Lecture 10 - Discrete Fourier Transform and Fast Fourier Transform (II)

James Barnes (James.Barnes@colostate.edu)

Spring 2014

Outline

- FFT algorithm (conclusion)
- Remarks on implementation
- Application: block convolution

Note: Chapter 6 of your text covers some of this material

Example: 8 pt FFT

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad W = e^{-j\frac{2*\pi}{N}kn}, \quad k, n = 0, 1, \ldots N-1, \quad N = 8. \tag{1}$$

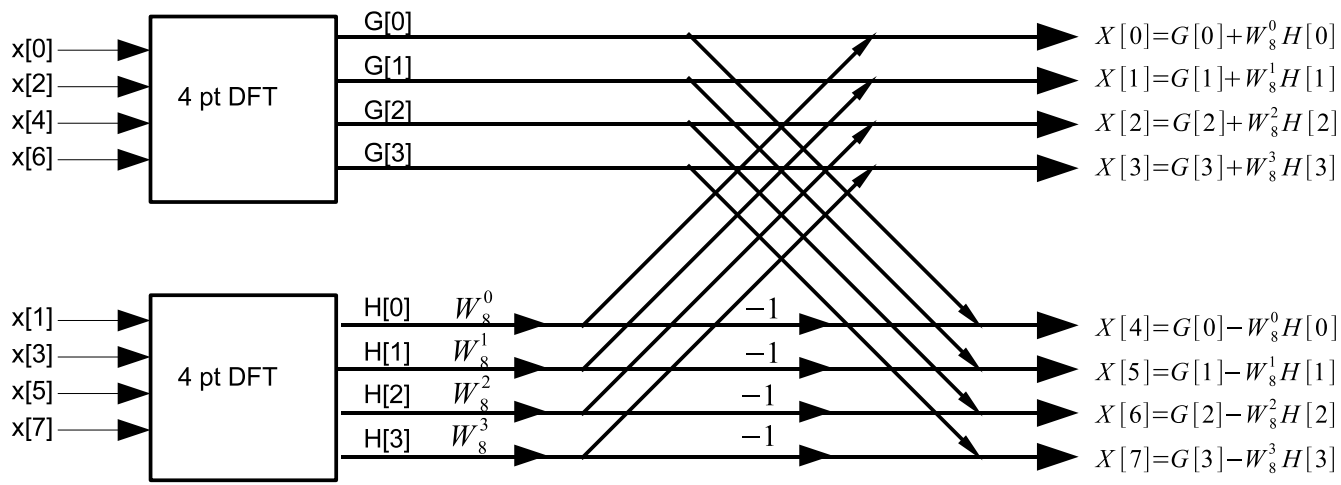
Split the summation into two sums over even and odd values of n, as shown below:

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

But using $W_N^2 = W_{N/2}$, easily shown from the definition of W_N, we can rewrite the last expression as:

$$X[k] = G[k] + W_N^k H[k], \quad G[k] = \sum_{r=0}^{(N/2)-1} x[2r] W_{N/2}^{k\cdot r}, \quad H[k] = \sum_{r=0}^{(N/2)-1} x[2r+1] W_{N/2}^{kr} \tag{3}$$

Example: 8 pt FFT - First stage Decimation in Time



Example: 8 pt FFT (2)

In eqn(3), $G[k]$and $H[k]$ are just the $(N/2)$ - order DFTs of the even and odd points of the original sequence. Therefore, $G[k]$ and $H[k]$ are periodic in k with period $(N/2)$. Using the fact that $W_N^0 = 1$ and $W_N^{N/2} = -1$ for all N, Eqn(3) can be completely written out:

$$X[0] = G[0] + H[0] \qquad X[4] = G[0] - H[0]$$

$$X[1] = G[1] + W_8^1 H[1] \qquad X[5] = G[1] - W_8^1 H[1]$$

$$X[2] = G[2] + W_8^2 H[2] \qquad X[6] = G[2] - W_8^2 H[2]$$

$$X[3] = G[3] + W_8^3 H[3] \qquad X[7] = G[3] - W_8^3 H[3]$$

Example: 8 pt FFT (3)

Now further divide $G[k]$ and $H[k]$ into sums over every other value of each sub-sequence. Going back to the definition of G[k] in eqn(22), we have for G[k],

$$G[k] = \sum_{r=0}^{(N/4)-1} x[2 \cdot 2r] W_N^{k \cdot 2 \cdot 2r} + \sum_{r=0}^{(N/4)-1} x[2 \cdot (2r+1)] W_N^{k \cdot 2 \cdot (2r+1)}, \tag{4}$$

or

$$G[k] = \sum_{r=0}^{(N/4)-1} x[4r] W_{N/4}^{kr} + W_N^k \sum_{r=0}^{(N/4)-1} x[4r+2] W_{N/4}^{kr} \tag{5}$$

Example: 8 pt FFT (4)

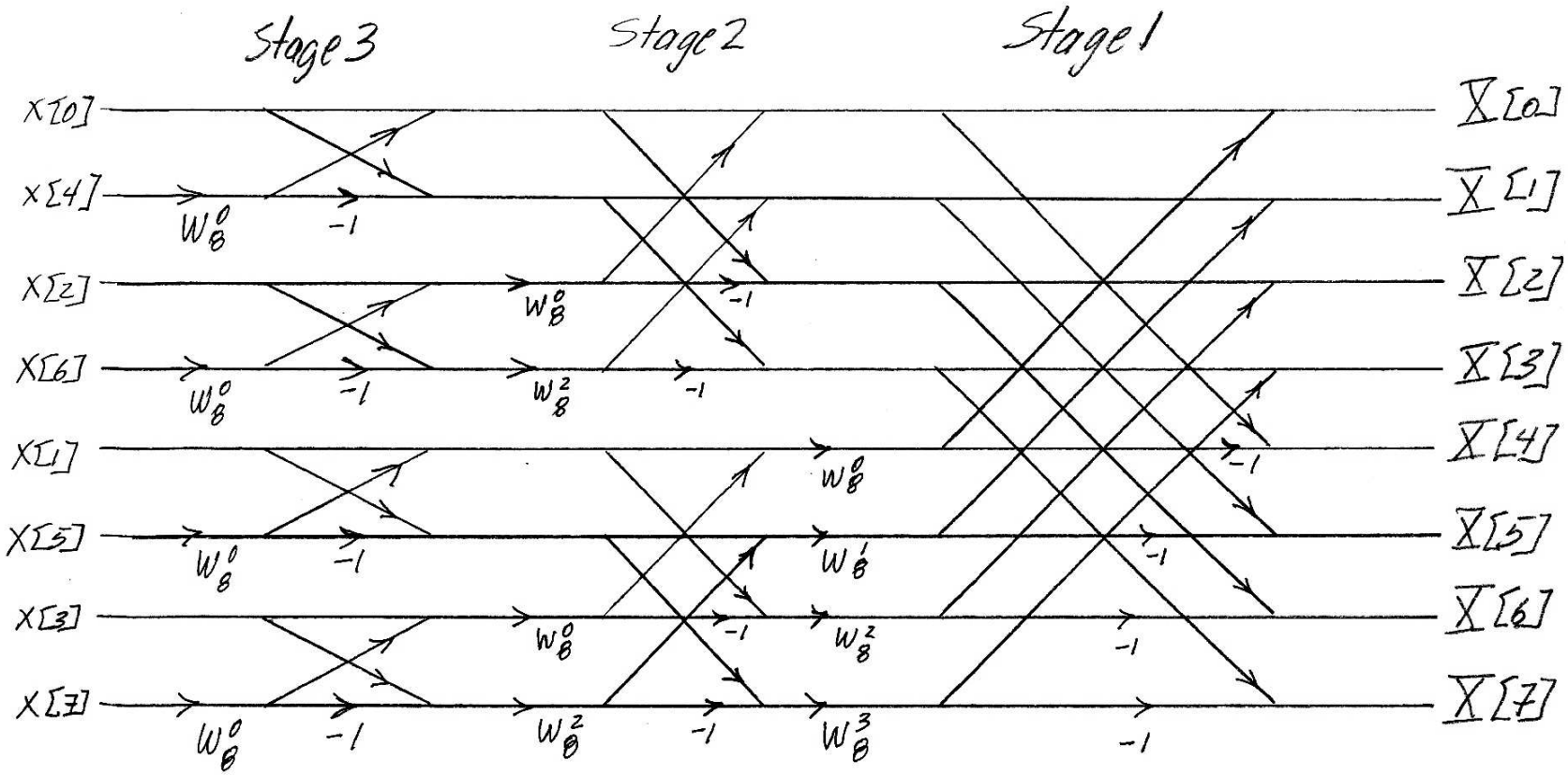
So $G[k]$ itself can be further broken down into two DFTs which involve every 4th value of the original sequence $x[n]$. Given that N=8, the two summations reduce to two terms each, and we can write:

$$G[0] = x[0] + x[4] + x[2] + x[6] \quad G[2] = x[0] + x[4] + W_8^2(x[2] + x[6])$$

$$G[1] = x[0] - x[4] + W_8^1(x[2] - x[6]) \quad G[3] = x[0] - x[4] + W_8^3(x[2] - x[6])$$

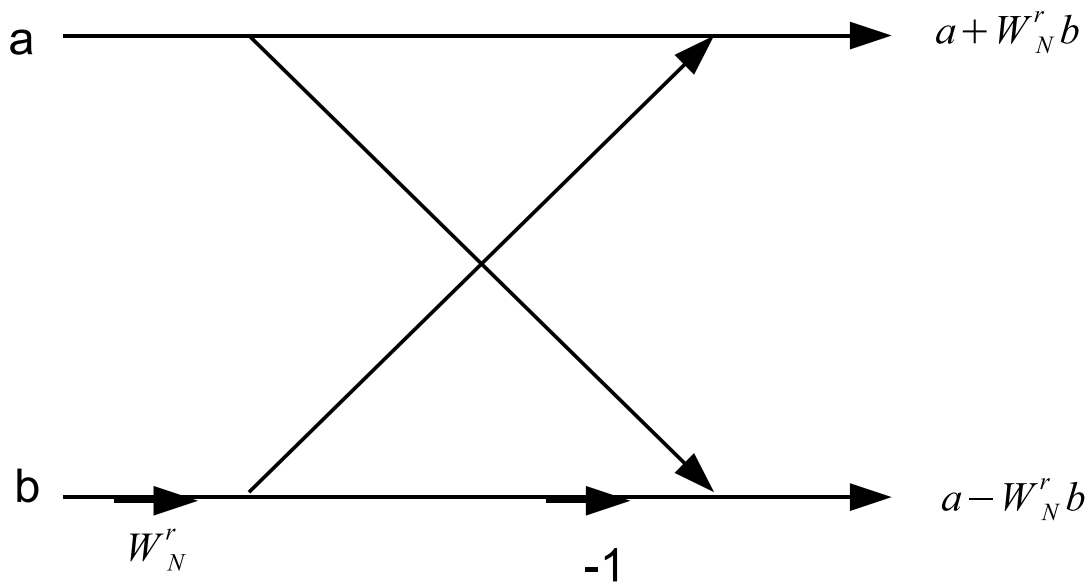
using $W_2^1 = W_2^3 = -1$ and $W_2^2 = 1$. The $H[k]$ term can similarly be broken down. This terminates the process for N=8.

Example: 8 pt FFT (5)



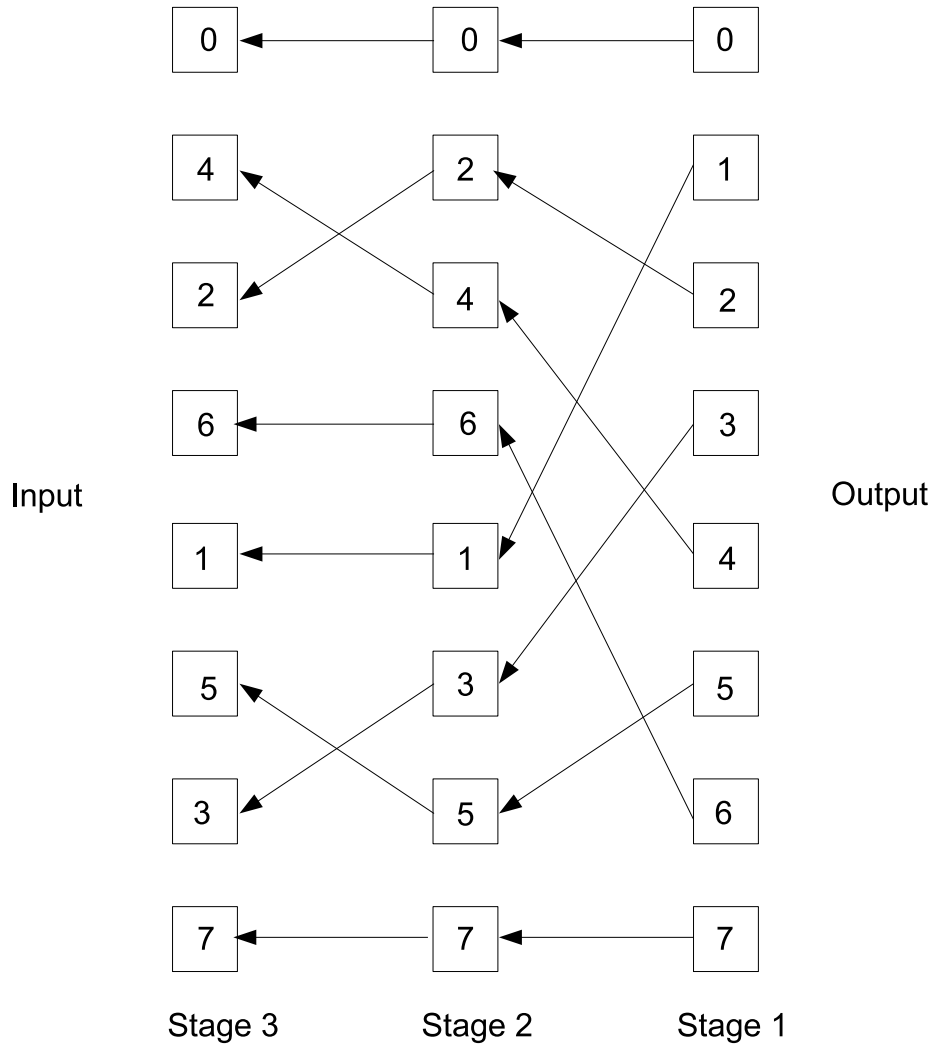
Butterfly Structure

Each stage consists of N/2 butterfly structures spanning ”track separations” of 1, 2, 4,... with possibly a multiply by a twiddle factor on the lower path:



Bit Reversal in 8 pt FFT

At each stage, partition into odd and even members.



FFT for FIR Filtering

We can compute the results of passing a sequence through a filter with the convolution sum:

$$y[n] = \sum_{k=0}^{N-1} h[k]\, x[n-k] \tag{6}$$

or by taking the Fourier Transform of h and x, multiplying, and taking :

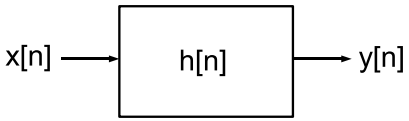
$$y[n] = FT^{-1}\{Y(\omega) = H(\omega)\, X(\omega)\}. \tag{7}$$

- For computation with DSP hardware, we must replace the Fourier Transform and its inverse with DFT/IDFT.
- If we use DFT/IDFT, the second method will require about as many arithmetic operations as the first,
- However, if we use FFT methods. the second method requires much fewer arithmetic ops.

FIR Filtering (2)

To be more specific:

$$y[n] = \Sigma_{k=0}^{M-1} h[k]x[n-k]$$



where $x[n] \neq 0,\ 0 \leq n < L-1, \quad h[n] \neq 0,\ 0 \leq n < M-1.$

Therefore, $y[n] \leq 0,\ 0 \leq n < L+M$, i.e. $y[n]$ is of length $L+M-1$.

Eqn(7) can be written in terms of DFTs

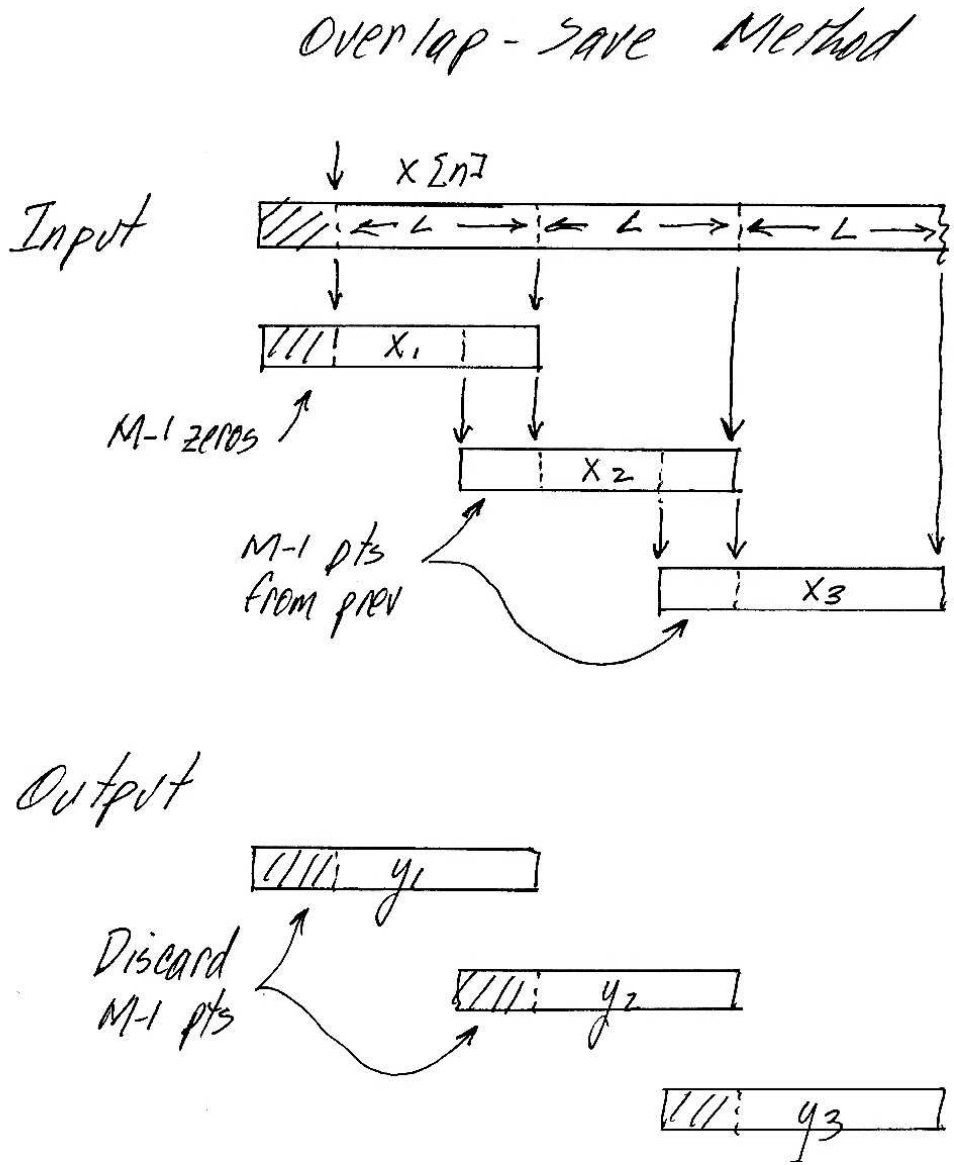
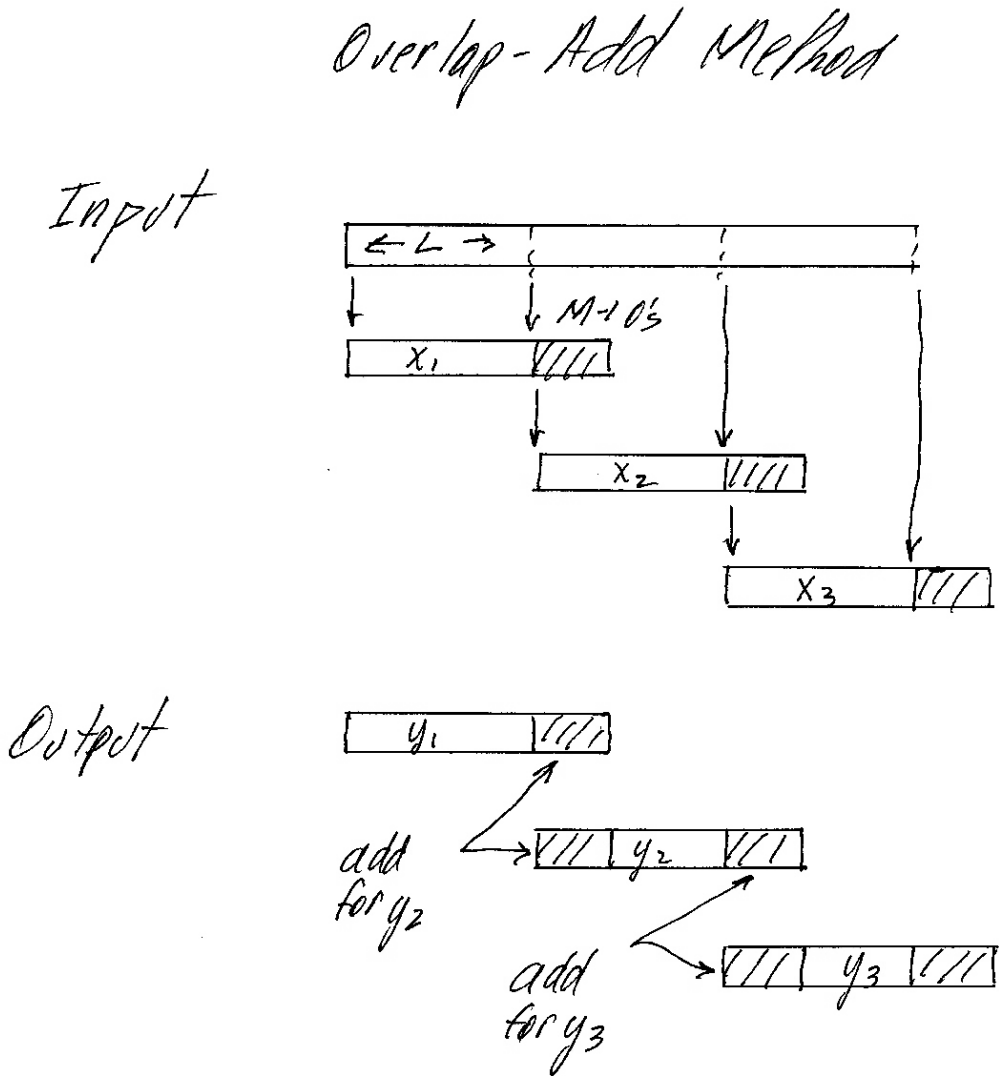
$$Y[k] = H[k] \cdot X[k] \tag{8}$$

ONLY IF the DFTs are of length N, where $N \geq L+M-1$. To make this work, the sequences {h[n]} and {x[n]} MUST be made to equal length N by padding with 0's. This will make H[k] and X[k] have the right length.

DFT/FFT Filtering of long data sequences

- Often it is required to filter a data sequence of long length – longer than can fit into memory
- The sequence can be broken up into fixed-length blocks which can be processed one at a time. Say we are implementing an FIR filter of length M with x[n] blocks of length L. Then the blocks must increased to length L+M-1 in order to avoid aliasing effects.
- There are two methods for increasing block size: overlap-save and overlap-add

Overlap-Save Method for Padding Input Sequence



Overlap-Add Method for Padding Input Sequence