The Fast Fourier Transform (FFT) Algorithm

Numerically efficient method to calculate DFT

- Gauss 1805, Cooley + Tukey 1965

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, \ldots N-1$$

N complex mults, N-1 complex adds } for each k

$O(N^2)$ computations for direct DFT

vs $O(N \log_2 N)$ for FFT

N	1000	10^6	10^9
N^2	10^6	10^{12}	10^{18}
$N \log_2 N$	10^4	20×10^6	30×10^9

10^{18} ns ~ 31.2 years

30×10^9 ns ~ 30 sec

FFT exploits symmetries of $e^{-j\frac{2\pi}{N}kn}$

Define $W_N = e^{-j2\pi/N}$

1) Complex conjugate symmetry $W_N^{k(N-n)} = W_N^{-kn} = (W_N^{kn})^*$

2) Periodicity in n, k $W_N^{kn} = W_N^{k(N+n)} = W_N^{(k+N)n}$

Decimation in Time FFT (one of many)

- build a big DFT from smaller ones
- Assume $N = 2^m$

separate $x[n]$ into even and odd-indexed sub sequences

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

even, odd indices $\}$ $n = 2r$, $n = 2r+1$; $r = 0, 1, 2, \ldots \frac{N}{2} - 1$

$$X[k] = \sum_{r=0}^{\frac{N}{2}-1} x[2r]\, W_N^{k2r} + \sum_{r=0}^{\frac{N}{2}-1} x[2r+1]\, W_N^{(2r+1)k}$$

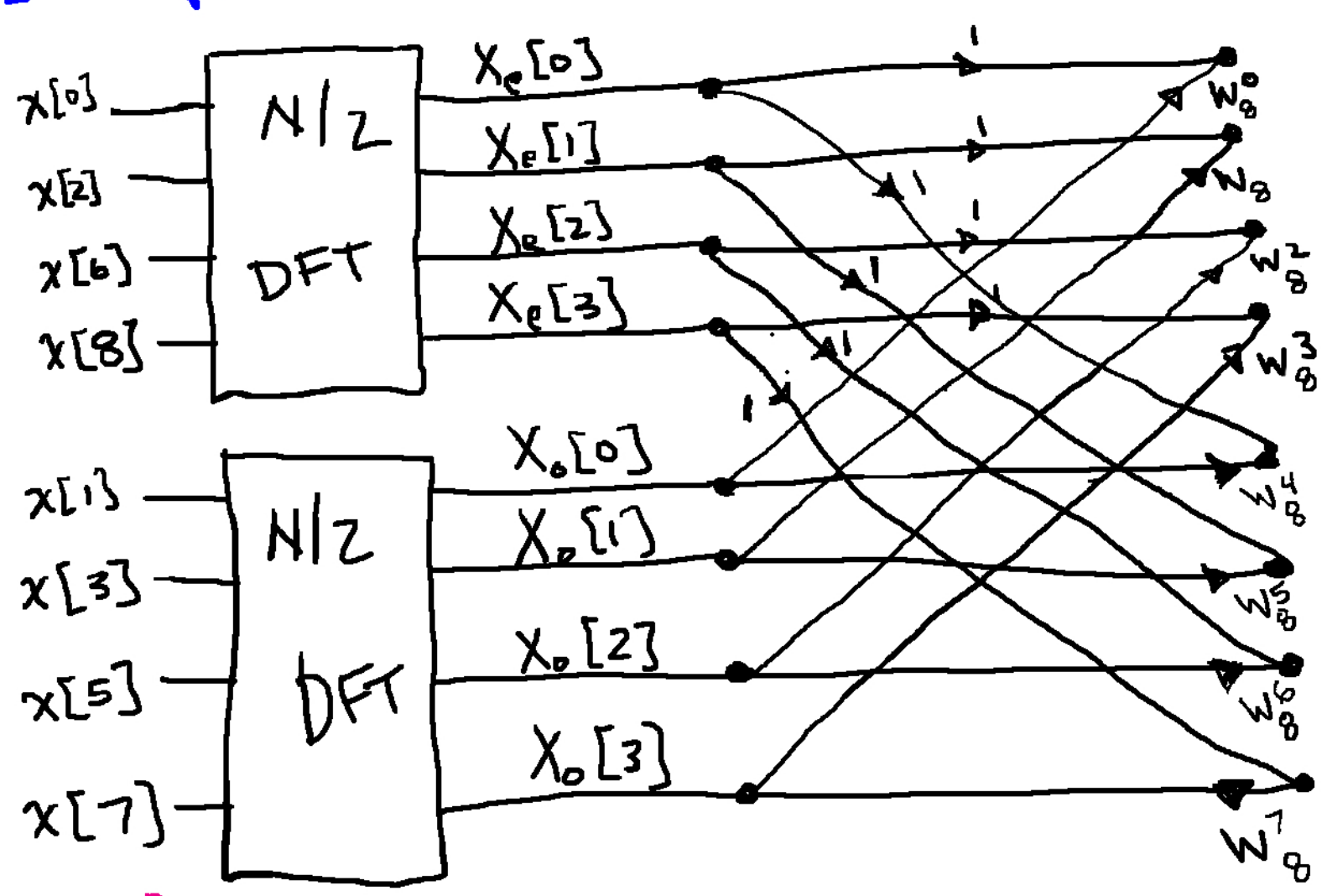
$$= \sum_{r=0}^{\frac{N}{2}-1} x[2r]\, (W_N^2)^{kr} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1]\, (W_N^2)^{kr}$$

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

$$X[k] = \underbrace{\sum_{r=0}^{N/2-1} x[2r]\, W_{N/2}^{kr}}_{\substack{N/2 \text{ DFT of even samples} \\ X_e[k]}} + W_N^k \underbrace{\sum_{r=0}^{N/2-1} x[2r+1]\, W_{N/2}^{kr}}_{\substack{N/2 \text{ DFT of odd samples} \\ X_o[k]}}$$

$X[k] = X_e[k] + W_N^k X_o[k]$ — sum of 2 $N/2$ point DFTs

Example: N = 8



$\left(\frac{N}{2}\right)^2 \cdot 2$ $+ N \approx \frac{N^2}{2} + N$ multipliers

Keep splitting: each $\frac{N}{2}$ pt $\rightarrow$ 2 $\frac{N}{4}$ pt DFTs

How many times? $N/2, \frac{N}{4}, \cdots \frac{N}{2^{m-1}}, \frac{N}{2^m} = 1$ $p = \log_2 N$ times

Cost

1: $\frac{N}{2} \rightarrow 2\left(\frac{N}{2}\right)^2 + N = \frac{N^2}{2} + N$

2: $N/4 \rightarrow 2\left(2\left(\frac{N}{4}\right)^2 + \frac{N}{2}\right) + N = \frac{N^2}{4} + 2N$

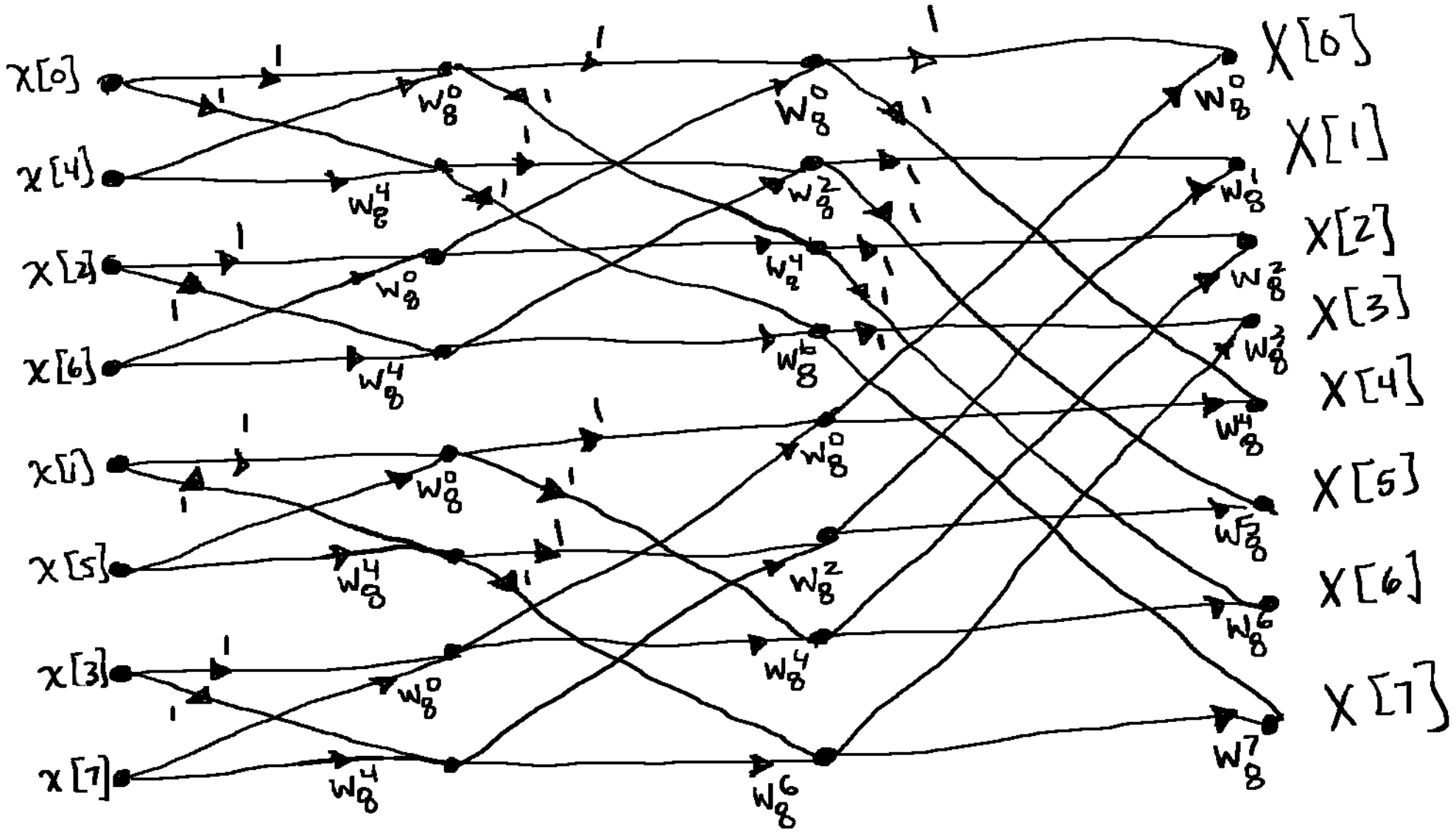
3: $N/8 \rightarrow 2\left[2\left(2\left(\frac{N}{8}\right)^2 + \frac{N}{4}\right) + \frac{N}{2}\right] + N = \frac{N^2}{8} + 3N$

$\vdots$

p: $\frac{N}{2^p} = 1 \rightarrow \frac{N^2}{2^p} + pN = \frac{N^2}{N} + N\log_2 N$

$\approx O(N \log_2 N)$ for N large

Example: $N = 8 = 2^3$ $(p = 3)$



Example: $N = 8 = 2^3$