

# 5. Ton und Klang

- 5.1 Ton: Physikalische und physiologische Aspekte
- 5.2 Kompression von Audio-Signalen: MPEG-Audio
- 5.3 Audio-Datenformate: Übersicht
- 5.4 Klangerzeugung und MIDI



Literatur:

Medieninformatik-Buch:  
Kapitel 4



D. Pan: A Tutorial on MPEG/Audio Compression,  
IEEE Multimedia 2(1995), 60–74

Arne Heyda, Marc Briede, Ulrich Schmidt: Datenformate im Medienbereich,  
Fachbuchverlag Leipzig 2003, Kapitel 5

John Watkinson: MPEG Handbook, 2nd ed., Butterworth-Heinemann 2004

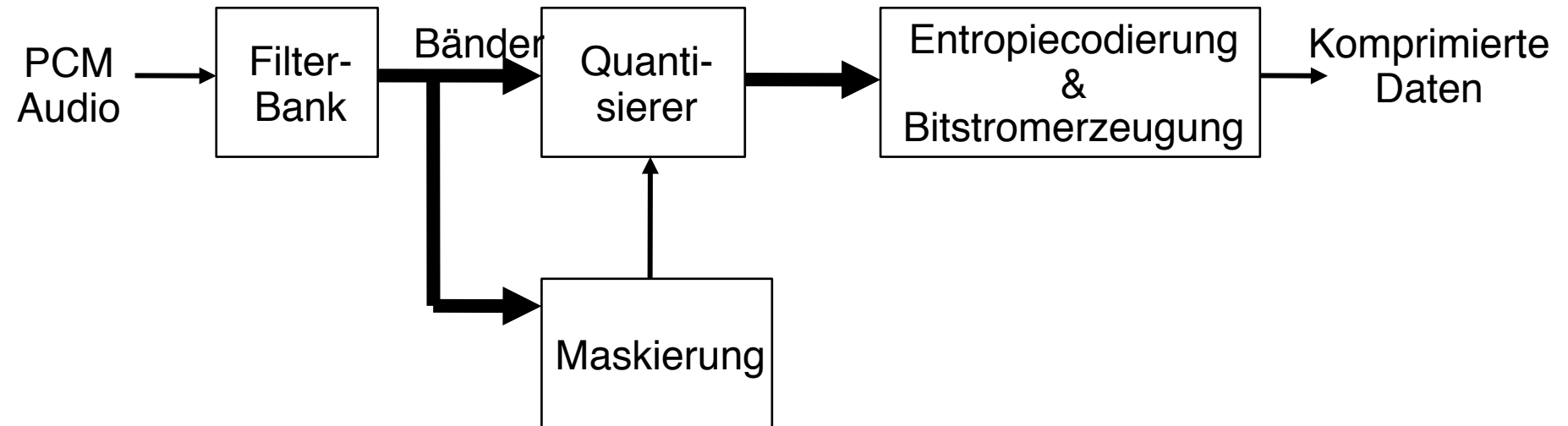
# Wiederholung und Abrundung: Akustische Illusionen

- Fehlender Grundton
  - Melodie mit künstlich entferntem Grundton bei den einzelnen Noten
  - Melodie dennoch gut wiedererkennbar: Grundton wird ergänzt

[http://commons.wikimedia.org/wiki/Image:Suppress\\_fundamental.ogg](http://commons.wikimedia.org/wiki/Image:Suppress_fundamental.ogg)
- Beliebige lange aufsteigende bzw. abfallende Tonleiter (Shepard-Effekt)

<http://www.cs.ubc.ca/nest/imager/contributions/flinn/Illusions/ST/st.html>

# MPEG-Audio Encoder: Grundlegender Aufbau

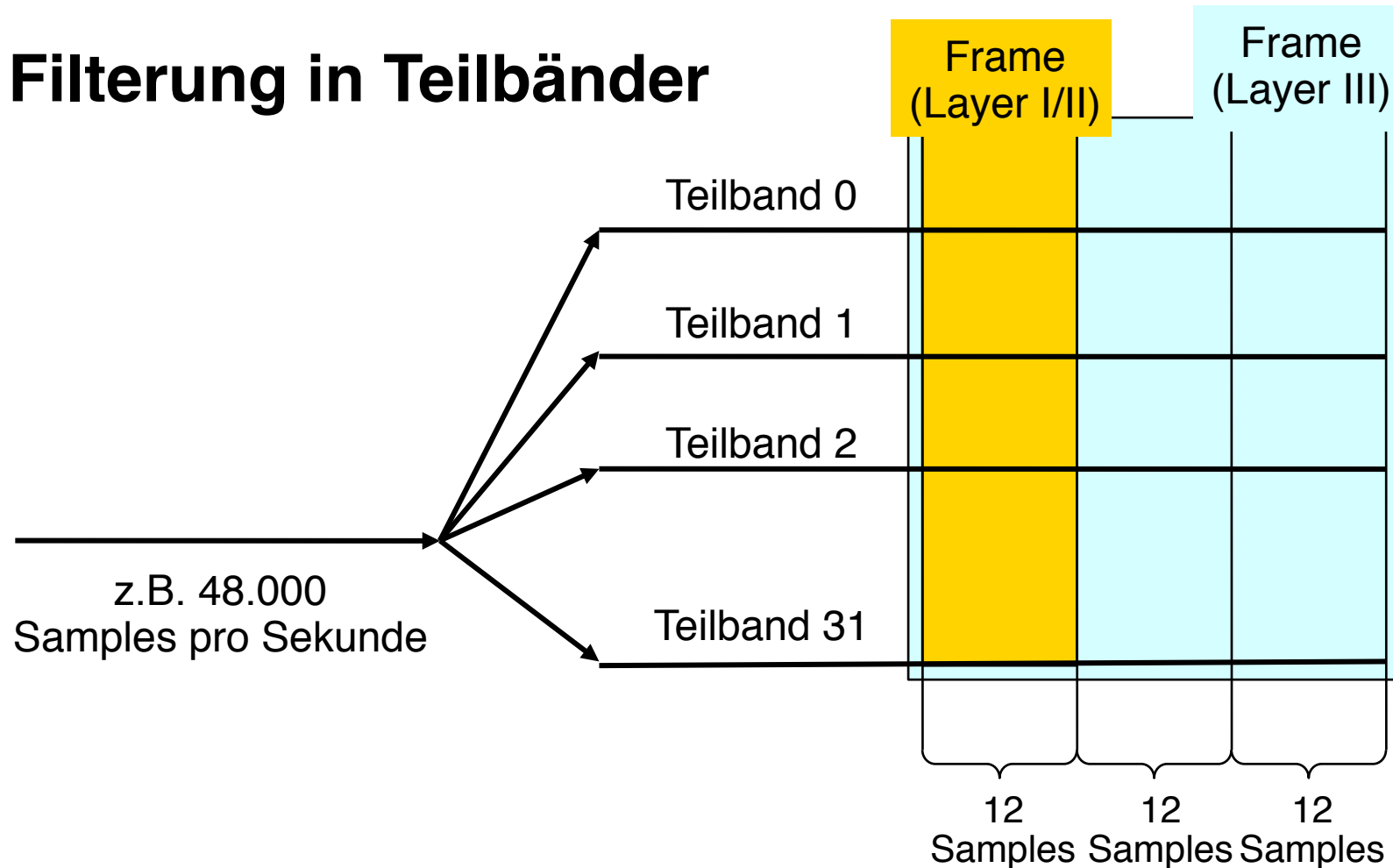


- Hinweis: Der MPEG-Standard definiert nicht den Aufbau eines Encoders, sondern nur die Decodierung!
- Signal wird in Frequenzbänder aufgeteilt
- Maskierung auf der Basis der Bänder mit einem psychoakustischen Modell

# Subband-Kodierung

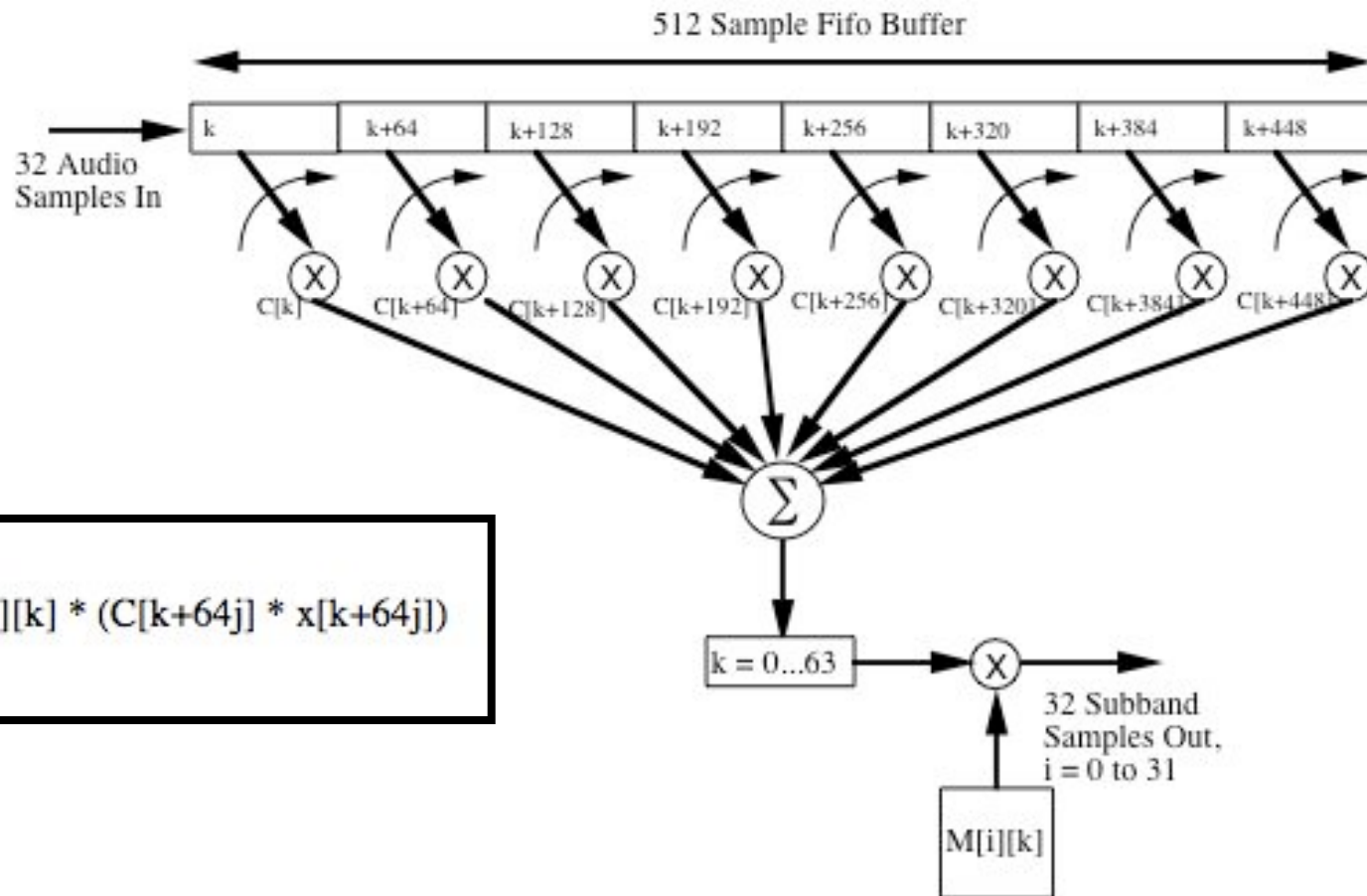
- Energie eines Tonsignals ist meist nicht gleichmäßig auf das Frequenzspektrum verteilt
- Idee:
  - Aufteilen des Signals in Teil-Frequenzbänder
  - Ermittlung des Signalpegels für jedes Teilband
  - Einzel-Codierung der Teilbänder mit jeweils angemessener Bitanzahl
    - » z.B. nicht belegtes Teilband: 0 Bit
  - Funktioniert optimal, wenn Teilbänder an kritische Bänder des Gehörs angepasst

# Filterung in Teilbänder



- 12 Samples entsprechen bei 48 kHz ca. 8 ms
- Ein Block von Samples in einem Teilband wird manchmal *bin* genannt
- *Frame*: Gesamtheit der Samples in allen Teilbändern  
 $12 \times 32 = 384$  Samples in Layer I/II,  $3 \times 12 \times 32 = 1152$  Samples in Layer III

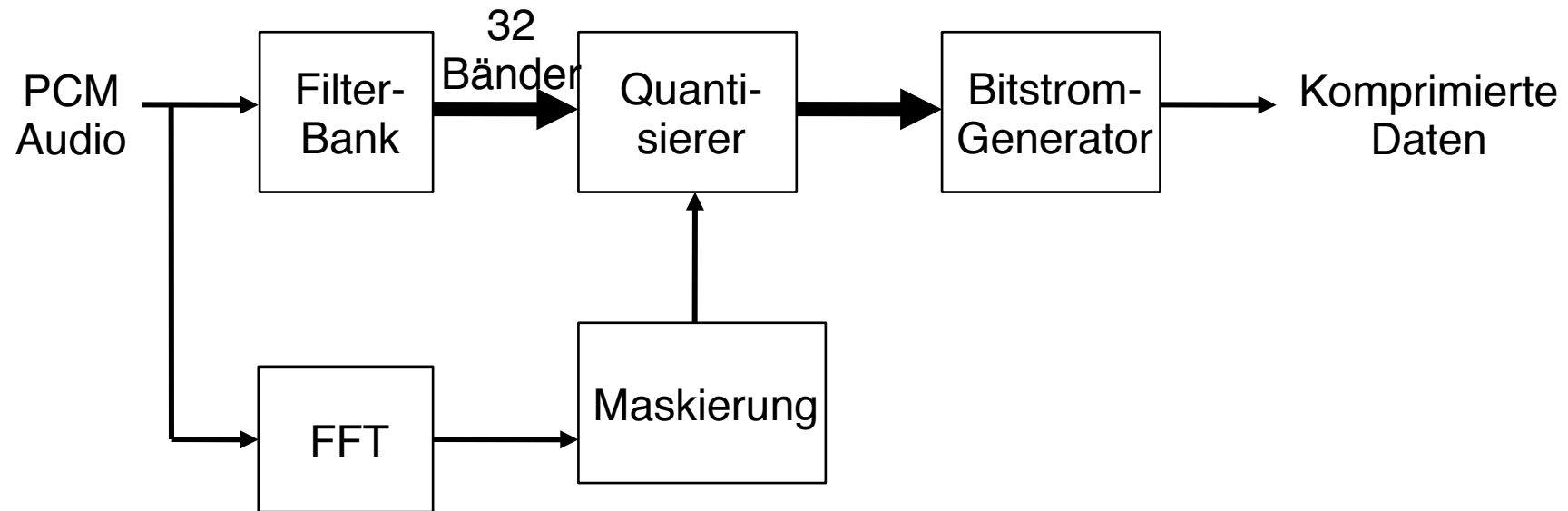
# Polyphasen-Filterbank



$$s_t[i] = \sum_{k=0}^{63} \sum_{j=0}^7 M[i][k] * (C[k+64j] * x[k+64j])$$

- Basiert auf Verfahren von Rothweiler
- Optimierter Algorithmus: Ca. 80 Multiplikationen und 80 Additionen pro Ausgabewert, gut in Hardware (DSP) realisierbar

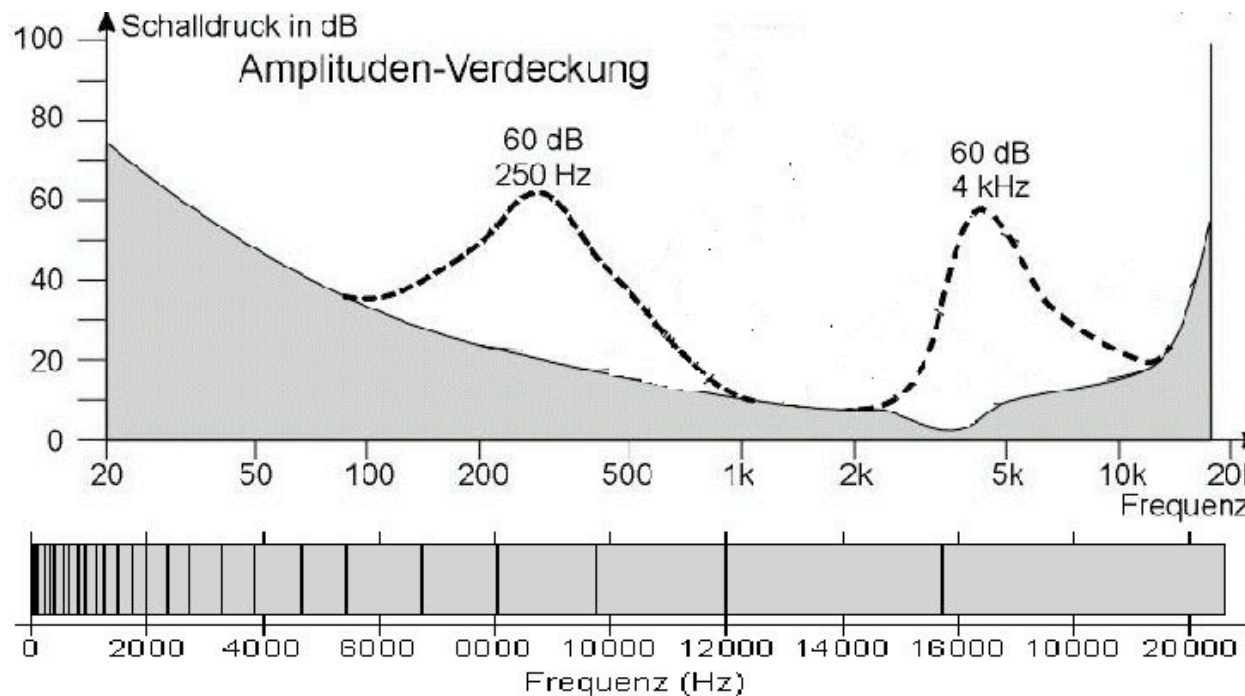
# Aufbau eines MPEG-Layer I/II Encoders



- Signal wird in 32 *gleich breite* Frequenzbänder aufgeteilt
  - Effektive Bandfilter funktionieren nur für gleich breite Teilbänder
  - Breite der Teilbänder bei Layer I/II: 750 Hz
  - „Unterabtastung“ der Subbänder: Keine zusätzliche Bandbreite benötigt
- Wegen der Eigenschaften des menschlichen Gehörs sind die Teilbänder ungeeignet für Maskierung
  - Zu breit bei niedrigen und zu schmal bei hohen Frequenzen
  - Einsatz einer zusätzlichen Frequenzanalyse (Fast Fourier Transform, FFT)

# Psychoakustisches Modell

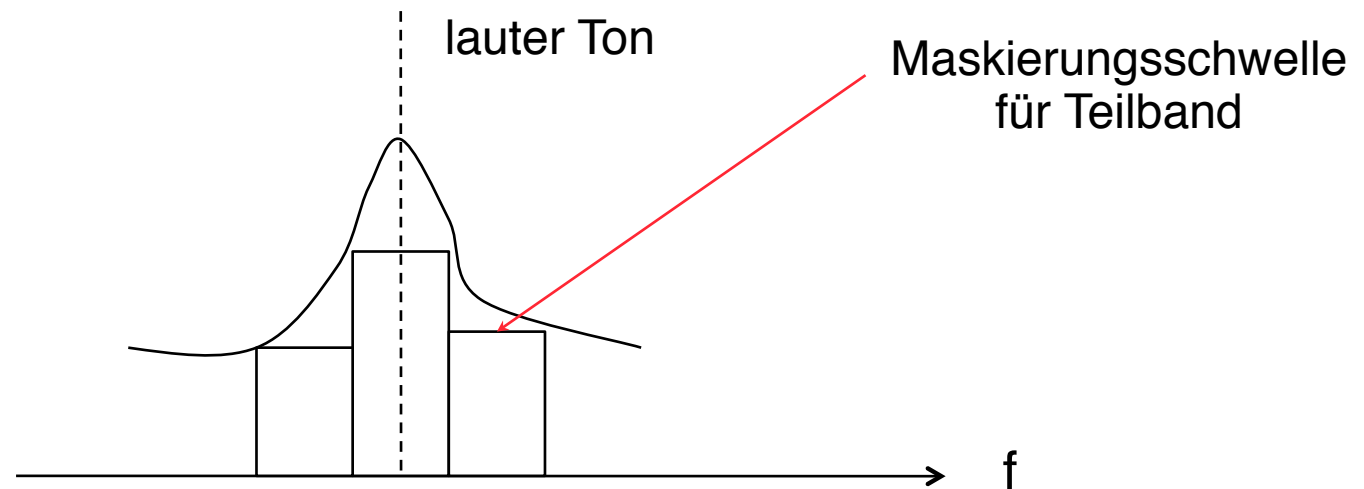
- Aus dem aktuellen Signalspektrum ergibt sich eine aktuelle Hörbarkeitskurve (wird berechnet)
  - Insbesondere: Für jedes Frequenzband eine Maskierungsschwelle, unter der der Ton nicht mehr hörbar ist
  - Details: z.B. tonale vs. geräuschartige Anteile verschieden behandelt





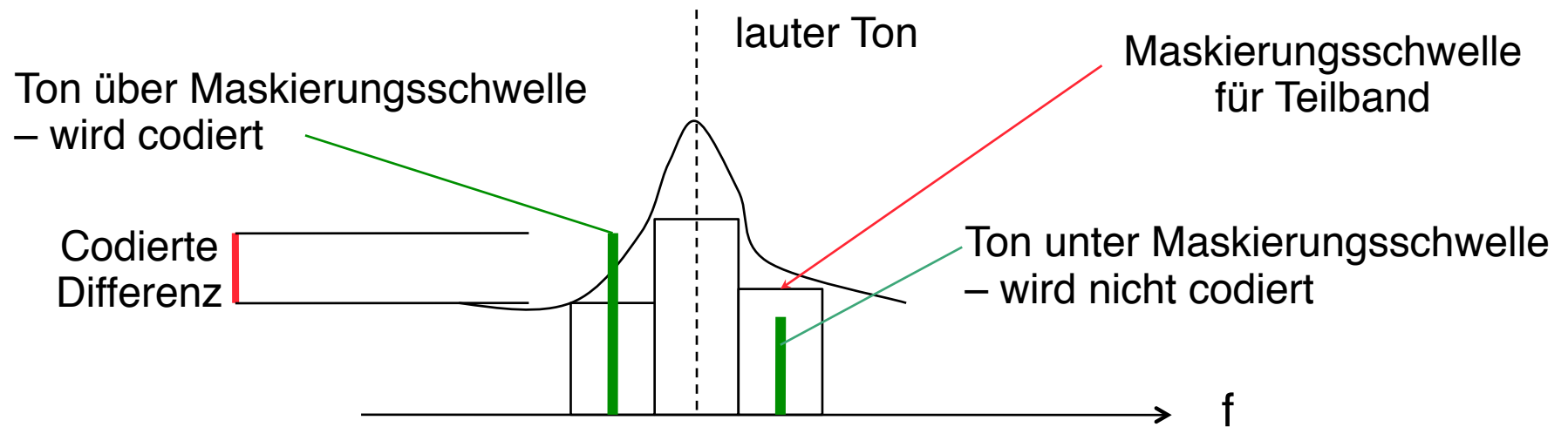
# FFT zur Berechnung der Maskierungsschwelle

- FFT = Fast Fourier Transform
- Umsetzung des Amplitudensignals in Frequenzspektrum
  - Angewandt auf die Länge eines Frames (12 Samples)
- Ergebnis:
  - Aufteilung des Signals auf viele (Layer I 512, Layer II 1024) Frequenzanteile
- Weiterverarbeitung:
  - Berechnung der Kurve für die (frequenzabhängige) Maskierungsschwelle



# Maskierung

- Die Maskierungsschwellen aus dem psychoakustischen Modell werden mit dem tatsächlichen Signalpegel (pro Teilband) verglichen
  - Verdeckte Signalanteile werden nicht codiert
- Es genügt bei teilweiser Maskierung eine geringere Bitauflösung
  - Nur „Differenz“ oberhalb der Maskierungsschwelle wird wahrgenommen!



# Maskierung: Beispiel

- Ergebnis nach der Analyse der ersten 16 Bänder:

Band	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Pegel (dB)	0	8	12	10	6	2	10	60	35	20	15	2	3	5	3	1

- Annahme: Psychoakustisches Modell liefert, dass der Pegel in Band 8 (60 dB) zu folgender Maskierung der Nachbarbänder führt:

- Maskierung um 12 dB in Band 9
- Maskierung um 15 dB in Band 7

- Pegel in Band 7 ist 10 dB  
--> Weglassen!

- Pegel in Band 9 ist 35 dB  
--> Codieren!

Wegen Maskierung 12 dB Ungenauigkeit (Rauschen) zulässig, d.h. mit zwei Bit weniger codierbar

1 Bit der Codierung = doppelter Amplitudenumfang = 6 dB Genauigkeit !

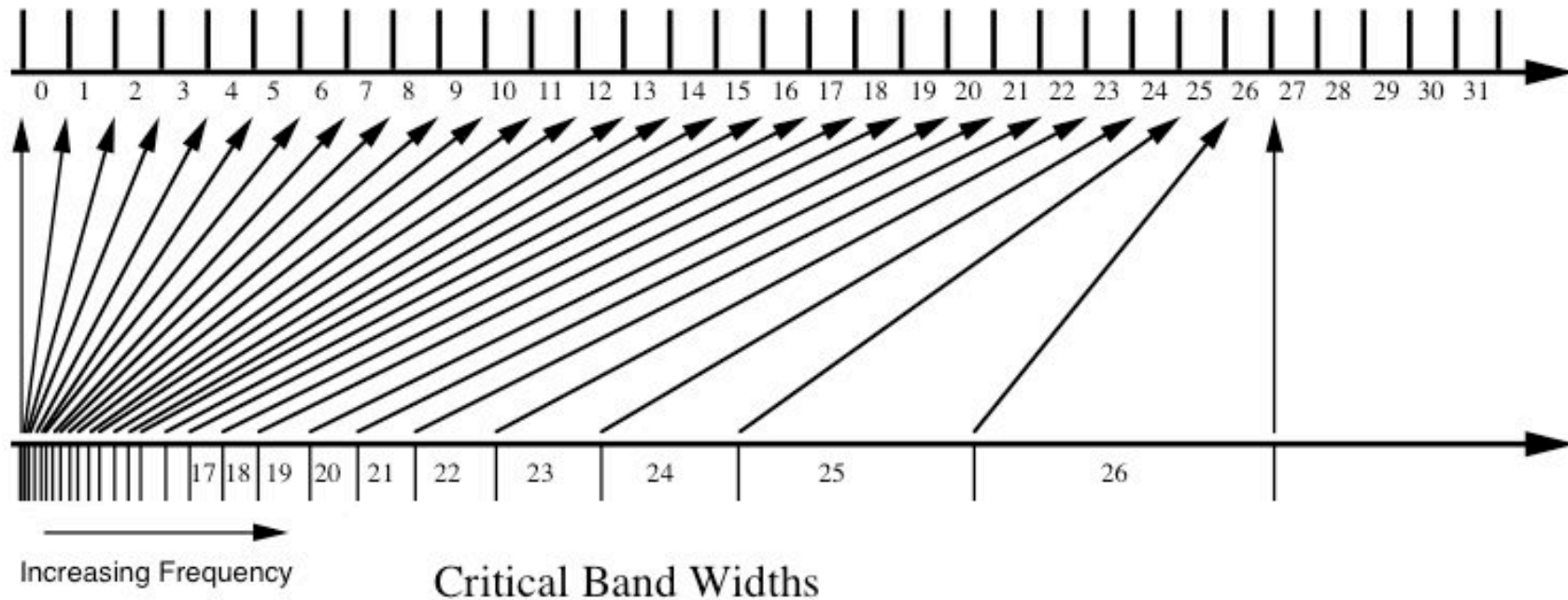
# Unterschiede der MPEG Layer

- Layer I:
  - 32 gleichbreite Teilbänder
  - FFT mit 512 Punkten
  - Betrachtung nur eines Frames
  - Psychoakustisches Modell benutzt nur Frequenzmaskierung
- Layer II:
  - 32 gleichbreite Teilbänder
  - FFT mit 1024 Punkten
  - Betrachtung von drei Frames (jetzt, vorher, nachher)
  - Einfache Zeitmaskierung, verfeinerte Bittiefenzuweisung
- Layer III:
  - Teilbänder verschiedener Breite, ähnlich zu den kritischen Bändern
  - Größere Frames (36 Samples)
  - (Modified) DCT der Teilbänder  
(in überlappenden „Fenstern“ variierender Breite)
  - Zusätzliche Entropiecodierung (Huffman)
  - Behandlung von Stereo-Redundanzen

# Kritische Bänder und Filterbänder

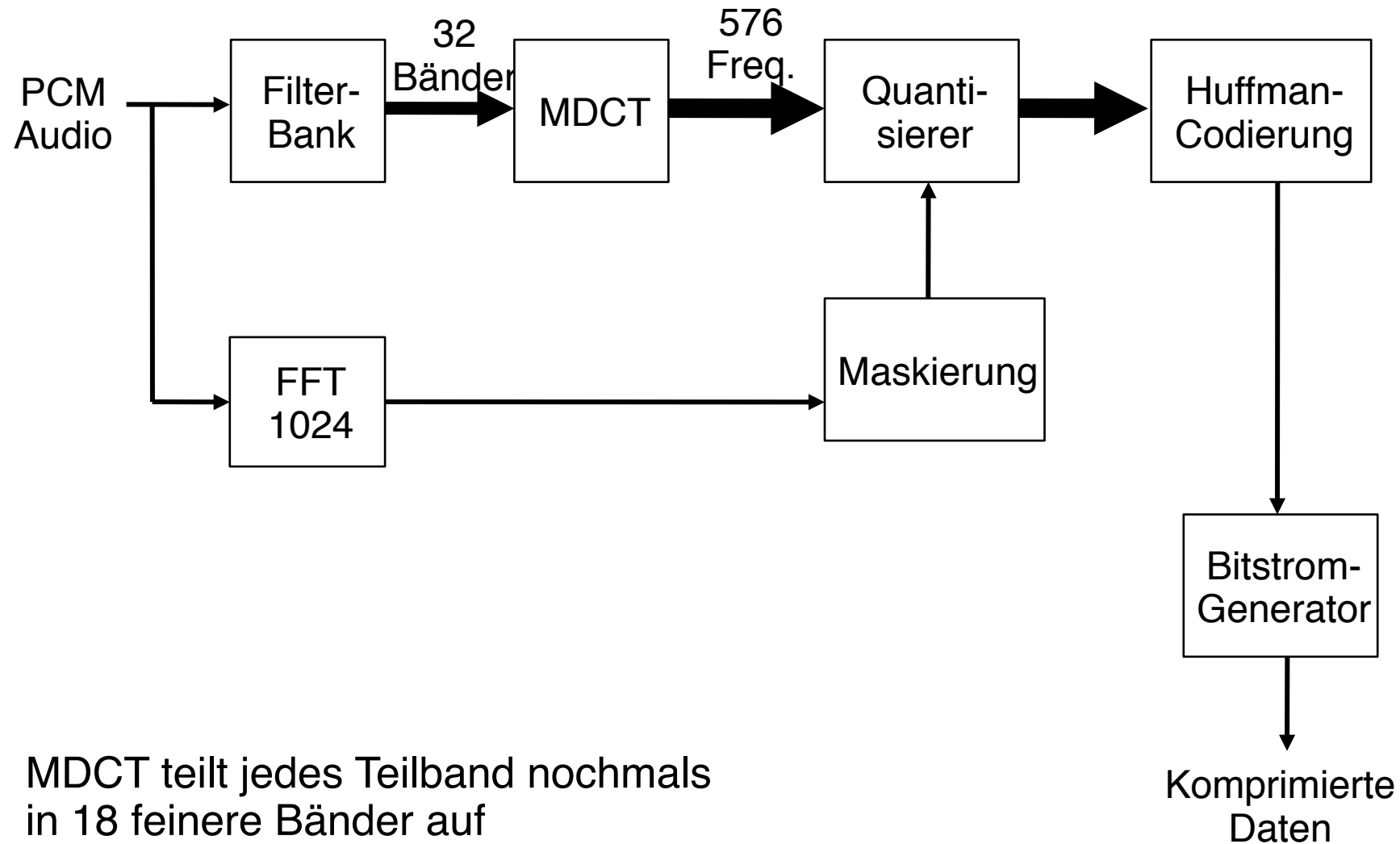
- Situation in MPEG Layer I/II:

MPEG/Audio Filter Bank Bands



Ziel: bessere Anpassung an die Bandbreite der kritischen Bänder  
Aber: Nicht durch Filterbank realisierbar

# Aufbau eines MPEG-Layer III Encoders



# DCT: Diskrete Cosinus-Transformation

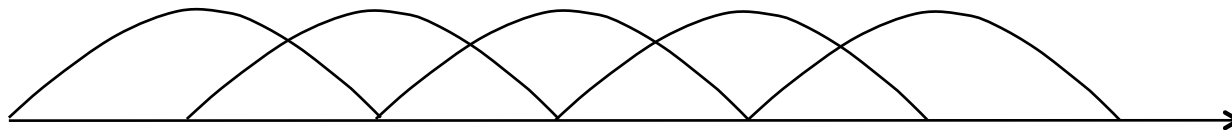
- Ähnlich zur Fourier-Transformation:
  - Gegebenes Signal wird durch Anteile bestimmter Grundfrequenzen beschrieben
- Diskrete Transformation:
  - $n$  Messwerte werden in  $n$  Anteilswerte (*Koeffizienten*) umgerechnet
  - Lineare Transformation (Matrixmultiplikation)
    - » D.h. sehr effizient zu berechnen
- Vorteile der Cosinus-Transformation
  - Besser geeignet für Kompression (Filtern von Frequenzen)
  - Bessere „Kompaktheits“-Eigenschaften (Energie auf wenige Grundfrequenzen konzentriert)
  - Glattere Übergänge

$$f_j = \sum_{k=0}^{n-1} x_k \cos \left[ \frac{\pi}{n} (j + 1/2)(k + 1/2) \right]$$

# Modified Discrete Cosine Transform MDCT (1)

- DCT
  - entspricht kleineren Teilbändern bei der Maskierungsanalyse
  - bei Audio Probleme mit Artefakten an Blockgrenzen
- Modified DCT
  - Überlappung der Cosinusfunktionen um 50%
  - Damit Vermeidung von Artefakten durch Blockgrenzen
  - Doppelt einbezogene Werte heben sich gegenseitig auf
  - Adaption der „Fenstergröße“ an Signalverlauf möglich

Überlappungen der Fenster bei MDCT:





# Modified Discrete Cosine Transform MDCT (2)

- Modified DCT
  - Adaption der „Fenstergröße“ an Signalverlauf möglich

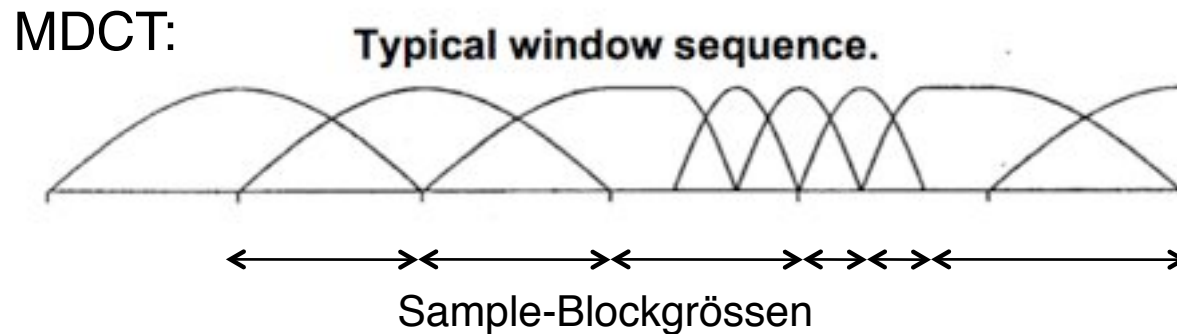


Bild: cnx.org

- Bei MP3: 6-Sample-Blöcke und 18-Sample-Blöcke
  - 6 Samples: Gut für schnelle Änderungen (Transienten)
  - 18 Samples: Gute Frequenzauflösung (wenn Signal relativ stationär)

# Stereophonie in MPEG-Audio

- Single Channel
  - Monosignale
- Dual Channel
  - Verschiedene Monosignale (z.B. Sprachsynchronisation)
- Stereo Coding
  - Separat codierte Stereosignale
- Joint Stereo Coding
  - Redundanzen im Stereosignal ausgenutzt
  - Linker Kanal und Differenz Links/Rechts
  - Frequenzabhängigkeit der Raumwahrnehmung
    - » Monosignal für tiefe Frequenzen
  
- Hinweis:
  - Räumliches Hören kann z.T. MPEG-Kompressionsverluste wahrnehmbar machen; spezielle Vorkehrungen nötig

# MPEG-2 Advanced Audio Coding AAC

- AAC = Advanced Audio Coding
  - Nachträglich zum MPEG-2 Standard hinzugefügt
  - Nicht rückwärtskompatibel
  - Bei gleicher Bitrate qualitativ überlegen zu MP3
- MPEG-2 AAC:
  - Größere Auswahl an Abtastfrequenzen (8 kHz bis 96 kHz)
  - 48 volle Audio-Kanäle
  - Reines MDCT-Filter, keine Filterbank mehr
  - Stark adaptierende Fenstergrößen
  - Joint Stereo Coding flexibilisiert (Methodenwahl frequenzabhängig)
  - Prädiktive Kodierung im Frequenzraum (Temporal Noise Shaping TNS)
    - » gute Kodierung für „Transiente“ (zeitweilige Pegelspitzen)

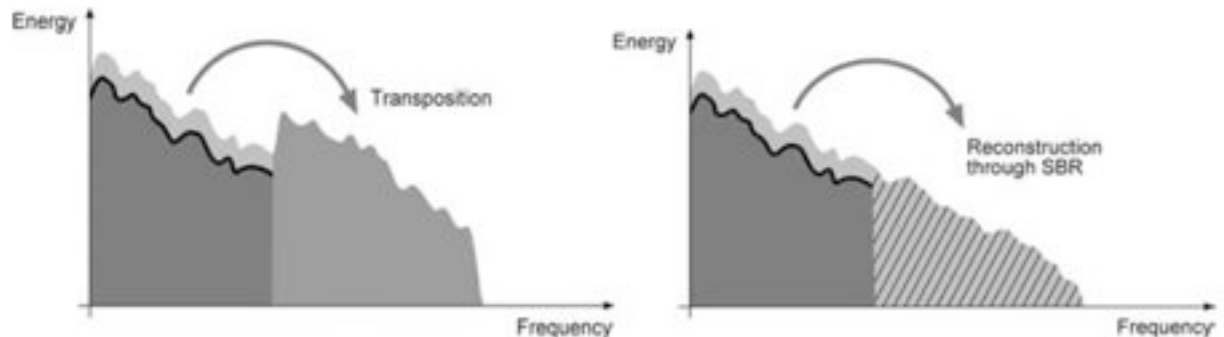
# MPEG-4 Advanced Audio Coding

- AAC = Advanced Audio Coding
  - Verbesserte Fassung des MPEG-2 Standards im aktuellen Video-/Audio-Standard MPEG-4
- MPEG-4 AAC:
  - alle Vorteile von MPEG-2 AAC
  - Perceptual Noise Substitution: Rauschen-ähnliche Teile des Signals werden beim Dekodieren synthetisiert
  - Long Term Prediction: Verbesserte Prädiktionskodierung
  - "Baukasten" zur Konstruktion verschiedener Kompressionsverfahren (effiziente Sprachcodierung bis hin zu sehr hoher Musikqualität)
  - "Profile", d.h. feste Kombinationen der Bausteine, Beispiele:
    - » Speech Audio Profile, Synthetic Audio Profile, High Quality Audio Profile, Low Delay Audio Profile, Mobile Audio Internetworking Profile

# High-Efficiency AAC (HE-AAC)

- Auch AAC+ (v1) genannt
  - v2 (eAAC+) mit zusätzlicher parametrischer Stereo-Kompression
- Gute Audioqualität bei niedrigen Bitraten (z.B. für Livestreams)
- SBR (Spektralband-Replikation):
  - Frequenzanteile oberhalb einer Grenzfrequenz (z.B. 8 kHz) werden nicht direkt codiert (Bandbreitenbegrenzung)
  - Enge Korrelation zwischen Signalverlauf in hohen und niedrigen Frequenzbändern
  - Akustische Wahrnehmung bei hohen Frequenzen weniger genau
  - Hohe Frequenzanteile aus niedrigeren synthetisiert

Henn et al.: Spectral Band Replication (SBR) Technology and its Application in Broadcasting,  
[www.broadcastpapers.com](http://www.broadcastpapers.com)



# Weitere Audiokompressionsverfahren

- Dolby AC-3 (Audio Code No. 3)
  - Prinzipiell sehr ähnlich zu den MPEG-Verfahren
  - Time-Domain Aliasing Cancellation (TDAC)
    - » Überlappende Fenster in einer MDCT
    - » Transformation so ausgelegt, dass sich Redundanzen im Folgefenster auslöschen
- ATRAC (Adaptive Transform Acoustic Encoding)
  - Sony-Verfahren, entwickelt für MiniDisc
  - Ebenfalls Aufteilung auf Teilbänder, MDCT, Skalierung
  - Hörbare Verzerrungen bei mehrfachem komprimieren/dekomprimieren
- Microsoft Windows Media Audio (WMA)
  - Nicht offengelegtes Verfahren mit recht hoher Kompression (CD-Qualität bei 64 kbit/s)

# VORBIS


- Meist in Zusammenhang mit dem "Container"-Format (zur Datenspeicherung) *Ogg* benutzt, deshalb auch *Ogg-Vorbis*
- Offenes und kostenloses Audio-Kompressionsverfahren
  - Xiph.org Stiftung, OpenSource-Projekt
  - Reaktion auf Patentansprüche aus MP3
- Ähnlich AAC:
  - Reine MDCT
  - Signal wird in "Basis-Rauschen" und Rest aufgeteilt
    - » Angenehmeres Verhalten bei zu niedriger Bitrate als MP3
  - "Bitrate Peeling":
    - » Vorhandene Dateien in der Bitrate reduzieren

# Einfachere verlustbehaftete Verfahren

- Stummunterdrückung (*silence compression*)
  - Ausblenden von Zeitbereichen mit Nullsignal
- $\mu$ -Gesetz-Codierung bzw.  $\alpha$ -Gesetz-Codierung (u.a. in G.711):
  - Nichtlineare Quantisierung: leise Töne angehoben
  - Ähnlich zu Dynamischer Rauschunterdrückung in Audiosystemen
- Adaptive Differential Pulse Code Modulation (ADPCM)
  - Prädiktives Verfahren
  - Vorhersage des Signalverlaufs durch Mittelung über bisherige Werte
  - Laufende Anpassung der Quantisierungstiefe an Signal
  - Kodierung der Differenzwerte zur Prädiktion
- Linear Predictive Coding (LPC)
  - Vergleicht Sprachsignal mit analytischem Modell der menschlichen Spracherzeugung, codiert Modellparameter und Abweichungen von der Vorhersage (militärische Entwicklung)
  - Nur für Sprache, klingt „blechern“, hohe Kompression
  - Weiterentwicklungen, z.B. Code Excited Linear Predictor (CELP)



# 5. Ton und Klang

- 5.1 Ton: Physikalische und physiologische Aspekte
- 5.2 Kompression von Audio-Signalen: MPEG-Audio
- 5.3 Audio-Datenformate: Übersicht 
- 5.4 Klangerzeugung und MIDI

Weiterführende Literatur:

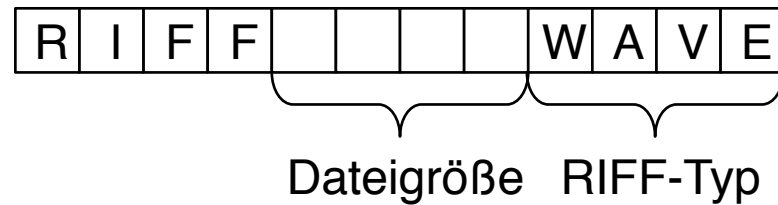
Arne Heyda, Marc Briede, Ulrich Schmidt: Datenformate im Medienbereich, Fachbuchverlag Leipzig 2003

# RIFF (Resource Interchange File Format)

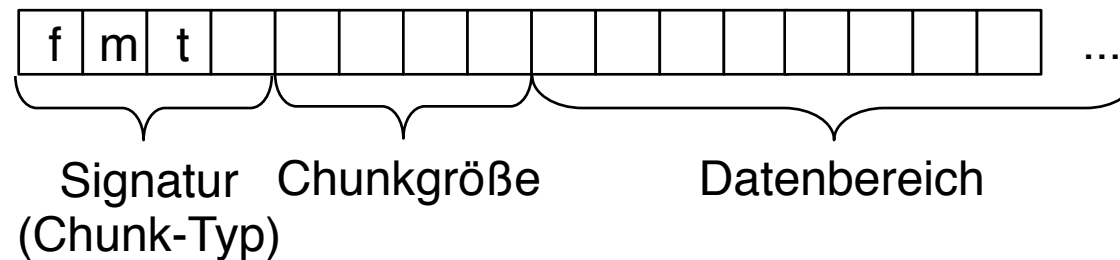
- **IFF:** 1985 von der Firma Electronic Arts eingeführt
  - Sehr einfaches Einheitsformat für verschiedene Arten von Multimedia-Daten, stark verbreitet auf AMIGA-Rechnern
  - Prinzip („Tagged File Format“):
    - » Header gibt Dateityp an
    - » Eigentliche Daten in einer Folge von ebenfalls (über Header) typisierten *chunks*
- **RIFF:**
  - Bestandteil der „Multimedia Programming Interface and Data Specifications“ von Microsoft und IBM, 1991
  - Basiert auf der Idee von IFF
  - Existiert prinzipiell in zwei Varianten:
    - » RIFF für Intel-Architektur („little-endian“)
    - » RIFX für Motorola-Architektur („big-endian“)(RIFX heutzutage auch auf Motorola-Prozessoren ungebräuchlich)

# Grundstruktur von RIFF-Dateien

RIFF-Header (in Bytes):



Chunk-Header (in Bytes):

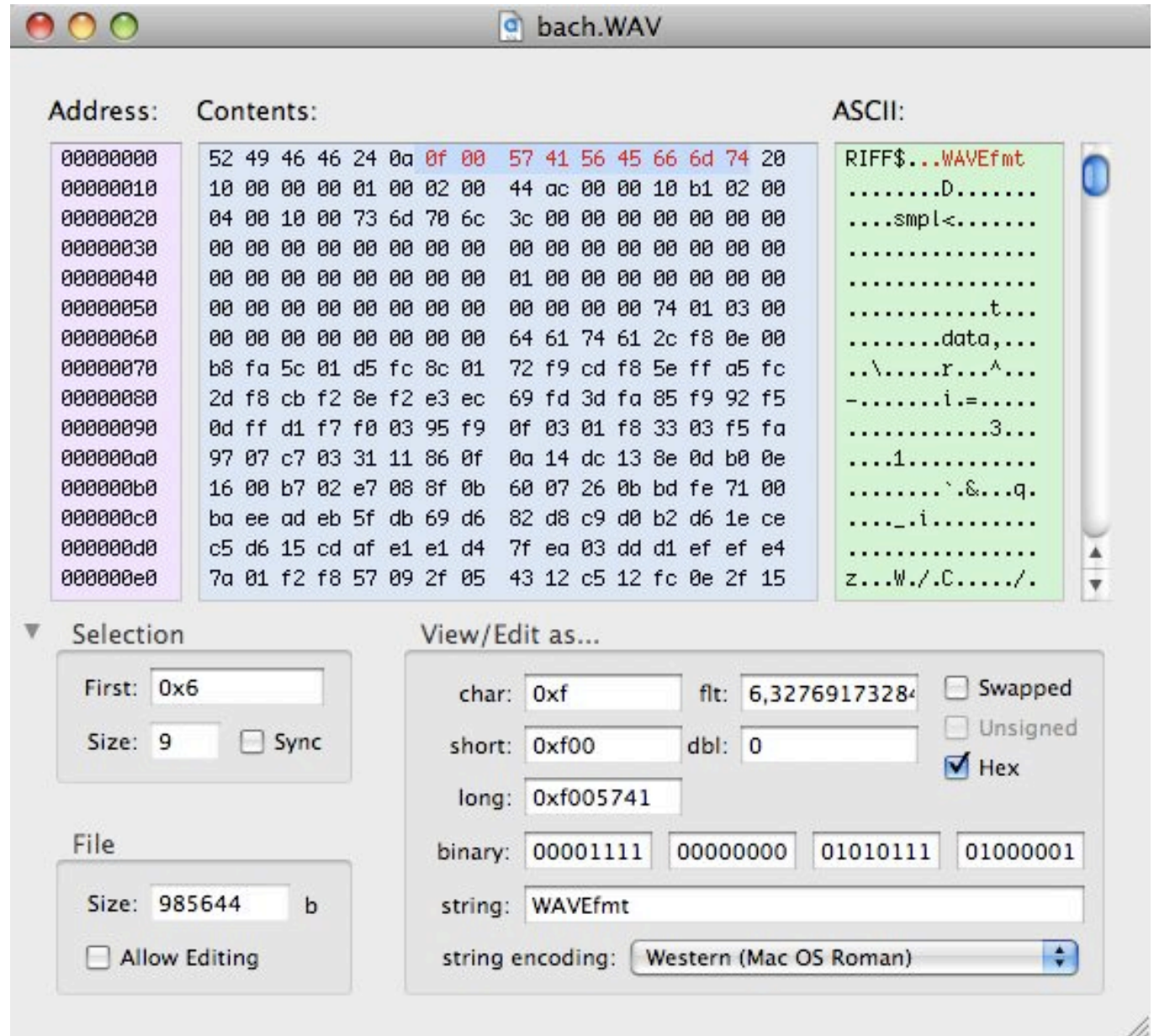


- Verbreitete RIFF-Datentypen (als eigenständige Dateiformate bekannt):
  - WAVE (oder .wav): Audio, unkomprimiert
  - AVI: Video (Audio/Video Interlaced), unkomprimiert
  - RMI: MIDI-Daten (sh. später)
  - BND: „Bündel“ von RIFF-Dateien

# Wave-Format

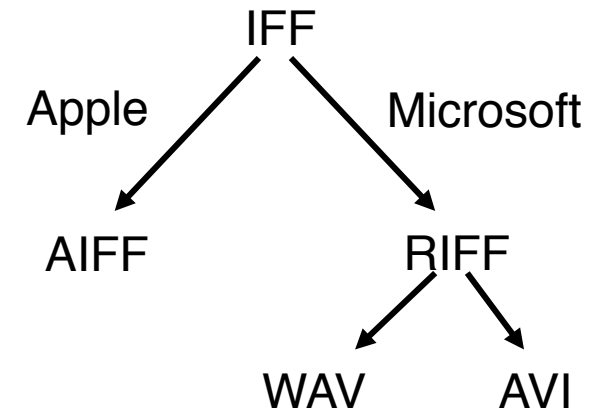
- Spezialfall des RIFF-Formats (RIFF-Typ „WAVE“)
- Zwei Arten von Chunks:
  - FMT-Chunk (Signatur „fmt“)
    - » Format-Typ (z.B. MS PCM, IBM ADPCM)
    - » Anzahl Kanäle
    - » Sampling-Rate (Hz)
    - » Datenrate (Bytes/s)
    - » Größe von Datenblöcken
    - » Formatspezifische Information  
(Z.B. bei MS PCM 2 Byte Sample-Größe (bits/Sample))
  - DATA-Chunk (Signatur „data“), meist nur ein solcher Chunk vorhanden
    - » Bei mehreren Kanälen „interleaving“,  
d.h. alle Kanäle für einen Zeitpunkt in Folge

# Beispiel: Hexadezimaler Editor



# AIFF (Audio Interchange File Format)

- Herstellerspezifische Erweiterung von IFF durch Apple für unkomprimiertes Audio
  - Format-Chunk
  - Daten-Chunks, byteweise gepackt
- Audiodaten für bis zu 6 Kanäle (Surround Sound)
- Möglichkeit zur Einstreuung von MIDI-Chunks und Instrumenten-Chunks
- Spezialvariante AIFF-C für komprimierte Audiodaten (ca. 6:1)



# AU (Audio File Format)

- Bei NeXT entwickelt, weit verbreitet im UNIX-Bereich (z.B. Sun)
- Header:
  - Abtastrate, Kanalzahl, Datenformat etc.
  - beliebig lange Textinformation
- Datenbereich:
  - Kanäle miteinander verschränkt
  - Viele Datenformate, z.B.:
    - » von 8 bis 32 Bit
    - »  $\mu$ -Law und linear
    - » Festkomma, Gleitkomma, doppelte Genauigkeit
- Unterstützung von Dateifragmentierung

# QuickTime

- Bibliothek von systemnahen Programmen für MacOS und Windows für die Bearbeitung von zeitbasierten Medien („movies“)
  - Entwickelt von Apple ca. 1991–heute
- Sehr allgemeines Konzept für Medienstrukturen
  - „Atom“: Allgemeiner Container für Mediendaten
  - Mehrere Tracks je Präsentation
  - Pro Track:
    - » Medienstruktur (Referenzen zu Medien verschiedenen Typs)
    - » „Edit List“ für Zeitsynchronisation
- QuickTime wurde als Basis für die MPEG-4 Dateistruktur gewählt.
- Viele verschiedene Dateitypen von QuickTime unterstützt
  - Wichtiges spezifisches QuickTime-Format: „Movie“ (MOV)



- 'moov' - Movie
  - 'mvhd' - Movie Header
  - 'trak' - Track
    - 'tkhd' - Track Header
    - 'edts' - Edits
      - 'elst' - Edit List
    - 'mdia' - Media
      - 'mdhd' - Media Handler Header
      - 'hdlr' - Handler Description
      - 'minf' - Media Information
        - 'vmhd' - Video Media Header
        - 'hdlr' - Handler Description
        - 'dinf' - Data Handler Information
          - 'dref' - Data Reference
        - 'stbl' - Sample Table
          - 'stsd' - Sample Descriptions
          - 'stts' - Sample to Time
          - 'stsc' - Sample to Chunk
          - 'stsz' - Sample Sizes
          - 'stco' - Chunk Offset Table
  - 'udta' - User Data
- 'trak' - Track
  - 'tkhd' - Track Header
  - 'edts' - Edits
    - 'elst' - Edit List
  - 'mdia' - Media
    - 'mdhd' - Media Handler Header
    - 'hdlr' - Handler Description
    - 'minf' - Media Information
      - 'smhd' - Sound Media Header
      - 'hdlr' - Handler Description
      - 'dinf' - Data Handler Information
        - 'dref' - Data Reference
      - 'stbl' - Sample Table
        - 'stsd' - Sample Descriptions
 

flags	\$00000000	numEntries	1
-------	------------	------------	---
  - 'stsd' - Sample Descriptions
 

descSize	52	numChannels	2
dataFormat	sowt	sampleSize	16
dataRefIndex	1	sampleRate	0.032000
packetSize	0	compressionID	-1
bytesPerPacket	2	bytesPerFrame	4
  - 'stts' - Sample to Time
  - 'stsc' - Sample to Chunk
  - 'stsz' - Sample Sizes
  - 'stco' - Chunk Offset Table
- 'udta' - User Data

- 'udta' - User Data
- 'VLOC'

# Beispiel: QuickTime Dateistruktur


2	version
16	revLevel
0.032000	vendor
-1	samplesPerPacket
4	bytesPerSample

1	reserved	0
0		0
0		
1		
2		

# Verlustfreie Audio-Kompression: Beispiele

- MPEG-4 Audio Lossless Coding (ALS)
  - TU Berlin, Real Networks, NTT
  - 2009 in MPEG-4 Standard aufgenommen
  - Basiert auf LPC-Codierung (und Golomb-Rice-Codierung)
- FLAC (Free Audio Lossless Coding)
  - Josh Coalson, jetzt bei Xiph.org Stiftung
  - Ca. 50% Reduktion der Dateigröße
  - Lineare Prädiktion, Lauflängen-Codierung, Golomb-Rice-Codierung
- ALAC (Apple Lossless Audio Codec)
  - 40-60 % Reduktion
  - In MPEG-4 Container gespeichert

# 5. Ton und Klang

- 5.1 Ton: Physikalische und physiologische Aspekte
- 5.2 Kompression von Audio-Signalen: MPEG-Audio
- 5.3 Audio-Datenformate: Übersicht
- 5.4 Klangerzeugung und MIDI 

Literatur:

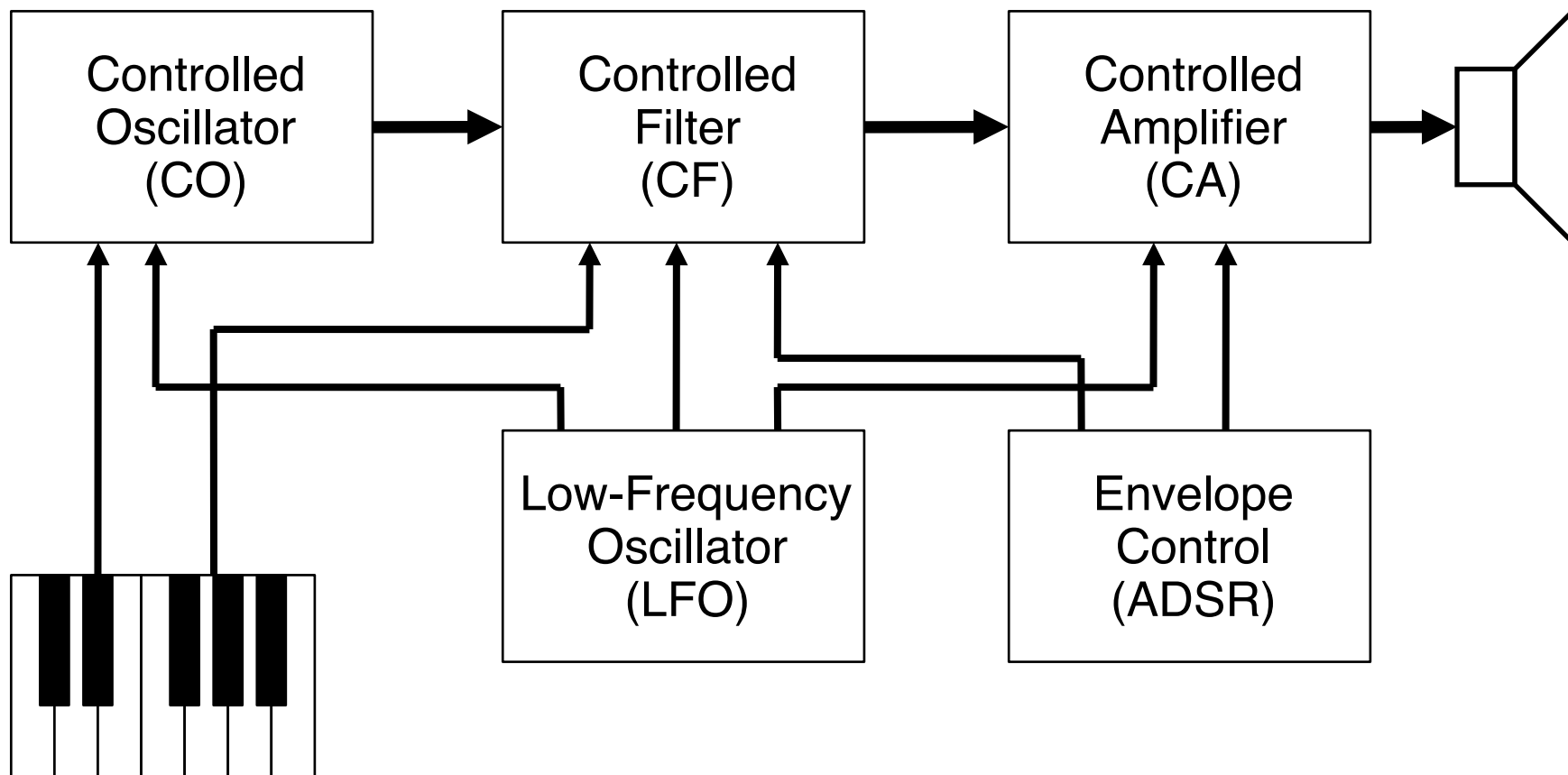
Hannes Raffaseder: Audiodesign, Fachbuchverlag Leipzig 2002

# Elektronische Klangerzeugung

- Klänge für Musik oder Sprache können künstlich produziert werden
  - Tonhöhe, Lautstärke, Klangfarbe (*timbre*) einstellbar
- Klangerzeuger:
  - Einfache Klangerzeuger in Soundkarten enthalten (Frequenzmodulation einfacher Wellenformen)
  - Hochwertige Klangerzeuger z.B. in elektronischen Musikinstrumenten („Synthesizer“, MIDI-Keyboards)
    - » mehrstimmig (z.B. 128)
    - » multitimbral (z.B. 64 Klangfarben)
- Historisch gesehen:
  - 1900 Dynamophone (Thaddeus Cahill), 1920 Termenvox (Lew Termen), 1930 Trautonium (Friedrich Trautwein), 1960 Mellotron
  - Anfang der 60er Jahre (Robert Moog): Moderne Synthesizer-Architektur
  - 1968 Walter Carlos „Switched-on Bach“



# Grundstruktur eines Synthesizers



# Grundelemente bei der Klangerzeugung

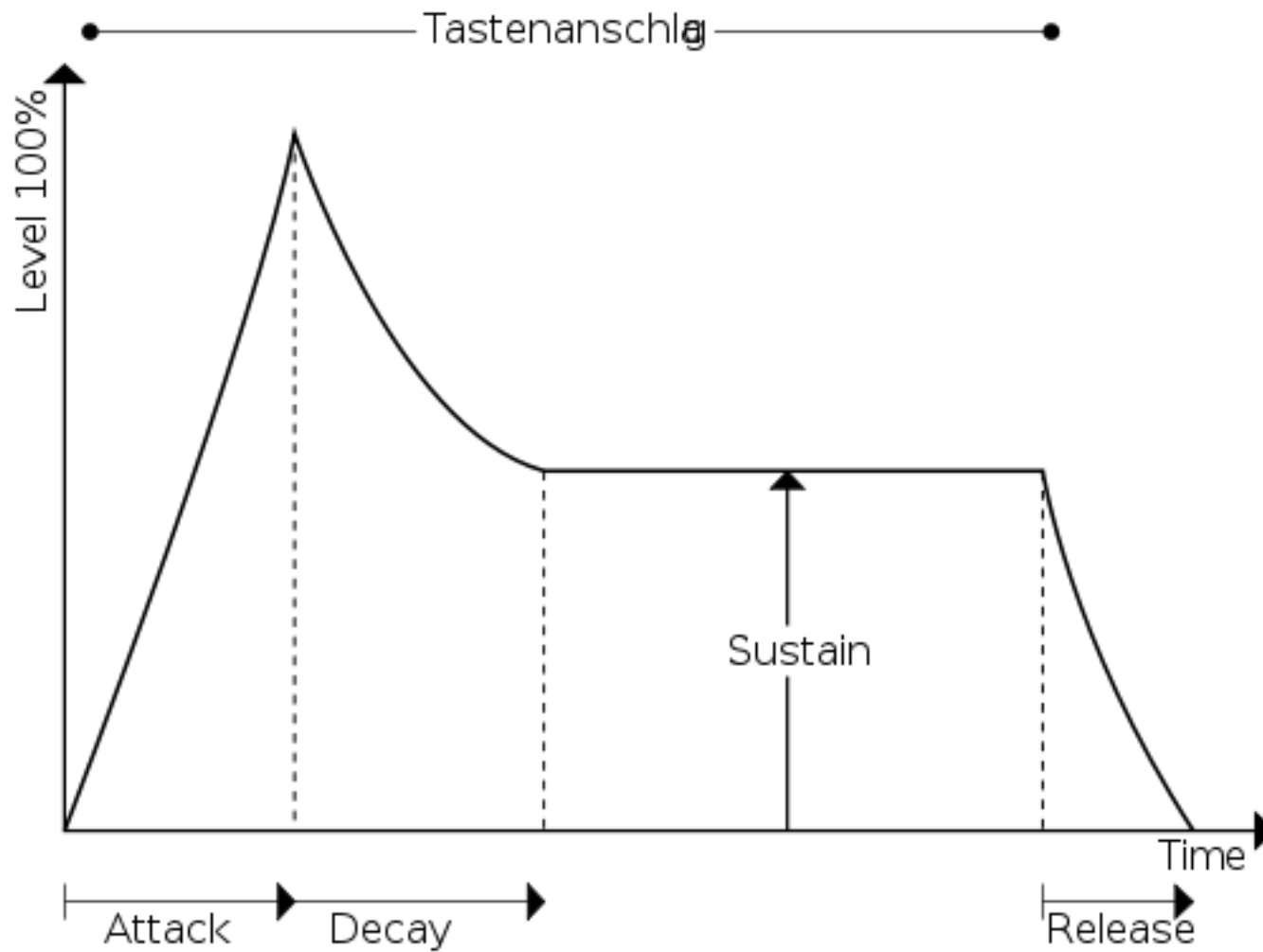
- Oszillator
  - Erzeugt mehr oder weniger obertonreiches Signal, das die Grundfrequenz und auch wesentlich den Klangcharakter bestimmt
- Filter
  - Z.B. Hochpass, Tiefpass, Bandfilter
- Verstärker (*Amplifier*)
  - Kann über zeitabhängigen Pegelverlauf Klangempfindung wesentlich beeinflussen
- Hüllkurvengenerator (*Envelope Control*)
  - Zeitlicher Verlauf eines Klangereignisses auf ein einmaliges erzeugendes Ereignis hin (z.B. Tastendruck), meist ADSR (siehe nächste Folie)
- *Low Frequency Oscillator LFO*
  - Dient zur kontinuierlichen Veränderung eines klangbestimmenden Parameters innerhalb eines bestimmten Zeitintervalls
  - Frequenzen typischerweise zwischen 0 und 20 Hz

# Beispiel: Software-Synthesizer



Software: Propellerhead Reason

# ADSR-Modell





# Verfahren zur Klangsynthese

- Additive Klangsynthese
  - Fourier-Analyse in der Praxis, zur Synthese umgewandelt
  - Realisierung eines Klangs als Überlagerung von Sinustönen
- Subtraktive Klangsynthese
  - Erzeugung obertonreicher Grundsignale (z.B. Sägezahn, Dreieck, ...)
  - Steuerung der Spektren mit Filtern und der Amplitude mit Verstärkern
- Wavetable-Synthese
  - Vordefinierte, gespeicherte Wellenformen
  - Oszillator durchläuft Wavetable in programmierter Weise (z.B. LFO)
- Sampling
  - Wiedergabe digital aufgezeichneter akustischer Ereignisse
  - Multisampling: Viele Aufnahmen mit verschiedenen Parameterwerte
- Granularsynthese
  - Zerlegung von Schallsignalen in *Grains* (wenige ms lange Abschnitte)
  - Entkopplung von Wiedergabegeschwindigkeit und Tonhöhe

# MIDI: Geschichte und Überblick

- Synthesizer: Revolutionäres Musikinstrument in den 70er Jahren
  - Beatles (White Album), Carlos (Switched-on Bach), ...
  - Technische Probleme:  
Polyphonie, Kombination verschiedener Geräte, Synchronisation
- 1983: Erste Interoperabilitäts-Vorführung
- MIDI (Musical Instrument Digital Interface) Standard
  - International MIDI Association (IMA)
  - MIDI Manufacturers Association (MMA)
- Bedeutung für Multimedia:
  - Standardisierte Sprache für
    - » Übernahme von Daten aus Endgeräten, die Musikinstrumenten entsprechen (insb. Keyboard)
    - » Ansteuerung von Peripheriegeräten (Synthesizer, Beleuchtung)
    - » Abstrahierte Darstellung von gespielter Musik

# MIDI-Grundbegriffe

- Ereignis (*event*):
  - Musikalische Aktion, z.B. Musiker drückt Taste auf Keyboard mit bestimmter Anschlagsstärke (*velocity*)
    - » etwa: „NOTE ON C3 velocity 100“
  - Jedes Ereignis findet zu einem bestimmten Zeitpunkt statt (Zeitstempel)
- Nachricht (*message*):
  - Binäre Codierung der in einem Ereignis enthaltenen Information
  - Kann gespeichert, weitergegeben, vervielfältigt, modifiziert werden
- Befehl (*command*):
  - Anweisung an ein externes Gerät, bestimmte musikalische Aktionen auszuführen
- Klangfarbe (*timbre*):
  - Charakteristik eines bestimmten wiederzugebenden Instruments
  - „Multitimbral“
- Kanal (*channel*):
  - Identifikator für bestimmten Empfänger (traditionell 16 Kanäle)
  - „Musikinstrument“ bzw. entsprechender Klangerzeugungsprozess

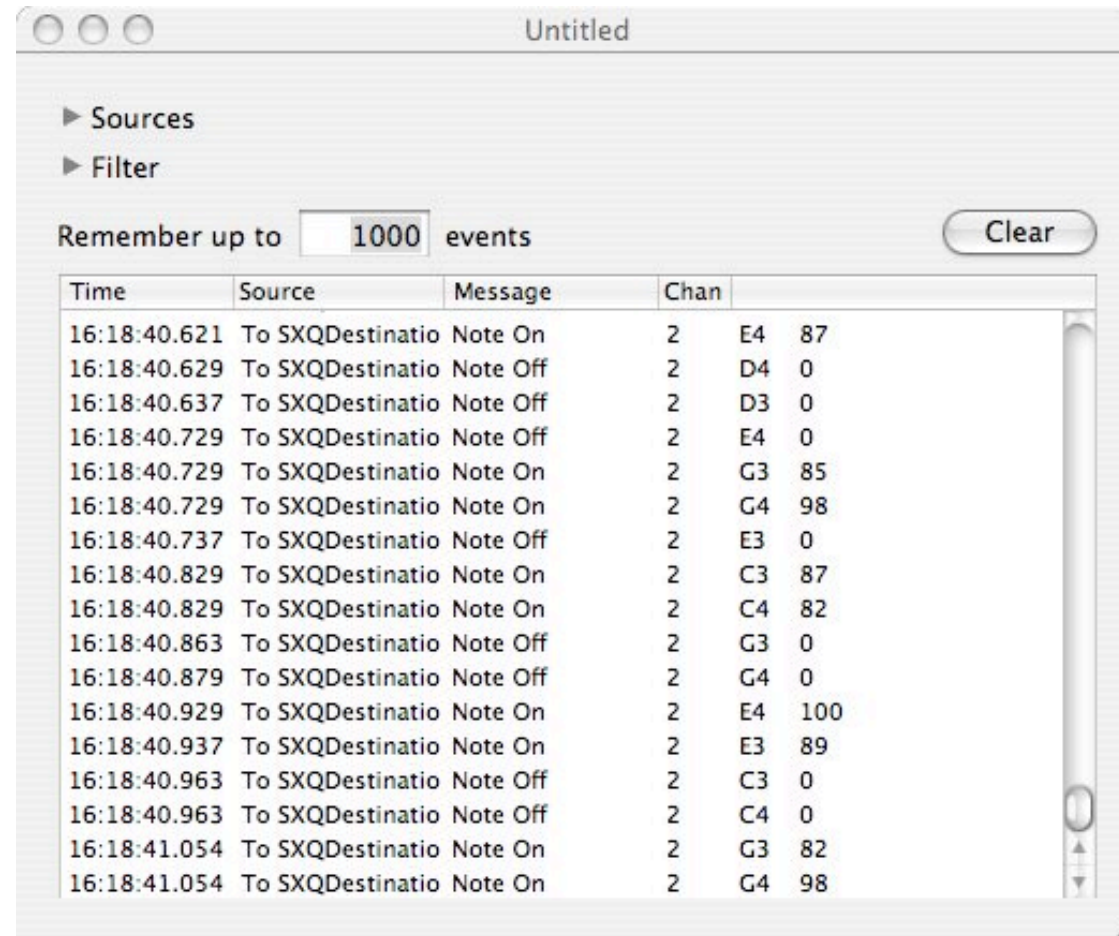
# MIDI-Nachrichten

- Channel Voice Messages
  - Eigentliche Musikdaten (sh. nächste Folie)
- Channel Mode Messages
  - Steuerung des Synthesizers
    - » Ein-/Ausschalten der eigenen Tastatur (z.B. bei Keyboard/Synthesizer)
    - » Testmodus
    - » Polyphonie-Steuerung
- System Real-Time Messages
  - Synchronisationstakt
  - Synchronisierte Sequenzen
  - Überprüfung der Verfügbarkeit von Geräten
- System Exclusive Messages (SysEx)
  - Weitergabe herstellerspezifischer Information an individuelle Geräte

# Inhalt einer MIDI-Datei: MIDI-Ereignisse

- Header-Information
- Track-Information
  - *Track* = Separat abspielbare und bearbeitbare Musikspur
- Track-Information Teil 1: Metainformation
  - Track-Nummer, -Name
  - Angaben zum Instrument (z.B. aus *General Midi*-Instrumenten)
  - Zeitbasis
- Track-Information Teil 2: Melodie
  - Folge von Channel Voice Messages, mit Zeitstempel relativ zur Zeitbasis
  - Note On (Parameter Notenwert, Anschlagstärke)
  - Note Off (Parameter Notenwert, Anschlagstärke)
  - Polyphonic Key Pressure (Parameter Notenwert, Anschlagstärke)  
(Änderung der Anschlagstärke über die Zeit)
  - Pitch Bend Change (Parameter Verschiebung)  
(Tonhöhenverstellung)

# MIDI Ereignisse: Beispiel



Remember up to  events Clear

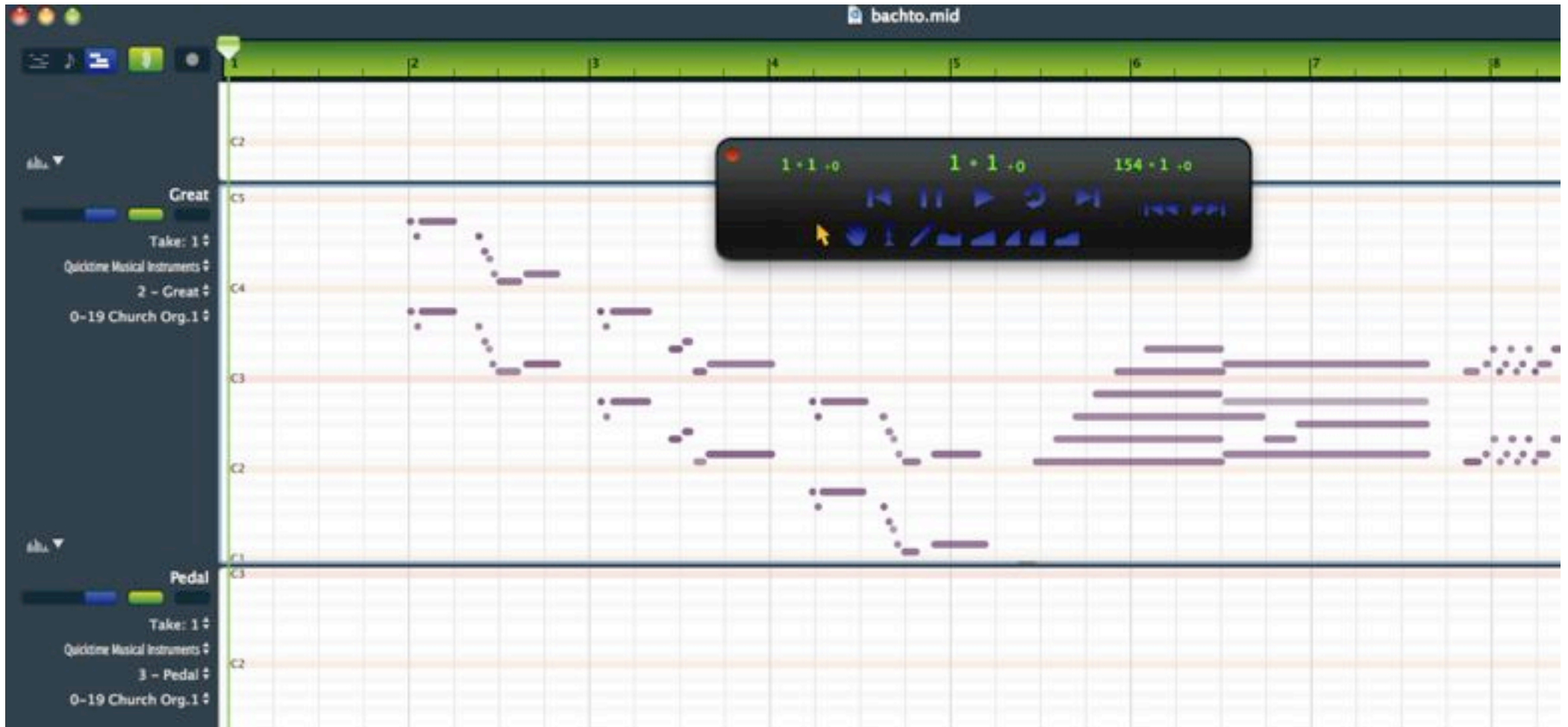
Time	Source	Message	Chan
16:18:40.621	To SXQDestinatio	Note On	2 E4 87
16:18:40.629	To SXQDestinatio	Note Off	2 D4 0
16:18:40.637	To SXQDestinatio	Note Off	2 D3 0
16:18:40.729	To SXQDestinatio	Note Off	2 E4 0
16:18:40.729	To SXQDestinatio	Note On	2 G3 85
16:18:40.729	To SXQDestinatio	Note On	2 G4 98
16:18:40.737	To SXQDestinatio	Note Off	2 E3 0
16:18:40.829	To SXQDestinatio	Note On	2 C3 87
16:18:40.829	To SXQDestinatio	Note On	2 C4 82
16:18:40.863	To SXQDestinatio	Note Off	2 G3 0
16:18:40.879	To SXQDestinatio	Note Off	2 G4 0
16:18:40.929	To SXQDestinatio	Note On	2 E4 100
16:18:40.937	To SXQDestinatio	Note On	2 E3 89
16:18:40.963	To SXQDestinatio	Note Off	2 C3 0
16:18:40.963	To SXQDestinatio	Note Off	2 C4 0
16:18:41.054	To SXQDestinatio	Note On	2 G3 82
16:18:41.054	To SXQDestinatio	Note On	2 G4 98

- MIDI-Dateien sind extrem kompakt.
- MIDI-Aufzeichnungen sind genauer als normale Notenschrift!

# Typische Funktionen von MIDI-Sequenzern

- „Sequencer“ = Software zur Bearbeitung von synthetisierter Musik, z.B. mit MIDI
- Aufnehmen und Wiedergeben von Tonspuren
- Verschiedene Ansichten der gleichen Information:
  - Partitur, Keyboard-Matrix
  - Zeitgenaue Liniendarstellung
  - Darstellung von Zusatzinformation (z.B. *velocity*)
- Musik-Editor:
  - Komponieren (Noten einsetzen und verschieben, Längen verändern, Transponieren, ...)
  - Instrumente variieren
  - Effekte einfügen
  - Synchronisieren von Spuren und Abmischen
  - Oft integriert mit klassischer Mischpult-Funktionalität
  - Oft integriert mit Notensatz-Funktionalität

# Beispiel: MIDI-Sequencer



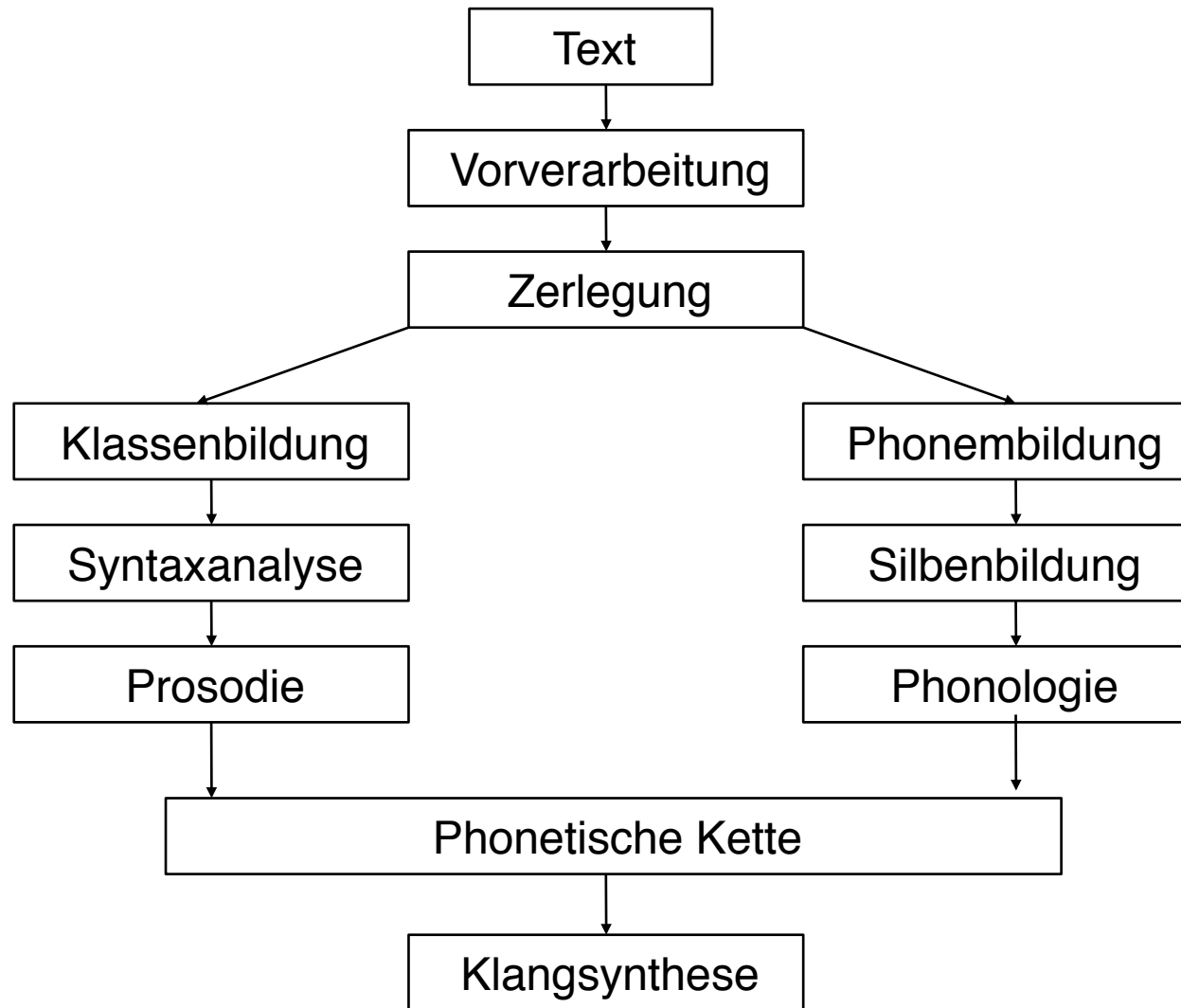
Intuem



# Sprachanalyse und Sprachsynthese

- Ein- und Ausgabe in natürlicher Sprache
  - Alter Traum der Informatik
  - Grenzgebiet zu Computerlinguistik, Künstlicher Intelligenz (KI)
- Sprachausgabe:
  - relativ stabile Technologie
  - Bestandteil vieler Standard-Betriebssysteme
- Spracheingabe:
  - immer noch relativ wenig beherrscht
  - Trainingsfreie Systeme noch störanfällig
  - Trainingsgebundene Systeme existieren mit akzeptabler Leistung

# Sprachsynthese: Grob Ablauf



# Weiterentwicklung im Bereich Klangerzeugung

- MPEG-4 Standard:
  - *Structured Audio Format* ermöglicht Spezifikation von Klangerzeugern
  - *SAOL (Structured Audio Orchestral Language)* zur Beschreibung von elektronischen Instrumenten und Audioeffekten
  - *SASL (Structured Audio Source Language)* erlaubt differenzierte Formulierung von Spielanweisungen (über MIDI hinaus)