Digital Signal Processing
ESD-5 & IV-5 (elektro), E24

10. Computing the DFT

Assoc. Prof. Peter Koch, AAU

Remember the Discrete Fourier Transform, DFT

Generally, the DFT analysis and synthesis equations are written as follows:

Analysis equation: $X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}$,

Synthesis equation: $x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}$.

That is, the fact that $X[k] = 0$ for k outside the interval $0 \leq k \leq N-1$ and that $x[n] = 0$ for n outside the interval $0 \leq n \leq N-1$ is implied, but not always stated explicitly.

The Twiddle Factor W_N represents the fundamental frequency ${}^{2\pi}/_{N}$, $W_N = e^{-j(2\pi/N)}$, and thus $\left(W_N^k\right)^n$ represents, for a given value of n, a rotation that matches the phase shift associated with the k'th harmonic of the fundamental frequency.

AALBORG UNIVERSITY
DENMARK

The Analysis equation in matrix form

$k = 0$ (DC)

$k = 1$ (fundamental freq.)

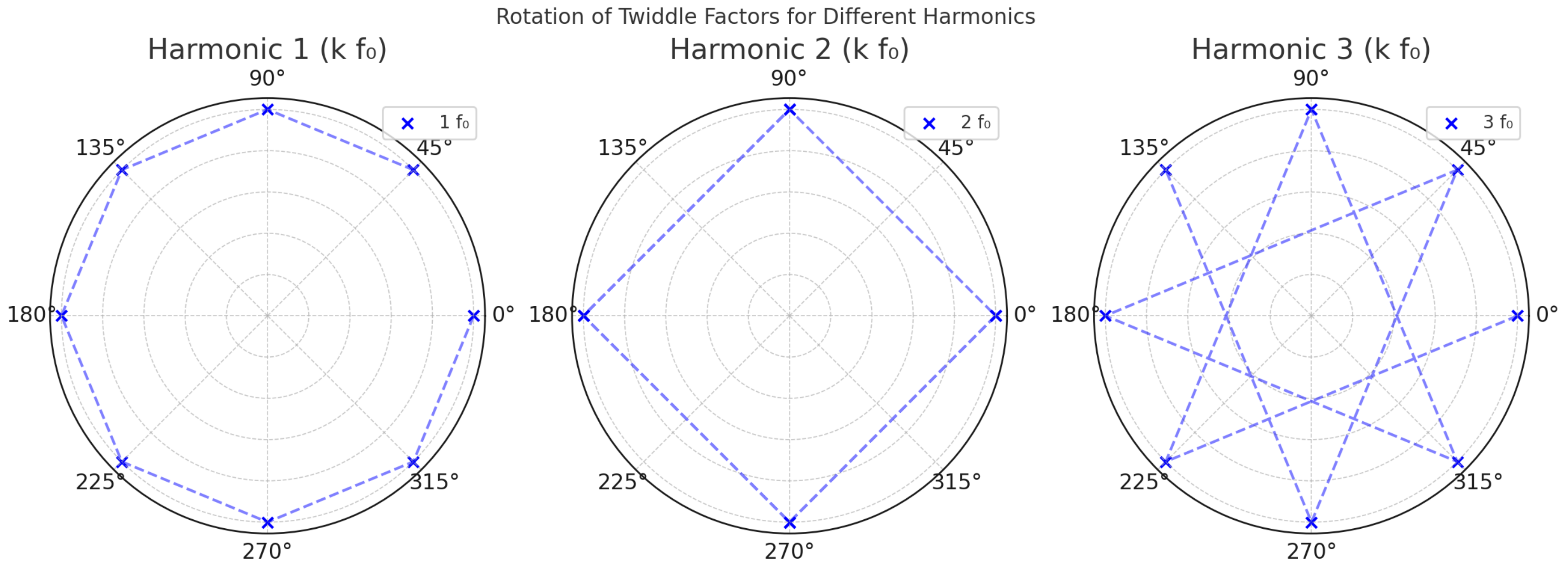
$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{bmatrix} = \underbrace{\begin{bmatrix} \boxed{1 \quad 1 \quad 1 \quad \cdots \quad 1} \\ \boxed{1 \quad e^{-j2\pi/N} \quad e^{-j4\pi/N} \quad \cdots \quad e^{-j2(N-1)\pi/N}} \\ 1 \quad e^{-j4\pi/N} \quad e^{-j8\pi/N} \quad \cdots \quad e^{-j4(N-1)\pi/N} \\ \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \\ \boxed{1 \quad e^{-j2(N-1)\pi/N} \quad e^{-j4(N-1)\pi/N} \quad \cdots \quad e^{-j2(N-1)(N-1)\pi/N}} \end{bmatrix}}_{\text{DFT matrix}} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix}$$

$k = N - 1$

Signal vector

Here, each row represents a frequency, i.e., DC, the fundamental frequency ω_0, $2\omega_0, 3\omega_0, \dots, (N-1)\omega_0$.

For each row, the time n is advancing from left (0) to right $(N-1)$, representing the phases $\theta_{k,n} = (k\omega_0)n$. Maybe better seen on the illustration on the next page…



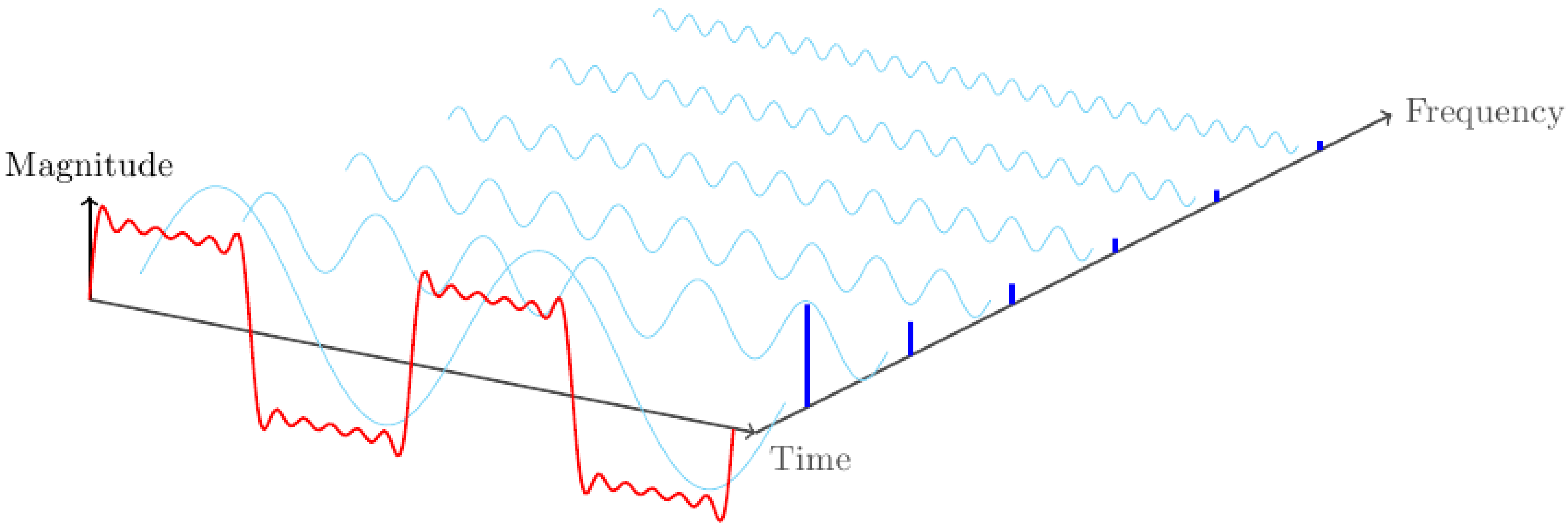
It is important to acknowledge, that despite that the phases may encounter several iterations on the circle, for $0 \leq n < N - 1$, there are ONLY N different values for the twittle factors.

Reason being that the sequence $\left(W_N^k\right)^n$ is periodic…

How should we understand the Analysis Equation

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \qquad 0 \leq k \leq N-1$$

So, this equation takes N samples of the time-domain signal and maps them into N Fourier coefficients (complex amplitudes), i.e., N samples representing one period of the spectrum in



Question now is, what should be the value of N in order to obtain a ”sufficiently good” **resolution** on the frequency axis…??

How should we understand the Analysis Equation

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \qquad 0 \leq k \leq N-1$$

Since $0 \leq n \leq N-1$ it means that we analyse only a segment (of length N) of the signal. We assume that this segment represents one period of $x[n]$.

The implication is that the segment can be considered as the (in principle) infinite signal ”seen through a window” of length N.

”Looking through” a window means that *in the time domain we multiply* the signal with a given window function which is identically equal to zero outside the interval.

Let’s have a look at this – in terms of the DTFT;

$$v[n] = x[n] \cdot w[n] \qquad 0 \leq n \leq N-1$$

$$V(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\theta}) W(e^{j(\omega-\theta)}) d\theta \qquad V(e^{j\omega}) \text{ is } 2\pi \text{ periodic in } \omega$$

This is the **D**iscrete **T**ime **F**ourier **T**ransform, see Table 2, p. 60 in O&S

AALBORG UNIVERSITY
DENMARK

How should we understand the Analysis Equation

As we have seen previously, the DFT is defined by sampling the DTFT, i.e.;

$$V[k] = V(e^{j\omega}) \quad \text{for} \quad \omega = \omega_k = \frac{2\pi}{N}k \quad 0 \leq k \leq N-1$$

Now, using the relation $\omega = T\Omega$ we then have $T\Omega_k = \omega_k = \frac{2\pi}{N}k$ and thus;

$$\boxed{\Omega_k = \frac{2\pi}{NT}k = 2\pi f_k \Rightarrow f_k = \frac{k}{NT} = k\frac{f_s}{N} \qquad 0 \leq k \leq N-1}$$

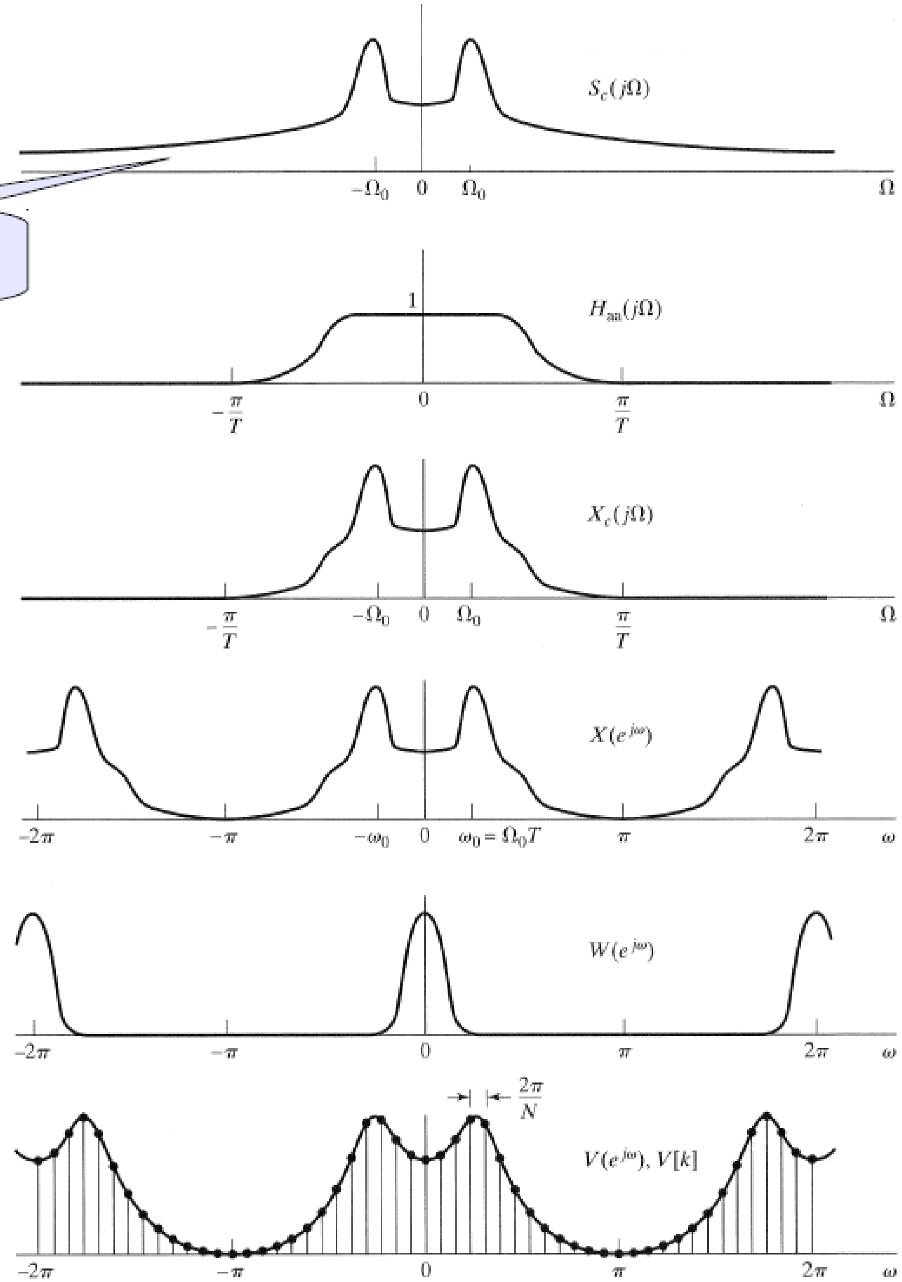
So, from this we realize that ”the size” of the DFT (i.e., the value of N) has to have a certain value in order to produce a spectrum with a ”sufficient” accuracy, i.e., *resolution*, in the interval from DC to $(f_s - \frac{f_s}{N})$.

The spectral resolution is defined as $\Delta f = \frac{f_s}{N}$, and thus $N \geq \frac{f_s}{\Delta f}$ to achieve a resolution which is Δf, or better. Alternatively, we could write $\Delta\omega = \frac{2\pi}{N}$.

AALBORG UNIVERSITY
DENMARK

From continuous time signal to the DFT spectrum

Wide-band



Considerations on the Computational Complexity

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \qquad 0 \le k \le N-1$$

For each of the N values of k we need to perform N complex multiplications and $N-1$ complex additions, i.e.;

N^2 **complex** multiplications, and $N(N-1)$ **complex** additions.

Now, since $(a+jb)\cdot(c+jd) = ac + jad + jbc - bd$ (i.e., 4 mult and 2 adds) and $(a+jb)+(c+jd) = (a+c)+j(b+d)$ (i.e., 2 adds), the total number of real arithmetic operations is;

$4N^2$ real multiplications

$N(2(N-1)+2N) = N(4N-2) = 4N^2 - 2N$ real additions

From the complex additions and multiplications

So, in conclusion the overall computational load of the DFT is $O(N^2)$

Are there any possible ways to reduce this complexity…???

AALBORG UNIVERSITY
DENMARK

The first approarch; The Goertzel Algorithm

This algorithm is based on a lot of ”creative” ideas – let’s get a brief overview…

The outset is the DFT analysis equation;

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{nk} \;, \quad 0 \leq k \leq N-1, \qquad W_N = e^{-j\frac{2\pi}{N}}$$

We now multiply with 1, and substitute n with r;

$$X[k] = 1 \cdot \sum_{r=0}^{N-1} x[r]W_N^{rk} = W_N^{-kN} \cdot \sum_{r=0}^{N-1} x[r]W_N^{rk}$$

$$X[k] = \sum_{r=0}^{N-1} x[r]W_N^{rk}W_N^{-kN} = \sum_{r=0}^{N-1} x[r]W_N^{rk-kN} = \sum_{r=0}^{N-1} x[r]W_N^{-k(N-r)}$$

AALBORG UNIVERSITY
DENMARK

The first approarch; The Goertzel Algorithm

So, we have…

$$X[k] = \sum_{r=0}^{N-1} x[r] W_N^{-k(N-r)} \qquad 0 \leq k \leq N-1$$

We now choose to change the summation limits to $]-\infty; \infty[$. This is a legal operation, because $x[r]$ is 0 outside the interval $0 \leq r \leq N-1$.

$$X[k] = \sum_{r=-\infty}^{\infty} x[r] W_N^{-k(N-r)} \qquad 0 \leq k \leq N-1$$

Since $X[k]$ is a sequence, we choose to denote it $y[n]$;

$$y[n] = \sum_{r=-\infty}^{\infty} x[r] W_N^{-k(N-r)} \qquad 0 \leq n \leq N-1$$

AALBORG UNIVERSITY
DENMARK

The first approarch; The Goertzel Algorithm

$$y[n] = \sum_{r=-\infty}^{\infty} x[r] W_N^{-k(N-r)} \qquad 0 \le n \le N-1$$

Since $x[r]$ is 0 outside the interval $[0; N-1]$, it is now legal to multiply with $u[N-r]$.

$$y[n] = \sum_{r=-\infty}^{\infty} x[r] (W_N^{-k(N-r)} \cdot u[N-r])$$

If we substitute N with n, we then recognize this sequence as the linear convolution between $x[n]$ and $W_N^{-nk} u[n]$

$$y[n] = \left(x[n] * (W_N^{-kn} u[n])\right)\big|_{n=N} \qquad 0 \le n \le N-1$$

AALBORG UNIVERSITY
DENMARK

The first approarch; The Goertzel Algorithm

$$y[n] = \left(x[n] * (W_N^{-nk}u[n])\right)\big|_{n=N} \qquad 0 \le n \le N-1$$

From this expression we conclude that the DFT can be derived by feeding a finite-length sequence $x[n]$ into a causal LTI system with impulse response;

$$h[n] = W_N^{-nk}u[n]$$

In this expression, k denotes the actual harmonics of the fundamental frequency.

$$x[n] \longrightarrow \boxed{h[n] = W_N^{-nk}u[n]} \longrightarrow y_k[n] = x[n] * \left(W_N^{-nk}u[n]\right)$$

How should we understand $n = N$..??

The interpretation is that $X[k] = y_k[n]$, i.e., the output of the LTI system h, when $n = N$.

Therefore; the corresponding N-point DFT coefficient $X[k] = y_k[N]$

The first approach; The Goertzel Algorithm

Now, let's represent the LTI system by its z-transform, i.e., the transfer function

$$h[n] = W_N^{-nk} u[n]$$

The impulse response

$$\begin{aligned} H(z) &= \sum_{m=0}^{\infty} W_N^{-mk} z^{-m} \\ &= \frac{1 - W_N^{-k} z^{-1}}{1 - W_N^{-k} z^{-1}} \sum_{m=0}^{\infty} W_N^{-mk} z^{-m} \\ &= \frac{\sum_{m=0}^{\infty} W_N^{-mk} z^{-m} - \sum_{m=0}^{\infty} W_N^{-(m+1)k} z^{-(m+1)}}{1 - W_N^{-k} z^{-1}} \\ &= \frac{W_N^0 z^0}{1 - W_N^{-k} z^{-1}} \\ &= \frac{1}{1 - W_N^{-k} z^{-1}}, \end{aligned}$$

Unilateral z-transform due to $u[n]$

…and now multiplied with 1

The only term left in the numerator is the term for $m = 0$

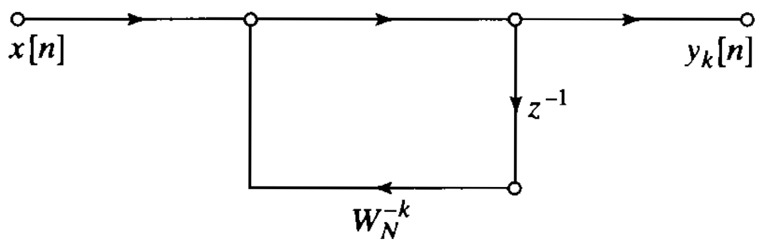
Another way of calculating the geometric series… ☺

We need one such filter for each harmonic frequency $\omega_k = k\,\frac{2\pi}{N}$

The first approach; The Goertzel Algorithm

$$x[n] \longrightarrow \boxed{\frac{1}{1 - W_N^{-k} z^{-1}}} \longrightarrow y_k[n]$$

We recognize this as a 1st order IIR filter. Since the filter has its zero in $z = 0$, we cannot distinguish Direct Form I and Direct Form II, and thus it can be represented by the following Signal Flow Graph;

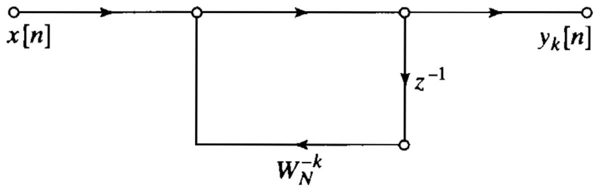


This is an example of a filter with a complex filter coefficient…!

requires 4 real multiplications and 4 real additions. All the intervening values $y_k[1], y_k[2], \ldots, y_k[N-1]$ must be computed in order to compute $y_k[N] = X[k]$, so the use of the system in Figure .1 as a computational algorithm requires $4N$ real multiplications and $4N$ real additions to compute $X[k]$ for a particular value of k. Thus, this procedure is slightly less efficient than the direct method. However, it avoids the computation or storage of the coefficients W_N^{kn}, since these quantities are implicitly computed by the recursion implied by Figure .1.

AALBORG UNIVERSITY
DENMARK

We can, however, lower the complexity…

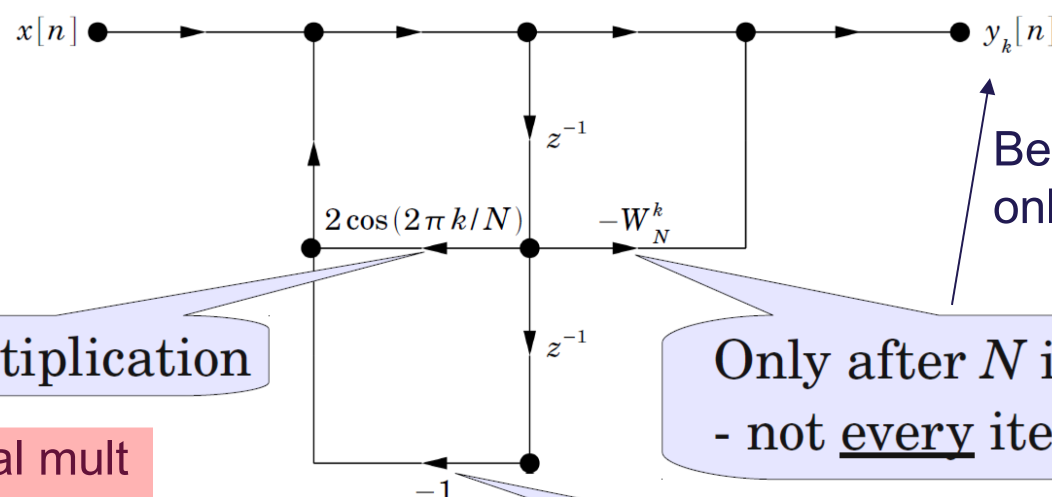


$$H_k(z) = \frac{1}{1 - W_N^{-k} z^{-1}}$$

Multiplying with $1 - W_N^{k} z^{-1}$ in numerator and denominator yields:

$$H_k(z) = \frac{1 - W_N^{k} z^{-1}}{(1 - W_N^{-k} z^{-1})(1 - W_N^{k} z^{-1})} \Rightarrow H_k(z) = \frac{1 - W_N^{k} z^{-1}}{1 - 2\cos(2\pi k / N) z^{-1} + z^{-2}}$$

with the flow graph (now a 2nd order recursive calculation of $X[k]$):



Because we need $y_k[n]$ only after N iterations

Real multiplication

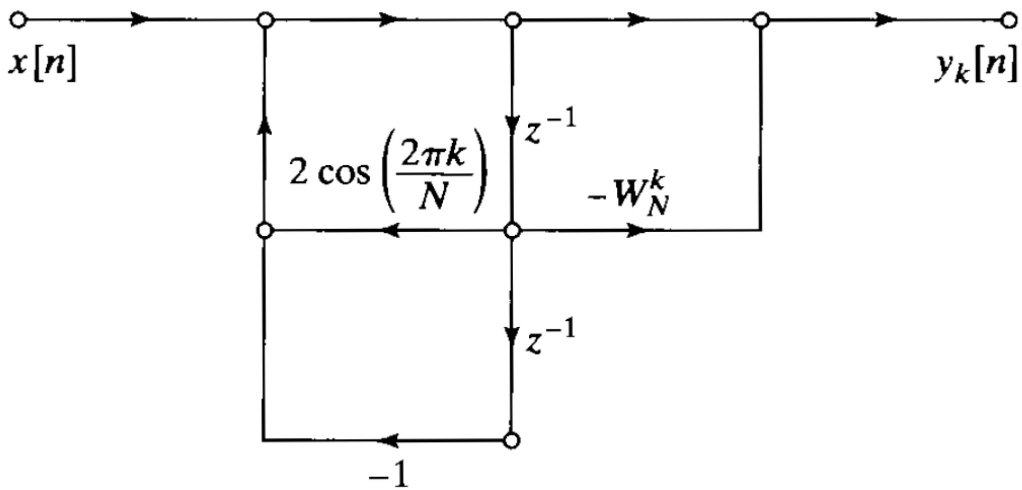
Only after N iterations - not <u>every</u> iteration!

N^2 real mult
$2N^2$ real add

Simple sign change

Yes, more efficient that direct DFT, but still $O(N^2)$…!!

The Goertzel Algorithm – in conclusion



One such filter calculates one complex Fourier coefficient. Therefore, Goertzel is to be considered as a Filter Bank approach – we need N filters working in parallel…

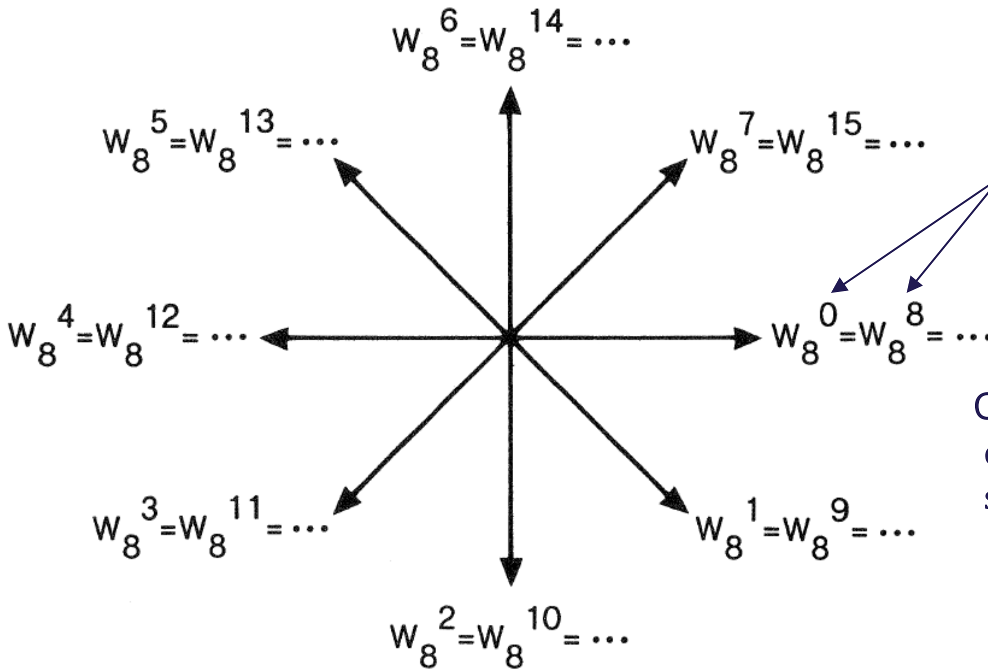
An interesting observation is, that we can choose the filter coefficients such that the filter bank calculates N harmonics which are not necessarily the DFT frequencies.

…and maybe better yet, we don't need to wait starting the calculations until we have collected N samples from $x[n]$ – we can start immediately once $x[0]$ is present at the output of the ADC.

Reducing the computational complexity
- utilize the symmetry and periodicity in the twiddle factor

$$\text{DFT:} \quad X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad k = 0, 1, \ldots N-1$$

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



Illustrated as vectors in the complex plane.

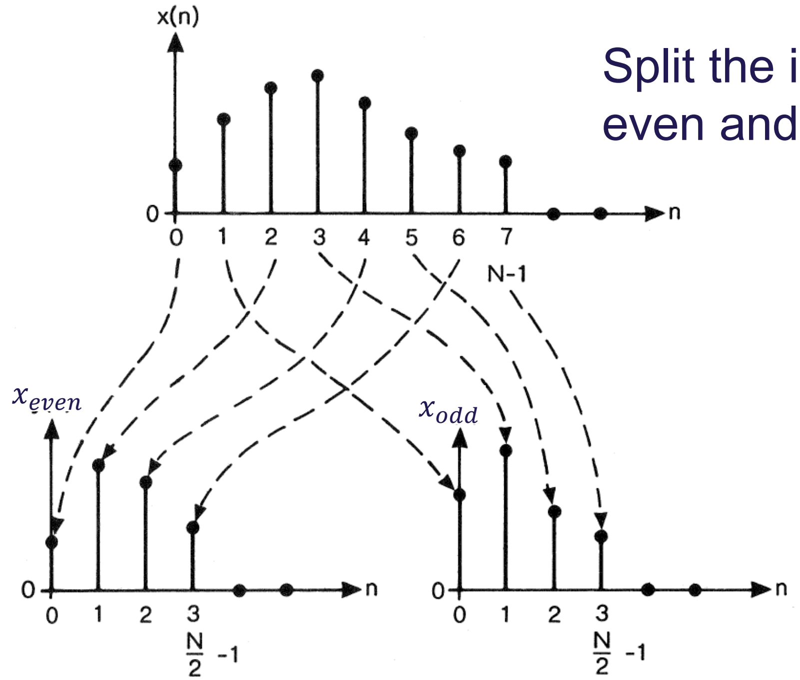
The twiddle factors repeat themselves with period N, here examplified for N=8.

Can be shown using Euler and $\cos(-x) = \cos(x)$ and $\sin(-x) = -\sin(-x)$

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

Periodicity in n og k: $W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n}$

The Fast Fourier Transform, FFT
- explained here a bit different than done in O&S



Split the input sequence into even and odd numbered samples.

$$X[k]=\sum_{n=0}^{N-1} x[n]W_N^{nk}, \quad k=0,1,\ldots N-1$$

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

The Fast Fourier Transform, FFT

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

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

We now use the following relation; $W_N^2 = (e^{-j\frac{2\pi}{N}})^2 = e^{-j\frac{2\pi}{N/2}} = W_{N/2}$

$$X[k] = \sum_{n=0}^{\frac{N}{2}-1} x[2n]W_{N/2}^{nk} + \sum_{n=0}^{\frac{N}{2}-1} x[2n+1]W_{N/2}^{nk}\, W_N^k$$

$$X[k] = \sum_{n=0}^{\frac{N}{2}-1} x_1[n]W_{N/2}^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_2[n]W_{N/2}^{nk}$$

Here we have introduced two new sequences, $x_1[n]$ and $x_2[n]$, holding the even and odd samples of $x[n]$.

AALBORG UNIVERSITY
DENMARK

The Fast Fourier Transform, FFT

$$X[k] = \sum_{n=0}^{\frac{N}{2}-1} x_1[n] W_{N/2}^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_2[n] W_{N/2}^{nk}$$

The two sums represent the $\frac{N}{2}$ - point DFT of $x_1[n]$ and $x_2[n]$, respectively, i.e., $X_1[k]$ and $X_2[k]$. Therefore, we can write $X[k]$ as;

$$X[k] = X_1[k] + W_N^k \; X_2[k]$$

Now, there is a little issue here that we need to consider…
…and it is here, I'm elaborating a bit more than O&S

$X_1[k]$ and $X_2[k]$ are defined for $0 \leq k \leq \frac{N}{2} - 1$, but $X[k]$ is defined for $0 \leq k \leq N - 1$, so how should we interprete $X[k]$ for $k \geq \frac{N}{2}$…???

Since $X_1[k]$ and $X_2[k]$ both are $\frac{N}{2}$ periodic in k, we can write $X[k]$ as;

$$X[k] = \begin{cases} X_1[k] + W_N^k \; X_2[k] & 0 \leq k \leq \frac{N}{2} - 1 \\ X_1\left[k - \frac{N}{2}\right] + W_N^k X_2\left[k - \frac{N}{2}\right] & \frac{N}{2} \leq k \leq N - 1 \end{cases}$$

AALBORG UNIVERSITY
DENMARK

The Fast Fourier Transform, FFT

$$X[k] = \begin{cases} X_1[k] + W_N^k \; X_2[k] & 0 \le k \le \frac{N}{2} - 1 \\ X_1\left[k - \frac{N}{2}\right] + W_N^k X_2\left[k - \frac{N}{2}\right] & \frac{N}{2} \le k \le N - 1 \end{cases}$$

Next, we note that $W_N^{k-\frac{N}{2}} = W_N^k W_N^{-\frac{N}{2}} = -W_N^k$ and therefore we can rewrite $X[k]$;

$$X[k] = \begin{cases} X_1[k] + W_N^k \; X_2[k] & 0 \le k \le \frac{N}{2} - 1 \\ X_1\left[k - \frac{N}{2}\right] - W_N^{k-\frac{N}{2}} X_2\left[k - \frac{N}{2}\right] & \frac{N}{2} \le k \le N - 1 \end{cases}$$

For all k, the twiddle factors in the two equations, respectively, are identical – only difference is the sign.

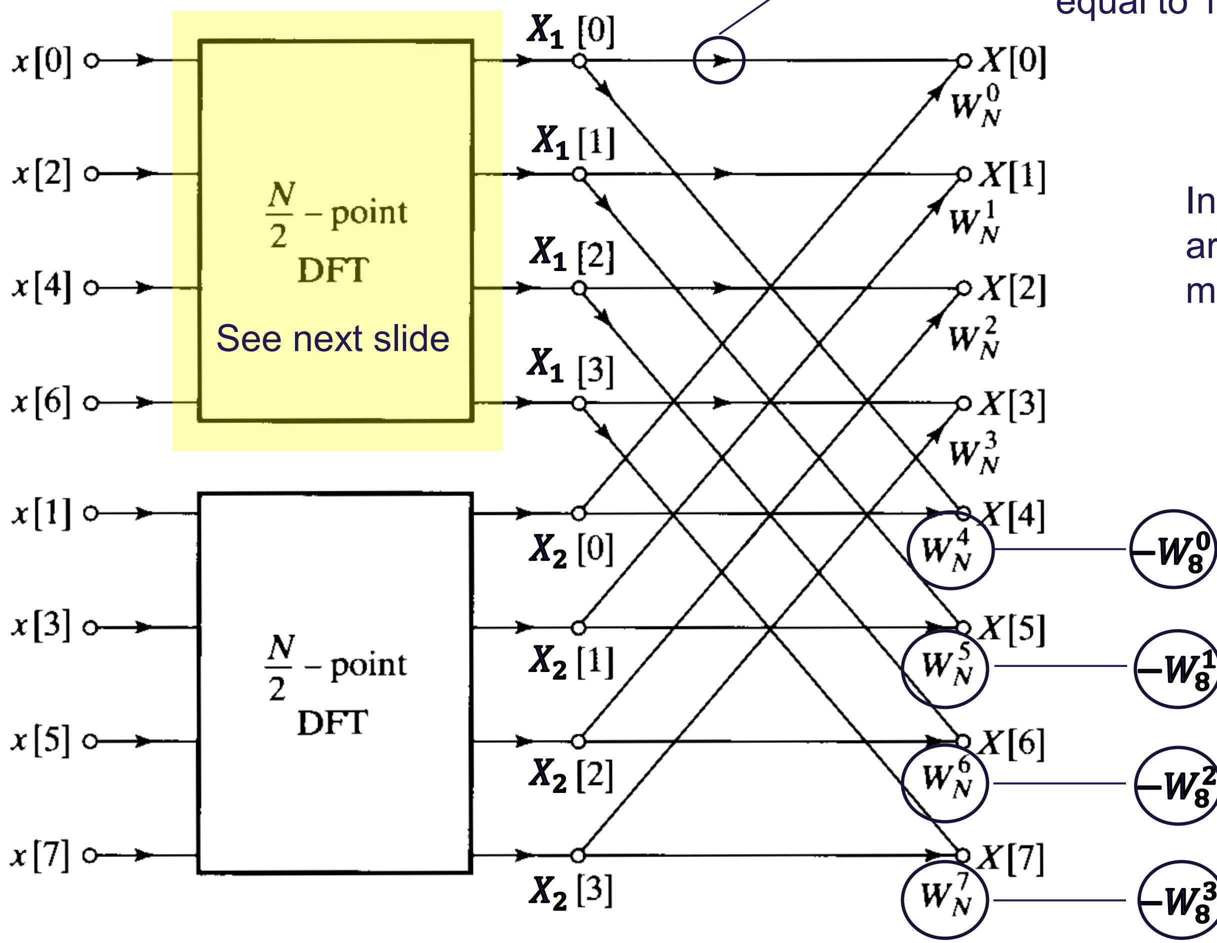
Let's draft a figure, illustrating how $X[k]$ is calculated for $N = 8$…

AALBORG UNIVERSITY
DENMARK

The Fast Fourier Transform, FFT

$$X[k] = \begin{cases} X_1[k] + W_N^k \, X_2[k] & 0 \le k \le \frac{N}{2} - 1 \\ X_1\left[k - \frac{N}{2}\right] - W_N^{k-\frac{N}{2}} X_2\left[k - \frac{N}{2}\right] & \frac{N}{2} \le k \le N - 1 \end{cases}$$

The arrows denote ”multiplication”, and if no multiplier is equal to 1, i.e., a direct transmission.



In the upper part, $X_1[k]$ and $X_2[k]$ are added after $X_2[k]$ has been multiplied with the twiddle factor

In the lower part, we multiply with the same twittle factors as in the upper part, but the two terms are subtracted.

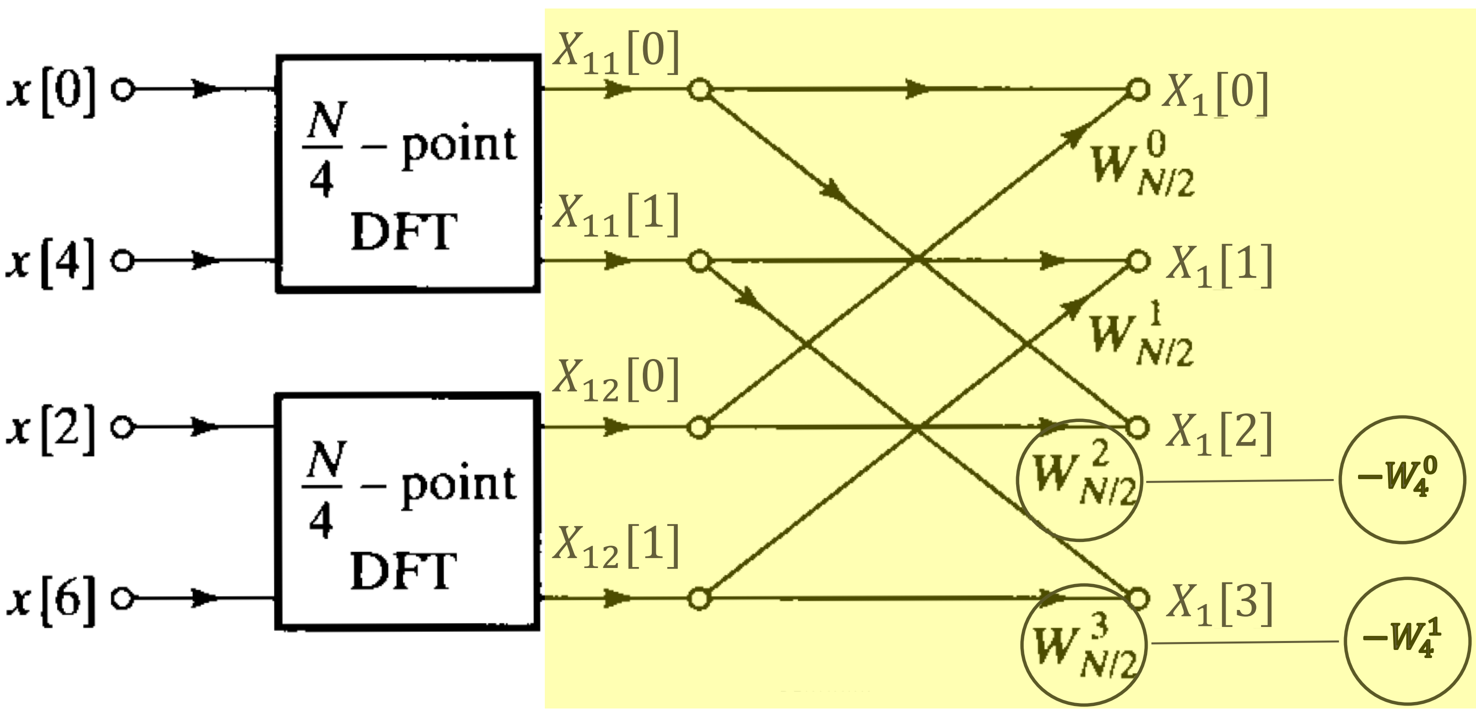
The Fast Fourier Transform, FFT

After this first "break-down" of the DFT, we now have two $\frac{N}{2}$ -points DFTs which we next re-organize using exactly the same procedure (see. p. 22), i.e.,

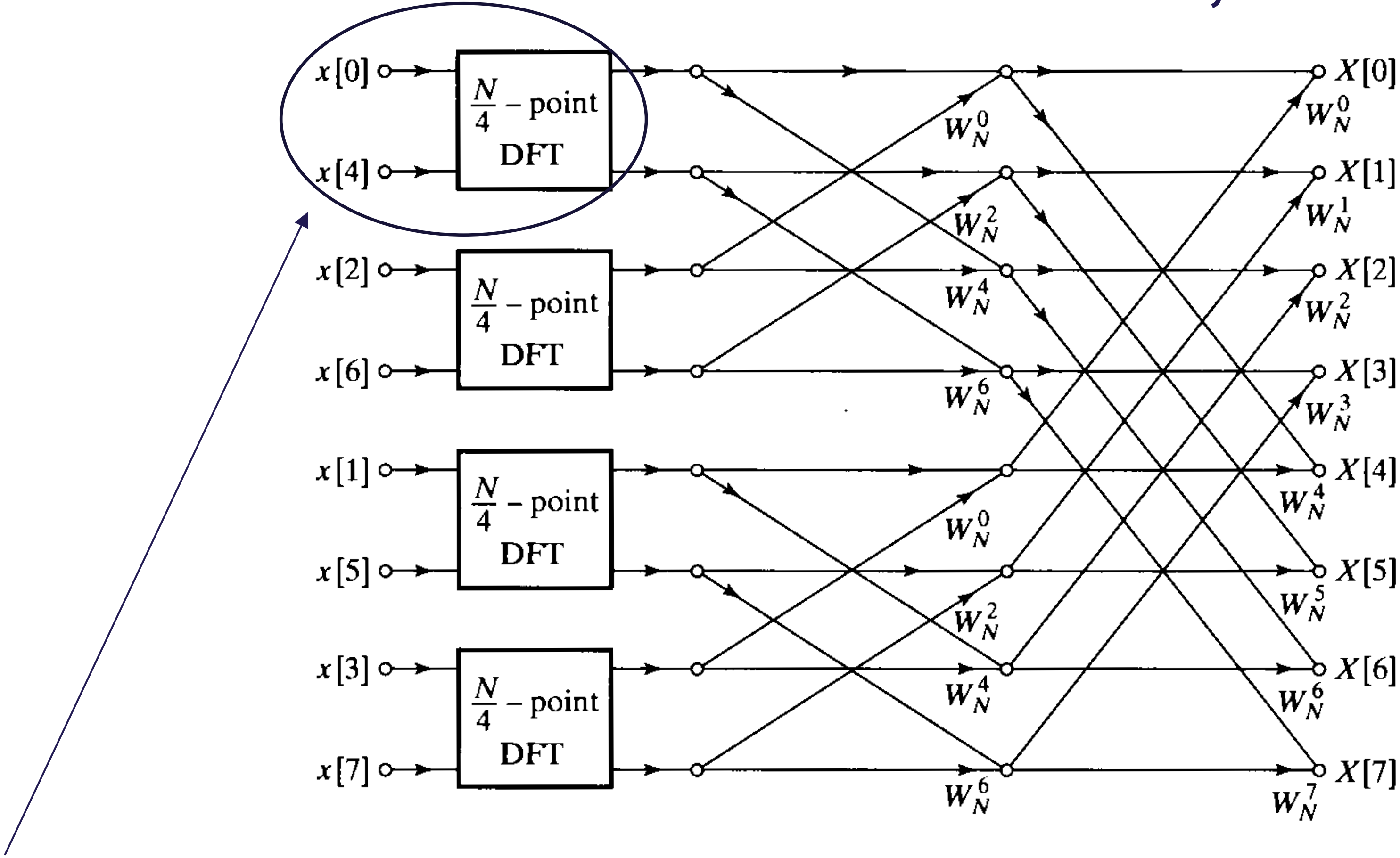
$$X_1[k] = X_{11}[k] + W_{N/_2}^{k} X_{12}[k] \quad \Rightarrow X_1[k] = \begin{cases} X_{11}[k] + W_{N/_2}^{k} X_{12}[k] & 0 \leq k \leq \frac{N}{4} - 1 \\ X_{11}\left[k - \frac{N}{4}\right] - W_{N/_2}^{k-\frac{N}{4}} X_{12}\left[k - \frac{N}{4}\right] & \frac{N}{4} \leq k \leq N - 1 \end{cases}$$

…and similarly for $X_2[k]$.

Again, let's illustrate the equations for $N = 8$, here the calculation of $X_1[k]$;



The Fast Fourier Transform, FFT



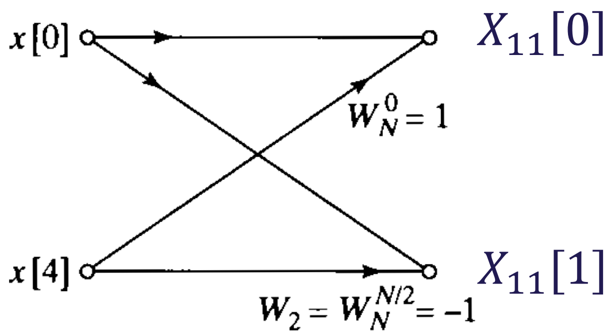
The "break-down" procedure is now repeated for the $\frac{N}{4}$ -point DFTs, and is continued until we obtain 2-point DFTs – then we cannot break it down any further…

For $N = 8$ we are already there…

The Fast Fourier Transform, FFT

The 2-point DFT, is often denoted ”a butterfly” computation…

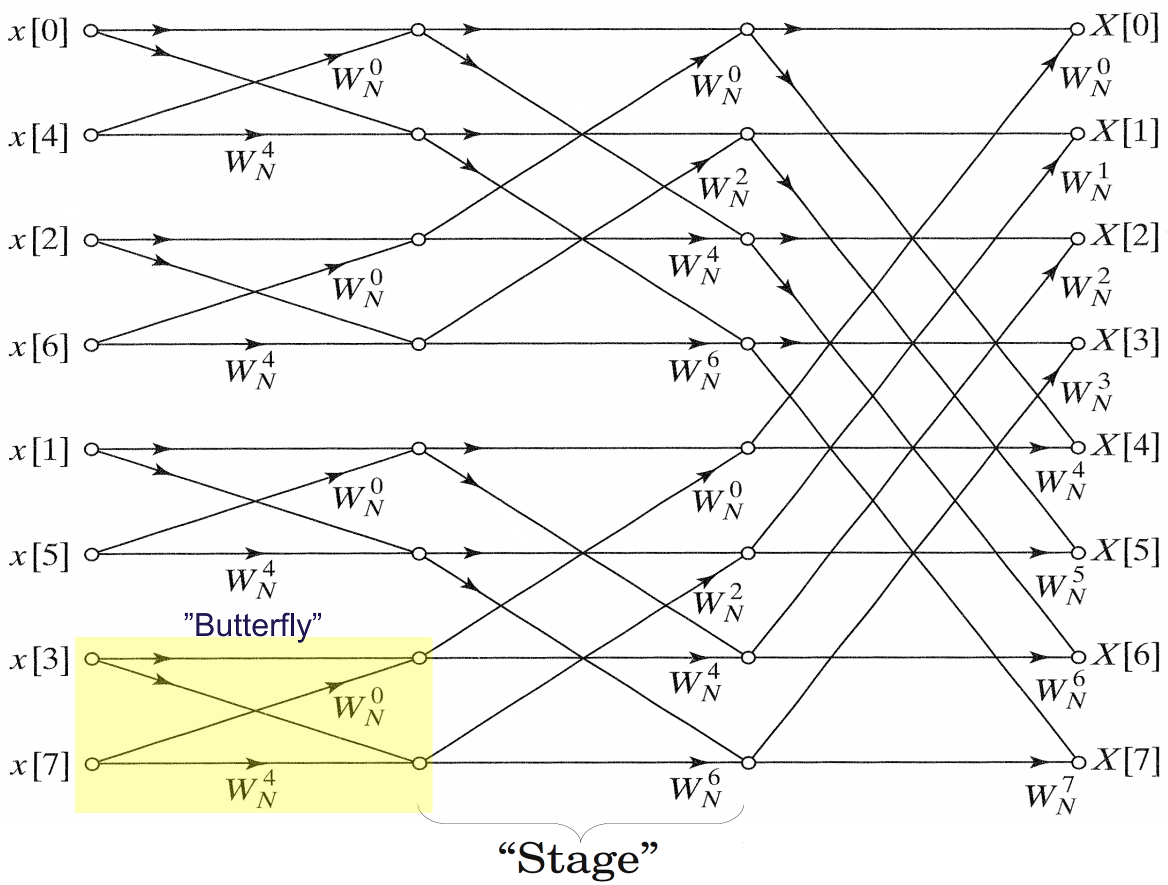
Here examplified by the circled 2-point DFT from p. 25.



The Fast Fourier Transform, FFT

What we see here in this complete break-down of the DFT is that ALL computations are butterfly computations – so, the butterfly is the core computation of the Fast Fourier Transform, i.e., the inner-loop…

For each break-down, we obtain what we call a ”stage” in the FFT, and there is a total of $\log_2 N$ stages – and therefore, we normally wish N to be a power of 2, e.g., 128, 256, 512, 1024, or…



The computational complexity of the FFT

The ordinary DFT: N^2 complex mult and add

1st break-down: $N + 2(N/2)^2$

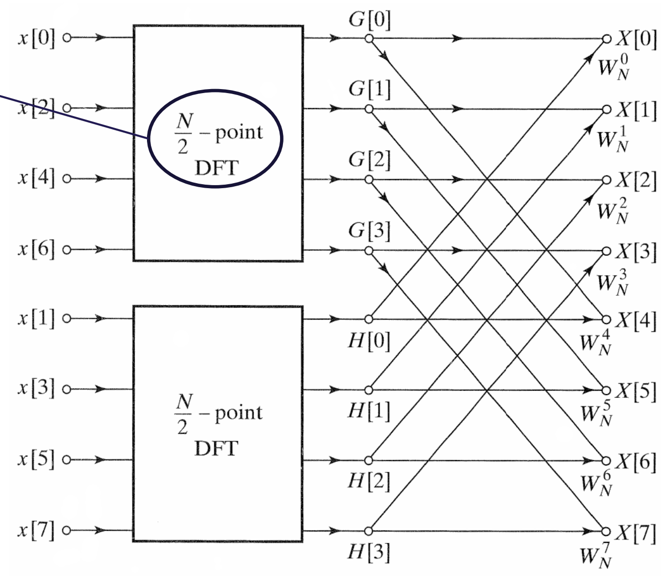
2nd break-down: $N + N + 4(N/4)^2$

.

.

.

With full break-down: $Nlog_2(N)$



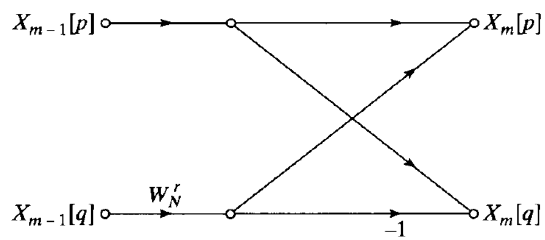
We have N mult/add for every stage, and after the full break-down we have a total of $log_2(N)$ stages, thus $\mathcal{O}(Nlog_2(N))$.

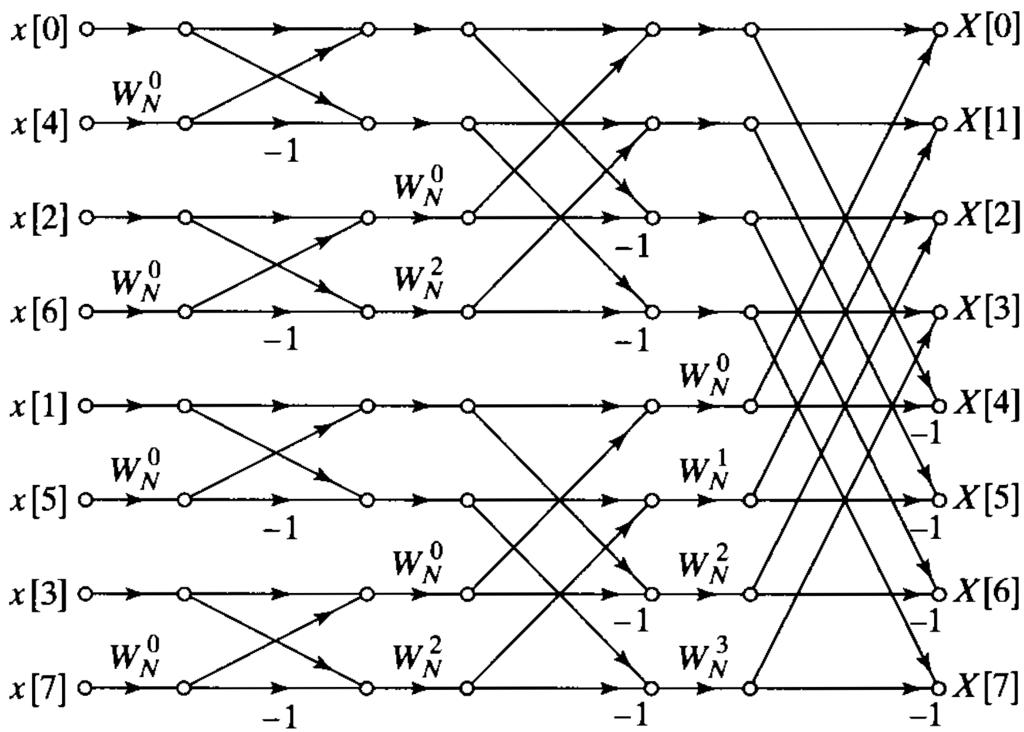
$Nlog_2(N) \ll N^2$ for "large" values of N, e.g., 2048 vs. 65536 for $N = 256$

AALBORG UNIVERSITY
DENMARK

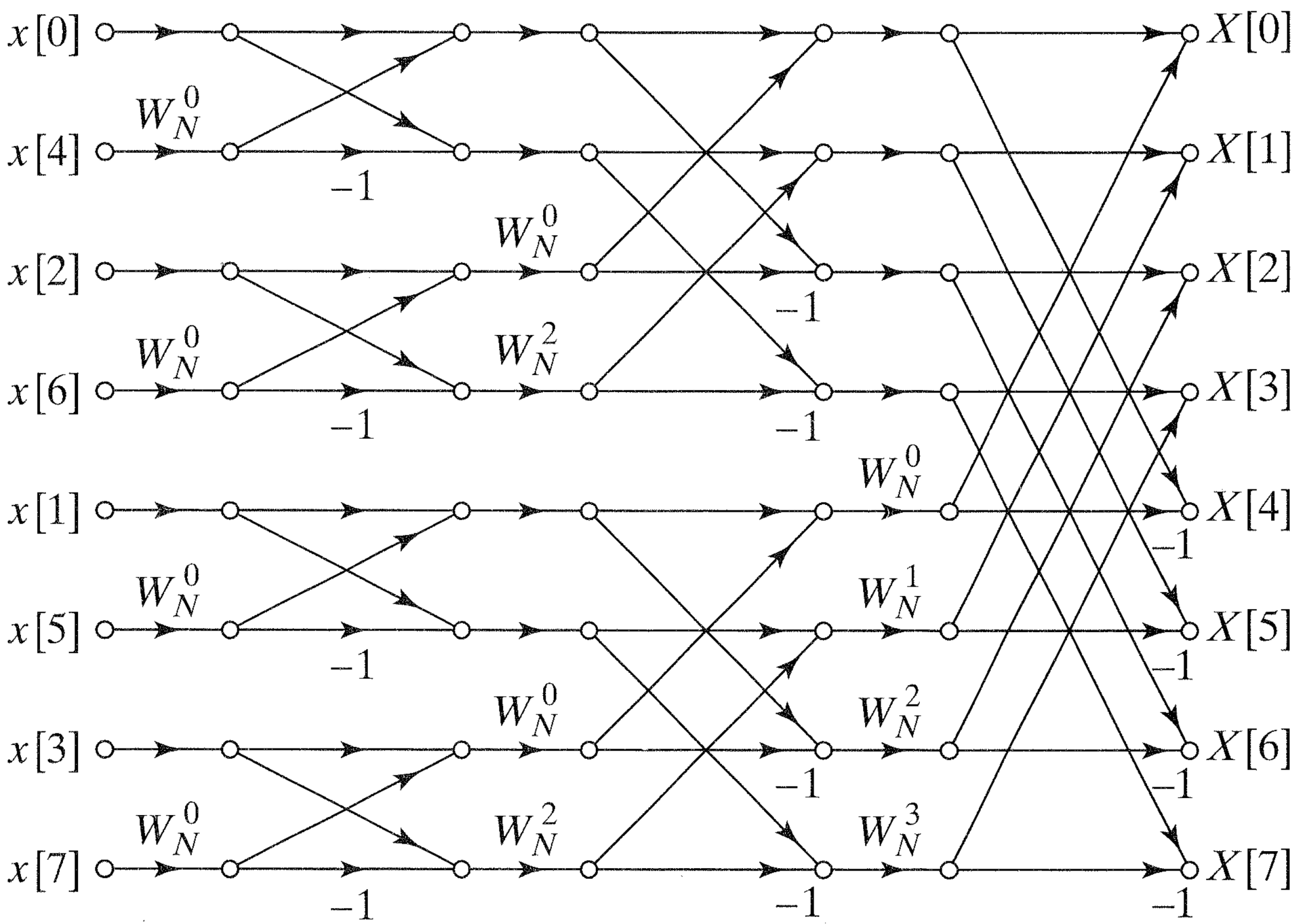
The Fast Fourier Transform, FFT

As we have discussed, for each stage, the upper and lower twiddle factors are identical, except for the sign. Therefore, we can rewrite the butterfly such that we multiply only once with the twiddle factor, and then do addition and subtraction.





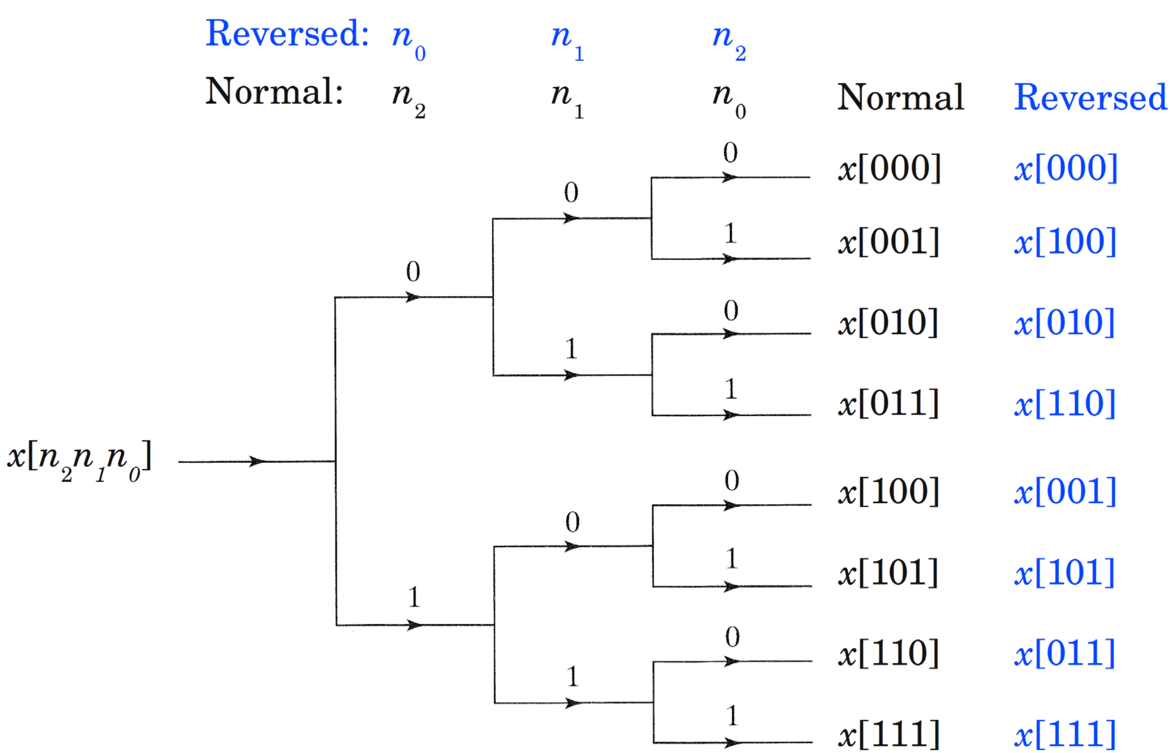
The final 8-point Decimation In Time FFT



AALBORG UNIVERSITY
DENMARK

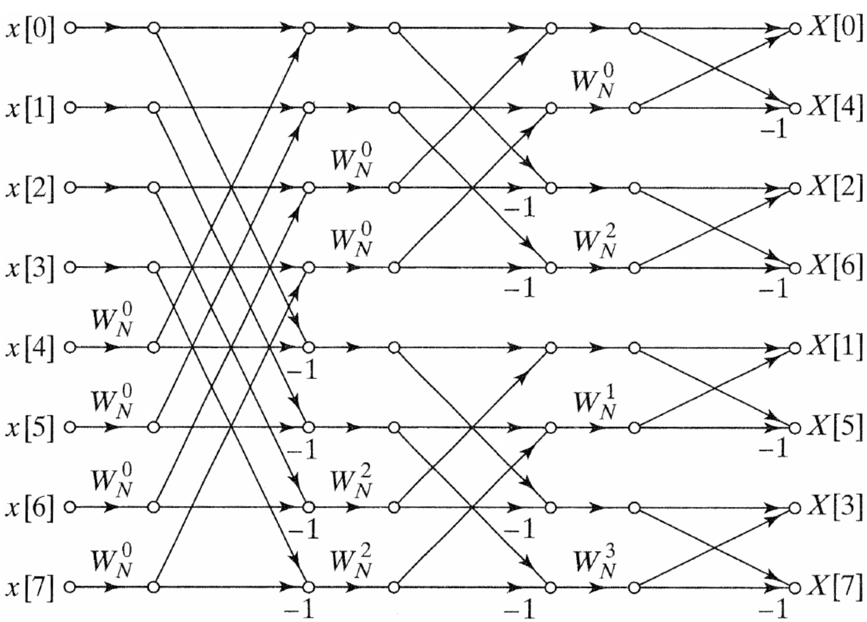
Scrambling (or coding) of input – Bit Reversing

Decimal	Binary	Binary reversed	Scrambed
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

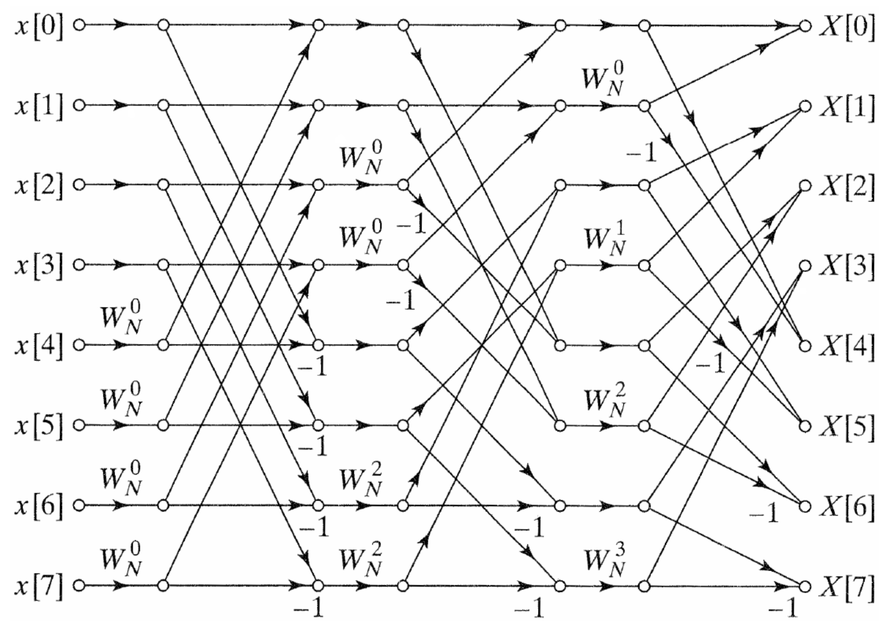


Alternative Representations
Decimation In Frequency

Simple change to sorted input and scrambled output:



...and with sorted output (messy!):



So, the price for ”order” externally is ”disorder” internally…
Normally it is much easier to do bit reversing of in- or output.

AALBORG UNIVERSITY
DENMARK

The Inverse FFT

It can be shown that

$$x[n] = \frac{1}{N}\left\{\sum_{k=0}^{N-1} X^*[k] W^{kn}\right\}^*$$

Hence, the exact same algorithm can be applied for inverse FFT, if you:

1) Complex conjugate the input to $X^*[k]$, i.e. change sign on the imaginary part of $X[k]$.

2) Complex conjugate the output and divide this output by N.

AALBORG UNIVERSITY
DENMARK

FFT – Practical Considerations

In practice you will normally find FFT algorithms included as ready-made routines (C, ASM, VHDL, …) in your IDE – both for CPU/MCU, DSP, FPGA, …

…and in most cases it is difficult to compete these packages…(!)

However, you may want to design your own FFT routine, and if so, then you may want to get inspiration from the FFTW package, which is a collection of free FFT routines inplemented in C, and which can handle any size N,
see http://fftw.org

…or simply just use Matlab…!