LECTURE [redacted]

From Ch. ~~8~~. 9.1, 9.3 ~~All Sections except sections~~ 5.8.

~~Ch. 6~~ Efficient computation of the DFT (FFT)

- Direct computation of the DFT (Let $W_N = e^{-j\frac{2\pi}{N}}$)

Then

$$\text{N-DFT pair:} \left\{ \begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) \cdot W_N^{kn} \\ x(n) &= \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, W_N^{-kn} \end{aligned} \right\}$$

$k, n = 0, 1, \ldots, N-1$

Both N-DFT and N-IDFT involve basically same # of operations

To calculate $X(k)$, $k = 0, 1, \ldots, N-1$

- N^2 complex multipl., $N(N-1)$ compl addit.
- or $4N^2$ real multiplic., $4N^2 - 2N$ real addit.
- $2N^2$ evaluations of trigonometric functions (avoid with look up tables)
- Indexing and addressing operations

Ex

	$N = 8$	$N = 2^{12} = 4096$
real ⊗	256	$6.7 \cdot 10^7$
real ⊕	240	$6.7 \cdot 10^7$

- Thus for N large the computational burden is significant
- Reduce # of operations by exploiting Symmetry and periodicity properties of W_N and Symmetry properties of $X(k)$

$W_N^N = 1$	$W_N^{kN} = 1$
$W_N^{N/2} = -1$	$W_N^{k} = W_{N/2}^{k/2}$
$W_N^{N/4} = -j$	$W_N^{kN+r} = W_N^r$
$W_N^{3N/4} = j$	

$X(k)$	Re	Im
Even	Re( ) Even	Im( ) Even
odd	Im( ) odd	Re( ) odd

$x(n)$

FFT algorithms: Efficient (fast) algorithms employing properties of W_N

Fundamental principle of FFT: Decompose the computation of N-DFT into successively smaller DFT's. Achieve that by exploiting the symmetry and periodicity properties of W_N.

Radix of an FFT: r: radix if $\underline{N = r^q}$

Decimation in time FFT: Decomposition of the computation into smaller DFT's is accomplished by decomposing (decimating) $x(n)$ into successively smaller sequences.

Radix-2, Decimation in time FFT algorithms. $(N = 2^q)$

1st step N: even integer. So for $k = 0, 1, \ldots, N-1$

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{\substack{n=2m \\ (\text{even})}} x(n) W_N^{nk} + \sum_{\substack{n=2m+1 \\ (\text{odd})}} x(n) W_N^{nk} =$$

$$= \sum_{m=0}^{N/2-1} x(2m) W_N^{2mk} + \sum_{m=0}^{\frac{N}{2}-1} x(2m+1) W_N^{(2m+1)k} = \sum_{m=0}^{\frac{N}{2}-1} x(2m) W_N^{2mk} + W_N^{k} \sum_{m=0}^{\frac{N}{2}-1} x(2m+1) W_N^{2mk}$$

From properties of W_N: $\boxed{W_N^2 = W_{N/2}}$, Thus,

$$X(k) = \underbrace{\sum_{m=0}^{N/2-1} x(2m) W_{N/2}^{mk}}_{A(k),\ \frac{N}{2}\text{-DFT} \left(\substack{\text{period} \\ N/2}\right)} + W_N^k \underbrace{\sum_{m=0}^{N/2-1} x(2m+1) W_{N/2}^{mk}}_{B(k),\ \frac{N}{2}\text{-DFT} \left(\substack{\text{period} \\ N/2}\right)}, \quad k = 0, 1, \ldots, N-1$$

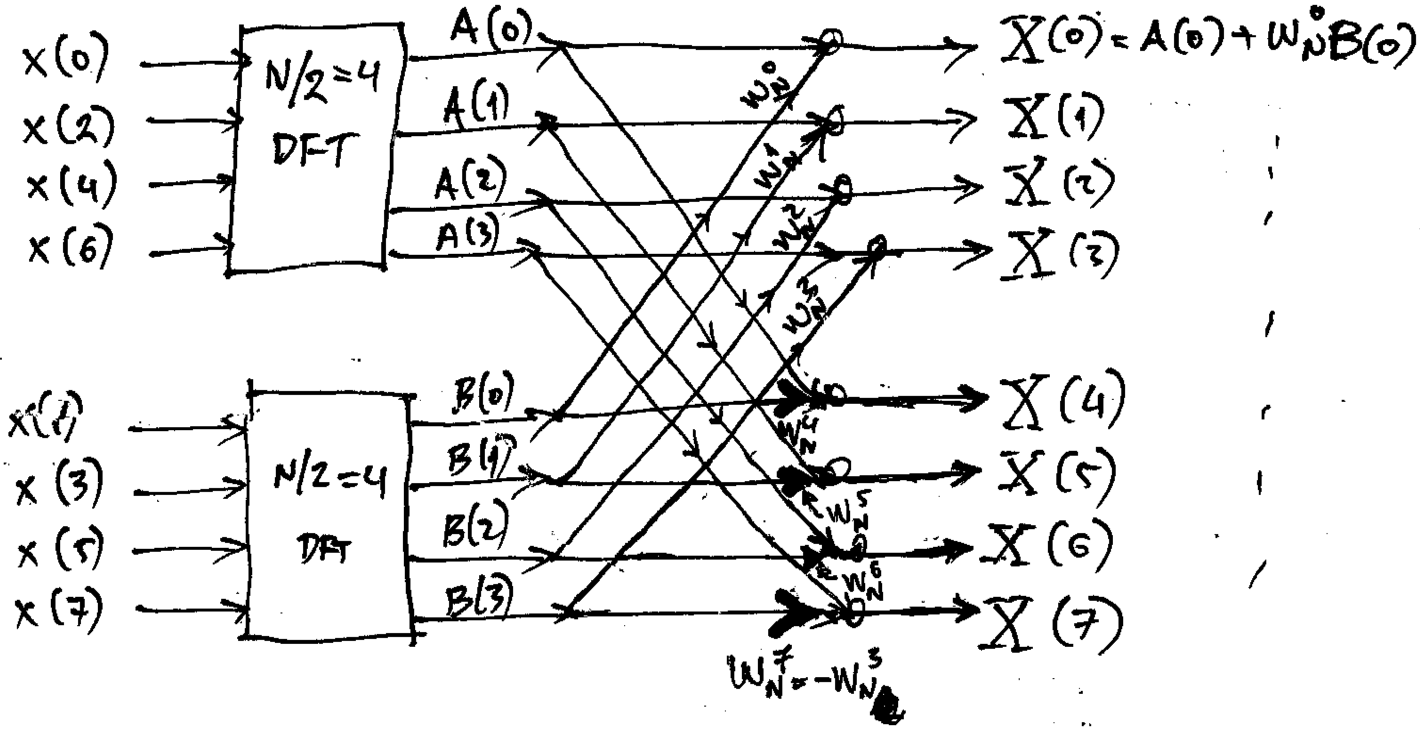
* $A(k) = A(k + N/2)$, $B(k) = B(k + \frac{N}{2})$ $\Rightarrow$ $A(k)$, $B(k)$ need to be computed only for $k = 0, \ldots, \frac{N}{2}-1$

* Also $\left[W_N^{k+N/2} = W_N^{N/2} W_N^k = -W_N^k \right]$

Thus, finally

$$X(k) = A(k) + W_N^k B(k) \quad k=0,1,\dots,N-1 \quad \equiv \quad \left[\begin{array}{l} X(k) = A(k) + W_N^k B(k) \\ X\left(k+\frac{N}{2}\right) = A(k) - W_N^k B(k) \end{array}\right] \quad k=0,1,\dots,\frac{N}{2}-1$$

Ex. Let $N = 2^3 = 8$



$$\#\otimes_c \quad 2\cdot\left(\frac{N}{2}\right)^2 + N = \frac{N^2}{2} + N < N^2 \quad \text{for } N>2$$

$$\#\oplus_c \sim 2\cdot\left(\frac{N}{2}\right)^2 + N = \frac{N^2}{2} + N < N^2 \quad \text{for } N>2$$

Reduction in storage requirement
place {A( )} {B( )} in place of {x( )}
Then you need only N storage positions
(computation in place) ..