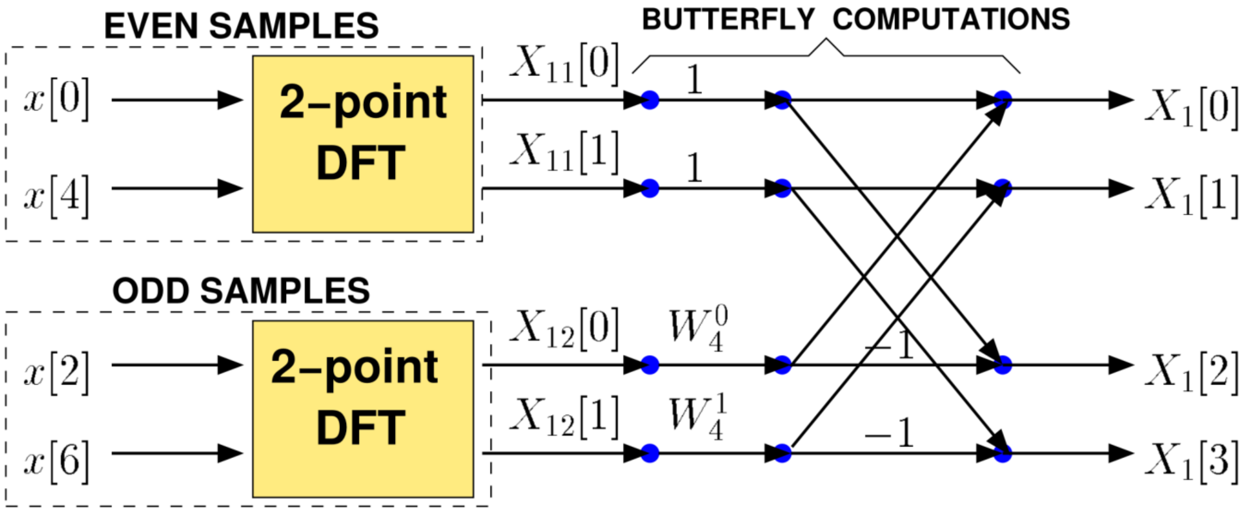
for $k = 0, 1, \dots, \frac{N}{2} - 1$

$X[k + \frac{N}{2}] = X_1[k] - W_N^k X_2[k]$

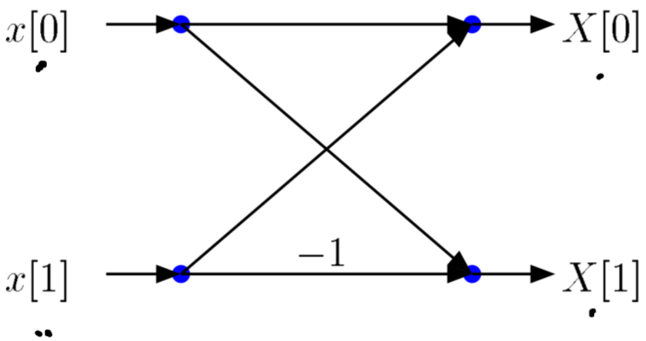
for $k = 0, 1, \dots, \frac{N}{2} - 1$



Four-point DFT



Two-point DFT



$$X[k] = \sum_{n=0}^{1} x[n] \cdot e^{-j \cancel{2}\pi k n / \cancel{2}}$$

$$= x[0] \cdot e^{0} + x[1] \cdot e^{-j\pi k}$$

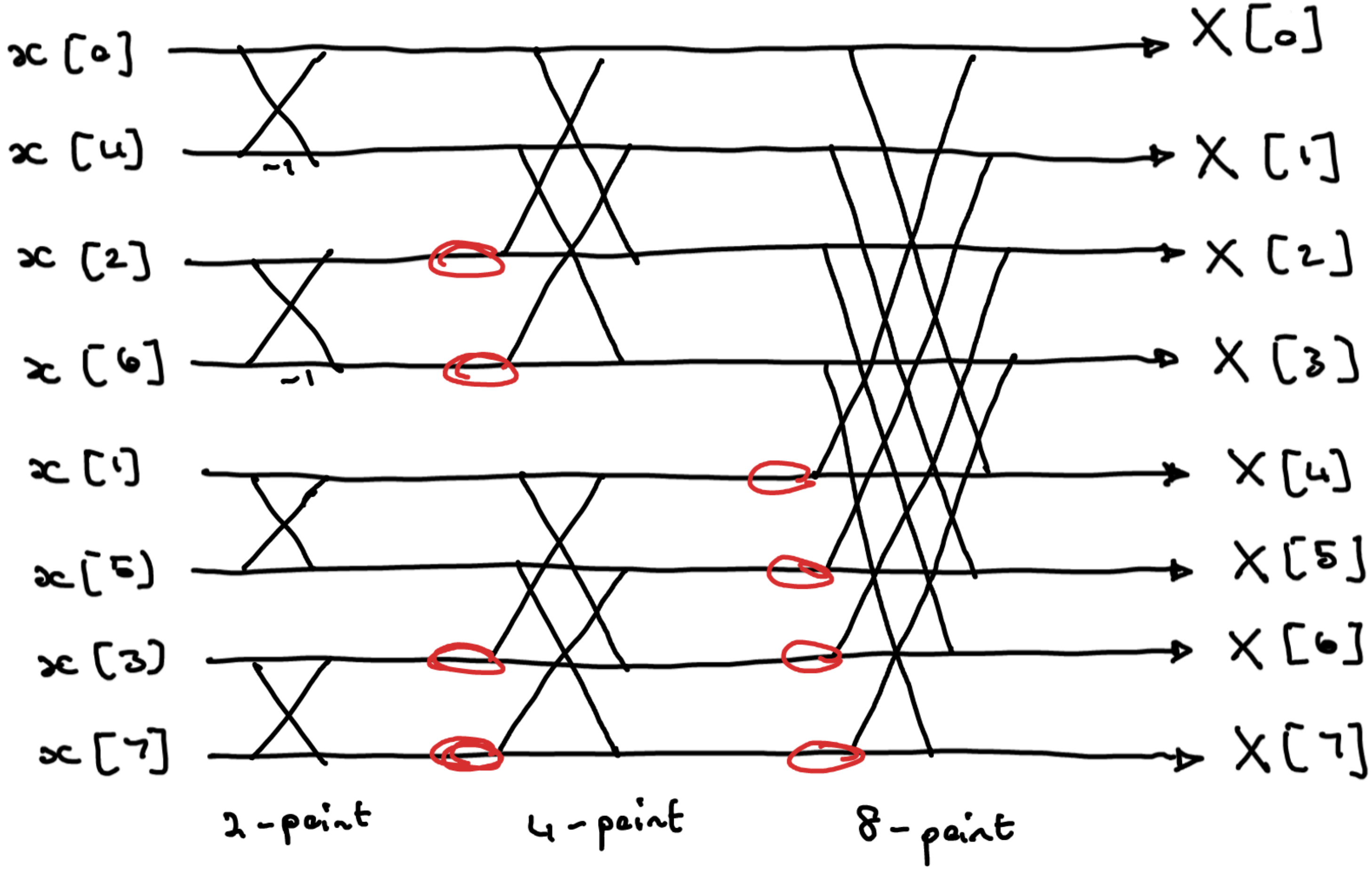
$$X[0] = x[0] + x[1]$$

$$X[1] = x[0] - x[1]$$

$2^{\text{layers}} = N$

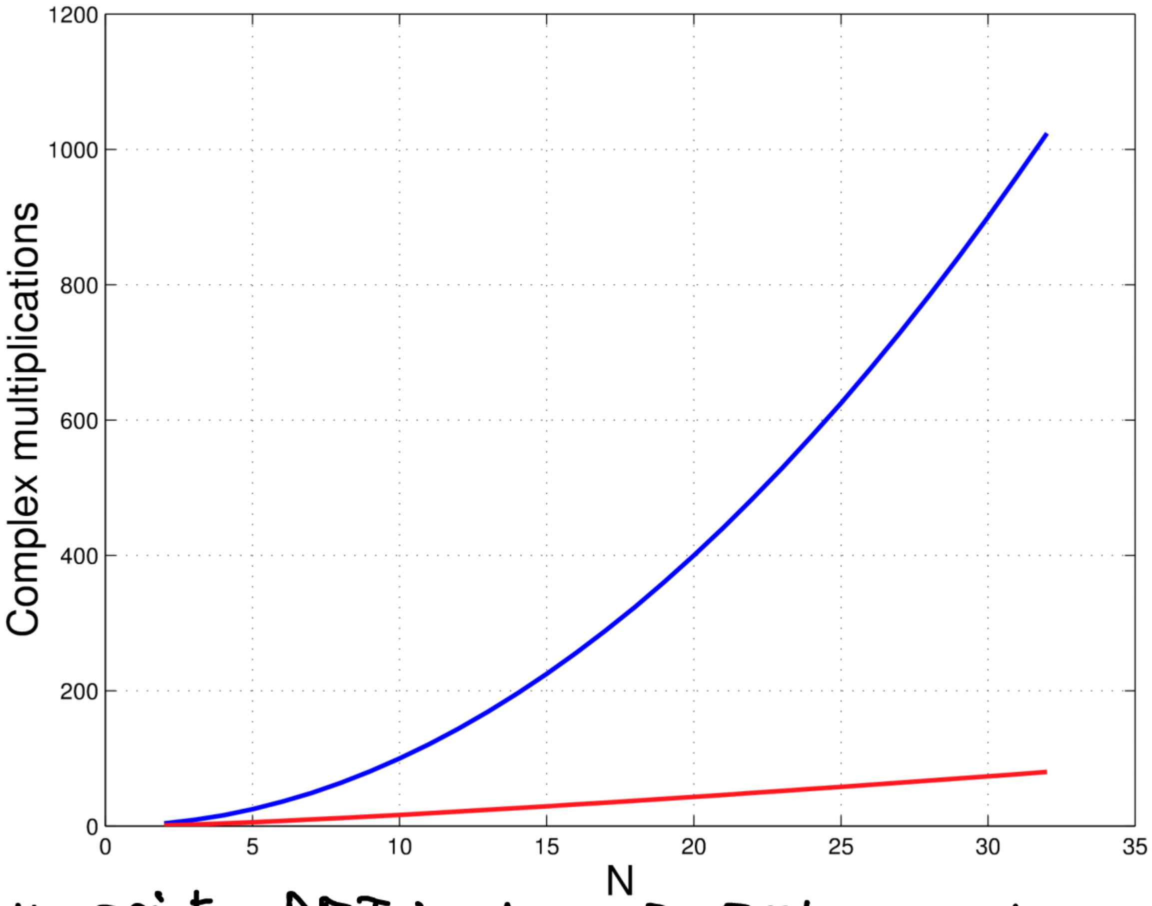
$\text{layers} = \log_2 N$

$\frac{N}{2}$ complex mults per layer



x[2,7]

Computational complexity



N-point DFT:

- $\log_2 N$ layers
- $\frac{N}{2}$ complex mult per layer

$\therefore \frac{N}{2}\log_2 N$ complex mults for FFT

vs

N^2 for direct DFT

1024-point DFT: 1 048 576 complex mults vs. 5 120 for FFT

In-place computation of the FFT

Decimal index	Binary index	Bit-reversed index
0: $x[0]$	$x[000]$	000
1: $x[4]$	$x[100]$	001
2: $x[2]$	$x[010]$	010
3: $x[6]$	$x[110]$	011
4: $x[1]$	$x[001]$	100
5: $x[5]$	$x[101]$	101
6: $x[3]$	$x[011]$	110
$x[7]$	$x[111]$	111

$x[3]$

↓

011

↓

110

↓

6

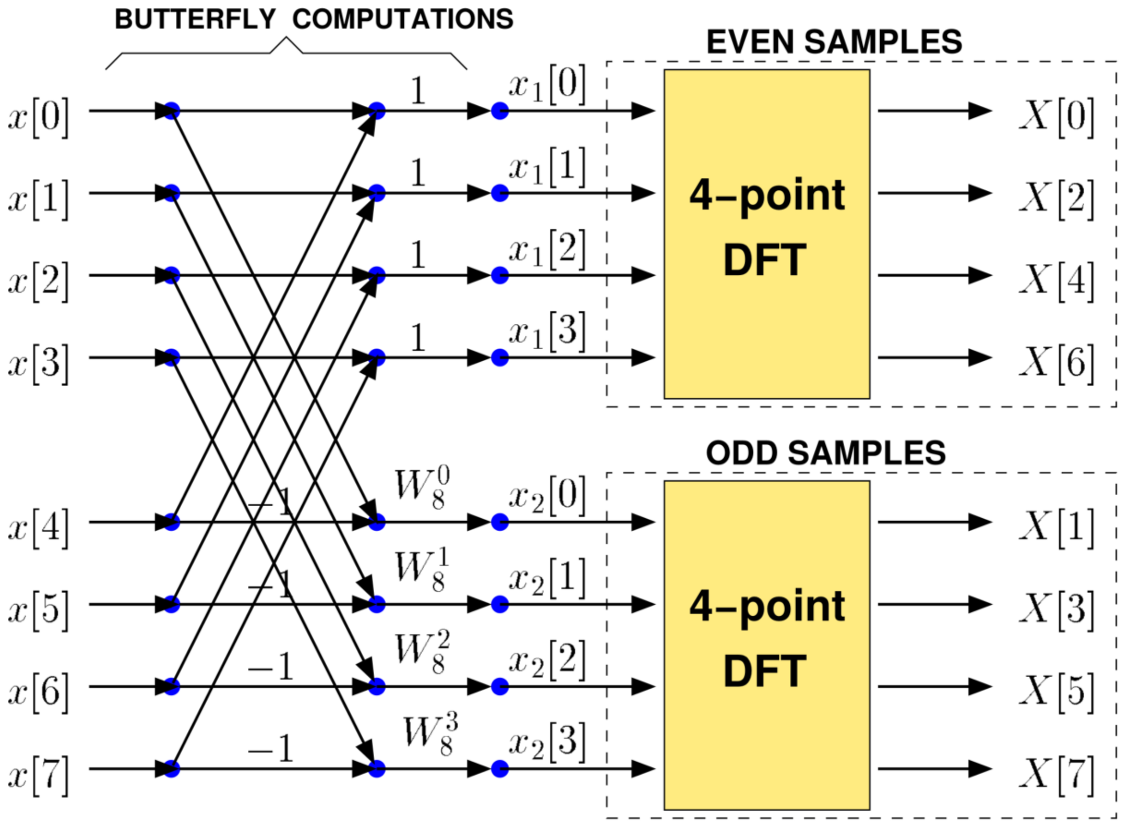
$x[217] = ?$

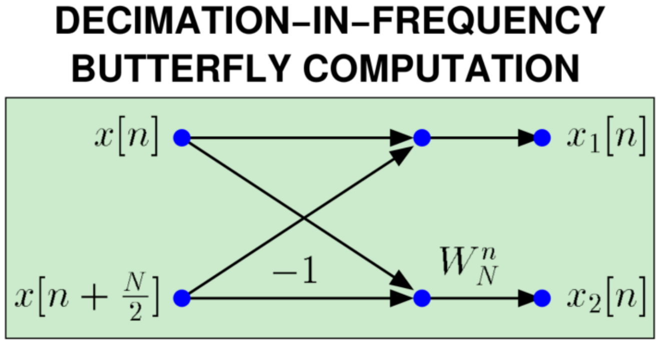
Decimation in frequency

Split $X[k]$ into even and odd sequences:

$X[2k]$ for $k = 0, 1, \ldots, \frac{N}{2} - 1$

$X[2k+1]$ for $k = 0, 1, \ldots, \frac{N}{2} - 1$





Summary

Decimation in time:

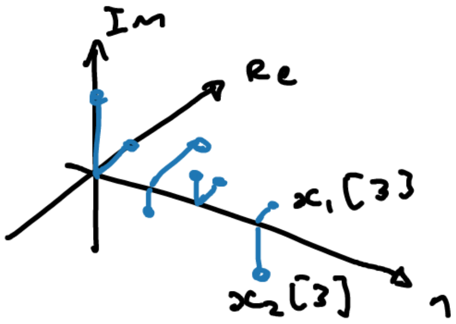
- Split $x[n]$ into even and odd parts
- Helpers: $X_1[k]$ and $X_2[k]$
- Input: Bit-reversed ; Output: Nice

Decimation in frequency:

- Split $X[k]$ into even and odd parts
- Helpers: $x_1[n]$ and $x_2[n]$
- Input: Nice ; Output: Bit-reversed

DFT of two real sequences

$x_1[n]$ and $x_2[n]$ are real



$$x[n] = x_1[n] + j\,x_2[n]$$

$\big\downarrow$ DFT

$$X[k] = X_1[k] + j\,X_2[k]$$

$$x_1[n] = \frac{x[n] + x^*[n]}{2} \qquad x_2[n] = \frac{x[n] - j x^*[n]}{2j}$$

$\big\downarrow$ DFT

$$X_1[k] = \frac{1}{2}X[k] + \frac{1}{2}X^*[N-k] \qquad X_2[k] = \frac{1}{2j}X[k] - \frac{1}{2j}X^*[N-k]$$

DFT of one real sequence

Real sequence $g[n]$ has $2N$ points

Let $x_1[n] = g[2n]$ and $x_2[n] = g[2n+1]$ for $n = 0, 1, \ldots, N-1$

$x[n] = x_1[n] + j x_2[n]$ $X[k]$

$$G[k] = \sum_{n=0}^{2N-1} g[n] \cdot e^{-j2\pi k/2N}$$

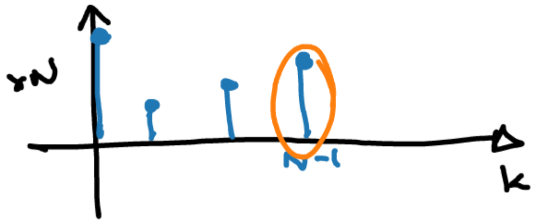
$$= \sum_{n=0}^{N-1} g[2n] \cdot e^{-j2\pi nk/N} + e^{-j\pi k/N} \sum_{n=0}^{N-1} g[2n+1] \cdot e^{-j2\pi nk/N}$$

$$= X_1[k] + e^{-j\pi k/N} \cdot X_2[k]$$

Calculating IFFTs using FFTs

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

- $\text{FFT}\{X[n]\} = N \cdot x[N-k]$



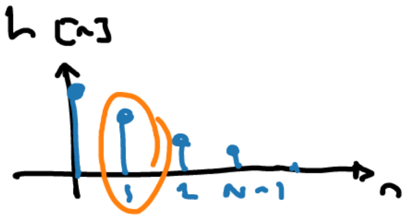
Symmetry for DFT:

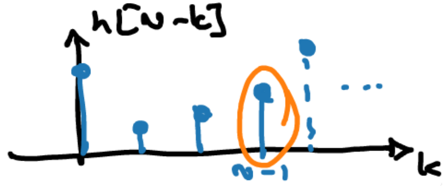
if $\text{DFT}\{h[n]\} = H[k]$

then

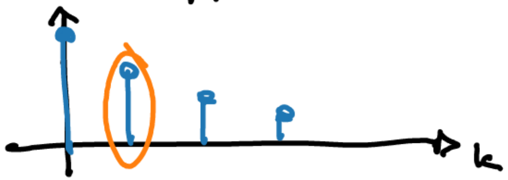
$\text{DFT}\{H[n]\} = N \cdot h[-k]$

$= N \cdot h[N-k]$





- Swap the indices: what happens at N−k now happens at k



- k becomes n
- Divide by N

$\boxed{\text{Another summary}}$

Can now quickly (fast) do:

1. DFT using FFT
2. DFT of real sequence using FFT
3. IDFT using FFT

What to take away

- Some technical tricks
- Divide and conquer: Store intermediate results and use them later
- You need to understand implications of the tricks:
    - When can you (not) use in-place computations?
    - Why radix-2?
    - Why are things slower with `np.fft.fft` when the input is not radix-2?