COMP ENG 4TL4:

Digital Signal Processing

Notes for Lecture #18

Friday, October 17, 2003

5.6 The Fast Fourier Transform (FFT)

Let us now introduce the following new notation:

$$X_N[k] = X[k]\,,\ k = 0, \ldots, N-1\,, \qquad W_N = e^{-j\frac{2\pi}{N}}.$$

Before, we used simpler notation $X[k]$ and W (without the subscript $_N$) because the sequence length was not important. Now, it is important!

Assume that:

$$N = 2^r\,, \quad r = \log_2 N.$$

i.e., that the transformed sequence length is an integer power of 2.

If the original sequence length does not satisfy this assumption, it can always be padded with the necessary number of zeros so that the length of the zero-padded sequence will satisfy this assumption.

The idea behind the FFT:

$$X_N[k] = \sum_{n=0}^{N-1} x[n]\, W_N^{kn} \qquad \longleftarrow \quad N\text{–point DFT}$$

$$= \sum_{n \text{ even}} x[n]\, W_N^{kn} + \sum_{n \text{ odd}} x[n]\, W_N^{kn}$$

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

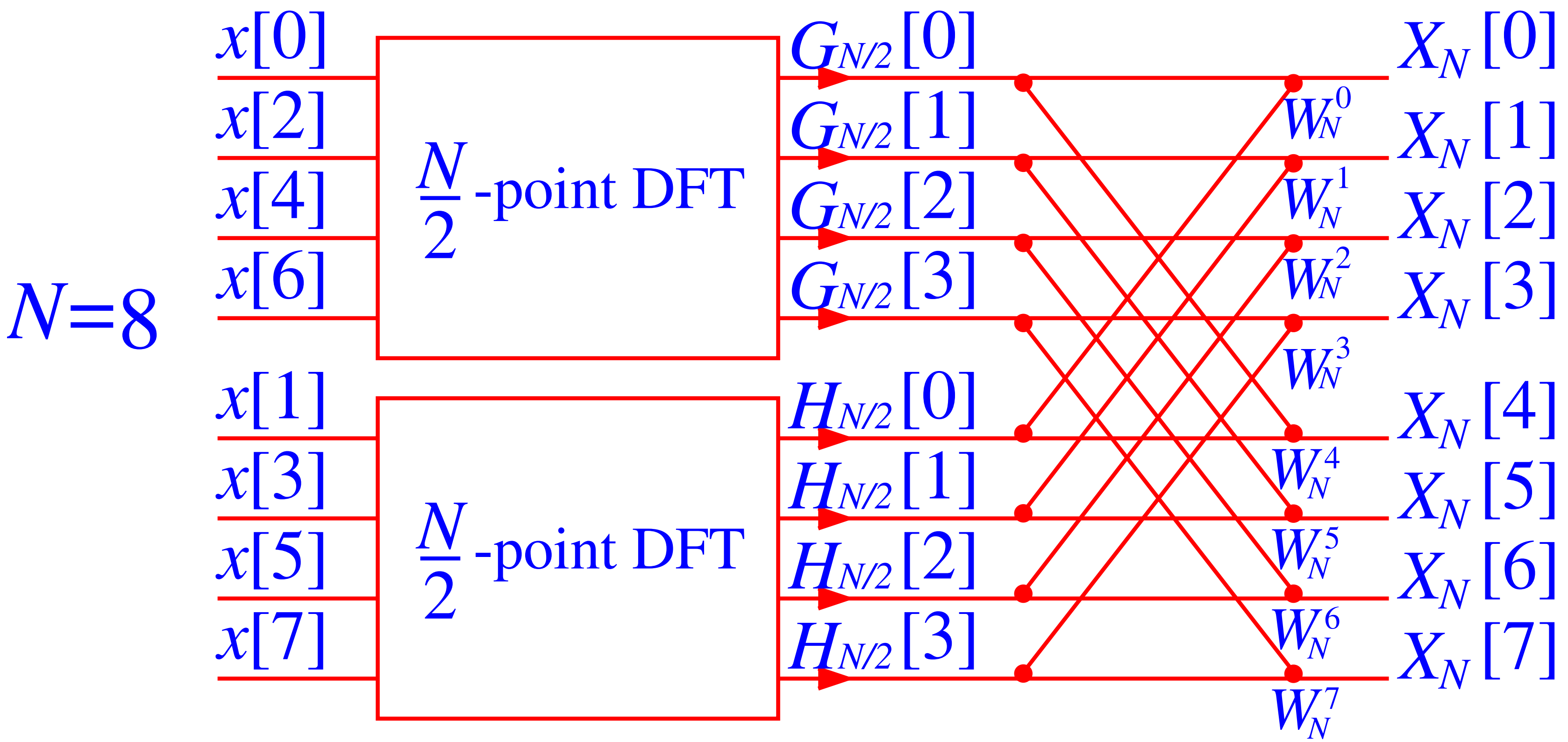
$$= \sum_{l=0}^{N/2-1} x[2l][\underbrace{W_N^2}_{W_{N/2}}]^{lk} + W_N^k \sum_{l=0}^{N/2-1} x[2l+1][\underbrace{W_N^2}_{W_{N/2}}]^{lk}$$

$$= \underbrace{\sum_{l=0}^{N/2-1} x[2l]\, W_{N/2}^{lk} + W_N^k \sum_{l=0}^{N/2-1} x[2l+1]\, W_{N/2}^{lk}}_{\text{two } N/2\text{–point DFT's!!!}}$$

Using a briefer system of notation:

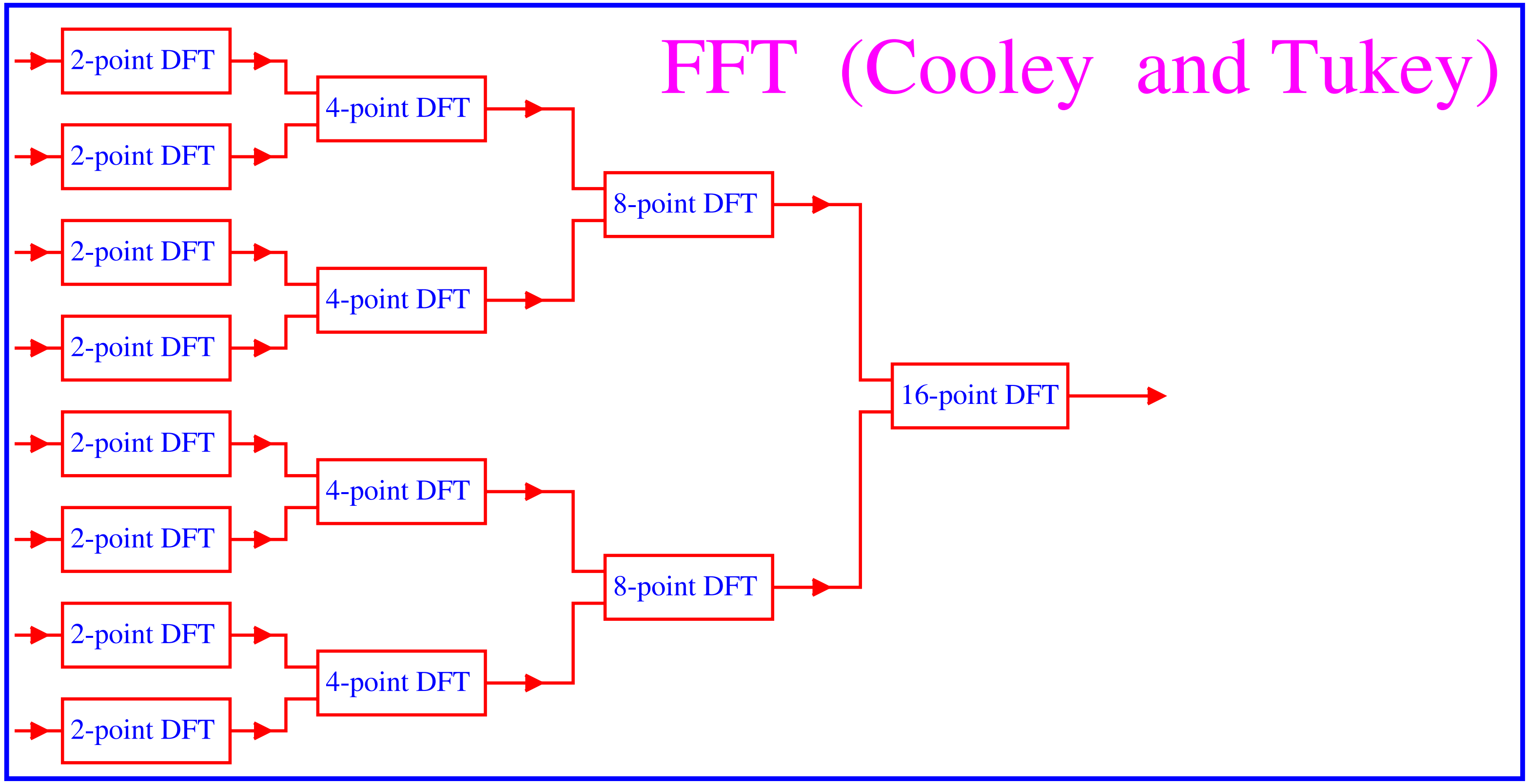
$$X_N[k] = G_{N/2}[k] + W_N^k H_{N/2}[k] \,,$$

where $G_{N/2}[k]$ and $H_{N/2}[k]$ are the $N/2$-point DFTs involving $x[n]$ with even and odd n, respectively.

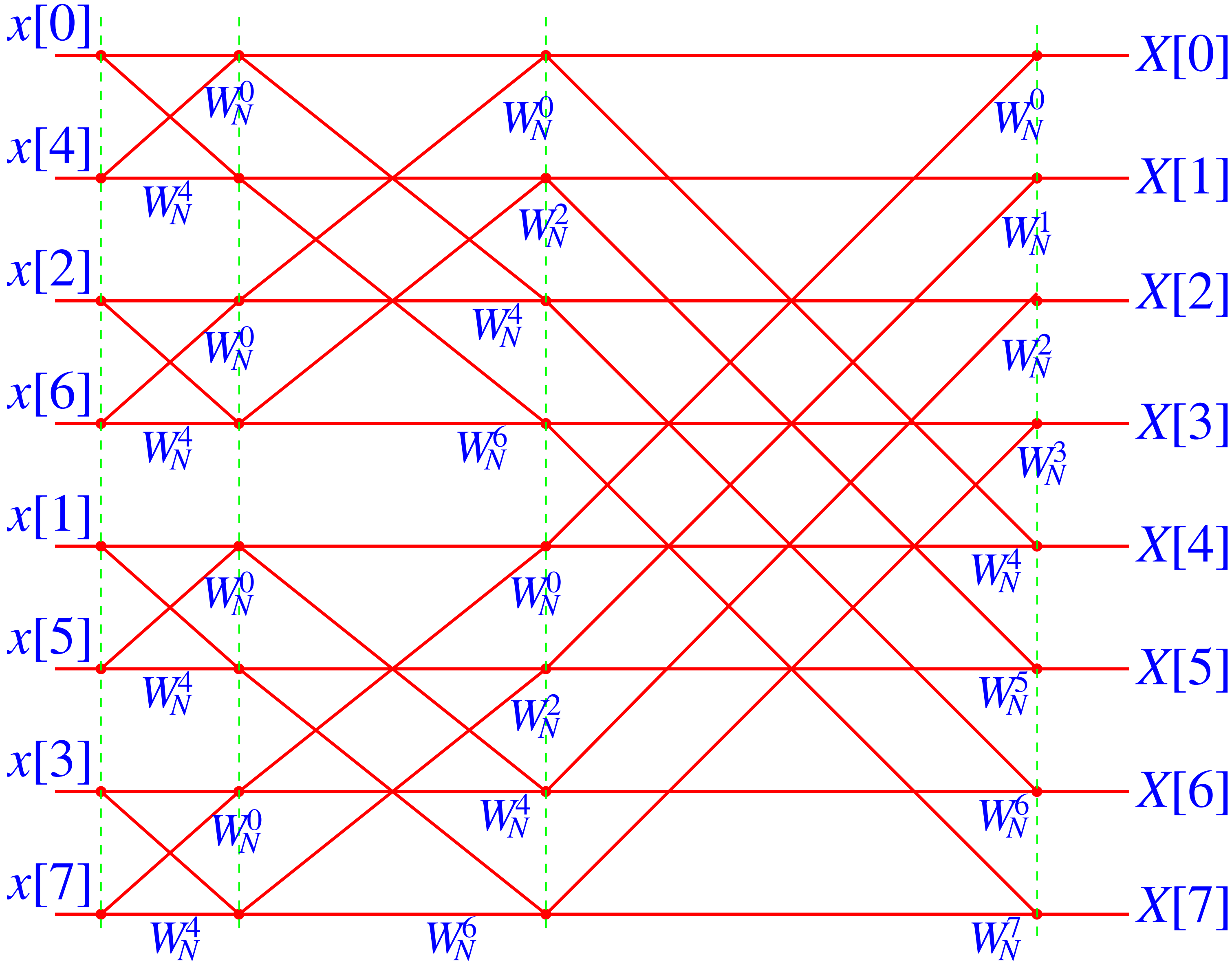


Corollary:

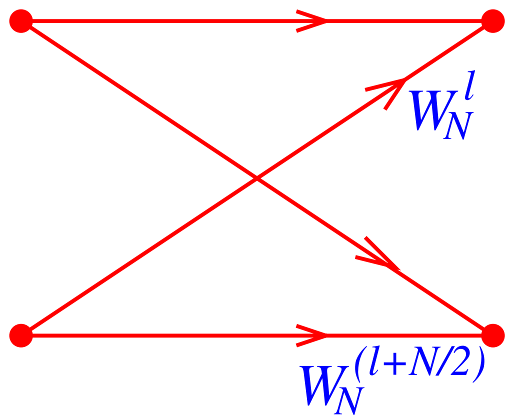
Any N-point DFT with even N can be computed via two $N/2$-point DFTs. In turn, if $N/2$ is even then each of these $N/2$-point DFTs can be computed via two $N/4$-point DFTs and so on. In the case of $N = 2^r$, all N, $N/2$, $N/4$ … are even and such a process of "splitting" ends up with all 2-point DFTs!



Flow graph of an 8-point FFT:



Flow graph of the basic "butterfly" computation:



Computational complexity: each stage has N complex multiplications and N complex additions and there are $\log_2 N$ stages $\Rightarrow$
the total complexity (additions and multiplications) is:

$$C_{\text{FFT}} = N \log_2 N,$$

compared with the complexity of the matrix DFT:

$$C_{\text{DFT}} \simeq N^2.$$

For large N (i.e., $N \gg 1$):

$$\log_2 N \ll N \quad \Rightarrow \quad C_{\mathsf{FFT}} \ll C_{\mathsf{DFT}}.$$

<u>Example:</u>

For $N = 2^{10} = 1024$, $C_{\mathrm{DFT}} \simeq 2^{20} \simeq 10^6$, whereas $C_{\mathrm{FFT}} = 10{\times}1024 \simeq 10^4$, i.e., the reduction is about <u>2 orders of magnitude</u>!