

Digitale Signalverarbeitung auf FPGAs

Kap. 3a – Diskrete Fouriertransformation (Grundlagen)

2016

Dr. Christian Münker

- Die Fourier-Familie
- DFT: Komplex aber nicht kompliziert
- Frequenzauflösung
- Schnell, schneller, FFT

Zusätzliches Material zur Wiederholung

- **edx-Kurs „ELEC301x: Discrete Time Signals and Systems“**

https://courses.edx.org/courses/RiceX/ELEC301x/T1_2014/

(Kostenloser Account erforderlich)

- **Videos von Jörg Lovisach:**

- 19.03.1: Diskrete Fourier-Transformation, FFT Teil 1

(www.youtube.com/watch?v=ls5MfqLZVbo)

- 19.03.2 Diskrete Fourier-Transformation, FFT Teil 2

(www.youtube.com/watch?v=wd6hAPS4ILc)

Digitale Signalverarbeitung auf FPGAs

Kap. 3a Diskrete Fouriertransformation (Grundlagen)

Teil 1 *Die Fourier-Familie*

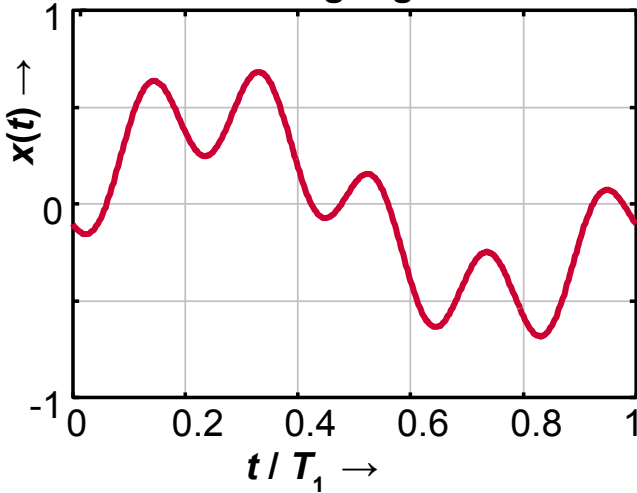
2016

Dr. Christian Münker

Spektrum abgetasteter Signale (Wdh.)

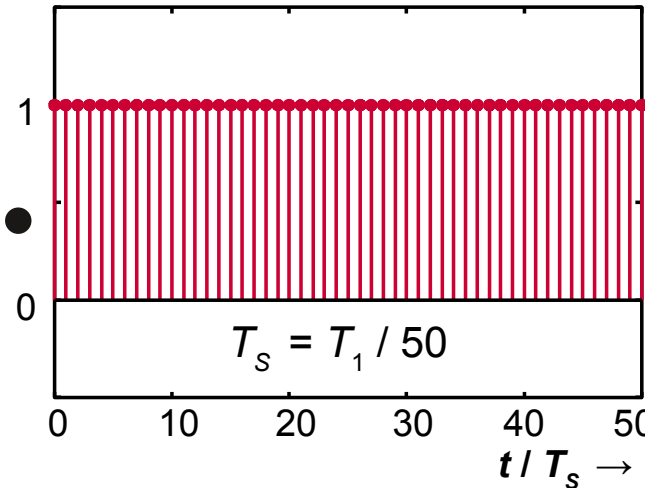


Analog Signal

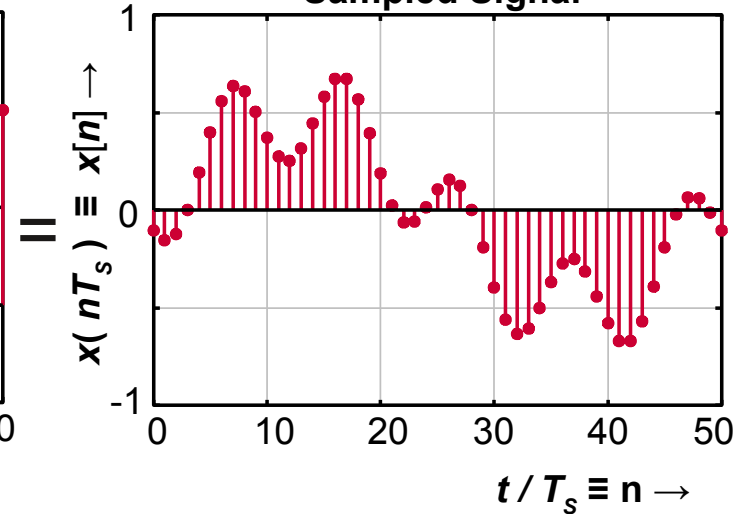


$$\sin\left(\frac{2\pi t}{T_1}\right)/2 + \cos\left(\frac{10\pi t}{T_1} + \varphi_0\right)/4$$

Ideal Sampler

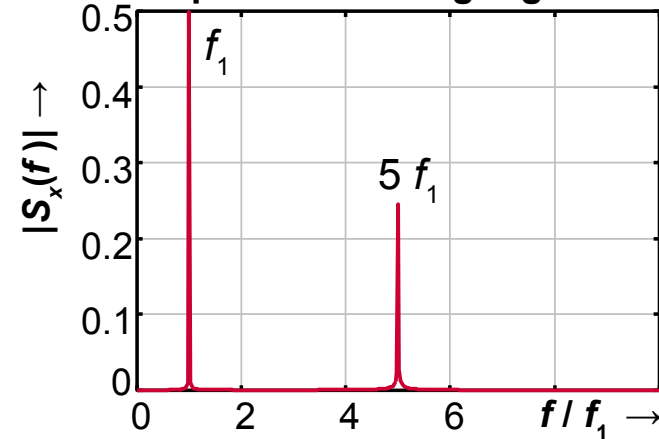


Sampled Signal

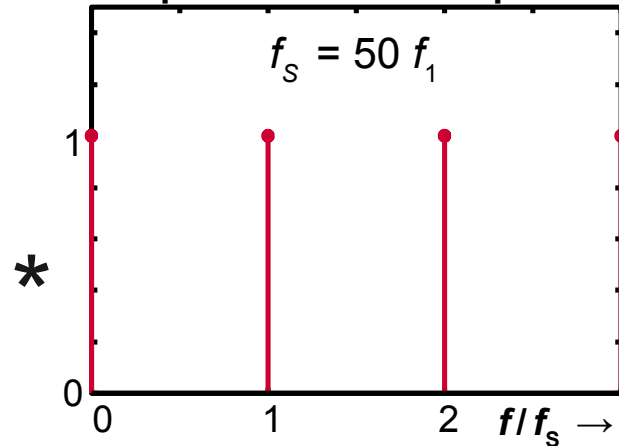


Periodisches Spektrum !

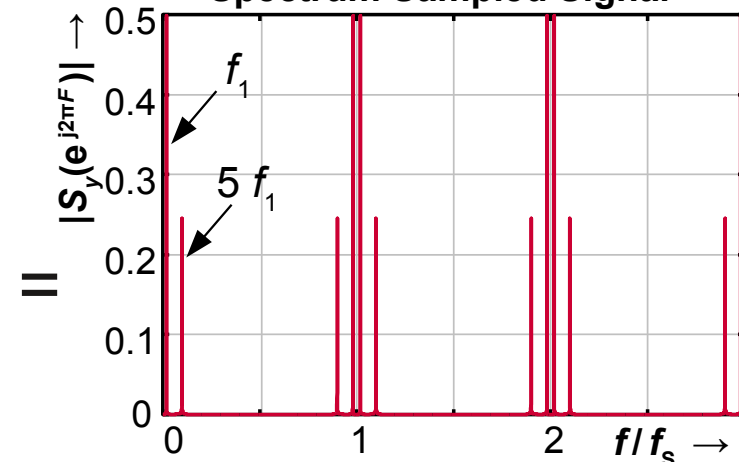
Spectrum Analog Signal



Spectrum Ideal Sampler



Spectrum Sampled Signal



Abschätzung des Spektrums eines analogen oder zeitdiskreten Signals $x(t)$ bzw. $x[n]$ (\rightarrow Matlab / Python, digitale Messwerterfassung)

Entwurf zeitdiskreter Filter in der Frequenzebene

Komprimierung zeitdiskreter Signale in der Frequenzebene

Schnelle Faltung (zeitdiskret, $\{ \ast \}$ \leftrightarrow $\{ \bullet \}$)

Robuste Breitband-Datenübertragung über parallele schmale Frequenzbänder (OFDM, PowerLine, LTE, ...)

Kenngößen bei Fourier-Transformation:

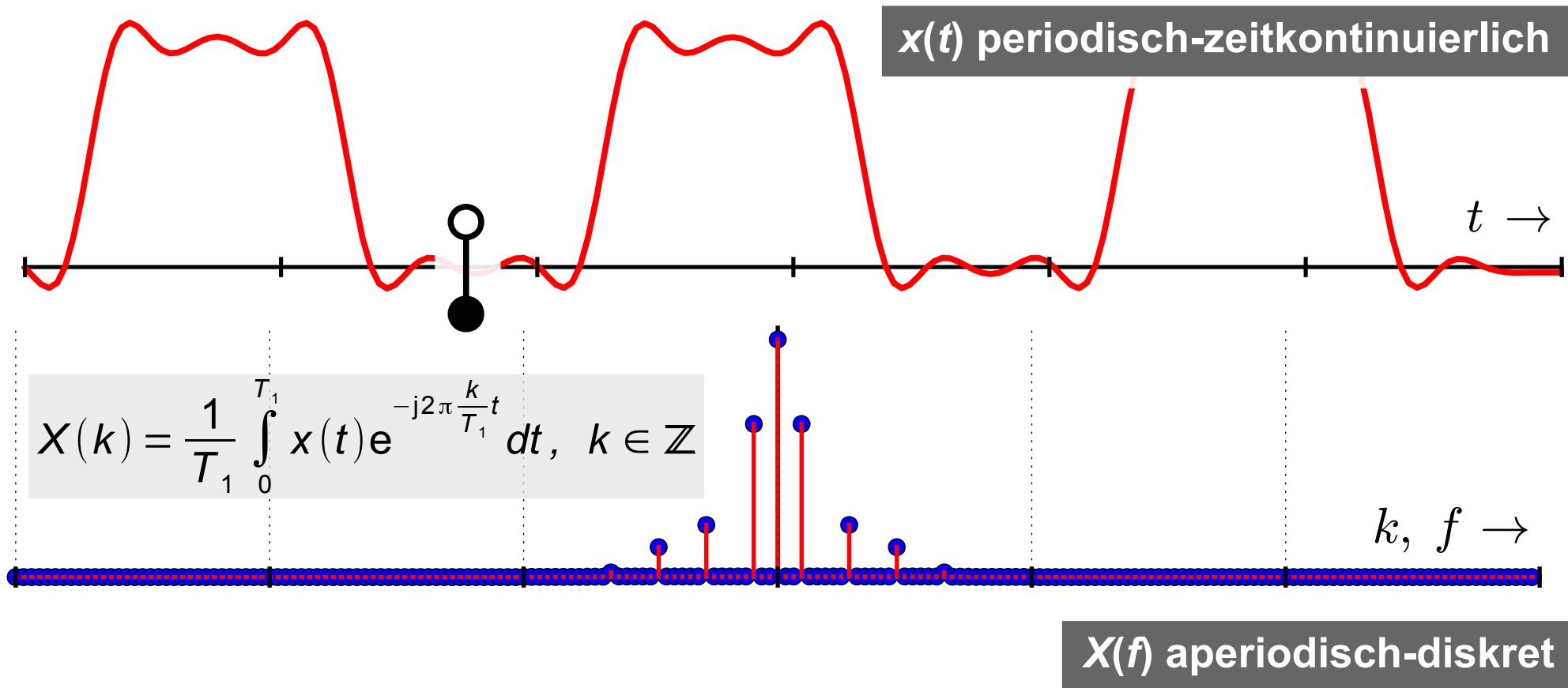
- Anzahl N der Zeit- / Frequenzpunkte
- Bandbreite Δf je Frequenzband
- Minimale Mess- bzw. Einschwingzeit T_E

Zeit-Bandbreite-Gesetz:

$$T_E \sim 1 / \Delta f !$$

(„Unschärferelation“
der Nachrichtentechnik)

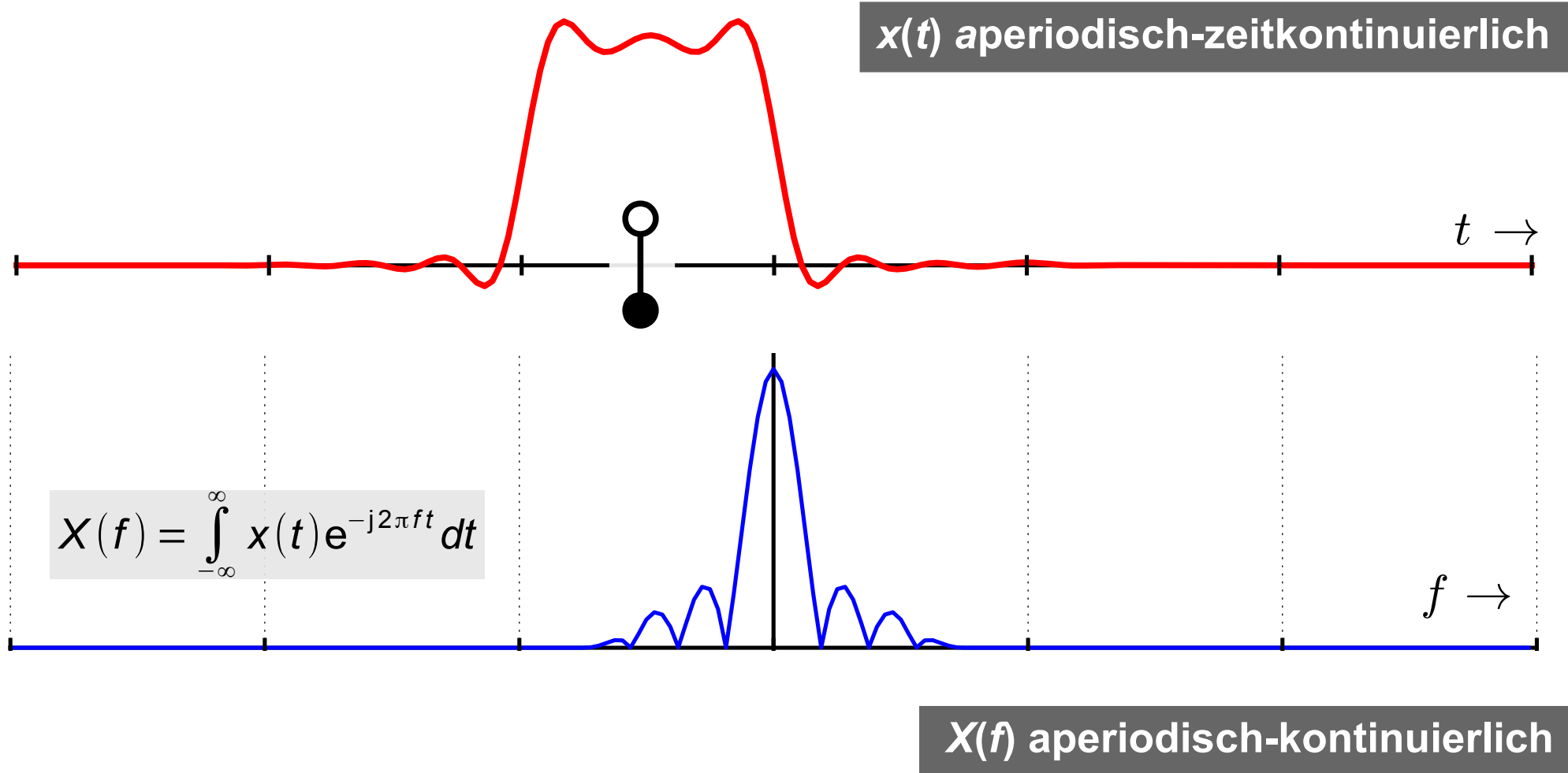
Fourier-Familie: Fourierreihe (CFT*)



Analoges Signal: Spektralschätzung nur mit analogen Methoden möglich
(Filterbank, durchstimmbares Bandpassfilter)

* Continuous Fourier Transform

Fourier-Familie: Fourierintegral (CFT)

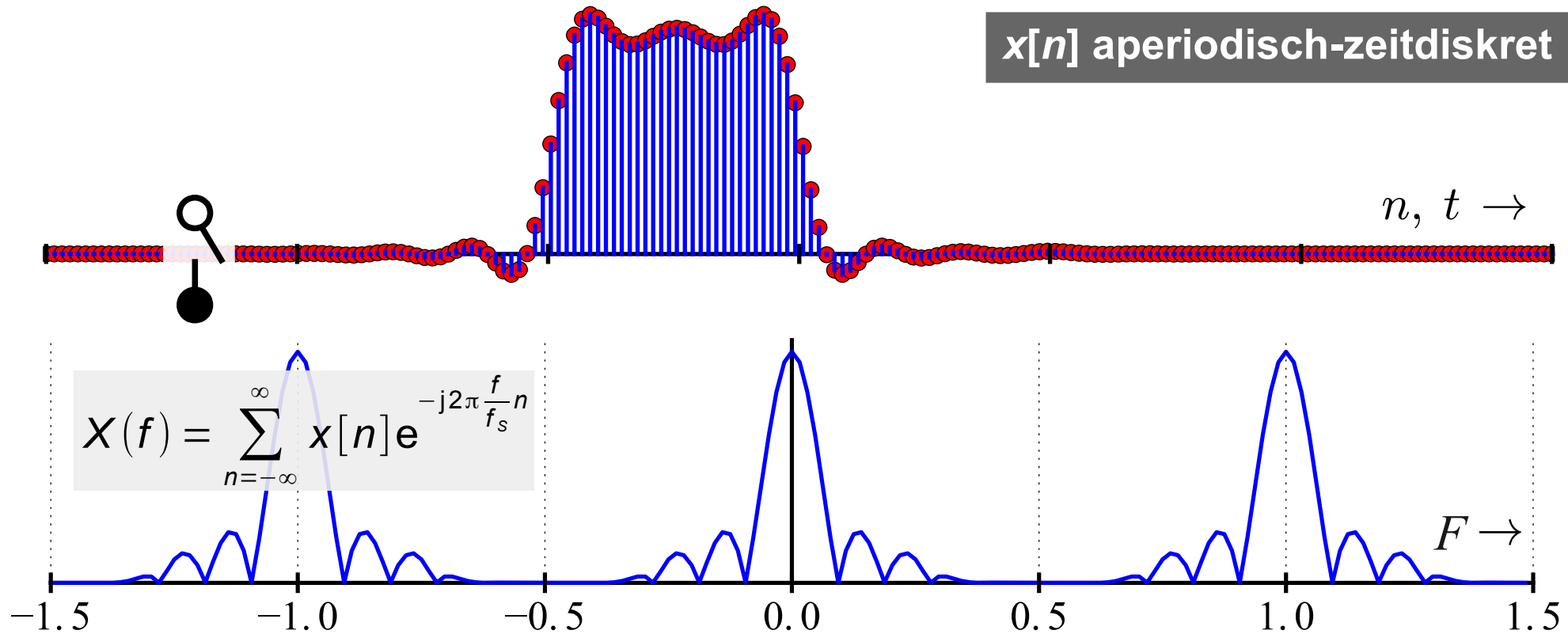


Analoges Signal: Spektralschätzung nur mit analogen Methoden möglich
(Filterbank, durchstimmbares Bandpassfilter)

Fourier-Familie: Discrete-Time Fourier Transform



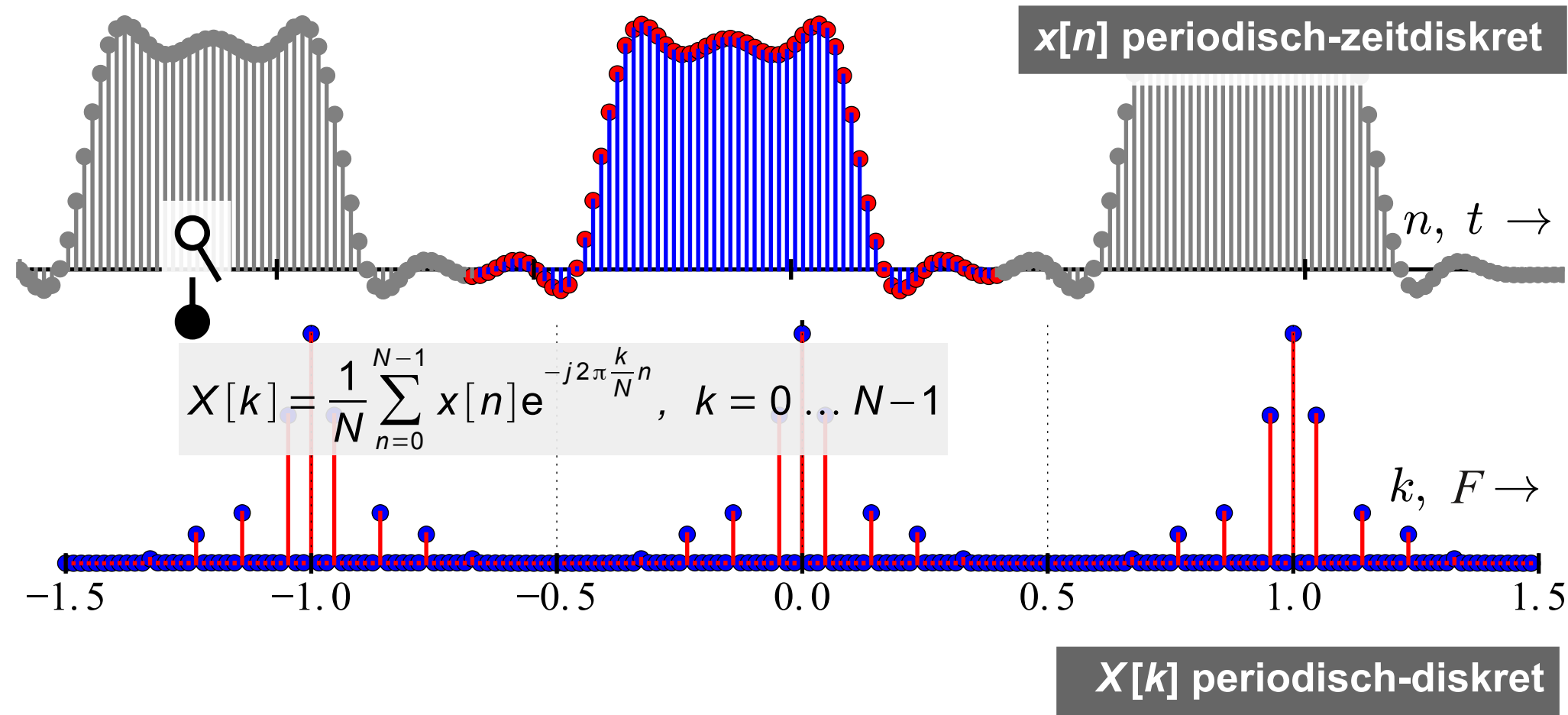
$x[n]$ aperiodisch-zeitdiskret



$X(f)$ periodisch-kontinuierlich

Zeitdiskret, aber keine numerische Berechnung möglich, da ∞ viele Samples.
Näherungsweise Darstellung mit DFT und Zero-Padding.

Fourier-Familie: Discrete Fourier Transform DFT



Zeitdiskret & endliche Anzahl Samples N herausgeschnitten: **Für periodisch (fortgesetzte) zeitdiskrete Signale ist numerische Berechnung möglich!**



Abtastung des Signals mit $f_s = 1 / T_s \rightarrow$ kein Informationsverlust

Quantisierung des Signals \rightarrow (hoffentlich) tolerierbarer Informationsverlust

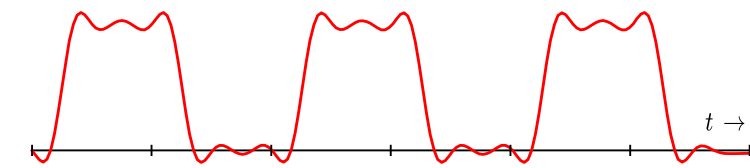
DTFT würde im Bereich $0 \dots f_s$ identisches Spektrum wie CFT liefern, aber nicht numerisch berechenbar (∞ viele Samples)

Fensterung = **Beschränkung auf N Messwerte**: Daten außerhalb der N Samples bzw. Ausschnitts mit Länge $T_1 = N T_s$ werden zu Null gesetzt (immer noch DTFT mit ∞ vielen Samples, nicht numerisch berechenbar)

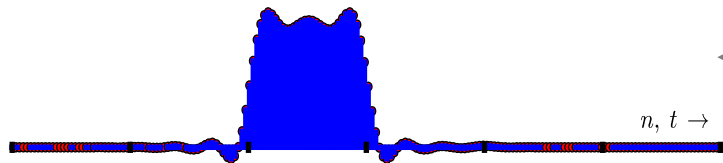
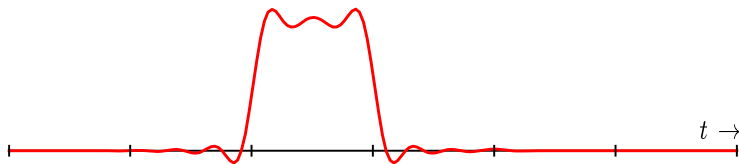
DFT des periodisch mit $T_1 = N T_s$ fortgesetzten Signalausschnitts liefert identische Spektrumswerte bei $k f_1 = k f_s / N$ wie **DTFT** !

N Datenpunkte $x[n]$ und N Frequenzpunkte $X[k] \rightarrow$ Block- oder Frame-Transformation \rightarrow **DFT ist numerisch gut berechenbar: Computer!**

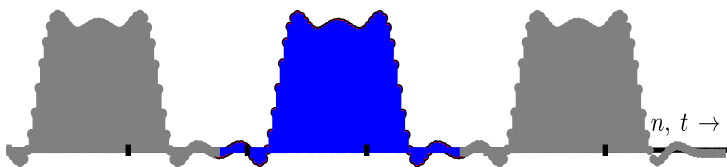
Numerische spektrale Analyse einer Messsequenz (2)



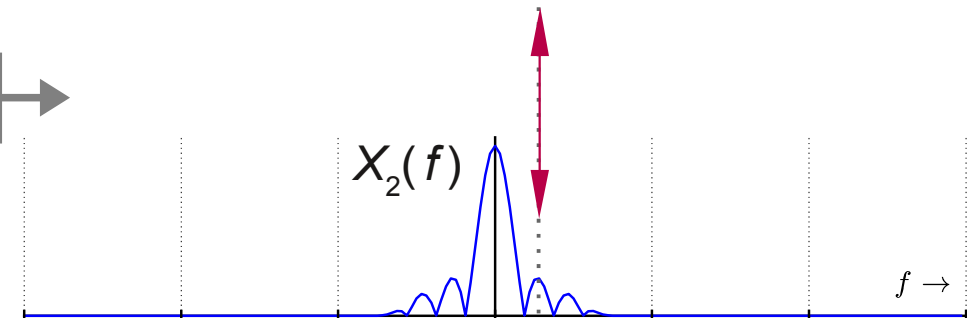
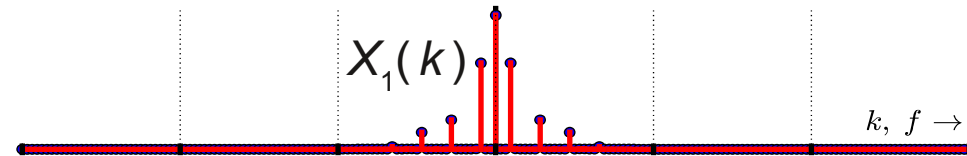
Fensterung ↔ Abtastung



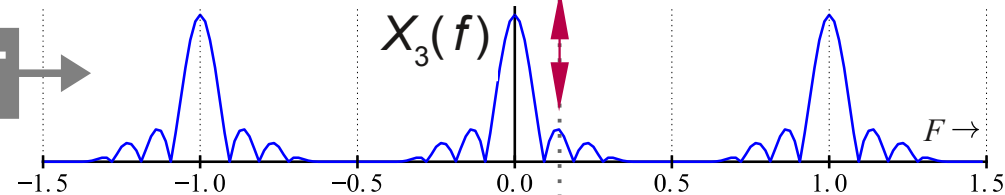
Wähle N Samples



CFT

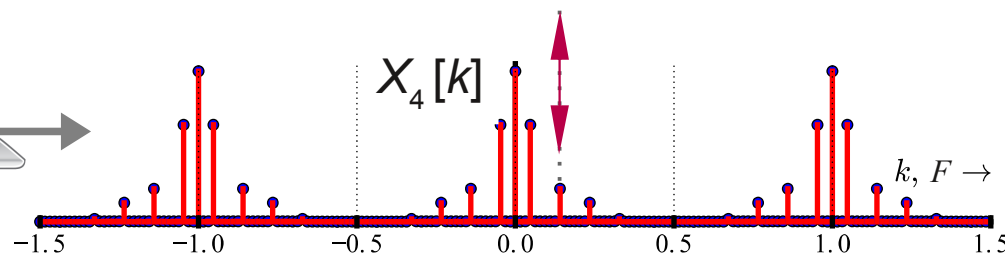


DTFT



$$X_1(k) = X_{2,3}(kf_s / N) = X_4[k]$$

DFT



Digitale Signalverarbeitung auf FPGAs

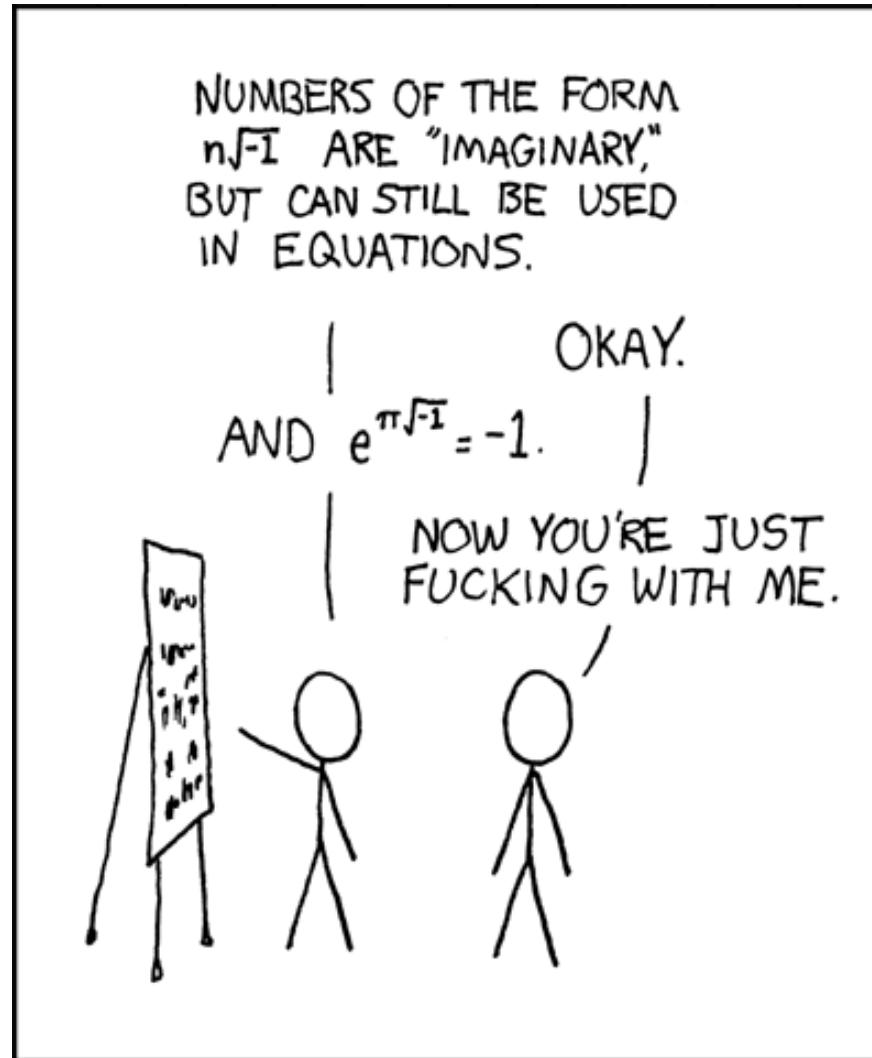
Kap. 3a Diskrete Fouriertransformation (Grundlagen)

Teil 2 *DFT: Komplex aber nicht kompliziert*

2016

Dr. Christian Münker

DFT: Komplex, aber nicht kompliziert

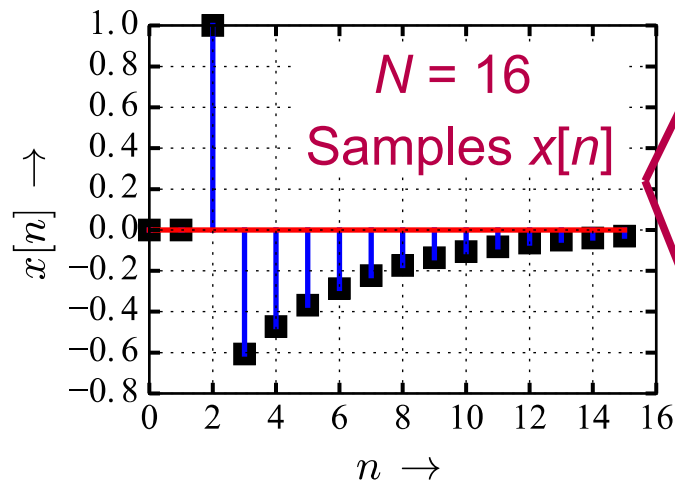


„e to the pi times i“ (<https://xkcd.com/179/>)
von Randall Munroe (xkcd@xkcd.com)
unter CC-BY-NC-2.5 Lizenz

Beispiel für DFT einer *reellwertigen* Sequenz



$N / 2 + 1$ Kosinusfunktionen (Realteil) \rightarrow



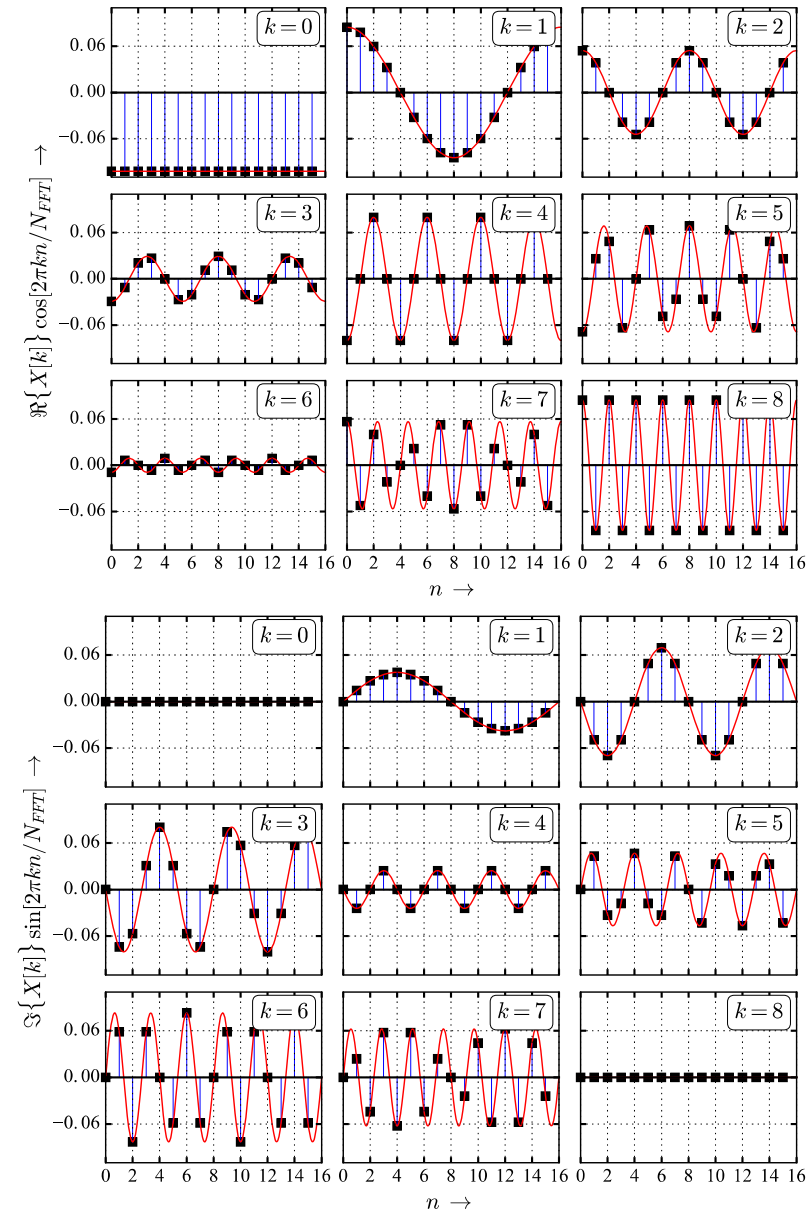
Analyse

Synthese

$N / 2 + 1$ Sinusfunktionen (Imaginärteil) \rightarrow

Aber: $k = 0$ und $N / 2$ liefern keinen Beitrag!

Σ : $N = 16$ Koeffizienten $X[k]$ für Basisfunktionen („Frequenzpunkte“)



mit $w_N = e^{-\frac{j2\pi}{N}}$ (Drehfaktor)
und $k, n = 0 \dots N-1$

- $f_k = f_s k / N$ [$F_k = k / N$] und $\Delta f = f_s / N = f_1 = 1 / T_1$
 Phys. Frequenzen Frequenzauflösung

Synthesegleichung der DFT (IDFT)



$$x[n] = \overbrace{\sum_{k=0}^{N-1} X[k] e^{+j2\pi \frac{kn}{N}}}^{N \cdot \text{Python / Matlab ifft}} = \sum_{k=0}^{N-1} X[k] w_N^{-kn} \quad \text{mit } k, n = 0 \dots N-1$$

und $w_N = e^{-\frac{j2\pi}{N}}$

- N Samples $\leftrightarrow N$ Frequenzpunkte \Rightarrow *Blocktransformation* !
- $X[k]$ im Allgemeinen komplex $\rightarrow N \times N$ komplexe Multiplikationen
- Aber aufgrund Symmetrie **$X[-k] = X^*[k]$ für reelle $x[n]$ genügt IDFT über die Hälfte des Spektrums**
- Periodizität der IDFT **$x[n + N] = x[n]$** : Zeitsignal kann periodisch fortgesetzt werden
- Wahl des Skalierungsfaktors (hier: $1/N$ bei DFT) so:
 - dass nacheinanderfolgende DFT und IDFT von $x[n]$ wieder $x[n]$ ergibt
 - dass DFT – Werte die Amplituden der Spektralkomponenten korrekt wiedergeben (optional, ist z.B. bei Matlab nicht so)

Eigenschaften der DTFT (CFT)



Eigenschaft	Zeitbereich $x[n]$	Frequenzbereich $X(e^{j\Omega})$
Definition	$x[n] \equiv x(nT_s)$	$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-jn\Omega}$
Linearität	$ax[n] + by[n]$	$aX(e^{j\Omega}) + bY(e^{j\Omega})$
Zeitverschiebung	$x[n-k], k \in \mathbb{Z}$	$X(e^{j\Omega}) e^{-j\Omega k}$
Frequenzverschiebung (Modulation)	$x[n] e^{jn\Omega_1}, \Omega_1 \in \mathbb{R}$	$X(e^{j(\Omega-\Omega_1)})$
Zeitumkehr	$x[-n]$	$X(e^{-j\Omega})$
Faltung im Zeitbereich	$x[n] \star y[n]$	$X(e^{j\Omega}) \cdot Y(e^{j\Omega})$
Multiplikation im Zeitbereich	$x[n] \cdot y[n]$	$X(e^{j\Omega}) \star Y(e^{j\Omega}) / 2\pi$
Inverse DTFT	$x[n] = T_s \int_{-f_s}^{f_s} X(e^{j\Omega}) e^{jn\Omega} d\Omega$	

Eigenschaften der DFT (N Punkte)



Eigenschaft	Zeitbereich $x[n]$	Frequenzbereich $X[k]$
Definition IDFT / DFT	$x[n] = \sum_{k=0}^{N-1} X[k] e^{+j2\pi \frac{kn}{N}}$	$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{kn}{N}}$
Periodizität	$x[n] = x[n - N]$	$X[k] = X[k - N]$
Linearität	$ax[n] + by[n]$	$aX[k] + bY[k]$
Zyklische Zeitverschiebung	$x[n-m], m \in \mathbb{Z}$	$X[k] e^{-j2\pi km / N}$
Modulation	$x[n] e^{j2\pi nm / N}$	$X[k-m]$
Zeitumkehr	$x[-n] = x[N-n]$	$X[-k] = X[N-k]$
Faltung im Zeitbereich	$x[n] \star y[n]$	$X[k] \cdot Y[k]$
Multiplikation im Zeitbereich	$x[n] \cdot y[n]$	$X[k] \star Y[k] / N$

Symmetrie der Fouriertransformationen



Bei *reellwertigen Zeitsignalen* x gelten für CFT / DTFT / DFT $x \circ \bullet X$ die folgenden Symmetrieeigenschaften des Spektrums:

Reellwertiges $x \circ \bullet X^*(e^{j\Omega}) = X(e^{-j\Omega})$ (konjugiert-symmetrisches Spektrum)

$\text{Re}\{X\}$ ist gerade

$\text{Im}\{X\}$ ist ungerade

$|X|$ ist gerade (Betragsgang)

$\angle X$ ist ungerade (Phasengang)

gerader Teil von $x \circ \bullet \text{Re}\{X\}$

ungerader Teil von $x \circ \bullet j \text{Im}\{X\}$

DFT reellwertiger und komplexer Sequenzen



Zeitbereich $x[n]$ mit $n = 0 \dots N-1$

Frequenzbereich $X[k]$; $k = 0 \dots N-1$

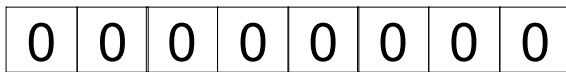
Reellwertig

N reellwertige Datenpunkte

reell: $x_R[n] = x[n]$

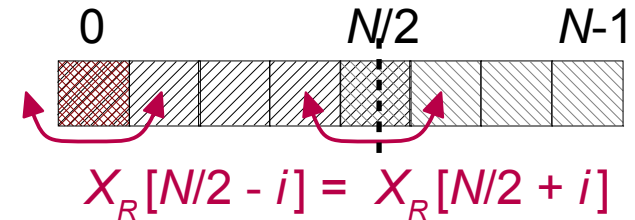


imaginär: $x_I[n] = 0$



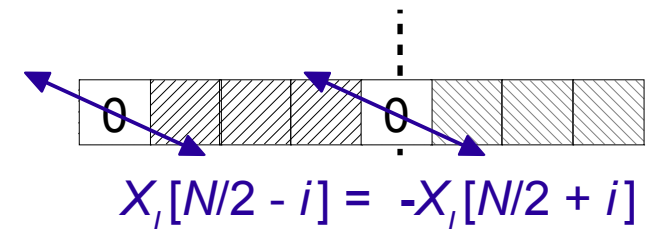
N relevante Datenpunkte

reell: $X_R[k]$



○—●

imag.: $X_I[k]$



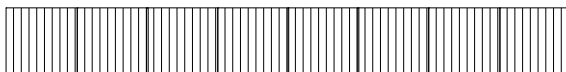
Komplex

N kompl. = $2N$ reellw. Datenpunkte

reell: $x_R[n]$

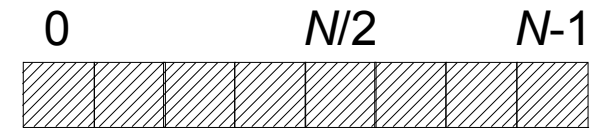


imaginär: $x_I[n]$



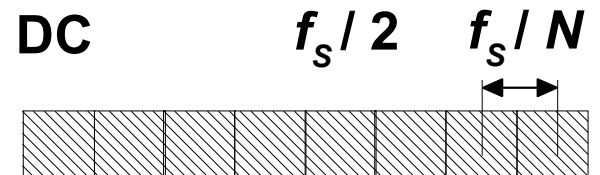
$2N$ relevante Datenpunkte

reell: $X_R[k]$



○—●

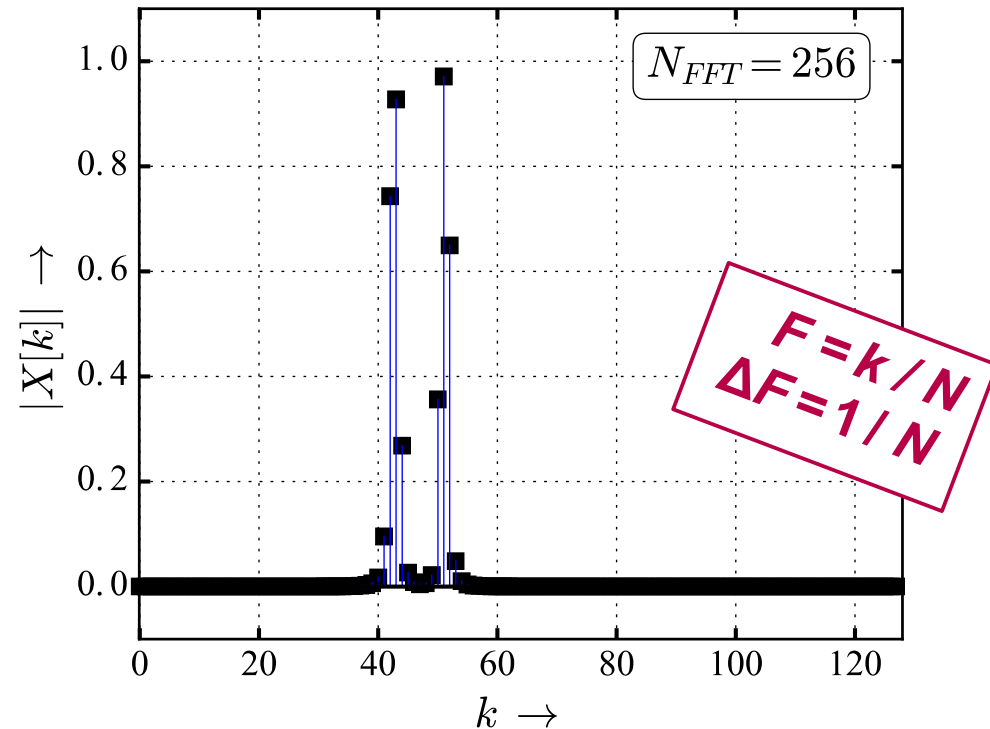
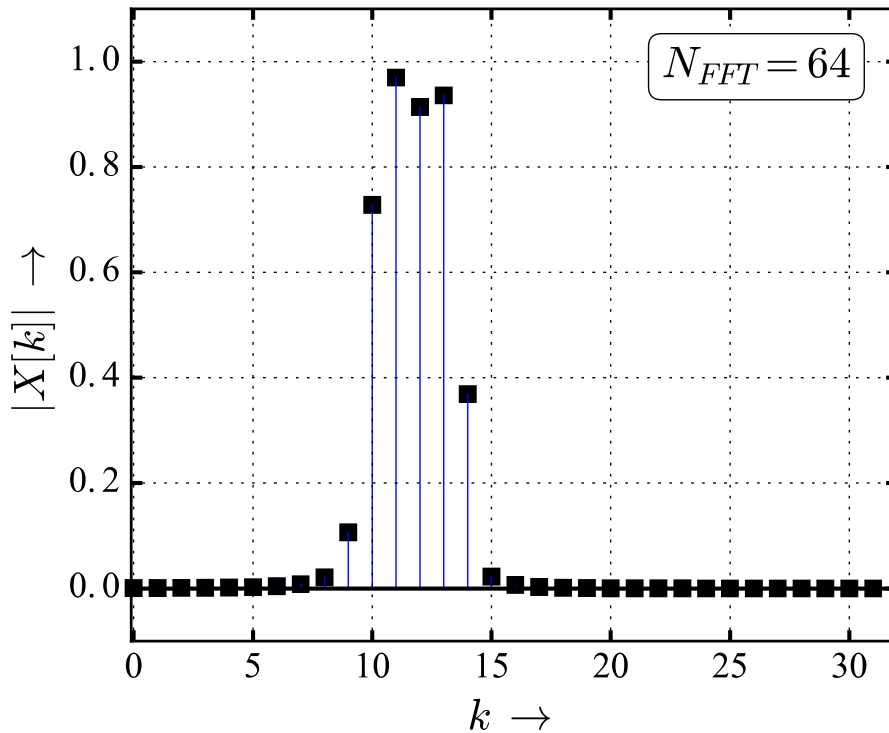
imag.: $X_I[k]$



Frequenzauflösung der DFT



Je größer die Länge N der Datenfolge, desto besser die
Frequenzauflösung der DFT: $\Delta f = 1 / NT_s$



Anmerkung: Die *höchste Frequenzkomponente* der DFT wird
natürlich durch die Abtastfrequenz bestimmt: $f_{k,max} = f_{N/2} = f_s / 2$

DFT des Dirac-Stoßes (1)



DFT des Dirac-Stoßes: $|X[k]| = ?$ $\varphi[k] = ?$

$$x[n] = \delta[n] \Rightarrow X[k] = \frac{1}{N} \sum_{n=0}^{N-1} \delta[n] e^{-j2\pi \frac{kn}{N}} = \frac{1}{N}$$

→ Alle Basisfunktionen gewichtet mit Faktor $1/N$, rein reell.

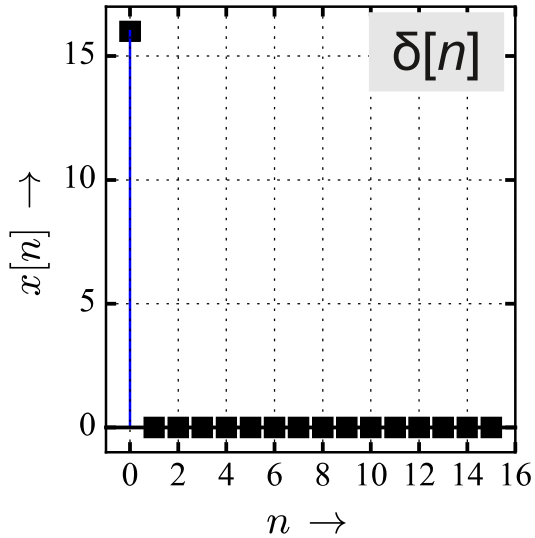
$$x[n] = \delta[n - \textcolor{red}{m}] \Rightarrow X[k] = \frac{1}{N} \sum_{n=0}^{N-1} \delta[n - m] e^{-j2\pi \frac{kn}{N}} = \frac{1}{N} e^{-j2\pi \frac{km}{N}}$$

→ Alle Basisfunktionen gewichtet mit Faktor $1/N$,
lineare Phase **$-2\pi km/N$** .

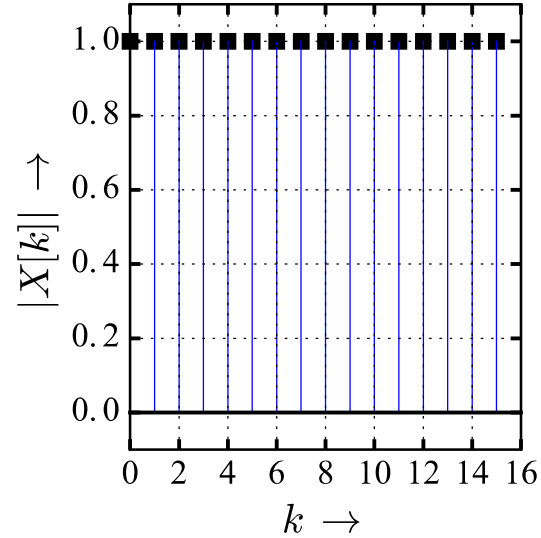
DFT des Dirac-Stoßes (2)



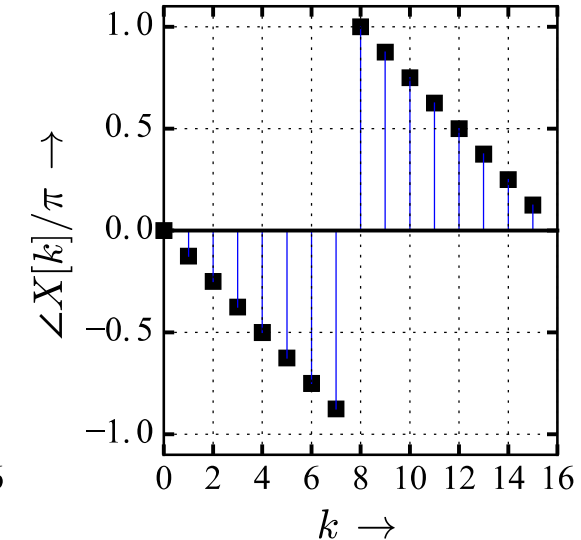
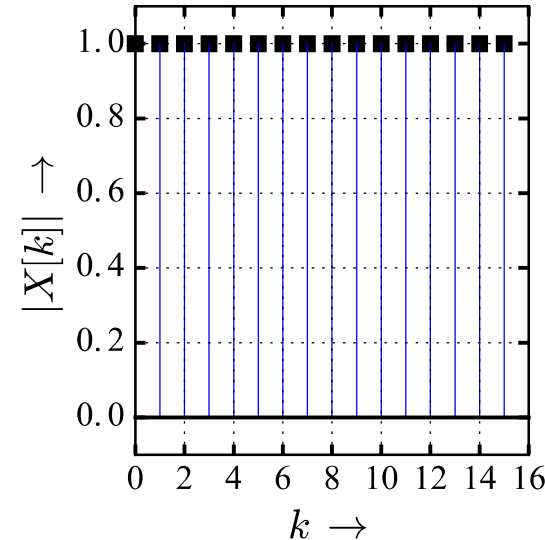
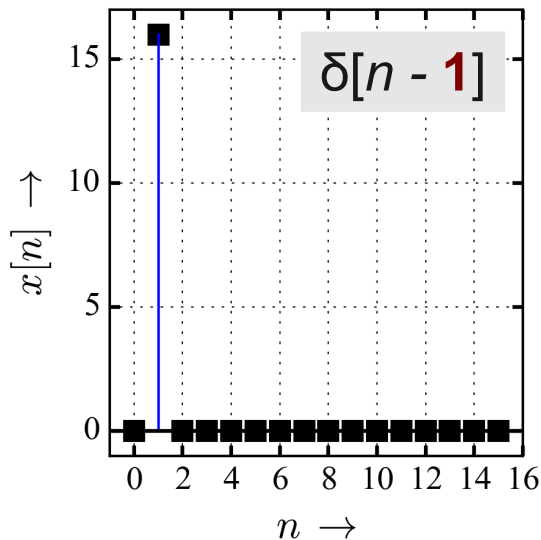
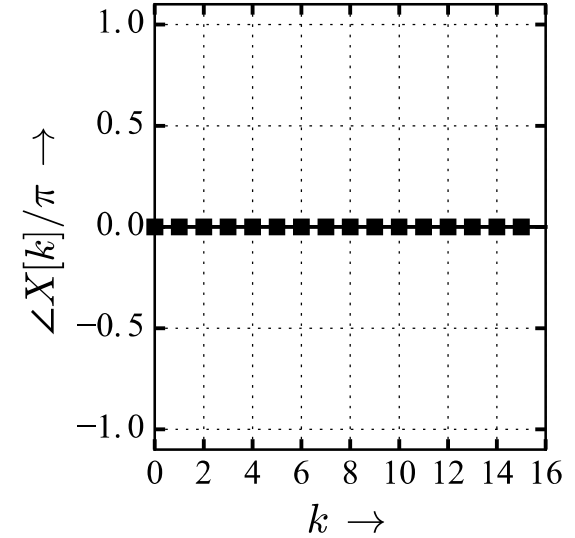
Zeit



Betrag



Phase



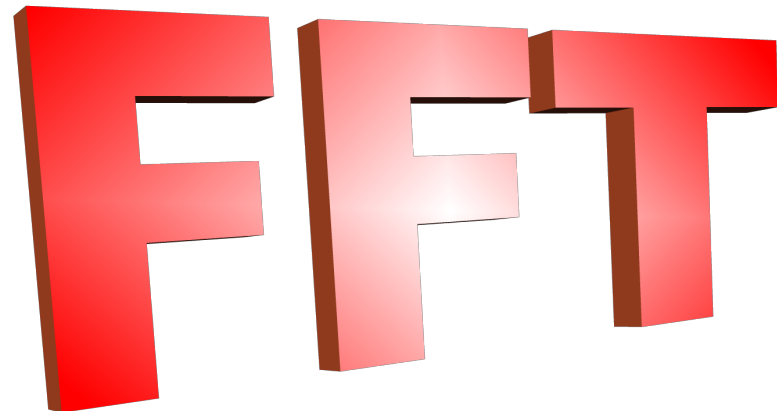
Digitale Signalverarbeitung auf FPGAs

Kap. 3a Diskrete Fouriertransformation (Grundlagen)

Teil 4 *Schnell, schneller ...*

2016

Dr. Christian Münker

A large, 3D, red logo for the Fast Fourier Transform (FFT). The letters 'F', 'F', and 'T' are rendered in a bold, blocky font with a slight shadow, giving them a three-dimensional appearance. The color is a vibrant red.

DFT als Signal Flow Graph für $N = 8$



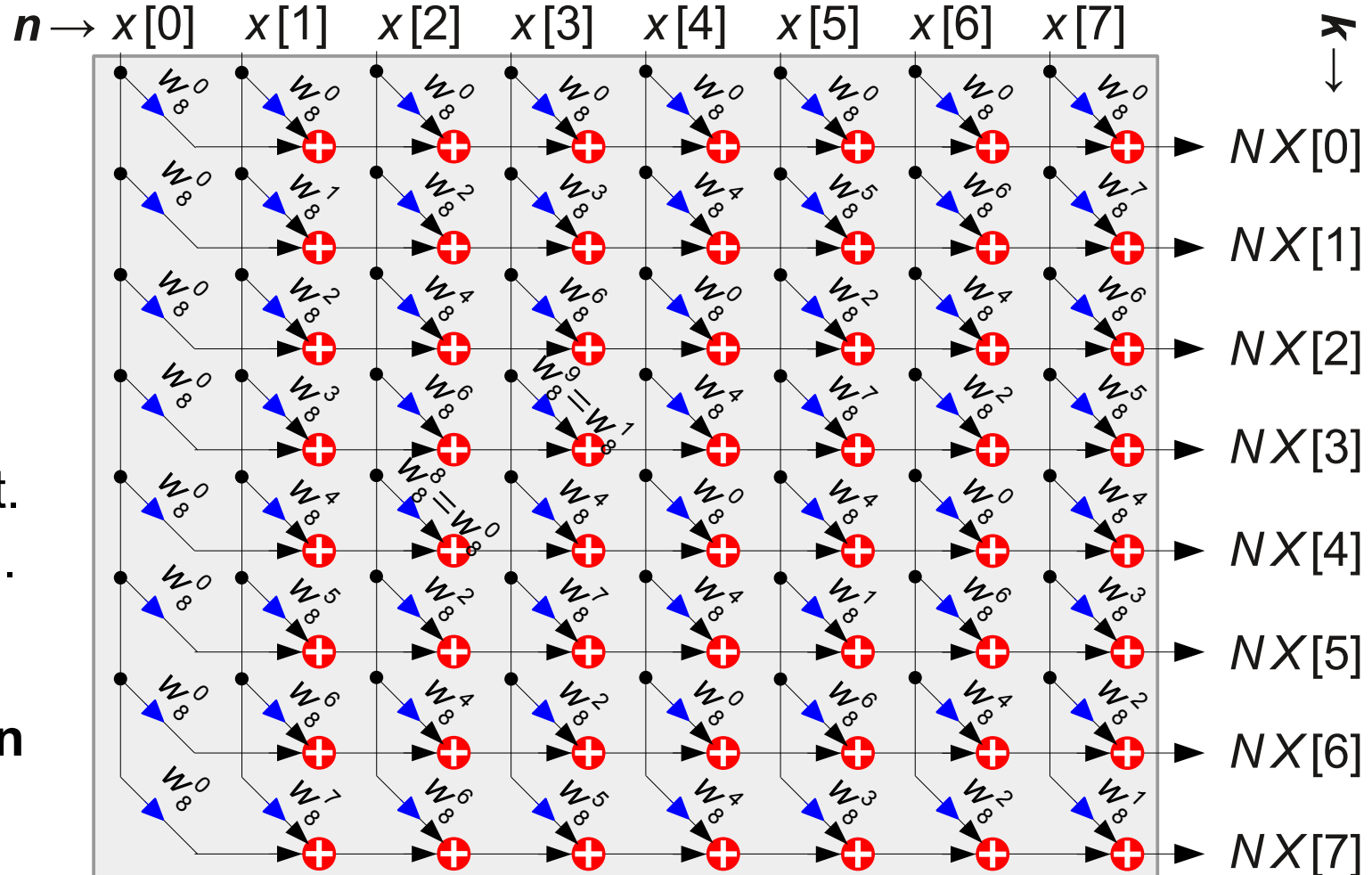
$$NX[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{kn}{N}} = \sum_{n=0}^{N-1} x[n] w_N^{kn} \quad \text{mit } w_N^{kn} = e^{-j2\pi \frac{kn}{N}} \quad (\text{Drehfaktor})$$

Matrix-
schreibweise:
 $NX = W_N x$

Je N^2 komplexe
Multiplikationen
und Additionen:

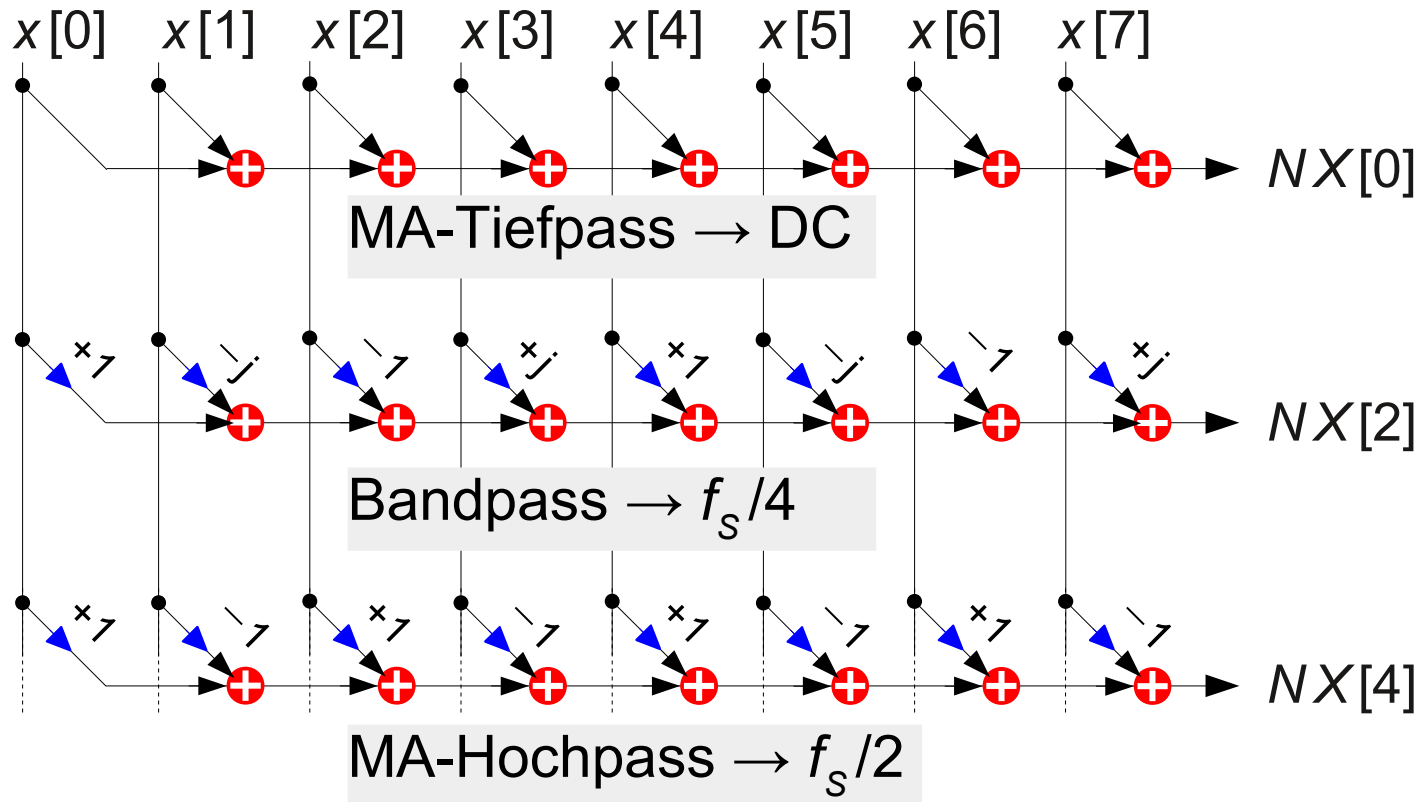
$4N^2$ reelle Mult.
+ $4N^2$ reelle Add.

**$\Sigma: 4N^2$ MAC-
Operationen**
(N^2 für reellwertige
Signale)

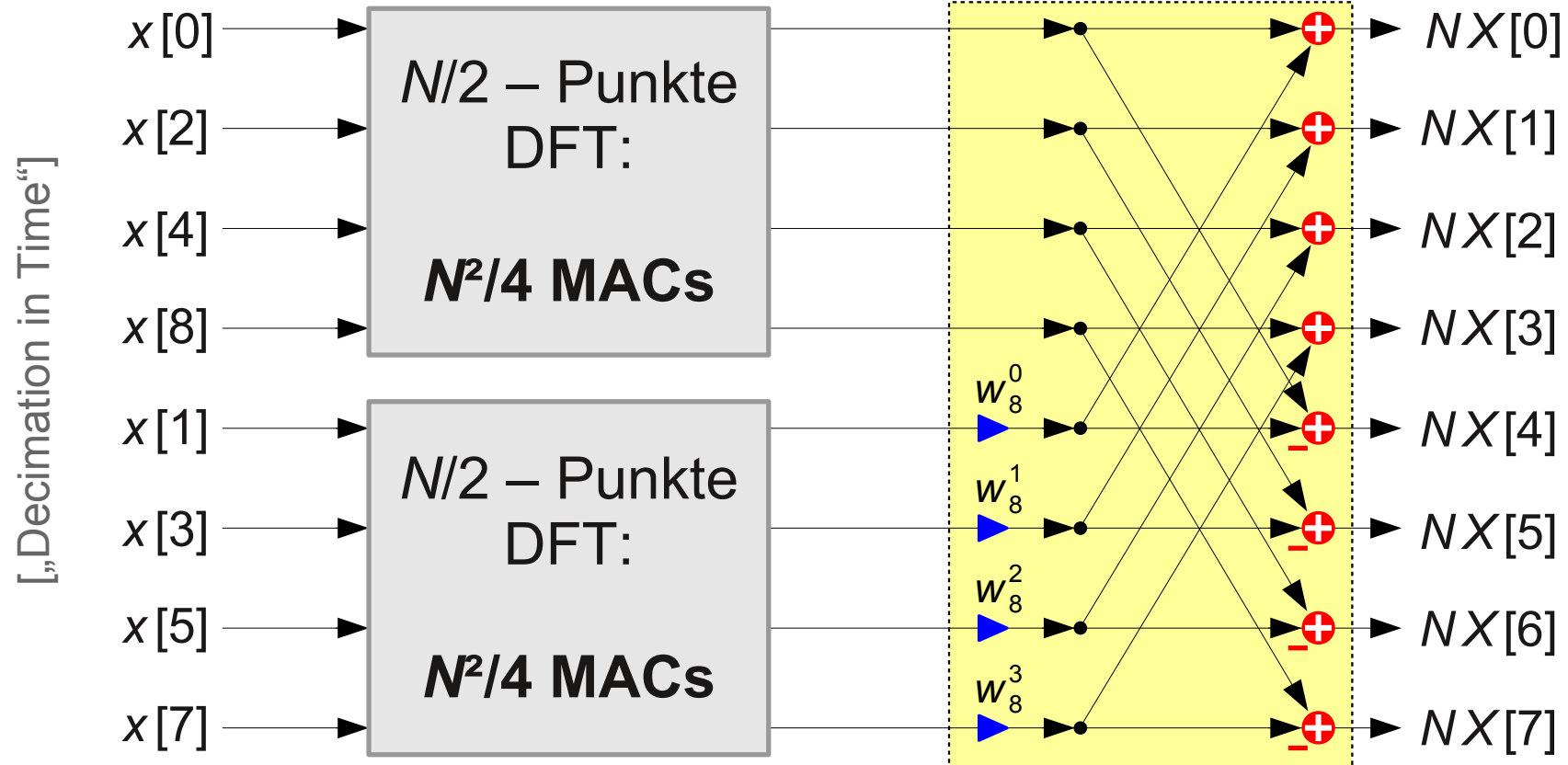


$$NX[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{kn}{N}} = \sum_{n=0}^{N-1} x[n] w_N^{kn} \quad \text{mit } w_N^{kn} = e^{-j2\pi \frac{kn}{N}}$$

DFT kann auch interpretiert werden als Filterbank aus N FIR-Filtern mit z.T. komplexwertigen Koeffizienten:

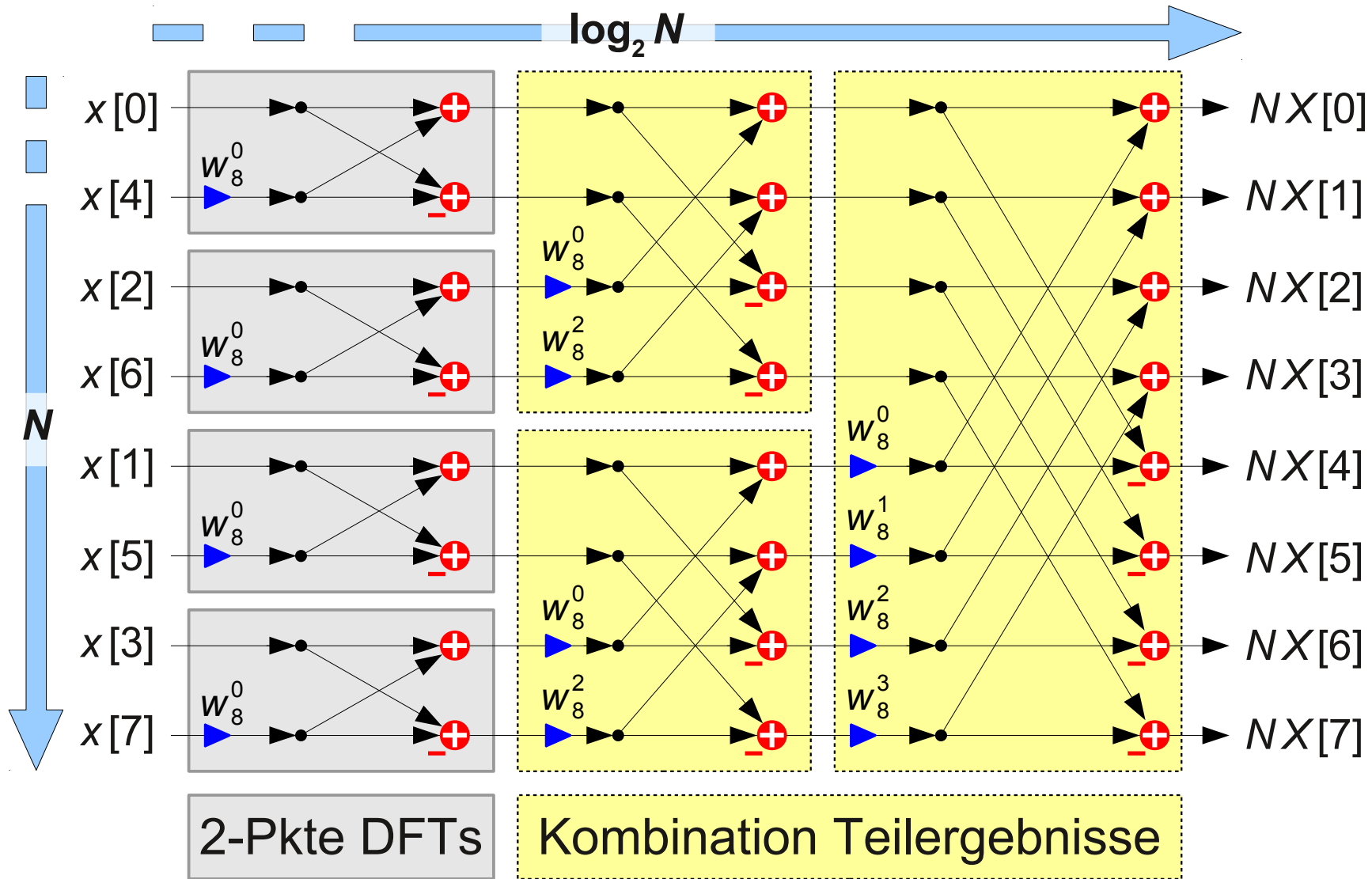


Aufteilung in zwei $N/2$ -Punkte DFTs



- Aufteilung in $N/2$ – Punkte DFTs für ungerade und gerade Samples und geeignete Kombination der Teilergebnisse reduziert Gesamt-MACs
- Umsortierung der Eingangssamples notwendig
- Herleitung nicht trivial, siehe Literatur oder 9. Kapitel (vielleicht)

FFT: Sukzessive Aufteilung in Teil-DFTs



ca. $2 \cdot N \log_2 N$ reellw. Mult. (abhängig von Algorithmus), vgl. DFT: N^2 reellw. Mult.

* „ $2N$ “, da kaum Einspareffekte für reellwertige Eingangssignale !

Symmetrien und „Butterfly“- Operation



$$w_N^k = e^{-j \frac{2k\pi}{N}}$$

Nutze Symmetrien des komplexen Drehfaktors („twiddle factor“) zur Vereinfachung der Rechnung:

$$w_N^N = e^{-j \frac{2\pi N}{N}} = 1; \quad w_N^{N/2} = e^{-j \frac{\pi N}{N}} = -1$$

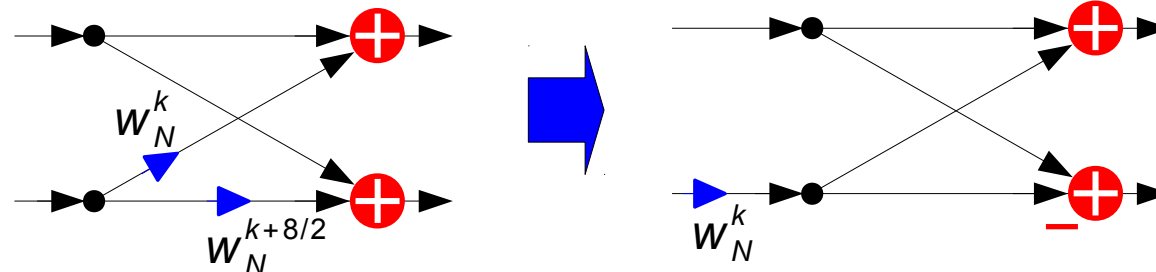
$$w_N^{N+k} = w_N^N w_N^k = w_N^k \quad (\text{Periodizität})$$

$$w_N^{N-m} = w_N^{-m} = w_N^{m*}$$

$$w_N^{N/4} = -j; \quad w_N^{3N/4} = j$$

$$w_N^{k+N/2} = w_N^k w_N^{N/2} = -w_N^k$$

Damit Optimierung der 2-Punkte DFT („butterfly computation“)



„Optimale“ FFT ?



- Für $N_{FFT} = 2^k$, lässt sich die FFT so zerlegen, dass man die minimale Anzahl Rechenoperationen erhält
- Aber: Verschiedene sehr gute FFT-Algorithmen wenn eine Zerlegung von N_{FFT} in kleine Primfaktoren möglich ist
- Für diese Fälle ist $N_{MAC} = O(N \log_2 N)$
- Weitere Optimierungsziele:
 - Minimaler Speicherverbrauch
 - minimale Anzahl von Sortiervorgängen
 - Eingangssamples werden in korrekter Reihenfolge verarbeitet
 - ...

Siehe auch <http://lighthouseinthesky.blogspot.de/2010/03/flops-and-fft.html>

FFT ist in Python / Matlab schnell aufgestellt:

Konstanten: `NFFT = 150; fs = 1000`

Signal: `n = np.arange(NFFT); y = np.sin(2*pi*n/16)`

FFT (skaliert): `Y = fft(y)/NFFT`

Frequenzen: `f = fftfreq(NFFT, 1/fs) # Angabe von fs optional`

`f = np.arange(NFFT) * fs/NFFT # Alternativ`

Zweiseit. Spektrum: `Y2 = fftshift(Y)`

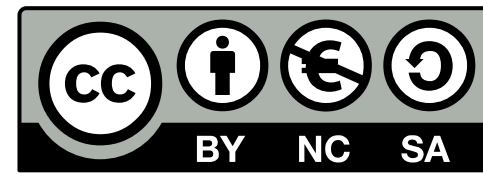
`f2 = fftshift(f)`

Plotten: `plot(f2, np.abs(Y2))`

Plotten bis $f_s/2$: `plot(f[0:NFFT/2], np.abs(Y[0:NFFT/2]))`

Python: `import fft, ... from numpy.fft`

Diese Folien und die zugehörigen Videos sind unter
Creative-Commons-Lizenz **CC-BY-NC-SA 3.0 de** veröffentlicht.



Bei Verwendung dieses Werks müssen Sie auf die entsprechende **CC-Lizenzurkunde** verweisen, in diesem Fall <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> .

Sie müssen ferner die folgenden Angaben machen ("BY", attribution)

- **Author** („Christian Münker“)
- **Titel** („Digitale Signalverarbeitung auf FPGAs“)
- **URL** zu Werk (https://github.com/chipmuenk/dsp_fpga)
und / oder Author (<http://www.chipmuenk.de>)

Außerdem ist die Verwendung auf folgende Weise eingeschränkt:

- Diese Materialien dürfen nur **nicht kommerziell** genutzt werden („NC“, non-commercial).
- Dieses Werk oder Teile daraus dürfen nur **unter gleichen Lizenzbedingungen** weiterverteilt werden („SA“, share alike).

Fragen, Anmerkungen, Anregungen, Bugs, Bierbons bitte an mail@chipmuenk.de.

Ich wünsche viel Erfolg und Spaß (!) mit den Materialien!