THE FAST FOURIER TRANSFORM: LECTURE 1

Paolo Favaro

Chapter 10 of Textbook

(original material from John Thompson)

Overview

- Fast Fourier transform (FFT) techniques
- Deriving the FFT
- The Butterfly Operation
- Properties of the FFT

Discrete Fourier Transform (DFT)

- The DFT is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \qquad k = 0 \ldots N-1$$

- For N bins, require N^2 complex multiplies and adds
- However, there are only N data samples $x(n)$ and twiddle factors W_N^{nk}!

DFT to FFT

- It turns out that:
  - Can subdivide an N-point DFT into two size $N/2$-point DFTs
  - Each size $N/2$ DFT can be subdivided into two size $N/4$-point DFTs and so on...
  - If N is a power of 2, can calculate the DFT using a 2-point DFT over and over again
- This simplified calculation is the FFT

Subdividing the DFT

- Can write the DFT equation as:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

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

- Now we make two substitutions:

$$m = n/2 \text{ (even } n\text{)}, \;\; m = (n-1)/2 \text{ (odd } n\text{)}$$

- So the DFT equation becomes:

$$
\begin{aligned}
X(k) &= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{(2m+1)k} \\
&= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk} + W_N^{k} \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{2mk} \\
&= X_1(k) + W_N^{k} X_2(k)
\end{aligned}
$$

- $X_1(k)$ is the $(N/2)$ point DFT of even $x(n)$
- $X_2(k)$ is the $(N/2)$ point DFT of odd $x(n)$

DFT Subdivision

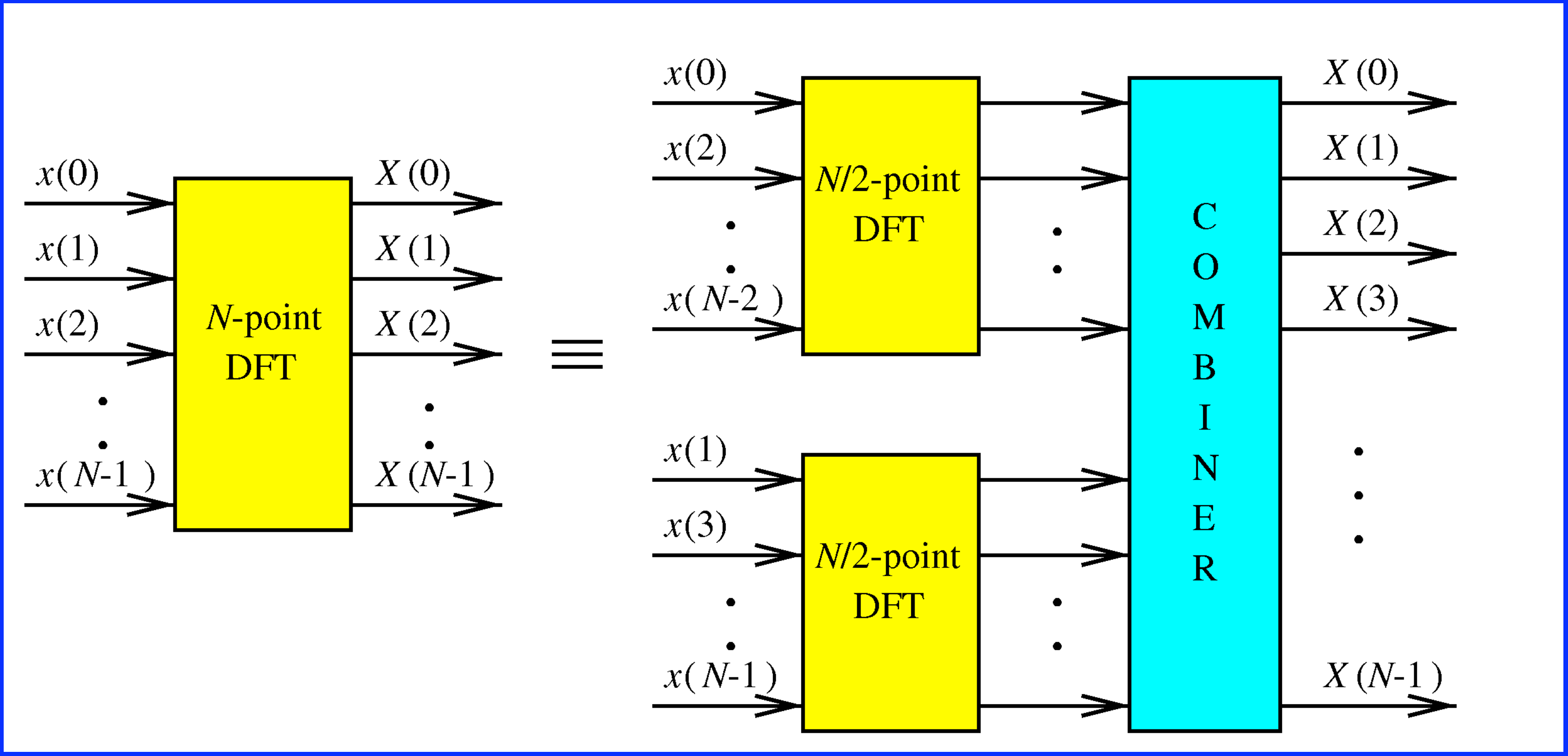
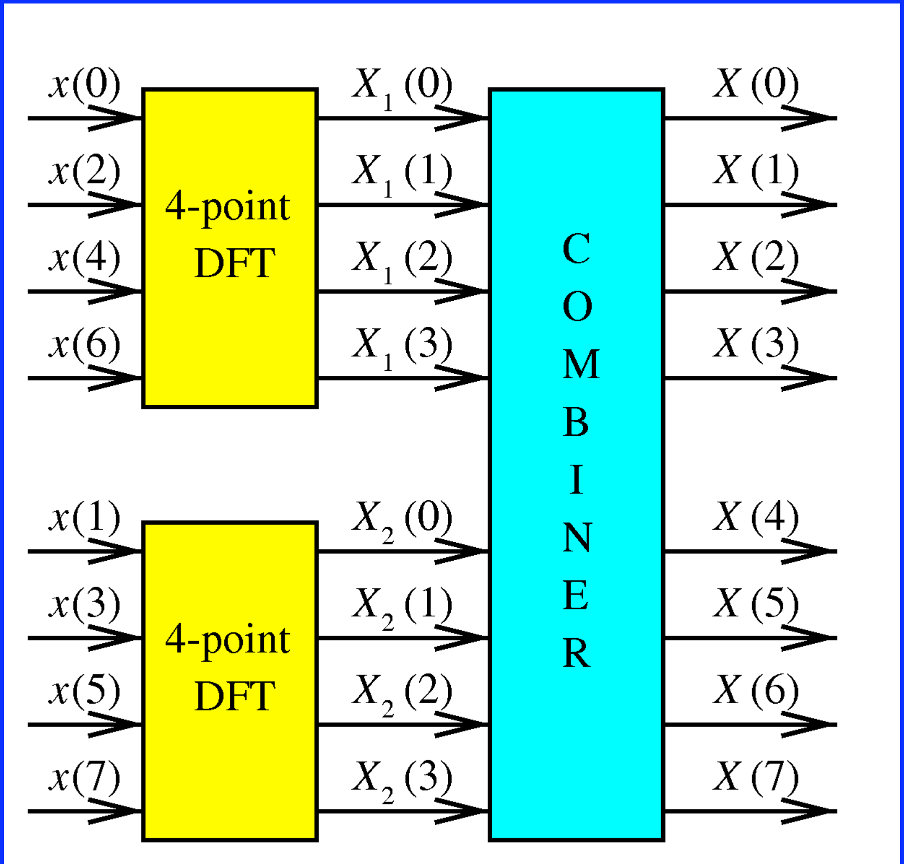


Fig 10.1

8-point DFT Example

Fig 10.2:

Split 8-point DFT into two 4-point DFTs, plus combiner



Repeating the Same Trick...

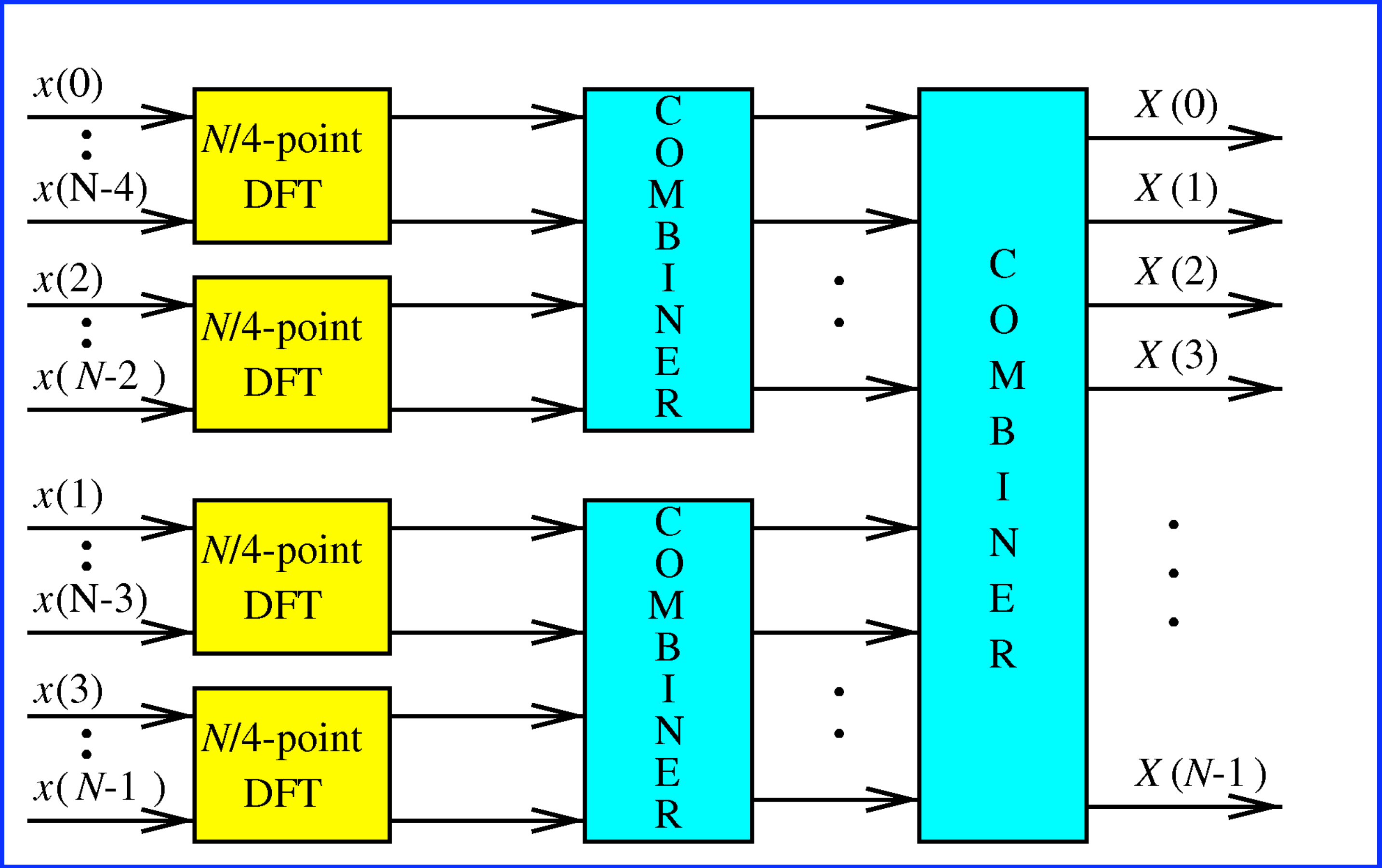
- We can also subdivide $X_1(k)$ and $X_2(k)$:

$$X_1(k) = \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk}$$

$$= \sum_{q=0}^{(N/4)-1} x(4q) W_N^{4qk} + W_N^{2k} \sum_{q=0}^{(N/4)-1} x(4q+2) W_N^{4qk}$$

$$= X_{11}(k) + W_N^{2k} X_{12}(k)$$

Subdividing the DFT Again



8-Point DFT Revisited

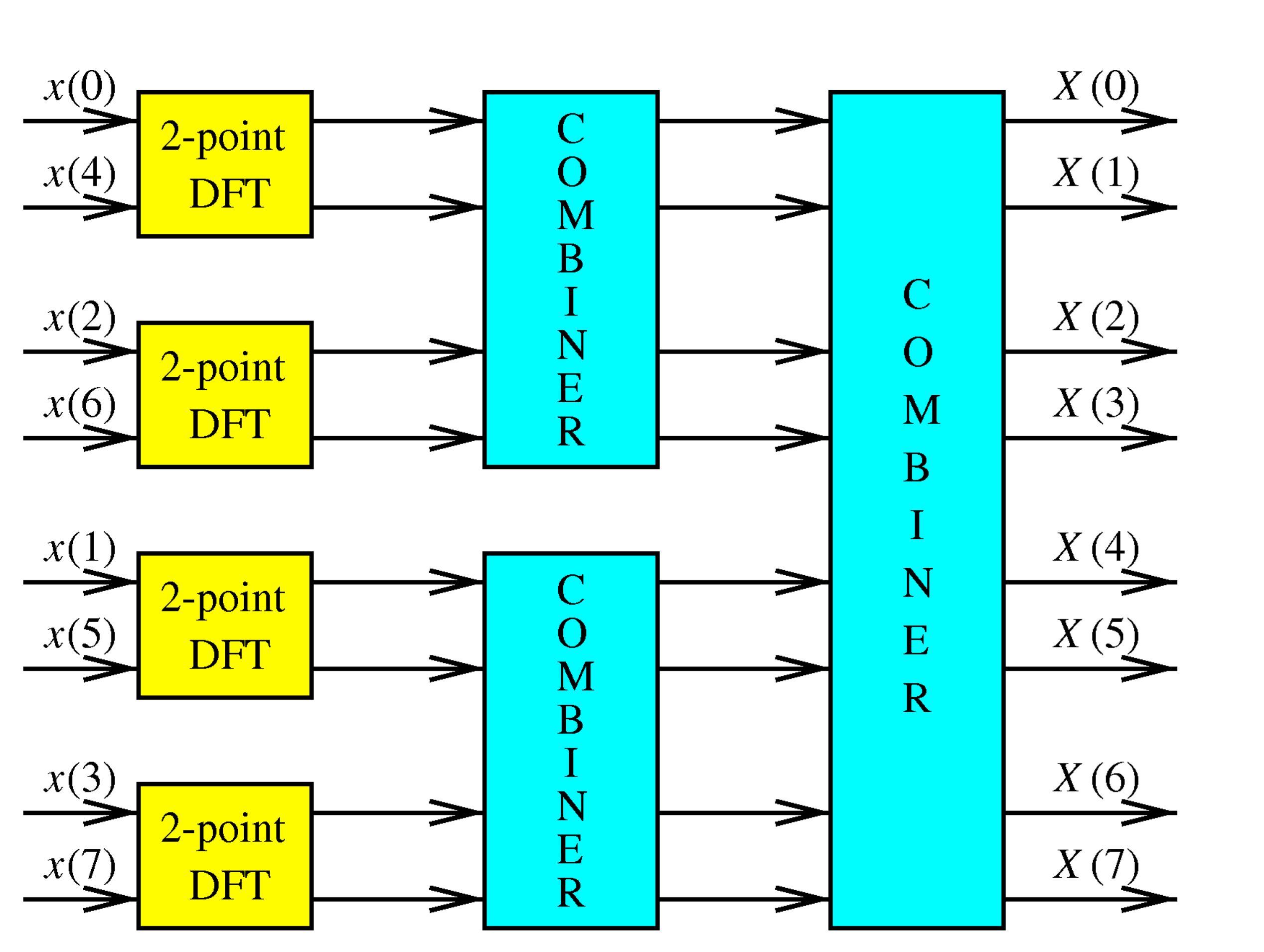


Fig 10.3

DFT Simplification

- If N is power of 2, can simplify DFT to:
  - N/2 separate 2-point DFTs
  - Followed by [$\log_2(N)-1$] combining stages
- So for the N=8 example, we have four 2-point DFTs and two combining stages

The 2-point DFT

- The 2-point DFT may be written as:

$$\begin{bmatrix} X(0) \\ X(1) \end{bmatrix} = \begin{bmatrix} W_2^0 & W_2^0 \\ W_2^0 & W_2^1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \end{bmatrix}$$

- *X(0)* & *X(1)* are sum and difference of *x(0)* and *x(1)*
- No multiplies needed for 2-point DFT!

- First stage of simplified DFT requires only additions/subtractions
- The 2-point DFT is also called the **Butterfly Operation**:

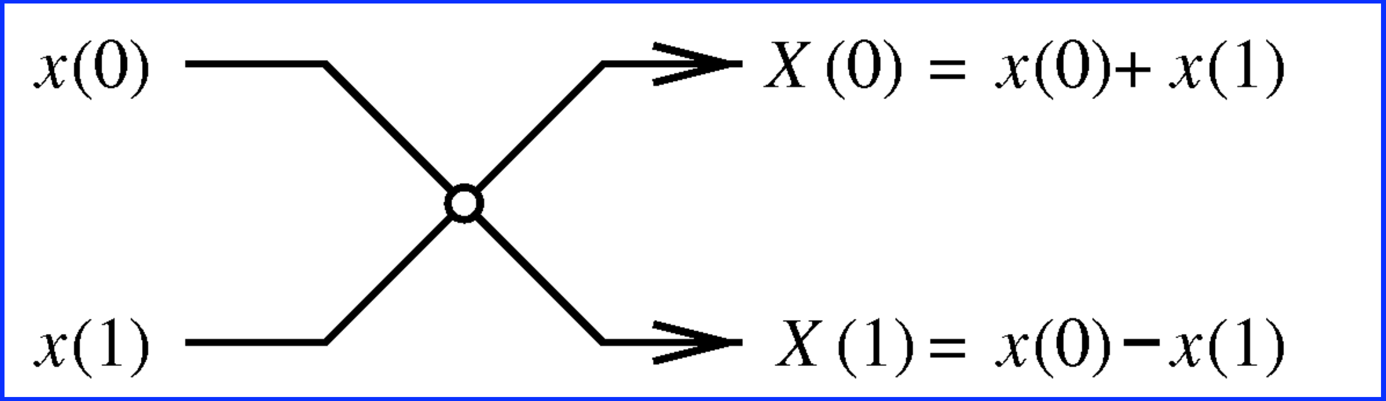


Fig 10.4

Last Combining Stage

- The first simplified DFT was

$$X(k) = X_1(k) + W_N^k X_2(k)$$

- $X_1(k)$ and $X_2(k)$ have period $N/2$, so:

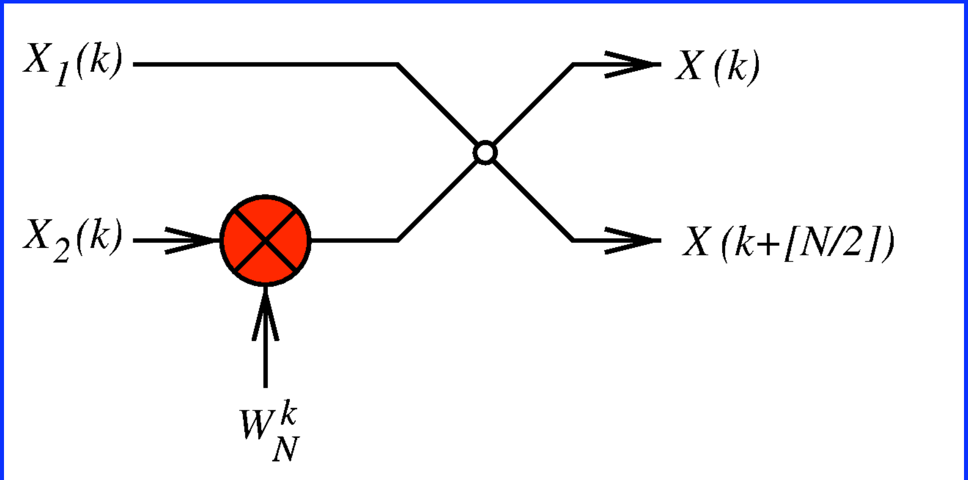
$$X_1(k) = X_1(k+(N/2)) \text{ and } X_2(k) = X_2(k+(N/2))$$

- Also can write $W_N^k = -W_N^{k+(N/2)}$

- So we can now write:

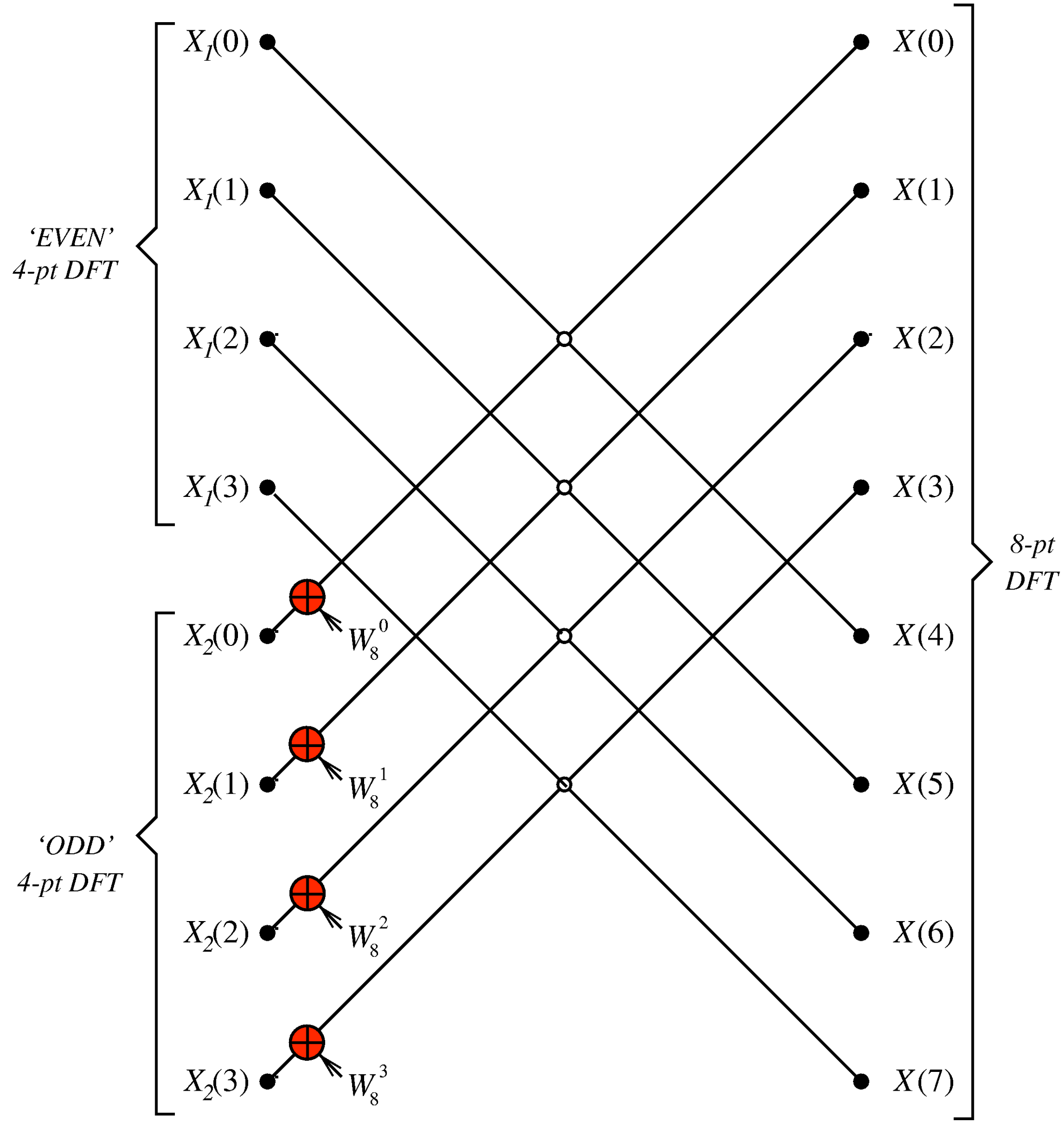
$$X(k) = X_1(k) + W_N^k X_2(k)$$
$$X(k+[N/2]) = X_1(k) - W_N^k X_2(k)$$

- Output is 2-point DFT of $X_1(k)$ and of $W_N^k X_2(k)$:



- Thus 'even' DFT $X_1(k)$ and 'odd' DFT $X_2(k)$ can produce $X(k)$ using:
  - $(N/2)$ multiplies of W_N^k and $X_2(k)$ (for $k=0...[N/2]-1$)
  - $N/2$ Butterfly operations are needed to calculate DFT outputs $X(k)$ and $X(k+[N/2])$ (for $k=0...[N/2]-1$)

Final Stage Combiner for 8-pt DFT example



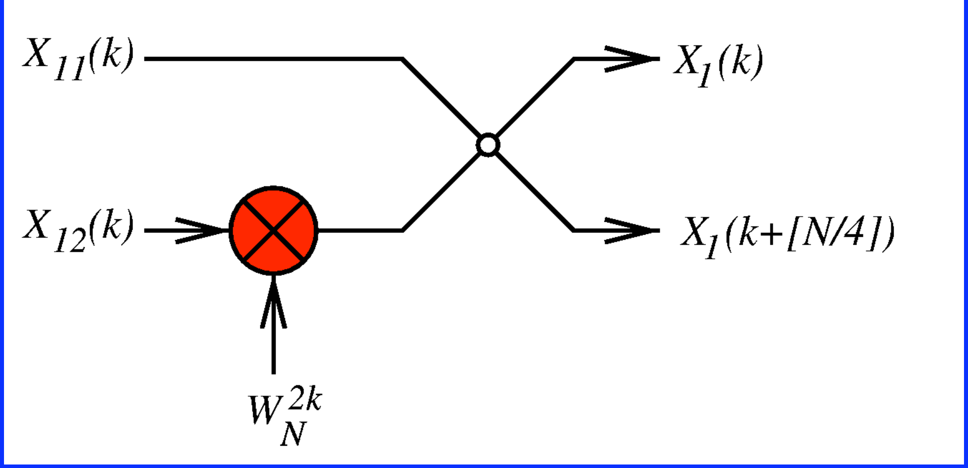
2nd Last Combining Stage

- Consider the ‘Even’ DFT $X_1(k)$:

$$X_1(k) = X_{11}(k) + W_N^{2k} X_{12}(k)$$

$$X_1(k + [N/4]) = X_{11}(k) - W_N^{2k} X_{12}(k)$$

- Butterfly of $X_{11}(k)$ and of $X_{12}(k)$:



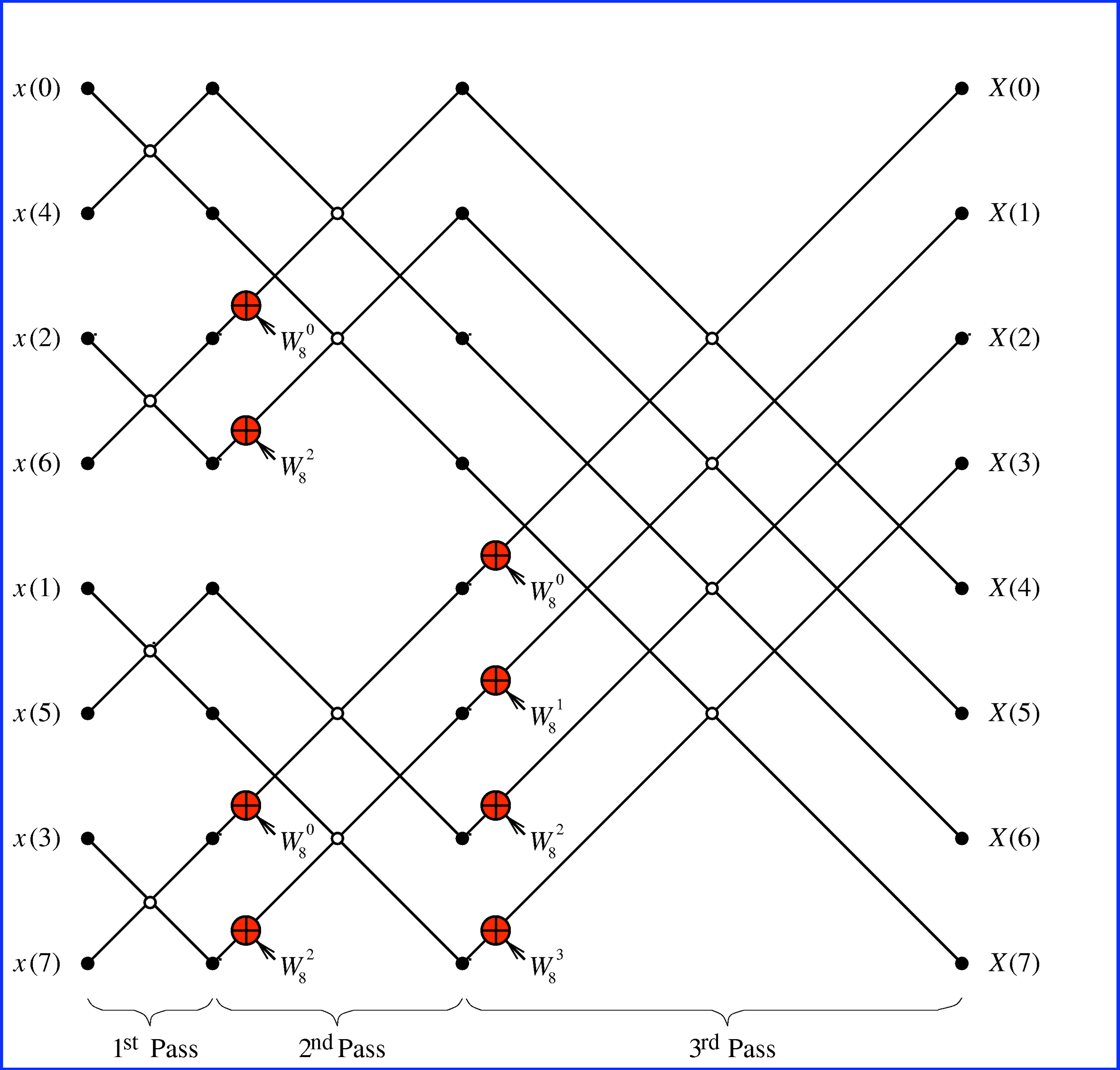


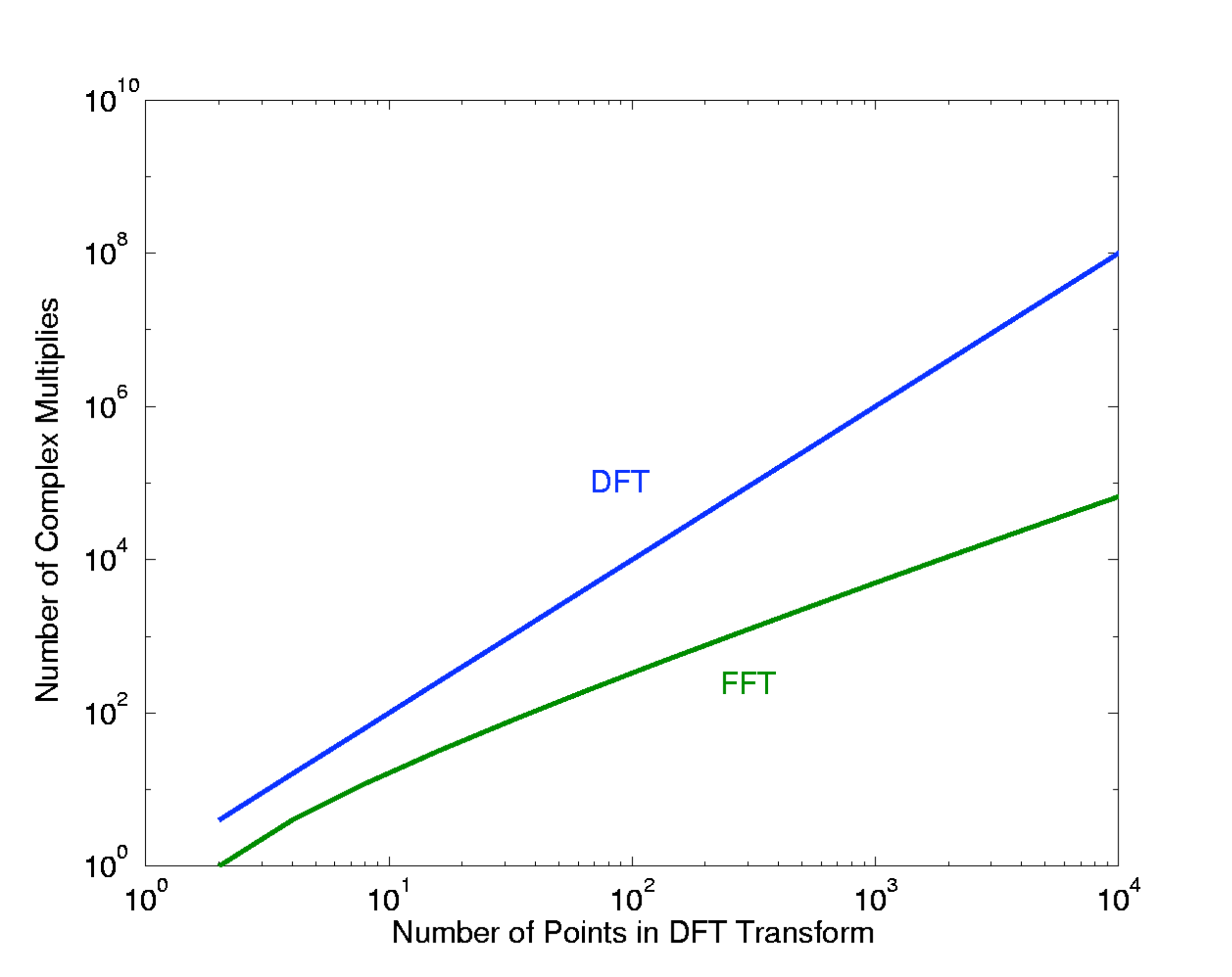
Fig 10.5:

8-pt FFT

FFT Complexity

- An *N*-point DFT can be simplified into $log_2(N)$ stages or ‘passes’
- Each stage requires *N/2* butterfly operations
- Thus require $(N/2)\ log_2(N)$ butterfly operations in total
- Each butterfly uses two complex adds plus one complex multiply

Complex Multiplies in DFT/FFT



FFT Characteristics

Our FFT derivation has several special properties:

- **Decimation in Time (DIT):**
  - Time domain signal is successively split into ‘even’ and ‘odd’ groups
  - Mirror reflection of the binary FFT position specifies address of data sample, eg: 000→x(0), 001 →x(4),010 →x(2), etc.

- **Radix-2:**
  - FFT is based on 2-point DFT Butterfly operation

- **In Place:**
  - Each butterfly operation applied to two locations in memory
  - The two outputs can be stored in the same locations to minimise memory requirements