Signalbehandling for computer-ingeniører

COMTEK-5, E22

&

Signalbehandling

EIT-5, E22

13. Efficient calculation of the DFT – the Goertzel Algorithm and the Fast Fourier Transform

Assoc. Prof. Peter Koch, AAU

Remember the Discrete Fourier Transform, DFT

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

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

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

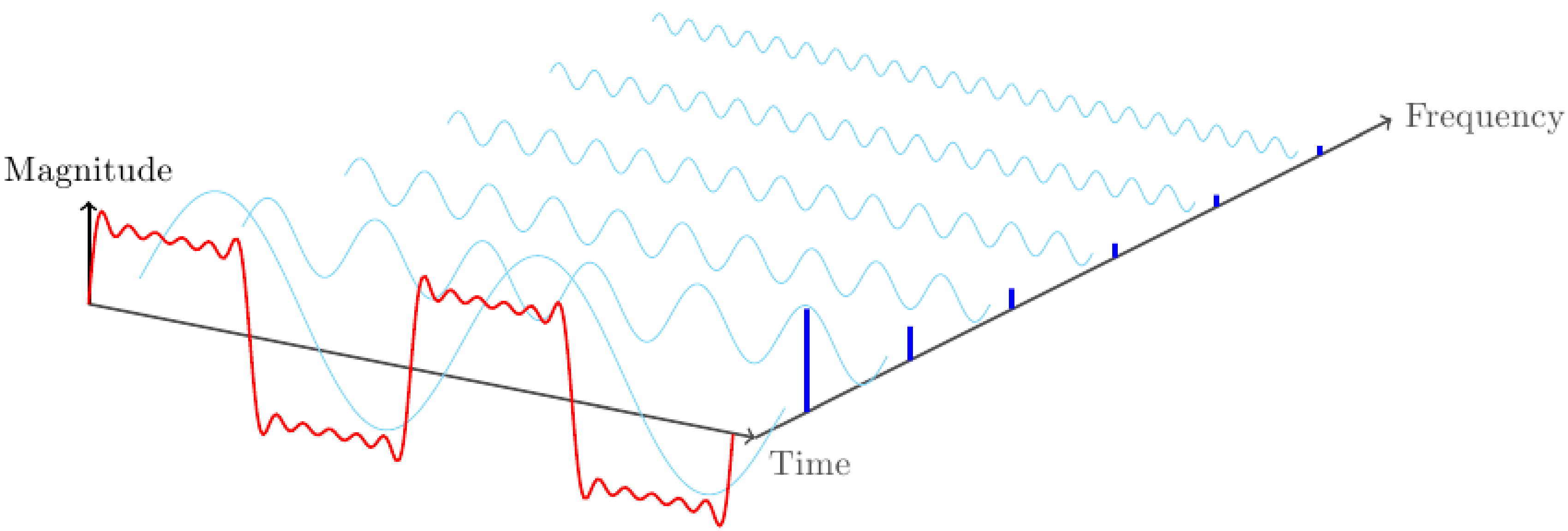
That is, the fact that $X[k] = 0$ for k outside the interval $0 \le k \le N-1$ and that $x[n] = 0$ for n outside the interval $0 \le n \le N-1$ is implied, but not always stated explicitly.

Twiddle Factor for the fundamental frequency ${}^{2\pi}/_{N}$; $W_N = e^{-j(2\pi/N)}$

How should we understand the Analysis Equation

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

Basically, this equation takes N samples from the time domain and maps them into N samples in the frequency domain.



This picture however, represents an ideal situation…

How should we understand the Analysis Equation

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

Since $0 \le n \le N-1$ it means that we analyse only a segment (of length N) of the signal.

This means essentially, that the segment can be considered as the infinite signal ”seen through a window” of length N.

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

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

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

This is the Discrete-Time Fourier Transform, see Table 2, p. 60

How should we understand the Analysis Equation

The DFT is derived by using the Analysis Equation – or alternatively we could also find it by sampling the Discrete-Time Fourier Transform, i.e.;

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

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

$$\boxed{\Omega_k = {}^{2\pi k}/_{TN} \;\Rightarrow\; f_k = k\frac{f_s}{N} \quad 0 \le k \le N-1}$$

So, from this we realize that ”the size” of the DFT (N point) has to have a certain value in order to produce a spectrum with a ”sufficient” accuracy in the interval from DC to the sample frequency f_s.

Numerical considerations – an example

Given a band-limited continuous-time signal $x_c(t)$, i.e., $X_c(j\Omega)$.

Assume $|X_c(j\Omega)| = 0$ for $|\Omega| > 2\pi \cdot 2500\ rad/s$

The sampling frequency is $f_s = 5\ kHz$, and thus $T = {}^{1}/_{5000}\ sec$.

We now want to calculate the DFT of $x_c[n]$ with a spectral resolution equal to or better than $10\ Hz$.

How many samples are needed in the DFT..???

$$\Omega_k = \frac{\Omega_s}{N} = \frac{2\pi f_s}{N} = \frac{2\pi}{NT} \leq 2\pi \cdot 10\ rad/sec$$

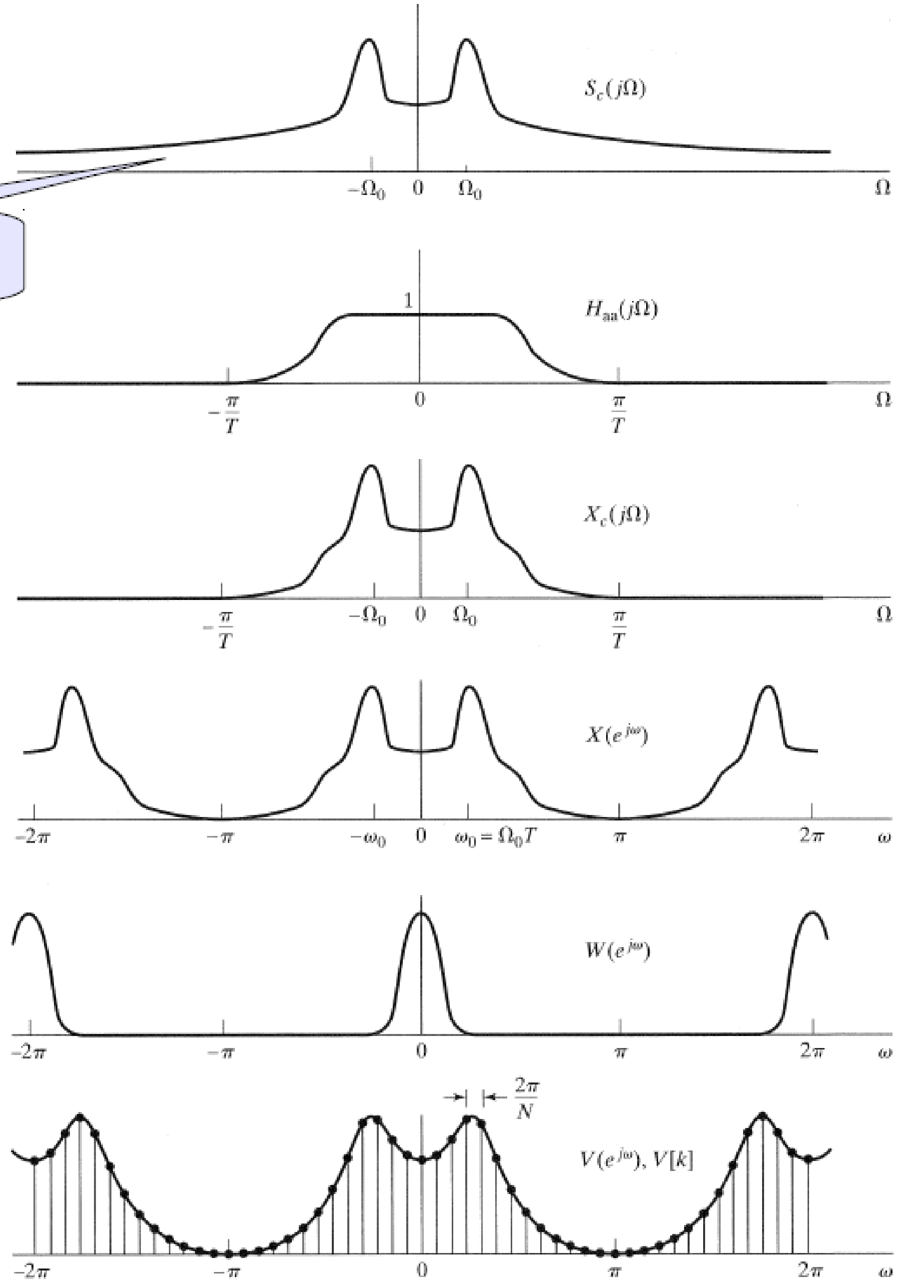
Spectral resolution

$$N \geq 500$$

AALBORG UNIVERSITY
DENMARK

From continuous time signal to the DFT spectrum

Wide-band



Considerations on the Computational Complexity

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

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 is $O(N^2)$

AALBORG UNIVERSITY
DENMARK

The first approach; The Goertzel Algorithm

First, we note that the DFT can be formulated in terms of a convolution

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

Twiddle factor

$$= \sum_{n=0}^{N-1} x[n] W_N^{-k(N-n)}, \qquad W_N^{-kN} = 1$$

Multiply with 1

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

Substitute *n* with *r*

$$= \left(x[n] * W_N^{-nk} \right) \Big|_{n=N}$$

Discrete-time convolution. In W, we may substitute N with n, if we at the same time claim that $n = N$.

$$= \left[x[n] * \left(W_N^{-nk} u[n] \right) \right] \Big|_{n=N}$$

Since $x[n] = 0$ for $n < 0$ and $n > N-1$, we are allowed to multiply *W* with *u*[*n*].

AALBORG UNIVERSITY
DENMARK

The first approarch; The Goertzel Algorithm

$$X[k] \;=\; \left[x[n] * \left(W_N^{-nk}u[n]\right)\right]\Big|_{n=N}$$

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

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

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

$$0 \le n \le N-1$$

Therefore;

$y_k[n]$, at sample $n = N$, will provide the corresponding N-point DFT coefficient $X[k] = y_k[N]$

AALBORG UNIVERSITY
DENMARK

The first approach; The Goertzel Algorithm

Now, let's represent the filter by its z-transform

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

The impulse response

$$x[n] \longrightarrow \boxed{\sum_{m=0}^{\infty} W_N^{-mk} z^{-m}} \longrightarrow y_k[n]$$

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

The transfer function

…multiplied with 1

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

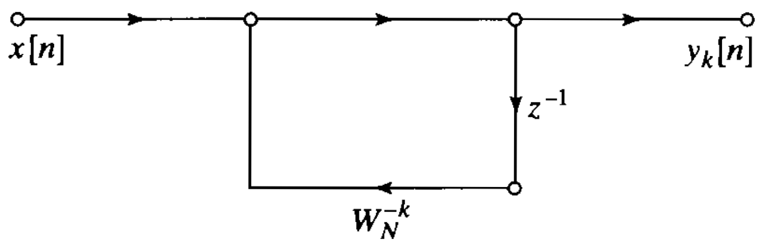
Another way of calculating the geometric series… ☺

AALBORG UNIVERSITY
DENMARK

The first approach; The Goertzel Algorithm

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

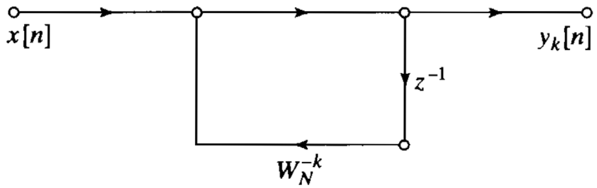
Such a system can be illustrated by the following signal flow graph;



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.

We can reduce the complexity…

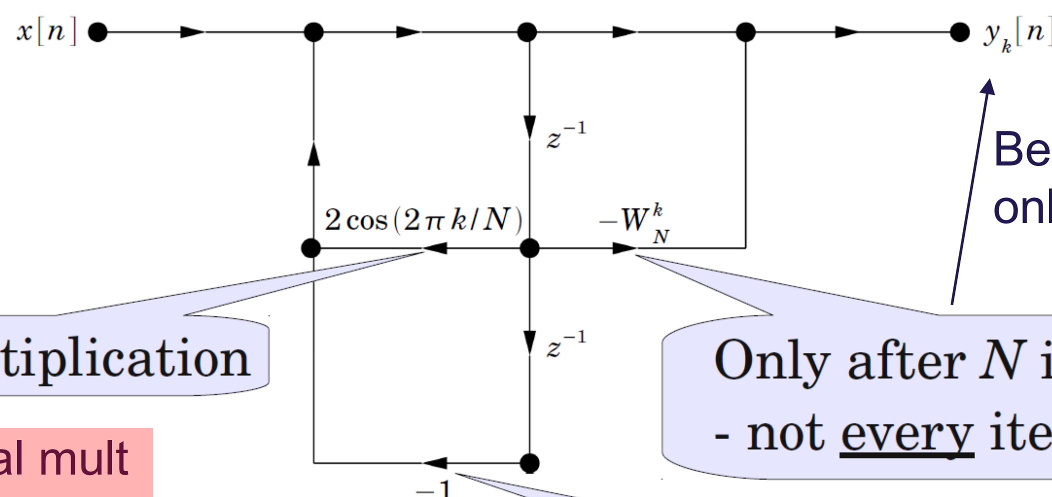


$$\mathrm{H}_k(z)=\frac{1}{1-W_N^{-k}z^{-1}}$$

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

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

with the flow graph (now a 2[nd] 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!

Simple sign change

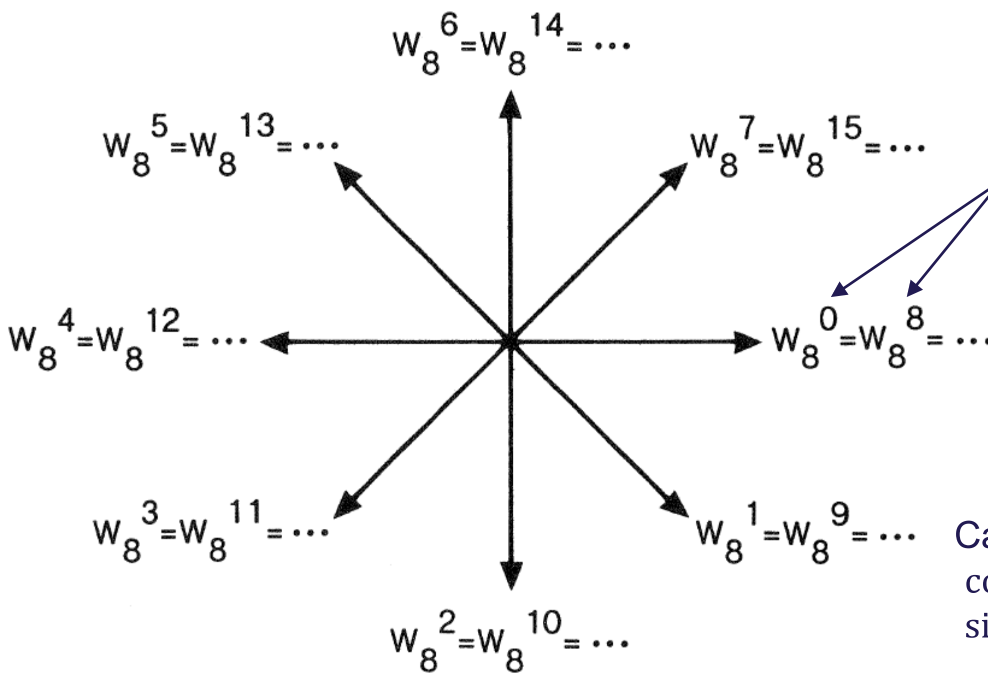
N^2 real mult
$2N^2$ real add

…but still $O(N^2)$…!!

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

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

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



Illustrated as vectors in the complex plane.

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

$N=8$

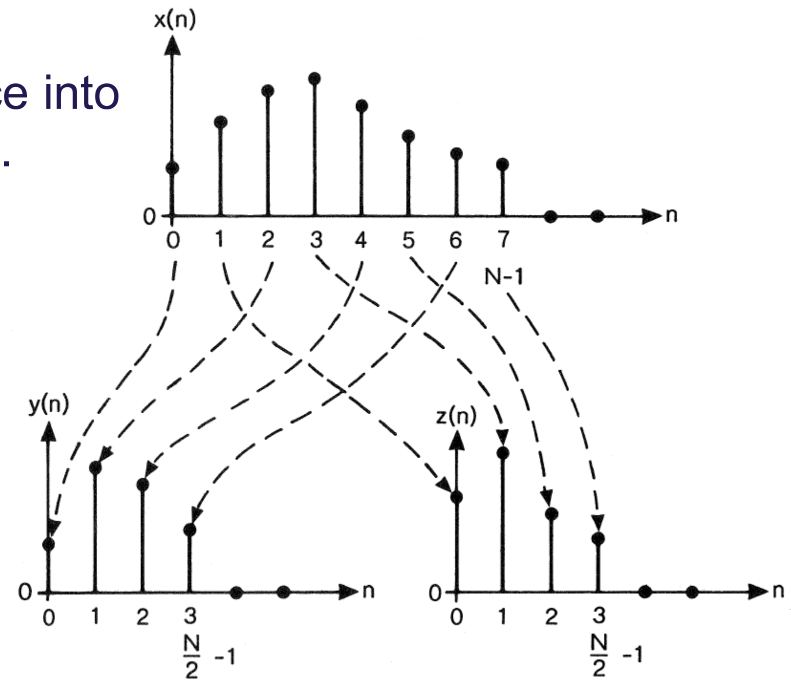
Can be shown using Euler and $\cos(-x) = \cos(x)$ and $\sin(-x) = -\sin(-x)$

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

Split the input sequence into even and odd samples.

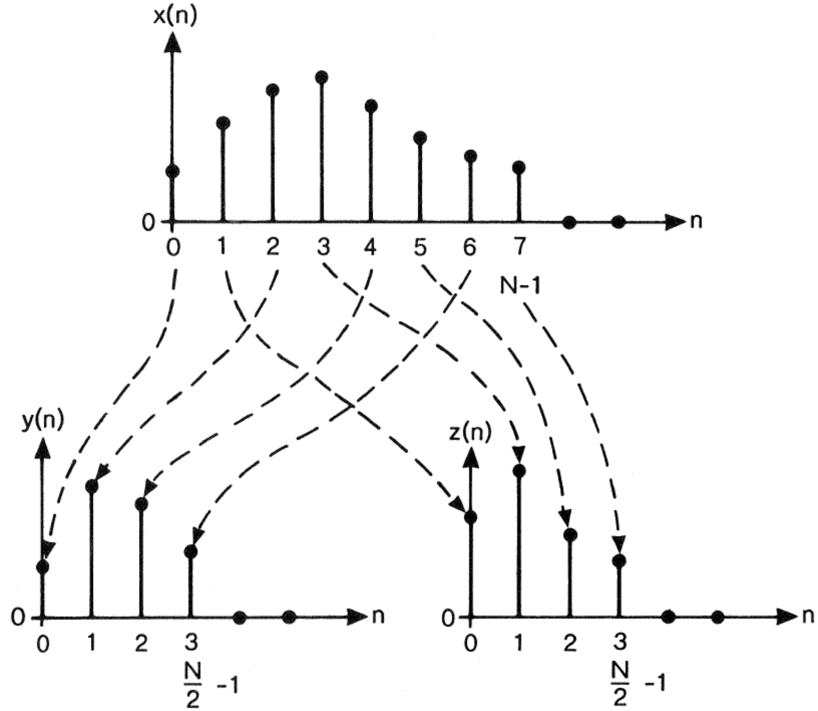


$$X[k]=\sum_{n=0}^{N-1} x[n]W_N^{nk},\quad k=0,1,\dots N-1$$

$$X[k]=\underbrace{\sum_{r=0}^{N/2-1} x[2r](W_N^2)^{rk}}_{\text{(even)}}+W_N^k\underbrace{\sum_{r=0}^{N/2-1} x[2r+1](W_N^2)^{rk}}_{\text{(odd)}},\quad \text{where } n=2r$$

We want to have identical twiddle factors in both sums, and thus it is needed to have this extra term in order to maintain $2r+1$ in the twiddle factor exponent.

Decimation in time



Selecting every M^{th} sample (here $M = 2$) in a sequence is denoted ”down sampling” or ”decimation”.

So, by decimation in time we obtain two new sequences, each with only half the number of samples as compared to the original sequence.

AALBORG UNIVERSITY
DENMARK

Decimation In Time FFT

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

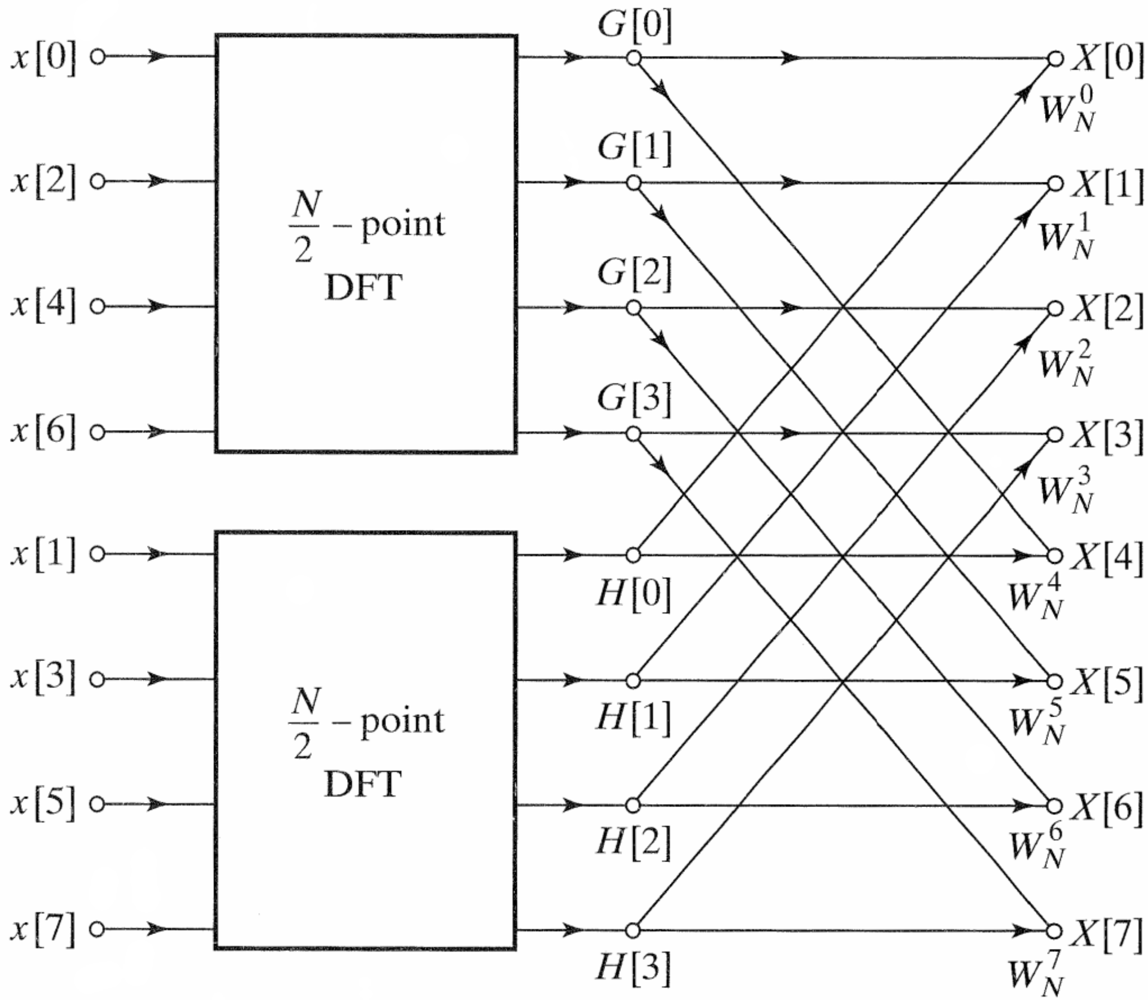
$$X[k]=\underbrace{\sum_{r=0}^{N/2-1} x[2r](W_N^2)^{rk}}_{\text{(even)}}+W_N^k \underbrace{\sum_{r=0}^{N/2-1} x[2r+1](W_N^2)^{rk}}_{\text{(odd)}}, \quad \text{where } n=2r$$

Since $W_N^2=e^{-2j(2\pi/N)}=e^{-j2\pi/(N/2)}=W_{N/2}$ the decimation can be considered as two DFTs with half length $N/2$:

$$X[k]=\sum_{r=0}^{N/2-1} x[2r]W_{N/2}^{rk}+W_N^k \sum_{r=0}^{N/2-1} x[2r+1]W_{N/2}^{rk} \Rightarrow$$

$$X[k]=G[k]+W_N^k H[k]$$

$$X[k] = G[k] + W_N^k H[k]$$

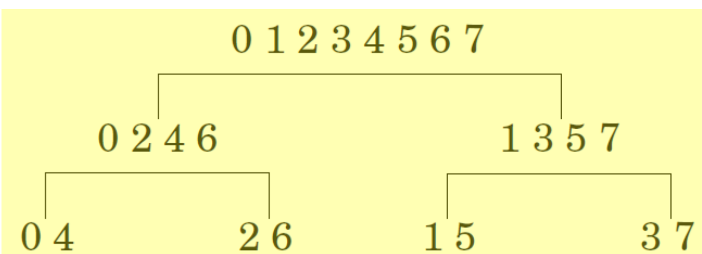


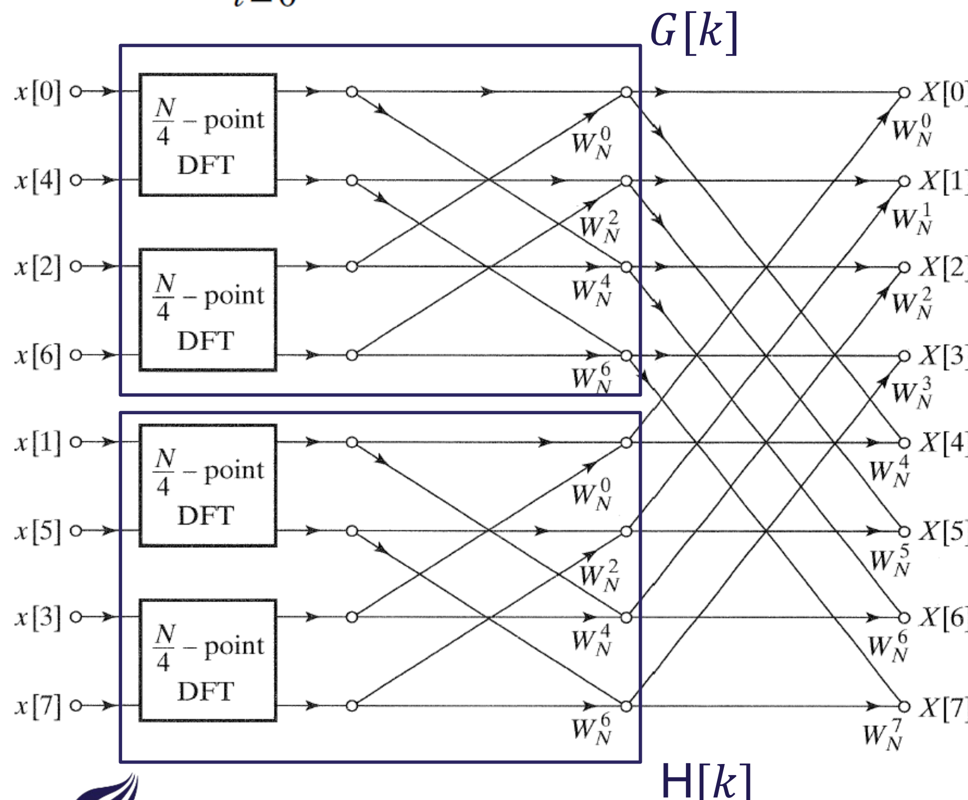
N=8

$G[k]$ and $H[k]$ can now be seperated in the same way:

$$G[k] = \sum_{l=0}^{N/4-1} g[2l] W_{N/4}^{lk} + W_{N/2}^{k} \sum_{l=0}^{N/4-1} g[2l+1] W_{N/4}^{lk} \quad \text{and}$$

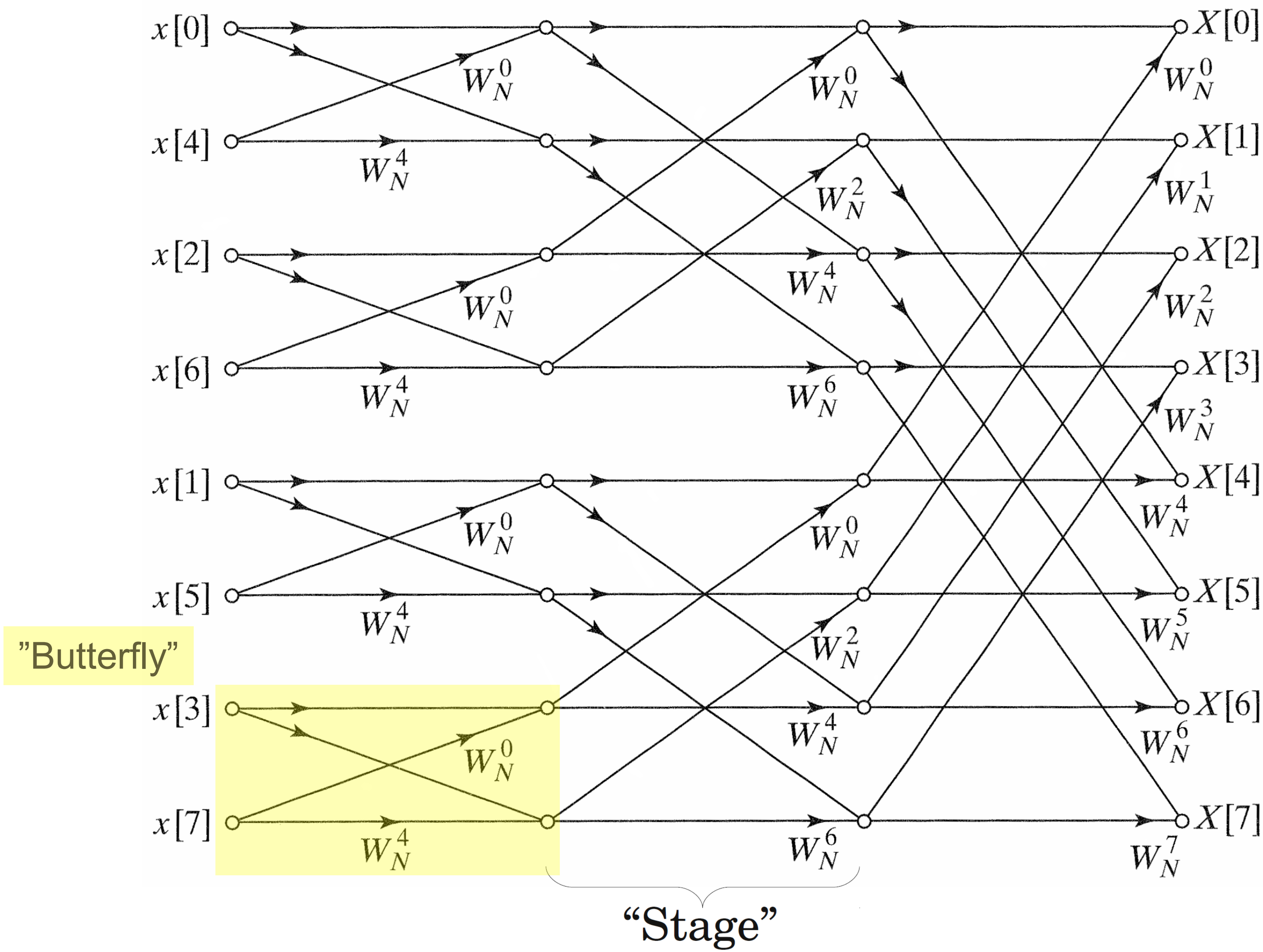
$$H[k] = \sum_{l=0}^{N/4-1} h[2l] W_{N/4}^{lk} + W_{N/2}^{k} \sum_{l=0}^{N/4-1} h[2l+1] W_{N/4}^{lk}$$





AALBORG UNIVERSITY
DENMARK

…the final break-down into an 8-point FFT



The computational complexity

The ordinary DFT: N^2

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

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

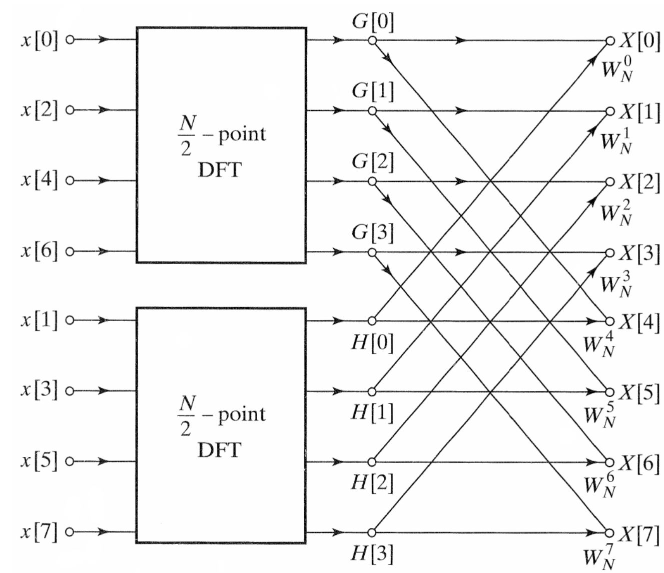
.

.

.

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

$$X[k] = G[k] + W_N^k H[k]$$

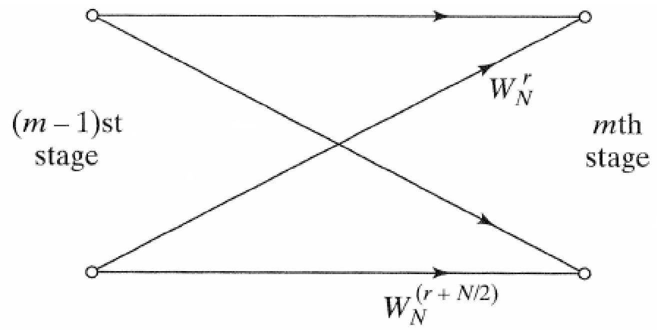


N=8

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

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

The 2-point FFT – the Butterfly



See e.g., p.20

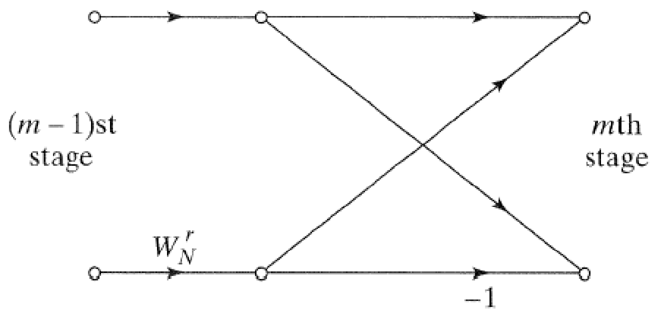
Since

$$W_N^{N/2} = e^{-j(2\pi/N)N/2} = e^{-j\pi} = -1$$

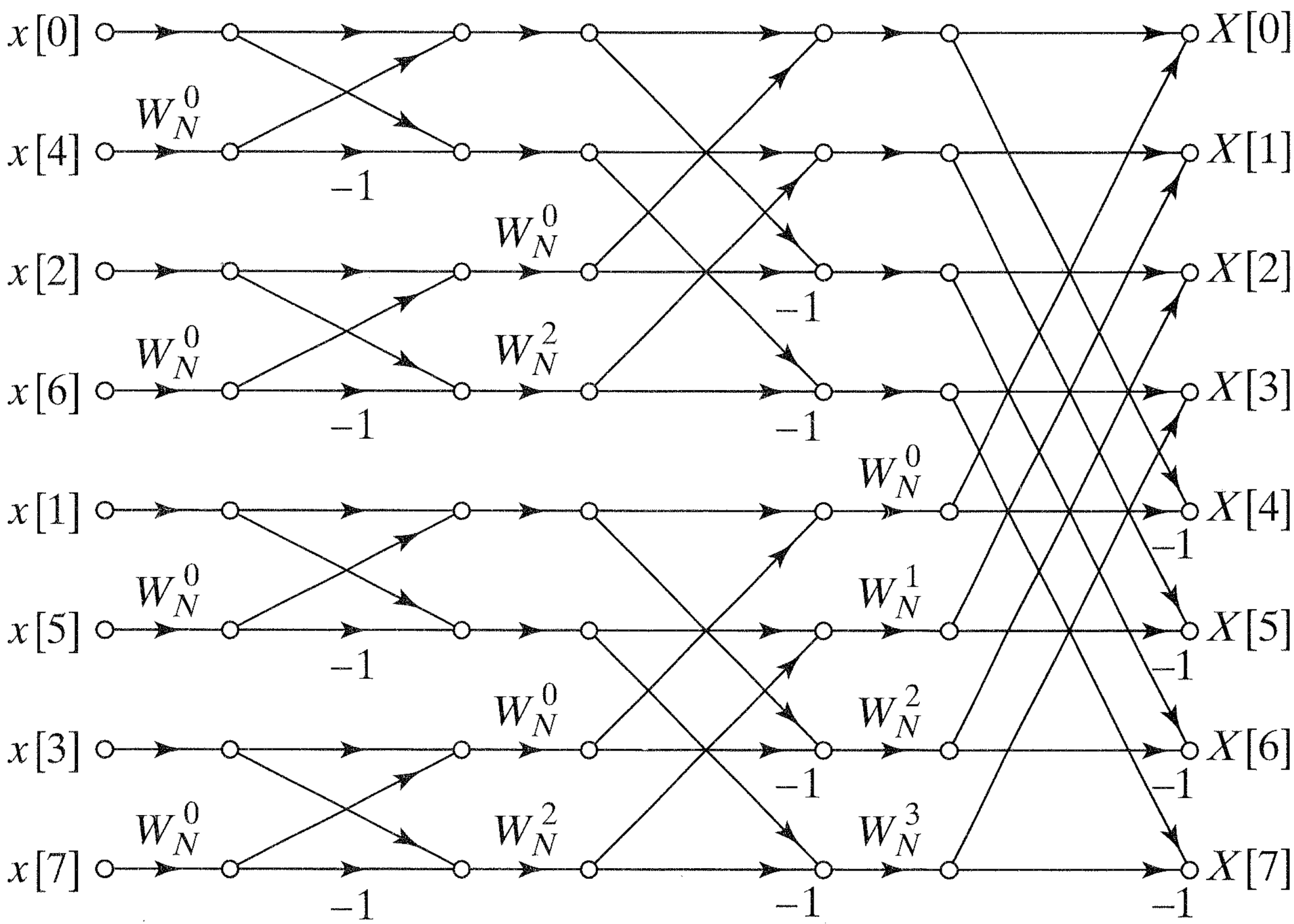
then $W_N^{r+N/2}$ can be written as

$$W_N^{r+N/2} = W_N^{N/2} W_N^r = -W_N^r$$

and the simplified Butterfly becomes:



The final 8-point Decimation In Time FFT



AALBORG UNIVERSITY
DENMARK

"In Place" computation in the FFT

Rewriting to $X_m[l]$, where m is the "stage" (column), and l is the row. In the 8-points case we then have:

$$X_0[0] = x[0]$$

$$X_0[1] = x[4]$$

$$X_0[2] = x[2]$$

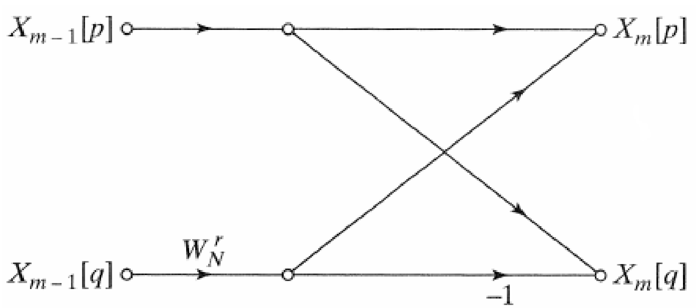
$$\vdots$$

$$X_0[7] = x[7]$$

All calculations follow:

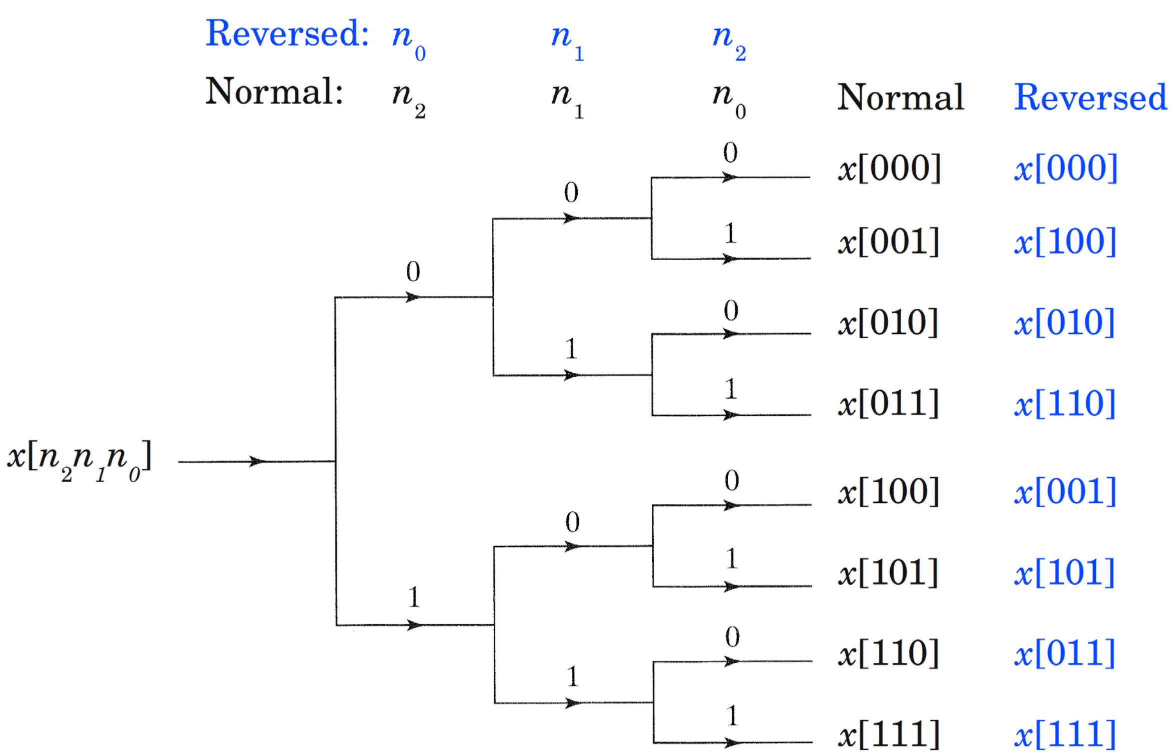
$$X_m[p] = X_{m-1}[p] + W_N^r X_{m-1}[q]$$

$$X_m[q] = X_{m-1}[p] - W_N^r X_{m-1}[q]$$



Scrambling (or coding) of input – Bit Reversing

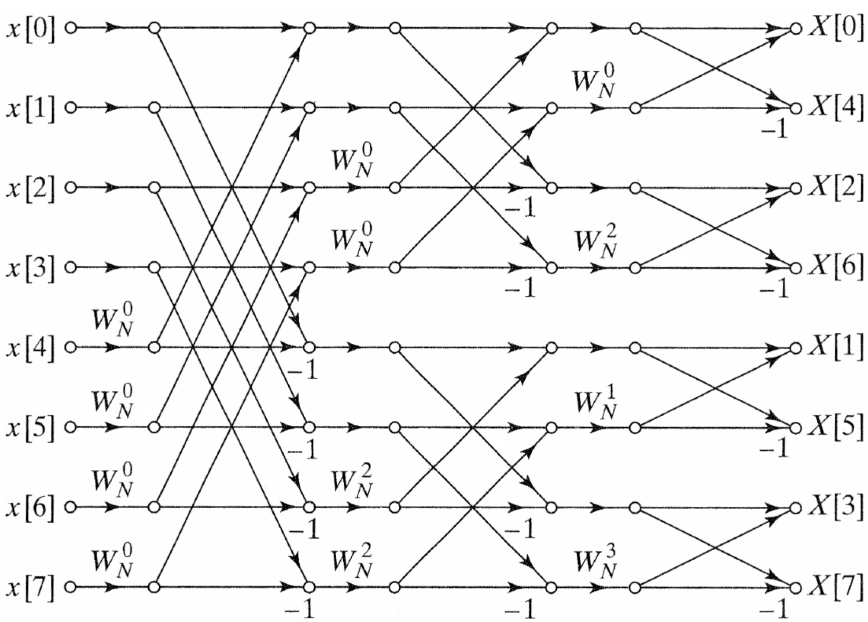
Decimal	Binary	Binary reversed	Scrambed
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7



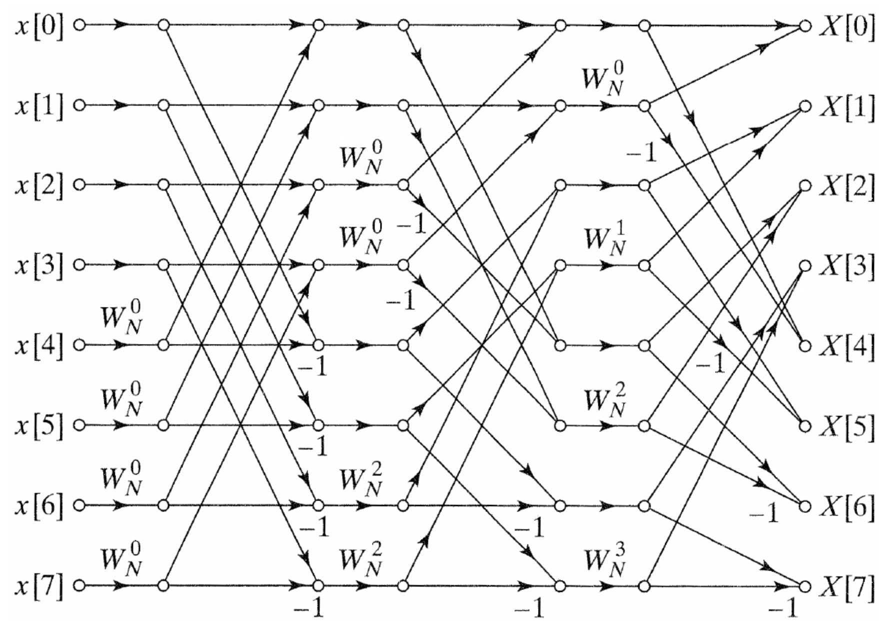
Alternative Representations

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.

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.

FFT – Practical Considerations

In practise you would rarely write your own FFT algorithm but use existing ones.

Efficient FFT algorithms usually come with commercial DSP development cards etc.

On "higher level" (e.g. C-programming) for execution on a PC, free software exists, most notably the FFTW package

http://fftw.org/

which can handle FFTs of <u>any</u> size N.

…or simply just use Matlab…!

AALBORG UNIVERSITY
DENMARK